



Device Network SDK (General)

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

Chapter 1 Overview	1
1.1 Introduction	1
1.1.1 Integrate Based on Private Protocol	1
1.1.2 Integrate Based on Text Protocol	2
1.2 System Requirements	3
1.3 SDK Component	4
1.4 Update History	8
1.4.1 Version 6.1.X.X	8
1.5 Notices	23
Chapter 2 Security Service	24
2.1 Configure Double Verification	24
2.2 Encrypt Stream	27
2.3 Encrypt Stream Transmission Link	29
2.3.1 (Optional) Verify Device	29
2.3.2 Encrypt HTTPS Link	30
2.3.3 Encrypt TLS Link	31
Chapter 3 User Management	32
Chapter 4 Device Management and Maintenance	33
4.1 Reset Password by Setting Recovery Email	33
4.2 Remotely Upgrade Device	35
4.3 Search for Logs	36
4.4 Configure Log Storage Mode	40
4.5 IoT Devices Access	42
4.5.1 Add IoT Device to NVR/DVR	43
4.5.2 Configure Alarm of IoT Device	46
4.6 Online Upgrade Device via Guarding Vision	48

4.7 Configuration and Maintenance	52
4.7.1 Automatic Maintenance	52
4.7.2 Basic Maintenance	53
4.7.3 Basic Settings	55
4.7.4 Device Calibration	57
4.7.5 Configure Camera Correction	58
Chapter 5 Video and Audio	61
5.1 Start Live View	61
5.2 Start Live View Based on ISAPI Protocol	69
5.3 Perform PTZ Control	70
5.3.1 PTZ Configuration	72
5.4 Search for Recording Start and End Time by Channel	78
5.5 Start Playback	80
5.6 Two-Way Audio	85
5.6.1 Start Two-Way Audio	85
5.6.2 Forward Audio	88
5.6.3 Broadcast Audio	92
5.6.4 Audio Settings	94
5.7 Stream Encoding	94
Chapter 6 Capture and Recording	97
Chapter 7 Alarm and Event Receiving	99
7.1 Supported Alarm/Event Types	99
7.2 Configure Exception Alarm	105
7.3 Configure Audio Exception Detection Alarm	107
7.4 Configure Certificate Expiry Alarm	109
7.5 Configure Network Traffic Pre-alarm	112
7.6 Configure Vibration Detection Alarm	114
7.7 Receive Alarm/Event in Arming Mode	118

7.8 Receive Alarm/Event in Listening Mode	121
7.9 Subscribe Alarm/Event in Arming Mode	124
7.10 Search for Alarms or Events	126
Chapter 8 File and Storage Management	128
8.1 Disk Quota	128
8.2 Download Video Files	128
8.3 HDD Management	132
8.4 SSD Management	136
8.5 Upload Files	138
8.6 File Search	139
8.6.1 Search Video Files by Time	139
8.6.2 Search Video Files by Event	143
8.6.3 Search Video Files by VCA Event	145
8.6.4 Search Video Files by Tag	147
8.6.5 Search for Pictures	147
8.6.6 Perform Picture Dual-VCA	147
8.7 Working Mode Configuration	148
8.8 Other File Operations	149
Chapter 9 Parameter Configuration	150
9.1 Channel Settings	150
9.2 I/O Settings	151
9.3 Video/Image Settings	152
9.4 Network Settings	158
9.4.1 Address Filter	159
9.4.2 Network Access	160
9.4.3 Network Interface	163
9.4.4 Network Service	164
9.4.5 Wireless Network	165

9.4.6 Multicast	169
9.5 Peripherals Settings	169
Chapter 10 System Configuration	173
Chapter 11 Advanced Functionality	174
11.1 Data Replenishment	174
11.2 Integrate by Transmitting Text Protocol	175
Chapter 12 API Reference	177
12.1 General APIs	177
12.1.1 NET_DVR_Cleanup	177
12.1.2 NET_DVR_CloseFormatHandle	177
12.1.3 NET_DVR_FormatDisk	178
12.1.4 NET_DVR_GetDeviceAbility	178
12.1.5 NET_DVR_GetDeviceConfig	179
12.1.6 NET_DVR_GetDeviceStatus	180
12.1.7 NET_DVR_GetDVRConfig	182
12.1.8 NET_DVR_GetErrorMsg	183
12.1.9 NET_DVR_GetFormatProgress	183
12.1.10 NET_DVR_GetIPCProtoList_V41	184
12.1.11 NET_DVR_GetLastError	185
12.1.12 NET_DVR_GetLastErrorModelCode	185
12.1.13 NET_DVR_GetLinkAddr	186
12.1.14 NET_DVR_GetNextRemoteConfig	186
12.1.15 NET_DVR_GetRemoteConfigState	187
12.1.16 NET_DVR_GetRtspConfig	188
12.1.17 NET_DVR_GetSDKLocalCfg	188
12.1.18 NET_DVR_GetSTDAbility	189
12.1.19 NET_DVR_GetSTDConfig	190
12.1.20 NET_DVR_GetUpnpNatState	191

12.1.21 NET_DVR_Init	191
12.1.22 NET_DVR_InquestGetEncryptState	192
12.1.23 NET_DVR_InquestSetSecretKey	192
12.1.24 NET_DVR_InquestStreamEncrypt	193
12.1.25 NET_DVR_Login_V40	193
12.1.26 NET_DVR_Logout	194
12.1.27 NET_DVR_RebootDVR	195
12.1.28 NET_DVR_RemoteControl	195
12.1.29 NET_DVR_RenderPrivateData	196
12.1.30 PlayM4_RenderPrivateDataEx	196
12.1.31 NET_DVR_SDKChannelToISAPI	197
12.1.32 NET_DVR_SendRemoteConfig	198
12.1.33 NET_DVR_SendWithRecvRemoteConfig	199
12.1.34 NET_DVR_SetConnectTime	200
12.1.35 NET_DVR_SetDeviceConfig	201
12.1.36 NET_DVR_SetDeviceConfigEx	202
12.1.37 NET_DVR_SetDVRConfig	203
12.1.38 NET_DVR_SetExceptionCallBack_V30	204
12.1.39 NET_DVR_SetRtspConfig	206
12.1.40 NET_DVR_SetSDKInitCfg	206
12.1.41 NET_DVR_SetSDKLocalCfg	208
12.1.42 NET_DVR_SetSDKSecretKey	208
12.1.43 NET_DVR_SetSTDConfig	209
12.1.44 NET_DVR_StartGetDevState	210
12.1.45 NET_DVR_StartRemoteConfig	210
12.1.46 NET_DVR_STDControl	211
12.1.47 NET_DVR_STDXMLConfig	212
12.1.48 NET_DVR_StopGetDevState	213

12.1.49 NET_DVR_StopRemoteConfig	214
12.2 Live View and PTZ APIs	214
12.2.1 NET_DVR_ChangeWndResolution	214
12.2.2 NET_DVR_GetCurrentAudioCompress_V50	215
12.2.3 NET_DVR_PTZControl	215
12.2.4 NET_DVR_PTZCruise	218
12.2.5 NET_DVR_PTZPreset	219
12.2.6 NET_DVR_RealPlay_V40	220
12.2.7 NET_DVR_RealPlaySpecial	221
12.2.8 NET_DVR_SetRealDataCallBack	222
12.2.9 NET_DVR_SetStandardDataCallBack	223
12.2.10 NET_DVR_StopRealPlay	224
12.3 Search and Playback APIs	224
12.3.1 NET_DVR_CloseFindPicture	224
12.3.2 NET_DVR_CloseSmartSearchPicture	225
12.3.3 NET_DVR_FindClose_V30	225
12.3.4 NET_DVR_FindFile_V50	226
12.3.5 NET_DVR_FindFileByEvent_V50	226
12.3.6 NET_DVR_FindNextEvent_V50	227
12.3.7 NET_DVR_FindNextFile_V50	228
12.3.8 NET_DVR_FindNextLabel	229
12.3.9 NET_DVR_FindNextPicture_V50	230
12.3.10 NET_DVR_FindNextSmartPicture_V50	231
12.3.11 NET_DVR_FindPicture	232
12.3.12 NET_DVR_FindRecordLabel	232
12.3.13 NET_DVR_GetPicture_V50	233
12.3.14 NET_DVR_PlayBackByName	233
12.3.15 NET_DVR_PlayBackByTime_V50	234

12.3.16 NET_DVR_PlayBackControl_V40	235
12.3.17 NET_DVR_SearchNextInfo	240
12.3.18 NET_DVR_SetPlayDataCallBack_V40	241
12.3.19 NET_DVR_SmartSearch_V40	242
12.3.20 NET_DVR_SmartSearchPicture	242
12.3.21 NET_DVR_StopFindLabel	243
12.3.22 NET_DVR_StopPlayBack	243
12.3.23 NET_DVR_StopSearch	244
12.3.24 NET_DVR_UpdateRecordIndex	244
12.4 File Uploading and Downloading APIs	245
12.4.1 NET_DVR_GetDownloadState	245
12.4.2 NET_DVR_GetFileByName	246
12.4.3 NET_DVR_GetFileByTime_V40	246
12.4.4 NET_DVR_GetUploadResult	247
12.4.5 NET_DVR_GetUploadState	248
12.4.6 NET_DVR_StartDownload	250
12.4.7 NET_DVR_StopDownload	251
12.4.8 NET_DVR_StopGetFile	251
12.4.9 NET_DVR_UploadClose	252
12.4.10 NET_DVR_UploadFile_V40	252
12.4.11 NET_DVR_UploadSend	253
12.5 Alarm APIs	254
12.5.1 NET_DVR_CloseAlarmChan_V30	254
12.5.2 NET_DVR_GetAlarmSubscribe	254
12.5.3 NET_DVR_SetAlarmSubscribe	255
12.5.4 NET_DVR_SetDVRMessageCallBack_V50	255
12.5.5 NET_DVR_SetupAlarmChan_V50	263
12.5.6 NET_DVR_StartListen_V30	264

12.5.7 NET_DVR_StopListen_V30	265
12.6 Two-Way Audio APIs	265
12.6.1 NET_DVR_AddDVR_V30	265
12.6.2 NET_DVR_ClientAudioStart_V30	266
12.6.3 NET_DVR_ClientAudioStop	267
12.6.4 NET_DelDVR_V30	267
12.6.5 NET_DVR_GetCurrentAudioCompress_V50	268
12.6.6 NET_DVR_EncodeMP3Frame	268
12.6.7 NET_DVR_DecodeMP3Frame	269
12.6.8 NET_DVR_InitG711Encoder	269
12.6.9 NET_DVR_EncodeG711Frame	270
12.6.10 NET_DVR_ReleaseG711Encoder	274
12.6.11 NET_DVR_InitG711Decoder	275
12.6.12 NET_DVR_DecodeG711Frame	275
12.6.13 NET_DVR_ReleaseG711Decoder	280
12.6.14 NET_DVR_InitG722Encoder	280
12.6.15 NET_DVR_EncodeG722Frame	281
12.6.16 NET_DVR_ReleaseG722Encoder	285
12.6.17 NET_DVR_InitG722Decoder	285
12.6.18 NET_DVR_DecodeG722Frame	286
12.6.19 NET_DVR_ReleaseG722Decoder	290
12.6.20 NET_DVR_InitG726Encoder	291
12.6.21 NET_DVR_EncodeG726Frame	291
12.6.22 NET_DVR_ReleaseG726Encoder	295
12.6.23 NET_DVR_InitG726Decoder	295
12.6.24 NET_DVR_DecodeG726Frame	296
12.6.25 NET_DVR_ReleaseG726Decoder	299
12.6.26 NET_DVR_SetVoiceDataCallBack	299

12.6.27 NET_DVR_StartVoiceCom_MR_V30	300
12.6.28 NET_DVR_StartVoiceCom_V30	302
12.6.29 NET_DVR_StopVoiceCom	303
12.6.30 NET_DVR_VoiceComSendData	304
12.7 Remote Maintenance APIs	305
12.7.1 NET_DVR_CloseUpgradeHandle	305
12.7.2 NET_DVR_FindDVRLog_V50	305
12.7.3 NET_DVR_FindLogClose_V30	306
12.7.4 NET_DVR_FindNextLog_V50	306
12.7.5 NET_DVR_GetUpgradeProgress	307
12.7.6 NET_DVR_GetUpgradeState	308
12.7.7 NET_DVR_SetLogFile	308
12.7.8 NET_DVR_Upgrade_V50	309
12.8 Callback Function	310
12.8.1 CHAR_ENCODE_CONVERT	310
12.8.2 DEV_WORK_STATE_CB	311
12.8.3 fExceptionCallBack	311
12.8.4 fLoginResultCallBack	313
12.8.5 fPlayDataCallBack_V40	314
12.8.6 fRealDataCallBack	315
12.8.7 fRemoteConfigCallback	316
12.8.8 fStdDataCallBack	317
12.8.9 fVoiceDataCallBack	319
12.8.10 MSGCallBack	319
12.8.11 REALDATACALLBACK	320
Appendix A. Data Structure	322
A.1 DATE_TIME	322
A.2 NET_ALARM_CVR_SUBINFO_UNION	322

A.3 NET_ALARM_RECORD_EXCEPTION	323
A.4 NET_ALARM_RECORDFILE LOSS	323
A.5 NET_ALARM_RESOURCE_USAGE	324
A.6 NET_ALARM_STREAM_EXCEPTION	324
A.7 NET_DVR_ADDRESS	325
A.8 NET_DVR_ADVANCE_COND_UNION	325
A.9 NET_DVR_AES_KEY_INFO	326
A.10 NET_DVR_AGC_PARAM	326
A.11 NET_DVR_ALRAM_FIXED_HEADER	327
A.12 NET_DVR_ALARM_HOT_SPARE	330
A.13 NET_DVR_ALARM_ISAPI_INFO	331
A.14 NET_DVR_ALARM_ISAPI_PICDATA	332
A.15 NET_DVR_ALARM_RS485CFG	332
A.16 NET_DVR_ALARM_SEARCH_COND	334
A.17 NET_DVR_ALARM_SEARCH_RESULT	335
A.18 NET_DVR_ALARMHOST_NETCFG_V50	336
A.19 NET_DVR_ALARMHOST_NETPARAM_V50	337
A.20 NET_DVR_ALARMINCFG_V40	338
A.21 NET_DVR_ALARMINFO_DEV	341
A.22 NET_DVR_ALARMINFO_DEV_V40	341
A.23 NET_DVR_ALARMINFO_V30	342
A.24 NET_DVR_ALARMINFO_V40	343
A.25 NET_DVR_ALARMER	345
A.26 NET_DVR_ALARMOUTCFG_V30	345
A.27 NET_DVR_AREA_SMARTSEARCH_COND_UNION	346
A.28 NET_DVR_ATMFINDINFO	347
A.29 NET_DVR_AUDIOENC_INFO	348
A.30 NET_DVR_AUDIODEC_INFO	348

A.31 NET_DVR_AUDIOENC_PROCESS_PARAM	349
A.32 NET_DVR_AUDIODEC_PROCESS_PARAM	350
A.33 NET_DVR_AUDIOEXCEPTION_ALARM	351
A.34 NET_VCA_AUDIO_ABNORMAL	352
A.35 NET_DVR_AUDIO_CHANNEL	352
A.36 NET_DVR_AUDIO_EXCEPTION	353
A.37 NET_DVR_AUDIO_STEEP_DROP	354
A.38 NET_DVR_AUXILIARY_DEV_UPGRADE_PARAM	354
A.39 NET_DVR_BACKLIGHT	355
A.40 NET_DVR_BEHAVIOR_INFO	356
A.41 NET_DVR_BEHAVIOUR_COND	356
A.42 NET_DVR_BINOC_RECTIFY_PARAM	357
A.43 NET_DVR_BUF_INFO	358
A.44 NET_DVR_BUILTIN_SUPPLEMENTLIGHT	359
A.45 NET_DVR_BV_CORRECT_PARAM	360
A.46 NET_DVR_CAMERAPARAMCFG	361
A.47 NET_DVR_CAMERAPARAMCFG_EX	365
A.48 NET_DVR_CERT_ADDITION_PARAM	372
A.49 NET_DVR_CERT_PARAM	373
A.50 NET_DVR_CETTIFICATE_INFO	375
A.51 NET_DVR_CHANNELSTATE_V30	375
A.52 NET_DVR_CHAN_FILTER_CFG	376
A.53 NET_DVR_CHANNEL_GROUP	378
A.54 NET_DVR_CHECK_DEV_STATE	379
A.55 NET_DVR_CMOSMODECFG	379
A.56 NET_DVR_CMS_PARAM	380
A.57 NET_DVR_COLOR	381
A.58 NET_DVR_COMPRESSION_AUDIO	382

A.59 NET_DVR_COMPRESSION_INFO_V30	383
A.60 NET_DVR_COMPRESSIONCFG_V30	386
A.61 NET_DVR_CONNECT_POS_CFG	387
A.62 NET_DVR_CORRIDOR_MODE_CCD	388
A.63 NET_DVR_CRUISECHAN_INFO	388
A.64 NET_DVR_DAYNIGHT	389
A.65 NET_DVR_DAYTIME	390
A.66 NET_DVR_DEC_DDNS_DEV	391
A.67 NET_DVR_DEFOGCFG	391
A.68 NET_DVR_DEVICECFG_V50	392
A.69 NET_DVR_DEV_DDNS_INFO	395
A.70 NET_DVR_DEC_STREAM_DEV_EX	396
A.71 NET_DVR_DEC_STREAM_MODE	397
A.72 NET_DVR_DEFOCUS_ALARM	398
A.73 NET_DVR_DEVICEINFO_V30	399
A.74 NET_DVR_DEVSERVER_CFG	403
A.75 NET_DVR_DEV_CHAN_INFO_EX	404
A.76 NET_DVR_DIRECTION	406
A.77 NET_DVR_DISKSTATE	406
A.78 NET_DVR_DISK_QUOTA	407
A.79 NET_DVR_DISK_QUOTA_CFG	408
A.80 NET_DVR_DISK_QUOTA_CFG_V60	408
A.81 NET_DVR_DISK_QUOTA_V60	409
A.82 NET_DVR_DDE_PARAM	410
A.83 NET_DVR_DDNS_STREAM_CFG	410
A.84 NET_DVR_DEVICEINFO_V40	412
A.85 NET_DVR_DNMODE	415
A.86 NET_DVR_DPC_PARAM	416

A.87 NET_DVR_ELECTRONICSTABILIZATION	417
A.88 NET_DVR_ETHERNET_V30	418
A.89 NET_DVR_EVENT_CAPTURE_V40	418
A.90 NET_DVR_EXCEPTION_V40	419
A.91 NET_DVR_EXPOSURE	420
A.92 NET_DVR_EZVIZ_ACCESS_CFG	421
A.93 NET_DVR_FACEDETECTION_SEARCHCOND	422
A.94 NET_DVR_FACE_PIC_DATA_INFO	423
A.95 NET_DVR_FACE_SUB_PIC_INFO	424
A.96 NET_DVR_FFC_PARAM	425
A.97 NET_DVR_FILECOND_V50	425
A.98 NET_DVR_FIND_LOG_COND	427
A.99 NET_DVR_FINDDATA_V50	428
A.100 NET_DVR_FIND_LABEL	430
A.101 NET_DVR_FINDLABEL_DATA	431
A.102 NET_DVR_FIND_PICTURE_PARAM	432
A.103 NET_DVR_FIND_PICTURE_V50	438
A.104 NET_DVR_FOCUSMODE_CFG	440
A.105 NET_DVR_GAIN	441
A.106 NET_DVR_GAMMACORRECT	442
A.107 NET_DVR_GENERIC_DATA_CFG	442
A.108 NET_DVR_GENERIC_END	443
A.109 NET_DVR_GENERIC_START	443
A.110 NET_DVR_GET_STREAM_UNION	444
A.111 NET_DVR_HANDLEEXCEPTION_V40	445
A.112 NET_DVR_HANDLEEXCEPTION_V41	446
A.113 NET_DVR_HDCFG_V50	447
A.114 NET_DVR_HIDDEN_INFORMATION_CFG	447

A.115 NET_DVR_HIDEALARM_V40	448
A.116 NET_DVR_HKDDNS_STREAM	449
A.117 NET_DVR_I_FRAME	451
A.118 NET_DVR_IGNORE_STRING	452
A.119 NET_DVR_IN_PARAM	452
A.120 NET_DVR_INIT_CFG_ABILITY	453
A.121 NET_DVR_INQUEST_SECRET_INFO	453
A.122 NET_DVR_INTRUSION_SEARCHCOND	454
A.123 NET_DVR_IPADDR_UNION	455
A.124 NET_DVR_IPCHANINFO	455
A.125 NET_DVR_IPCHANINFO_V40	456
A.126 NET_DVR_IPC_PROTO_LIST_V41	457
A.127 NET_DVR_IPDEVINFO_V31	458
A.128 NET_DVR_IPPARACFG_V40	459
A.129 NET_DVR_IPSERVER_STREAM	460
A.130 NET_DVR_ISP_CAMERAPARAMCFG	461
A.131 NET_DVR_JPEGPARA	463
A.132 NET_DVR_JPEG_CAPTURE_CFG_V40	463
A.133 NET_DVR_LABEL_IDENTIFY	464
A.134 NET_DVR_LASER_PARAM_CFG	464
A.135 NET_DVR_LINK_ADDR	465
A.136 NET_DVR_LOCAL_ABILITY_PARSE_CFG	466
A.137 NET_DVR_LOCAL_ASYNC_CFG	467
A.138 NET_DVR_LOCAL_BYTE_ENCODE_CONVERT	467
A.139 NET_DVR_LOCAL_CERTIFICATION	468
A.140 NET_DVR_LOCAL_CFG_TYPE_PTZ	469
A.141 NET_DVR_LOCAL_CHECK_DEV	469
A.142 NET_DVR_LOCAL_GENERAL_CFG	470

A.143 NET_DVR_LOCAL_LOG_CFG	471
A.144 NET_DVR_LOCAL_MEM_POOL_CFG	471
A.145 NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG	472
A.146 NET_DVR_LOCAL_PORT_MULTI_CFG	473
A.147 NET_DVR_LOCAL_PROTECT_KEY_CFG	473
A.148 NET_DVR_LOCAL_SDK_PATH	474
A.149 NET_DVR_LOCAL_STREAM_CALLBACK_CFG	474
A.150 NET_DVR_LOCAL_TALK_MODE_CFG	474
A.151 NET_DVR_LOCAL_TCP_PORT_BIND_CFG	475
A.152 NET_DVR_LOCAL_UDP_PORT_BIND_CFG	475
A.153 NET_DVR_LOG_V50	476
A.154 NET_DVR_MACFILTER_CFG	478
A.155 NET_DVR_MESSAGE_CALLBACK_PARAM_V51	479
A.156 NET_DVR_MIME_UNIT	479
A.157 NET_DVR_MONITOR_INFO	480
A.158 NET_DVR_MOTION_MODE_PARAM	481
A.159 NET_DVR_MOTION_MULTI_AREA	481
A.160 NET_DVR_MOTION_MULTI_AREAPARAM	482
A.161 NET_DVR_MOTION_SINGLE_AREA	483
A.162 NET_DVR_MOTION_V40	484
A.163 NET_DVR_MRД_SEARCH_PARAM	486
A.164 NET_DVR_MRД_SEARCH_RESULT	487
A.165 NET_DVR_MULTI_STREAM_COMPRESSIONCFG	487
A.166 NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND	488
A.167 NET_DVR_NAT_CFG	489
A.168 NET_DVR_NAT_PORT	490
A.169 NET_DVR_NETCFG_V50	491
A.170 NET_DVR_NET_RECEIVE	492

A.171 NET_DVR_NET_SNIFF	493
A.172 NET_DVR_NOISEREMOVE	494
A.173 NET_DVR_OBJECT_FEATURE	495
A.174 NET_DVR_ONLINEUPGRADE_SERVER	495
A.175 NET_DVR_ONLINEUPGRADE_VERSION_COND	496
A.176 NET_DVR_ONLINEUPGRADE_VERSION_RET	497
A.177 NET_DVR_OPTICAL_DEHAZE	497
A.178 NET_DVR_OUT_PARAM	498
A.179 NET_DVR_PHONECFG_V50	498
A.180 NET_DVR_PHY_DISK_INFO	499
A.181 NET_DVR_PICCFG_V40	500
A.182 NET_ITS_PICTURE_INFO	503
A.183 NET_DVR_PIC_FEATURE_UNION	505
A.184 NET_DVR_PIC_PARAM	506
A.185 NET_DVR_PIRIS_PARAM	507
A.186 NET_DVR_PLAYCOND	507
A.187 NET_DVR_POE_CFG	508
A.188 NET_DVR_POS_AVE	508
A.189 NET_DVR_POS_CONNECTMODE_UNION	509
A.190 NET_DVR_POS_FILTER_CFG	509
A.191 NET_DVR_POS_GENERIC	510
A.192 NET_DVR_POS_HIDDEN_INFORMATION	511
A.193 NET_DVR_POS_NUCLEUS	511
A.194 NET_DVR_POS_OSD_REGION	512
A.195 NET_DVR_POS_PROTOCOL_UNION	512
A.196 NET_DVR_POSINFO_OVERLAY	513
A.197 NET_DVR_PPPOECFG	514
A.198 NET_DVR_PRESETCHAN_INFO	514

A.199 NET_DVR_PRESET_NAME	515
A.200 NET_DVR_PREVIEWINFO	516
A.201 NET_DVR_PREVIEWINFO_SPECIAL	519
A.202 NET_DVR_PROTO_TYPE	520
A.203 NET_DVR_RTSPCFG	523
A.204 NET_DVR_RTSP_PARAMS_CFG	524
A.205 NET_DVR_PTZTRACKCHAN_INFO	524
A.206 NET_DVR_PTZPOS	524
A.207 NET_DVR_PTZPOS_INFO	525
A.208 NET_DVR_PTZ_POSITION	526
A.209 NET_DVR_PU_STREAM_CFG_V41	527
A.210 NET_DVR_PU_STREAM_URL	528
A.211 NET_DVR_REL_CAPTURE_CHAN_V40	528
A.212 NET_DVR_RGB_COLOR	529
A.213 NET_DVR_RECORD_PASSBACK_MANUAL_COND	530
A.214 NET_DVR_RECORD_PASSBACK_MANUAL_TASK_RET	530
A.215 NET_DVR_SCENE_INFO	531
A.216 NET_DVR_SCENECHANGE_DETECTION_RESULT	532
A.217 NET_DVR_SCHEDTIME	532
A.218 NET_DVR_SCHEDULE_DAYTIME	533
A.219 NET_DVR_SEARCH_EVENT_PARAM_V50	533
A.220 NET_DVR_SEARCH_EVENT_RET_V50	544
A.221 NET_DVR_SECURITY_CFG_FILE_COND	553
A.222 NET_DVR_SEND_PARAM_IN	554
A.223 NET_DVR_SETUPALARMS_PARAM	555
A.224 NET_DVR_SETUPALARMS_PARAM_V50	557
A.225 NET_DVR_SHELTER	560
A.226 NET_DVR_SHOWSTRINGINFO	561

A.227 NET_DVR_SHOWSTRING_V30	562
A.228 NET_DVR_SIMXML_LOGIN	562
A.229 NET_DVR_SINGLE_HD_V50	563
A.230 NET_DVR_SINGLE_RS232	565
A.231 NET_DVR_SIP_CFG_V50	566
A.232 NET_DVR_SMARTIR_PARAM	569
A.233 NET_DVR_SMARTSEARCH_PIC_UNION	569
A.234 NET_DVR_SMART_SEARCH_PARAM_V40	570
A.235 NET_DVR_SMART_SEARCH_PIC_PARA	572
A.236 NET_DVR_SMART_SEARCH_PIC_RET_V50	573
A.237 NET_DVR_SMART_SEARCH_RET	575
A.238 NET_DVR_SMSRELATIVEPARAM_V50	576
A.239 NET_DVR_SNAP_CAMERAPARAMCFG	576
A.240 NET_DVR_SPECIAL_FINDINFO_UNION	578
A.241 NET_DVR_STD_ABILITY	578
A.242 NET_DVR_STD_CONFIG	579
A.243 NET_DVR_STD_CONTROL	580
A.244 NET_DVR_STOP_LINE_PARAM	581
A.245 NET_DVR_STREAM_INFO	581
A.246 NET_DVR_STREAMING_COND	582
A.247 NET_DVR_STREAM_MEDIA_SERVER	583
A.248 NET_DVR_STREAM_MODE	583
A.249 NET_DVR_THERMOMETRY_AGC	584
A.250 NET_DVR_TIME	585
A.251 NET_DVR_TIME_EX	585
A.252 NET_DVR_TIME_SEARCH	586
A.253 NET_DVR_TIME_SEARCH_COND	587
A.254 NET_DVR_TIME_V30	588

A.255 NET_DVR_TIME_V50	588
A.256 NET_DVR_TIMING_CAPTURE	590
A.257 NET_DVR_TRAVERSE_PLANE_SEARCHCOND	590
A.258 NET_DVR_UPGRADE_PARAM	591
A.259 NET_DVR_UPLOAD_RECORD_INFO	593
A.260 NET_DVR_UPNP_NAT_STATE	595
A.261 NET_DVR_UPNP_PORT_STATE	595
A.262 NET_DVR_USB_RS232	596
A.263 NET_DVR_USER_LOGIN_INFO	597
A.264 NET_DVR_USER_INFO_V52	598
A.265 NET_DVR_USER_V52	602
A.266 NET_DVR_VCA_VERSION	602
A.267 NET_DVR_VCA_VERSION_LIST	603
A.268 NET_DVR_VEHICLE_PARA	604
A.269 NET_DVR_VICOLOR	605
A.270 NET_DVR_VIDEOEFFECT	605
A.271 NET_DVR_VILOST_V40	606
A.272 NET_DVR_VOD_DRAWFRAME_PARA	607
A.273 NET_DVR_VOD_PARA_V50	608
A.274 NET_DVR_WDR	610
A.275 NET_DVR_WHITEBALANCE	611
A.276 NET_DVR_WIRELESSDIAL_CFG	612
A.277 NET_DVR_WIRELESSDIAL_CONNECT_PARAM	614
A.278 NET_DVR_WIRELESSDIAL_SCHEDULE	614
A.279 NET_DVR_WIRELESSDIAL_STATUS	615
A.280 NET_DVR_WIRELESSSERVER	617
A.281 NET_DVR_WORKSTATE_V40	618
A.282 NET_DVR_XML_CONFIG_INPUT	619

A.283 NET_DVR_XML_CONFIG_OUTPUT	619
A.284 NET_IPC_AUXALARM_RESULT	620
A.285 NET_IPC_AUXALARM_UPLOAD_UNION	621
A.286 NET_IPC_CALLHELP_ALARMCFG	622
A.287 NET_IPC_PIR_ALARMCFG	623
A.288 NET_IPC_SINGLE_WIRELESS_ALARMCFG	623
A.289 NET_PTZ_INFO_EX	624
A.290 NET_VCA_AUDIO_ABNORMAL	625
A.291 NET_VCA_DEV_INFO	626
A.292 NET_VCA_MASK_REGION	627
A.293 NET_VCA_MASK_REGION_LIST	627
A.294 NET_VCA_POINT	628
A.295 NET_VCA_POLYGON	628
A.296 NET_VCA_RECT	628
A.297 STREAM_FRAME_INFO_S	629
A.298 STREAM_FS_SUPPLE_INFO_S	630
A.299 STREAM_RT_DATA_INFO_S	630
Appendix B. Enumeration	632
B.1 BITRATE_ENCODE_INDEX	632
B.2 COUNTRY_INDEX	633
B.3 CR_INDEX	654
B.4 FIRE_ALARM	674
B.5 NET_DVR_CHARSET_ENUM	674
B.6 NET_SDK_CALLBACK_STATUS_NORMAL	676
B.7 NET_DVR_LINK_KIND	676
B.8 NET_SDK_LOCAL_CFG_TYPE	677
B.9 NET_SDK_DOWNLOAD_TYPE	679
B.10 NET_SDK_UPLOAD_TYPE	683

B.11 PRIDATA_RENDER	686
B.12 STREAM_TYPE_E	687
B.13 TEM_FLAG	687
B.14 VCA_PLATE_COLOR	688
B.15 VLR_VEHICLE_CLASS	689
Appendix C. HCNetSDK Log Types	694
Appendix D. Device Network SDK Errors	730
Appendix E. Request URIs	774
E.1 /ISAPI/AUXInfo/attributes/Channels	776
E.2 /ISAPI/AUXInfo/attributes/Channels/<ID>	776
E.3 /ISAPI/ContentMgmt/capabilities	777
E.4 /ISAPI/ContentMgmt/download	777
E.5 /ISAPI/ContentMgmt/download/capabilities	778
E.6 /ISAPI/ContentMgmt/InputProxy/channels/<ID>	778
E.7 /ISAPI/ContentMgmt/logSearch?format=json	779
E.8 /ISAPI/ContentMgmt/record/control/locks/capabilities	779
E.9 /ISAPI/ContentMgmt/record/control/locks/name	780
E.10 /ISAPI/ContentMgmt/record/control/remark/capabilities?format=json	781
E.11 /ISAPI/ContentMgmt/record/control/remark?format=json	781
E.12 /ISAPI/ContentMgmt/record/tracks	781
E.13 /ISAPI/ContentMgmt/record/tracks/<ID>	782
E.14 /ISAPI/ContentMgmt/record/tracks/<ID>/capabilities	783
E.15 /ISAPI/ContentMgmt/search	784
E.16 /ISAPI/ContentMgmt/search/profile	786
E.17 /ISAPI/ContentMgmt/storage	787
E.18 /ISAPI/ContentMgmt/Storage/capabilities	787
E.19 /ISAPI/ContentMgmt/Storage/dataReconstruction/progress?format=json	788
E.20 /ISAPI/ContentMgmt/Storage/dataReconstruction?format=json	788

E.21 /ISAPI/ContentMgmt/Storage/extension	789
E.22 /ISAPI/ContentMgmt/Storage/extension/capabilities	790
E.23 /ISAPI/ContentMgmt/Storage/hdd	790
E.24 /ISAPI/ContentMgmt/Storage/hdd/<ID>	790
E.25 /ISAPI/ContentMgmt/Storage/hdd/<ID>/encryptFormat?format=json	791
E.26 /ISAPI/ContentMgmt/Storage/hdd/<ID>/encryptVerfy?format=json	792
E.27 /ISAPI/ContentMgmt/Storage/hdd/<ID>/logProTest?format=json	792
E.28 /ISAPI/ContentMgmt/Storage/hdd/<ID>/syncStatus?format=json	793
E.29 /ISAPI/ContentMgmt/Storage/hdd/capabilities	793
E.30 /ISAPI/ContentMgmt/storage/hdd/clearingSpace/capabilities?format=json	794
E.31 /ISAPI/ContentMgmt/storage/hdd/clearingSpace?format=json	794
E.32 /ISAPI/ContentMgmt/Storage/hdd/specifyHddFormat?format=json	795
E.33 /ISAPI/ContentMgmt/Storage/ssd	796
E.34 /ISAPI/ContentMgmt/Storage/ssd/<ID>	796
E.35 /ISAPI/ContentMgmt/Storage/ssd/<ID>/capabilities?format=json	796
E.36 /ISAPI/ContentMgmt/Storage/ssd/<ID>/SMARTTest/start	797
E.37 /ISAPI/ContentMgmt/Storage/ssd/<ID>/SMARTTest/status	798
E.38 /ISAPI/ContentMgmt/Storage/ssd/<ID>/upgrade/status?format=json	798
E.39 /ISAPI/ContentMgmt/Storage/ssd/<ID>/upgrade?format=json	799
E.40 /ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/capabilities	799
E.41 /ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/PictureByUrl/capabilities?format=json	800
E.42 /ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/PictureByUrl?format=json	800
E.43 /ISAPI/ContentMgmt/time/search/capabilities?format=json	801
E.44 /ISAPI/ContentMgmt/time/search?format=json	801
E.45 /ISAPI/Event/capabilities	802
E.46 /ISAPI/Event/channels/<ID>/capabilities	802
E.47 /ISAPI/Event/channels/capabilities	803

E.48 /ISAPI/Event/IOT/channels/<ID>/capabilities?format=json	803
E.49 /ISAPI/Event/notification/subscribeEventCap	804
E.50 /ISAPI/Event/schedules/vibrationDetection/<ID>	804
E.51 /ISAPI/Event/schedules/<EventType>/<ID>	805
E.52 /ISAPI/Event/triggers/hdBadBlock	806
E.53 /ISAPI/Event/triggers/hdImpact	806
E.54 /ISAPI/Event/triggers/highHDTemperature	807
E.55 /ISAPI/Event/triggers/lowHDTemperature	808
E.56 /ISAPI/Event/triggers/severeHDFailure	808
E.57 /ISAPI/Event/triggers/<eventType>-<channelID>	809
E.58 /ISAPI/Event/triggers/<eventType>-<channelID>/notifications	810
E.59 /ISAPI/Event/triggers/<ID>/zone/<ID>	811
E.60 /ISAPI/Event/triggers/notifications/AudioAlarm/capabilities?format=json	812
E.61 /ISAPI/Event/triggers/notifications/AudioAlarm?format=json	812
E.62 /ISAPI/Event/triggers/notifications/AudioAlarm/customAudio?format=json	813
E.63 /ISAPI/Event/triggers/notifications/AudioAlarm/customAudioInfo?format=json	814
E.64 /ISAPI/Event/triggers/notifications/AudioAlarm/customAudioTypeInfo/channels?format=json	814
E.65 /ISAPI/Event/triggers/notifications/whiteLightAlarm/capabilities?format=json	815
E.66 /ISAPI/Event/triggers/notifications/whiteLightAlarm?format=json	815
E.67 /ISAPI/Event/triggersCap	816
E.68 /ISAPI/Event/triggersCap/IOT	816
E.69 /ISAPI/Image/channels/<ID>	817
E.70 /ISAPI/Image/channels/<ID>/capabilities	818
E.71 /ISAPI/Image/channels/<ID>/color	818
E.72 /ISAPI/Image/channels/<ID>/color/capabilities	819
E.73 /ISAPI/Image/channels/<ID>/EPTZ	819
E.74 /ISAPI/Image/channels/<ID>/EPTZ mode/capabilities?format=json	820

E.75 /ISAPI/Image/channels/<ID>/EPTZ mode?format=json	820
E.76 /ISAPI/Image/channels/<ID>/focusConfiguration	821
E.77 /ISAPI/Image/channels/<ID>/focusConfiguration/capabilities	822
E.78 /ISAPI/Image/channels/<ID>/focusConfiguration/status?format=json	822
E.79 /ISAPI/Image/channels/<ID>/lensDistortionCorrection	823
E.80 /ISAPI/Image/channels/<ID>/lensDistortionCorrection/capabilities	823
E.81 /ISAPI/Image/channels/<ID>/mountingScenario	824
E.82 /ISAPI/Image/channels/<ID>/mountingScenario/capabilities	825
E.83 /ISAPI/Image/channels/<ID>/supplementLight	825
E.84 /ISAPI/Image/channels/<ID>/supplementLight/capabilities	826
E.85 /ISAPI/Image/channels/<ID>/targetEnhancement	826
E.86 /ISAPI/Image/channels/<ID>/targetEnhancement/capabilities	827
E.87 /ISAPI/Image/channels/<ID>/tempRange	828
E.88 /ISAPI/Image/channels/<ID>/tempRange/capabilities	828
E.89 /ISAPI/Intelligent/channels/<ID>/Shield/EventType	829
E.90 /ISAPI/MasterSlaveTracking/capabilities	830
E.91 /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/<ID>	830
E.92 /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/<ID>/tracking	831
E.93 /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/<ID>/trackingRatio	832
E.94 /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/capabilities	832
E.95 /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCameraStatus	833
E.96 /ISAPI/MasterSlaveTracking/channels/<ID>/tracking/capabilities	833
E.97 /ISAPI/MasterSlaveTracking/linkedTracking/capabilities?format=json	833
E.98 /ISAPI/MasterSlaveTracking/linkedTracking?format=json	834
E.99 /ISAPI/PTZCtrl/channels/<ID>	835
E.100 /ISAPI/PTZCtrl/channels/<ID>/auxcontrols/<ID>	835
E.101 /ISAPI/PTZCtrl/channels/<ID>/capabilities	836
E.102 /ISAPI/PTZCtrl/channels/<ID>/EagleFocusing/auto/capabilities?format=json	837

E.103 /ISAPI/PTZCtrl/channels/<ID>/EagleFocusing/auto?format=json	837
E.104 /ISAPI/PTZCtrl/channels/<ID>/EagleFocusing/capabilities	838
E.105 /ISAPI/PTZCtrl/channels/<ID>/GPS?format=json	838
E.106 /ISAPI/PTZCtrl/channels/<ID>/gyroVerify/capabilities?format=json	839
E.107 /ISAPI/PTZCtrl/channels/<ID>/gyroVerify?format=json	839
E.108 /ISAPI/PTZCtrl/channels/<ID>/maxelevation	840
E.109 /ISAPI/PTZCtrl/channels/<ID>/maxelevation/capabilities	840
E.110 /ISAPI/PTZCtrl/channels/<ID>/moveAutoTracking?format=json	841
E.111 /ISAPI/PTZCtrl/channels/<ID>/PTZOSDDisplay	841
E.112 /ISAPI/PTZCtrl/channels/<ID>/zoomFocus	842
E.113 /ISAPI/SDT/Management/capabilities?format=json	843
E.114 /ISAPI/SDT/Management/EventSearch?format=json	843
E.115 /ISAPI/SDT/Management/EventSearch/capabilities?format=json	844
E.116 /ISAPI/SDT/Management/IntelligentSearch/capabilities?format=json	844
E.117 /ISAPI/SDT/Management/IntelligentSearch?format=json	845
E.118 /ISAPI/Security/adminAccesses	845
E.119 /ISAPI/Security/adminAccesses/capabilities	846
E.120 /ISAPI/Security/adminAccesses/<ID>	846
E.121 /ISAPI/Security/capabilities	847
E.122 /ISAPI/Security/CloudUserManage/users/<ID>?format=json	847
E.123 /ISAPI/Security/CloudUserManage/users/byType?format=json	848
E.124 /ISAPI/Security/CloudUserManage/users/capabilities?format=json	849
E.125 /ISAPI/Security/CloudUserManage/users?format=json	849
E.126 /ISAPI/Security/deviceCertificate/certificateRevocation/capabilities?format=json	850
E.127 /ISAPI/Security/deviceCertificate/certificateRevocation?format=json	851
E.128 /ISAPI/Security/doubleVerification/users/<ID>?format=json	851
E.129 /ISAPI/Security/doubleVerification/users/capabilities?format=json	853
E.130 /ISAPI/Security/doubleVerification/users?format=json	853

E.131 /ISAPI/Security/doubleVerification/UsersPermission/<ID>?format=json	854
E.132 /ISAPI/Security/doubleVerification/UsersPermission/capabilities?format=json	855
E.133 /ISAPI/Security/doubleVerification?format=json	855
E.134 /ISAPI/Security/email/parameter/capabilities?format=json	856
E.135 /ISAPI/Security/email/parameter?format=json	857
E.136 /ISAPI/Security/email/qrCode?format=json	858
E.137 /ISAPI/Security/emailCertification?format=json	858
E.138 /ISAPI/Security/extern/capabilities	859
E.139 /ISAPI/Security/illegalLoginLock	859
E.140 /ISAPI/Security/questionConfiguration	860
E.141 /ISAPI/Security/RTSPCertificate	861
E.142 /ISAPI/Security/RTSPCertificate/capabilities	861
E.143 /ISAPI/Security/UserPermission	862
E.144 /ISAPI/Security/UserPermission/adminCap	862
E.145 /ISAPI/Security/UserPermission/capabilities	863
E.146 /ISAPI/Security/UserPermission/<ID>	863
E.147 /ISAPI/Security/users	864
E.148 /ISAPI/Security/users/<ID>	868
E.149 /ISAPI/Security/users/<ID>/capabilities	869
E.150 /ISAPI/Security/webCertificate	870
E.151 /ISAPI/Streaming/channels/<ID>	871
E.152 /ISAPI/Streaming/channels/<ID>/capabilities	872
E.153 /ISAPI/Streaming/channels/<ID>/smartOverlap/capabilities?format=json	874
E.154 /ISAPI/Streaming/channels/<ID>/smartOverlap?format=json	875
E.155 /ISAPI/System/accessDevice/associatedChannel?format=json	876
E.156 /ISAPI/System/algorithmsVersion	876
E.157 /ISAPI/System/autoMaintenance/capabilities?format=json	877
E.158 /ISAPI/System/autoMaintenance?format=json	877

E.159 /ISAPI/System/autoSwitch/capabilities?format=json	878
E.160 /ISAPI/System/autoSwitch?format=json	878
E.161 /ISAPI/System/BVCorrect/channels/<ID>/capabilities?format=json	879
E.162 /ISAPI/System/BVCorrect/channels/<ID>?format=json	879
E.163 /ISAPI/System/capabilities	880
E.164 /ISAPI/System/channels/<ID>/vibrationDetection/capabilities?format=json	880
E.165 /ISAPI/System/channels/<ID>/vibrationDetection?format=json	881
E.166 /ISAPI/System/deviceInfo	882
E.167 /ISAPI/System/deviceInfo/capabilities	882
E.168 /ISAPI/System/deviceInfo/characteristicCode?format=json	883
E.169 /ISAPI/System/DeviceLanguage	883
E.170 /ISAPI/System/DeviceLanguage/capabilities	884
E.171 /ISAPI/System/diagnosedData/parameter	884
E.172 /ISAPI/System/diagnosedData/parameter/capabilities	885
E.173 /ISAPI/System/diagnosedData/server?format=json	886
E.174 /ISAPI/System/diagnosedData/server/capabilities?format=json	887
E.175 /ISAPI/System/diagnosedData/server/test?format=json	887
E.176 /ISAPI/System/externalDevice/capabilities	888
E.177 /ISAPI/System/factoryReset?mode=	888
E.178 /ISAPI/System/GPSCalibration/channels/<ID>/capabilities?format=json	889
E.179 /ISAPI/System/GPSCalibration/channels/<ID>/points/<PID>?format=json	889
E.180 /ISAPI/System/GPSCalibration/channels/<ID>/points?format=json	890
E.181 /ISAPI/System/GPSPTZCalibration/channels/<ID>/calibration?format=json	892
E.182 /ISAPI/System/GPSVerification/channels/<ID>/display?format=json	892
E.183 /ISAPI/System/GPSVerification/channels/<ID>/points/<ID>?format=json	893
E.184 /ISAPI/System/GPSVerification/channels/<ID>/points/capabilities?format=json	894
E.185 /ISAPI/System/GPSVerification/channels/<ID>/points/resultInfo?format=json	894
E.186 /ISAPI/System/GPSVerification/channels/<ID>/points?format=json	895

E.187 /ISAPI/System/guideConfig/<guideEvent>/capabilities?format=json	896
E.188 /ISAPI/System/guideConfig/<guideEvent>?format=json	897
E.189 /ISAPI/System/installationAngleCalibration/channels/<ID>/capabilities?format=json .	897
E.190 /ISAPI/System/installationAngleCalibration/channels/<ID>?format=json	898
E.191 /ISAPI/System/IOT/channelConfig?format=json	899
E.192 /ISAPI/System/IOT/channels/<ID>/<EventType>/capabilities?format=json	900
E.193 /ISAPI/System/IOT/channels/<ID>/<EventType>?format=json	900
E.194 /ISAPI/System/IOT/channels/<ID>/alarmStatistics?format=json	901
E.195 /ISAPI/System/IOT/channels/<ID>/all?format=json	901
E.196 /ISAPI/System/IOT/channels/<ID>/basicParam?format=json	902
E.197 /ISAPI/System/IOT/channels/<ID>/OSD/capabilities?format=json	902
E.198 /ISAPI/System/IOT/channels/<ID>/OSD?format=json	903
E.199 /ISAPI/System/IOT/channels/<ID>/status?format=json	903
E.200 /ISAPI/System/IOT/channels/<ID>?format=json	904
E.201 /ISAPI/System/IOT/channels/status?format=json	905
E.202 /ISAPI/System/IOT/channels/status?format=json&deviceInductiveType=	906
E.203 /ISAPI/System/IOT/channels?format=json	907
E.204 /ISAPI/System/IOT/channels?format=json&deviceInductiveType=	908
E.205 /ISAPI/System/IOT/linkageChannels?format=json	908
E.206 /ISAPI/System/IOT/search?format=json	909
E.207 /ISAPI/System/IOT/sourceCapabilities?format=json	909
E.208 /ISAPI/System/IOT/sourceSupport/capabilities?format=json	910
E.209 /ISAPI/System/laserOpticalAxis/capabilities?format=json	910
E.210 /ISAPI/System/laserOpticalAxis/goto?format=json	911
E.211 /ISAPI/System/laserOpticalAxis?format=json	911
E.212 /ISAPI/System/logServer	912
E.213 /ISAPI/System/logServer/capabilities	913
E.214 /ISAPI/System/Network/capabilities	913

E.215 /ISAPI/System/Network/Ehome/<ISUPID>	913
E.216 /ISAPI/System/Network/Ehome/capabilities	914
E.217 /ISAPI/System/Network/EhomeList?centerID=	915
E.218 /ISAPI/System/Network/extension	915
E.219 /ISAPI/System/Network/EZVIZ	916
E.220 /ISAPI/System/Network/EZVIZ/capabilities	917
E.221 /ISAPI/System/Network/EZVIZ/QRCode	917
E.222 /ISAPI/System/Network/EZVIZ/QRCode?format=json	918
E.223 /ISAPI/System/Network/EZVIZ/secretKey?format=json	919
E.224 /ISAPI/System/Network/ftp/<ID>	919
E.225 /ISAPI/System/Network/ftp/capabilities	920
E.226 /ISAPI/System/Network/Integrate	920
E.227 /ISAPI/System/Network/Integrate/capabilities	921
E.228 /ISAPI/System/Network/ipFilter	922
E.229 /ISAPI/System/Network/ipFilter/capabilities	922
E.230 /ISAPI/System/Network/resourceStatistics?format=json	923
E.231 /ISAPI/System/Network/SNMP	923
E.232 /ISAPI/System/Network/SNMP/advanced	924
E.233 /ISAPI/System/Network/SNMP/advanced/trapReceivers	925
E.234 /ISAPI/System/Network/SNMP/advanced/trapReceivers/<ID>	926
E.235 /ISAPI/System/Network/socketIP	927
E.236 /ISAPI/System/Network/verificationCodeCheck	927
E.237 /ISAPI/System/Network/WirelessDial/Interfaces	927
E.238 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>	928
E.239 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial	929
E.240 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial/capabilities	930
E.241 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive/capabilities? format=json	930

E.242 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive/test?format=json ...	931
E.243 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive?format=json	932
E.244 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/schedule	932
E.245 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor/capabilities?format=json	933
E.246 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor/status?format=json	933
E.247 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor?format=json	934
E.248 /ISAPI/System/onlineUpgrade/deviceParameter?format=json	935
E.249 /ISAPI/System/onlineUpgrade/downloadPackage/pause?format=json	935
E.250 /ISAPI/System/onlineUpgrade/downloadPackage/resume?format=json	936
E.251 /ISAPI/System/onlineUpgrade/downloadPackage/status?format=json	936
E.252 /ISAPI/System/onlineUpgrade/downloadPackage?format=json	936
E.253 /ISAPI/System/onlineUpgrade/status	937
E.254 /ISAPI/System/onlineUpgrade/upgradeWithoutDownload?format=json	937
E.255 /ISAPI/System/Serial/authentication/capabilities?format=json	938
E.256 /ISAPI/System/Serial/authentication?format=json	938
E.257 /ISAPI/System/Serial/capabilities	939
E.258 /ISAPI/System/Software/channels/<ID>	939
E.259 /ISAPI/System/Software/channels/<ID>/capabilities	940
E.260 /ISAPI/System/SSDFileSystem/capacity?format=json	941
E.261 /ISAPI/System/SSDFileSystem/format/status?format=json	941
E.262 /ISAPI/System/SSDFileSystem/format?format=json	942
E.263 /ISAPI/System/SSDFileSystem/upgrade/status?format=json	942
E.264 /ISAPI/System/SSDFileSystem/upgrade?format=json	942
E.265 /ISAPI/System/status	943
E.266 /ISAPI/System/time/capabilities	944
E.267 /ISAPI/System/TwoWayAudio/channels	944
E.268 /ISAPI/System/TwoWayAudio/channels/<ID>	945

E.269 /ISAPI/System/unitConfig/capabilities?format=json	946
E.270 /ISAPI/System/unitConfig?format=json	946
E.271 /ISAPI/System/usb?format=json	947
E.272 /ISAPI/System/usb/capabilities?format=json	947
E.273 /ISAPI/System/Video/capabilities	948
E.274 /ISAPI/System/Video/inputs/channels/<ID>/privacyMask/regions/<ID>	948
E.275 /ISAPI/System/Video/inputs/channels/<ID>/VCAResource	949
E.276 /ISAPI/System/Video/inputs/channels/<ID>/VCAResource/capabilities	951
E.277 /ISAPI/System/Video/inputs/channels/counting/collection/capabilities?format=json .	951
E.278 /ISAPI/System/Video/inputs/channels/counting/collection?format=json	951
E.279 /ISAPI/System/Video/inputs/channels/heatMap/collection/capabilities?format=json	952
E.280 /ISAPI/System/Video/inputs/channels/heatMap/collection?format=json	952
E.281 /ISAPI/System/Video/inputs/OSDLanguage	953
E.282 /ISAPI/System/workingstatus?format=json	953
E.283 /ISAPI/System/zeroBiasCalibration/channels/<ID>?format=json	954
E.284 /ISAPI/Thermal/capabilities	954
E.285 /ISAPI/Thermal/temperature/collection/capabilities?format=json	955
E.286 /ISAPI/Thermal/temperature/collection?format=json	955
E.287 /ISAPI/Traffic/ContentMgmt/InputProxy/channels/<ID>/ocrScene/capabilities	956
Appendix F. Request and Response Messages	957
F.1 JSON_AlarmStatistics	957
F.2 JSON_AssociatedChannelList	957
F.3 JSON_AudioAlarm	957
F.4 JSON_AudioAlarmCap	959
F.5 JSON_Authentication	961
F.6 JSON_AuthenticationCap	962
F.7 JSON_AutoEagleFocusing	962

F.8 JSON_AutoEagleFocusingCap	963
F.9 JSON_AutoMaintenance	964
F.10 JSON_AutoMaintenanceCap	964
F.11 JSON_AutoSwitch	965
F.12 JSON_AutoSwitchCap	965
F.13 JSON_BasicParam	965
F.14 JSON_BVCorrectParam	965
F.15 JSON_BVCorrectParamCap	967
F.16 JSON_CalibrationStatus	971
F.17 JSON_Cap_CalibrationStatus	971
F.18 JSON_Cap_CaputreDescription	971
F.19 JSON_Cap_ClearingSpaceConfig	972
F.20 JSON_Cap_IntelliManagement	972
F.21 JSON_Cap_RemarkList	975
F.22 JSON_Cap_SSD	976
F.23 JSON_Cap_TimeSearchCond	976
F.24 JSON_CaputreDescription	977
F.25 JSON_CaputreResult	978
F.26 JSON_CertificateRevocation	978
F.27 JSON_CertificateRevocationCap	978
F.28 JSON_ChangedStatus	979
F.29 JSON_Characteristic	980
F.30 JSON_ClearingSpaceConfig	980
F.31 JSON_CloudUserManage	981
F.32 JSON_CloudUserManageCap	982
F.33 JSON_CloudUserManageList	986
F.34 JSON_CustomAudioID	986
F.35 JSON_CustomAudioTypeInfoList	987

F.36 JSON_DataReconstruction	987
F.37 JSON_DiagnosedDataServerCap	987
F.38 JSON_DiagnosedDataServerList	989
F.39 JSON_DoubleVerification	990
F.40 JSON_DownloadPackageStatus	990
F.41 JSON_EmailCertification	991
F.42 JSON_EncryptFormat	991
F.43 JSON_EncryptVerfy	991
F.44 JSON_EPTZMode	991
F.45 JSON_EPTZModeCap	993
F.46 JSON_EventNotificationAlert_Alarm/EventInfo	995
F.47 JSON_EventNotificationAlert_CertificateExpiryAlarmMsg	995
F.48 JSON_EventNotificationAlert_HDDBadSectorEventMsg	996
F.49 JSON_EventNotificationAlert_HDDHighTemperatureEventMsg	997
F.50 JSON_EventNotificationAlert_HDDImpactEventMsg	998
F.51 JSON_EventNotificationAlert_HDDLowTemperatureEventMsg	999
F.52 JSON_EventNotificationAlert_HDDSevereFaultEventMsg	999
F.53 JSON_EventNotificationAlert_NetworkTrafficPrealarmMsg	1000
F.54 JSON_EventNotificationAlert_vibrationDetection	1001
F.55 JSON_EventSearchCap	1002
F.56 JSON_EventSearchCond	1004
F.57 JSON_EventSearchResult	1006
F.58 JSON_EZVIZSecretKey	1008
F.59 JSON_EZVIZQRCode	1008
F.60 JSON_FocusStatus	1009
F.61 JSON_GetCustomAudio	1009
F.62 JSON_GetUserInfoByType	1009
F.63 JSON_GPSCalibration	1010

F.64 JSON_GPSCalibrationCap	1010
F.65 JSON_GPSCalibrationList	1011
F.66 JSON_GPSCalibrationStatus	1012
F.67 JSON_GPSInfo	1012
F.68 JSON_GPSVerification_ResultInfo	1012
F.69 JSON_GPSVerification_ResultInfoList	1013
F.70 JSON_GyroVerify	1013
F.71 JSON_GyroVerifyCap	1013
F.72 JSON_GuideConfig	1014
F.73 JSON_GuideConfigCap	1014
F.74 JSON_HddFormatList	1016
F.75 JSON_HeatMap_CollectionDescription	1016
F.76 JSON_HeatMap_CollectionDescriptionCap	1017
F.77 JSON_HeatMap_CollectionResult	1018
F.78 JSON_id	1019
F.79 JSON_ImportCustomAudio	1019
F.80 JSON_IntelligentSearchCap	1020
F.81 JSON_IntelligentSearchCondition	1025
F.82 JSON_IntelligentSearchResult	1029
F.83 JSON_IOT_ChannelInfo	1036
F.84 JSON_IOT_ChannelInfoList	1036
F.85 JSON_IOT_ErrorList	1037
F.86 JSON_IOTChannel	1037
F.87 JSON_IOTChannelEventCap	1039
F.88 JSON_IOTChannelList	1041
F.89 JSON_IOTChannelStatus	1041
F.90 JSON_IOTChannelStatusList	1042
F.91 JSON_IOTSourceDescription	1042

F.92 JSON_IOTSourceList	1043
F.93 JSON_IOTSourceSupport	1044
F.94 JSON_KeepAlive	1045
F.95 JSON_KeepAliveCap	1045
F.96 JSON_LaserOpticalAxis	1046
F.97 JSON_LaserOpticalAxis_direction	1046
F.98 JSON_LaserOpticalAxisCap	1046
F.99 JSON_LinkageChansCond	1047
F.100 JSON_LinkedTracking	1047
F.101 JSON_LinkedTrackingCap	1048
F.102 JSON_logProTestResult	1050
F.103 JSON_MatchList	1050
F.104 JSON_MoveAutoTracking	1052
F.105 JSON_OnlineUpgradeParameter	1052
F.106 JSON OSD	1052
F.107 JSON_OSDCap	1053
F.108 JSON_PeopleCounting_CollectionDescriptionCap	1055
F.109 JSON_PeopleCounting_CollectionDescription	1055
F.110 JSON_PeopleCounting_CollectionResult	1056
F.111 JSON_Reconstruction_ProgressLists	1056
F.112 JSON_RemarkList	1057
F.113 JSON_resourceStatistics	1057
F.114 JSON_ResponseStatus	1058
F.115 JSON_Result	1058
F.116 JSON_SearchCondition	1059
F.117 JSON_SecurityEmail	1059
F.118 JSON_SecurityEmailCap	1060
F.119 JSON_SecurityEmailQrCode	1060

F.120 JSON_SmartOverlap	1061
F.121 JSON_SmartOverlapCap	1061
F.122 JSON_SourceCapabilities	1061
F.123 JSON_SSDCapacity	1062
F.124 JSON_SSDFormatStatus	1062
F.125 JSON_SSDUpgrade	1063
F.126 JSON_SSDUpgradeStatus	1063
F.127 JSON_SSD_UpgradeStatus	1063
F.128 JSON_SyncStatus	1064
F.129 JSON_CollectionDescription	1064
F.130 JSON_Temperature_CollectionDescriptionCap	1064
F.131 JSON_CollectionResult	1065
F.132 JSON_TestDescription	1066
F.133 JSON_TestResult	1067
F.134 JSON_TimeSearchCond	1067
F.135 JSON_TimeSearchResult	1067
F.136 JSON_TrafficMonitor	1068
F.137 JSON_TrafficMonitorCap	1069
F.138 JSON_TrafficMonitorStatus	1069
F.139 JSON_unitConfig	1070
F.140 JSON_unitConfigCap	1071
F.141 JSON_USB	1071
F.142 JSON_USBCap	1071
F.143 JSON_User	1072
F.144 JSON_UserCap	1072
F.145 JSON_UserList	1072
F.146 JSON_UserPermission	1073
F.147 JSON_UserPermissionCap	1074

F.148 JSON_VerificationDisplay	1074
F.149 JSON_VerificationPoint	1075
F.150 JSON_VerificationPointList	1075
F.151 JSON_VerificationPointsCap	1076
F.152 JSON_VibrationDetection	1077
F.153 JSON_VibrationDetectionCap	1077
F.154 JSON_WhiteLightAlarm	1078
F.155 JSON_WhiteLightAlarmCap	1078
F.156 JSON_WorkingStatus	1079
F.157 JSON_XX	1080
F.158 JSON_XXCap	1081
F.159 XML_AccessProtocolAbility	1081
F.160 XML_AdminAccessProtocol	1084
F.161 XML_AdminAccessProtocolList	1085
F.162 XML_AlgorithmsVersion	1085
F.163 XML_BasicCapability	1086
F.164 XML_CAMERAPARA	1099
F.165 XML_Cap_AdminAccessProtocolList	1117
F.166 XML_Cap_AudioVideoCompressInfo	1118
F.167 XML_Cap_Color	1158
F.168 XML_Cap_DefaultParam	1158
F.169 XML_Cap_DeviceInfo	1159
F.170 XML_Cap_DeviceLanguage	1160
F.171 XML_Cap_DiagnosedDataParameter	1161
F.172 XML_Cap_Dial	1161
F.173 XML_Cap_EagleFocusing	1162
F.174 XML_Cap_EHome	1163
F.175 XML_Cap_ExternalDevice	1164

F.176 XML_Cap_EZVIZ	1164
F.177 XML_Cap_FocusConfiguration	1165
F.178 XML_Cap_FTPNotification	1165
F.179 XML_Cap_FTPNotificationList	1169
F.180 XML_Cap_HardwareService	1169
F.181 XML_Cap_hddList	1170
F.182 XML_Cap_HiddenInformation	1171
F.183 XML_Cap_ImageChannel	1171
F.184 XML_Cap_Integrate	1174
F.185 XML_Cap_IPFilter	1175
F.186 XML_Cap_ItemList	1175
F.187 XML_Cap_LensDistortionCorrection	1176
F.188 XML_Cap_MACFilter	1176
F.189 XML_Cap_MaxElevation	1177
F.190 XML_Cap_MountingScenario	1177
F.191 XML_Cap_SalveCamera	1177
F.192 XML_Cap_Schedule	1178
F.193 XML_Cap_storage	1179
F.194 XML_Cap_storageExtension	1179
F.195 XML_Cap_StreamingChannel	1180
F.196 XML_Cap_SoftwareService	1182
F.197 XML_Cap_SupplementLight	1183
F.198 XML_Cap_tempRange	1184
F.199 XML_Cap_Time	1184
F.200 XML_Cap_Track	1185
F.201 XML_Cap_Tracking	1187
F.202 XML_Cap_User	1188
F.203 XML_Cap_UserPermissionList	1190

F.204 XML_Cap_VCAResource	1193
F.205 XML_Cap_WirelessServer	1193
F.206 XML_ChannelEventCap	1195
F.207 XML_ChannelEventCapList	1200
F.208 XML_ChannelInfo	1201
F.209 XML_ChannelInfoList	1203
F.210 XML_CheckInfo	1203
F.211 XML_CMSearchProfile	1203
F.212 XML_Color	1204
F.213 XML_DefaultParam	1204
F.214 XML_Desc_AccessProtocolAbility	1205
F.215 XML_Desc_AudioVideoCompressInfo	1205
F.216 XML_Desc_EventAbility	1205
F.217 XML_Desc_JpegCaptureAbility	1205
F.218 XML_Desc_NetAppAbility	1206
F.219 XML_Desc_POSAbility	1206
F.220 XML_Desc_UserAbility	1206
F.221 XML_DeviceCap	1206
F.222 XML_DeviceInfo	1217
F.223 XML_DeviceLanguage	1220
F.224 XML_DeviceStatus	1220
F.225 XML_DiagnosedDataParameter	1222
F.226 XML_Dial	1222
F.227 XML_DownloadAbility	1223
F.228 XML_downloadRequest	1224
F.229 XML_Ehome	1224
F.230 XML_EhomeList	1225
F.231 XML_EPTZ	1226

F.232 XML_EventAbility	1226
F.233 XML_EventCap	1245
F.234 XML_EventNotificationAlert_AlarmEventInfo	1247
F.235 XML_EventTrigger	1248
F.236 XML_EventTriggerCapType	1249
F.237 XML_EventTriggerNotification	1250
F.238 XML_EventTriggerNotificationList	1251
F.239 XML_EventTriggersCap	1251
F.240 XML_externSecurityCap	1254
F.241 XML_EZVIZ	1254
F.242 XML_FocusConfiguration	1255
F.243 XML_FTPNotification	1256
F.244 XML_hdd	1259
F.245 XML_hddList	1260
F.246 XML_IllegalLoginLock	1261
F.247 XML_ImageChannel	1261
F.248 XML_Integrate	1263
F.249 XML_IntelliCap	1263
F.250 XML_IOTTriggersCap	1267
F.251 XML_IPFilter	1268
F.252 XML_IPFilterAddress	1268
F.253 XML_IPFilterAddressList	1268
F.254 XML_InputProxyChannel	1269
F.255 XML_IpViewDevAbility	1269
F.256 XML_ItemList	1276
F.257 XML_JpegCaptureAbility	1277
F.258 XML_Language	1283
F.259 XML_LensDistortionCorrection	1283

F.260 XML_localPermission	1284
F.261 XML_localPermissionCap	1285
F.262 XML_LockByName	1286
F.263 XML_LockCap	1286
F.264 XML_LogServer	1287
F.265 XML_LogServerCap	1287
F.266 XML_PrivacyMaskRegion	1288
F.267 XML_MasterSlaveTrackingCap	1288
F.268 XML_MaxElevation	1289
F.269 XML_MountingScenario	1289
F.270 XML_NetAppAbility	1289
F.271 XML_NetworkCap	1295
F.272 XML_networkExtension	1299
F.273 XML_NetworkSetting	1299
F.274 XML_OcrConfigCap	1303
F.275 XML_OnlineUpgradeCap	1304
F.276 XML_OnlineUpgradeServer	1305
F.277 XML_OnlineUpgradeStatus	1306
F.278 XML_OnlineUpgradeVersion	1306
F.279 XML_ParkAction	1306
F.280 XML_POSAbility	1307
F.281 XML_PrivacyMaskRegion	1309
F.282 XML_PTZAux	1310
F.283 XML_PTZChannel	1310
F.284 XML_PTZChanelCap	1311
F.285 XML_PTZOSDDisplay	1318
F.286 XML_RacmCap	1318
F.287 XML_RecordAbility	1322

F.288 XML_remotePermission	1323
F.289 XML_remotePermissionCap	1326
F.290 XML_ResponseStatus	1327
F.291 XML_RTSPCertificate	1328
F.292 XML_RTSPCertificateCap	1328
F.293 XML_Schedule	1329
F.294 XML_SecurityCap	1330
F.295 XML_SecurityQuestion	1332
F.296 XML_SerialCap	1333
F.297 XML_Shield	1333
F.298 XML_SlaveCamera	1334
F.299 XML_SlaveCameraStatus	1334
F.300 XML_SNMP	1335
F.301 XML_SNMPAdvanced	1335
F.302 XML_SNMPTrapReceiver	1336
F.303 XML_SNMPTrapReceiverList	1336
F.304 XML(ssd)	1336
F.305 XML(ssdList)	1337
F.306 XML_SSDSMARTTest	1337
F.307 XML_SSD_SmartTestStatus	1337
F.308 XML_StreamingChannel	1338
F.309 XML_storage	1341
F.310 XML_storageExtension	1341
F.311 XML_SoftwareService	1341
F.312 XML_SocketIP	1342
F.313 XML_SubscribeEvent	1342
F.314 XML_SubscribeEventCap	1343
F.315 XML_SubscribeEventResponse	1344

F.316 XML_SupplementLight	1345
F.317 XML_TargetEnhancement	1346
F.318 XML_TargetEnhancementCap	1347
F.319 XML_tempRange	1347
F.320 XML_ThermalCap	1347
F.321 XML_TimeTaskList	1349
F.322 XML_Track	1350
F.323 XML_Tracking	1352
F.324 XML_TrackList	1353
F.325 XML_TwoWayAudioChannel	1353
F.326 XML_TwoWayAudioChannelList	1354
F.327 XML_User	1354
F.328 XML_UserAbility	1356
F.329 XML_UserList	1358
F.330 XML_UserPermission	1358
F.331 XML_UserPermissionCap	1359
F.332 XML_UserPermissionList	1359
F.333 XML_VCAResource	1359
F.334 XML_VideoCap	1360
F.335 XML_VideoPic_CMSearchDescription	1361
F.336 XML_VideoPic_CMSearchResult	1364
F.337 XML_WebCertificate	1366
F.338 XML_WirelessDialInterface	1366
F.339 XML_WirelessDialInterfaceList	1366
F.340 XML_ZoomFocus	1366
Appendix G. Response Codes of Text Protocol	1367
Appendix H. Error Codes Categorized by Functional Modules	1406

Chapter 1 Overview

The Device Network SDK is a software development kit for the third-party platform to access the network devices (e.g., DVR, NVR, encoding devices, decoding devices, network cameras, speed dome, etc.) and integrate multiple functions, such as live view, playback, alarm, PTZ control, two-way audio, fisheye expansion, access control, ANPR, face picture comparison, and so on.

1.1 Introduction

The network communication of the SDK adopts private protocol, which is designed based on TCP (Transmission Control Protocol)/IP (Internet Protocol). To make it available for more third-party platforms and simplify the integration progresses, the SDK provides several APIs and structures to pass through operation methods, request URLs, and request or response messages to implement different functions based on the text protocol communication.

- For the general applications based on private protocol, refer to [Integrate Based on Private Protocol](#).
- For the general applications based on text protocol, refer to [Integrate Based on Text Protocol](#).
For the detailed process, refer to [Integrate by Transmitting Text Protocol](#).

1.1.1 Integrate Based on Private Protocol

The general integration applications of SDK based on private protocol includes live view, PTZ control, playback, video downloading, basic configuration, remote maintenance, two-way audio, alarm/event, decoder, abnormal event detection, and so on.

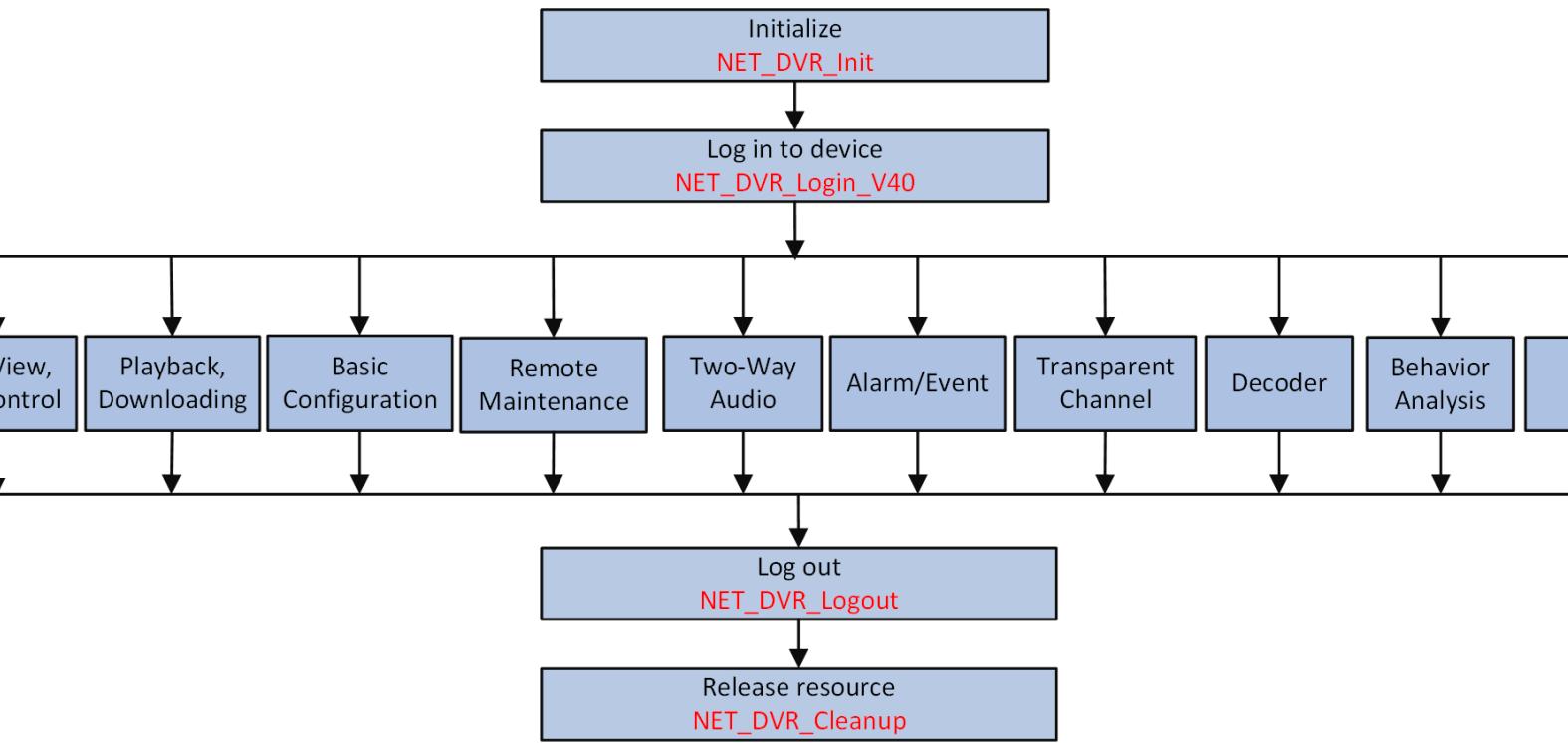


Figure 1-1 Overall Process of Integrating Based on Private Protocol



- Before and after integrating each application, the initialization, login, logout, and releasing resources are required.
- For detailed flows of each applications, refer to other chapters in this manual.

1.1.2 Integrate Based on Text Protocol

The general integration applications of SDK based on text protocol includes log search, live view, playback, video search, alarm/event, device upgrade, protocol transmission, and so on.

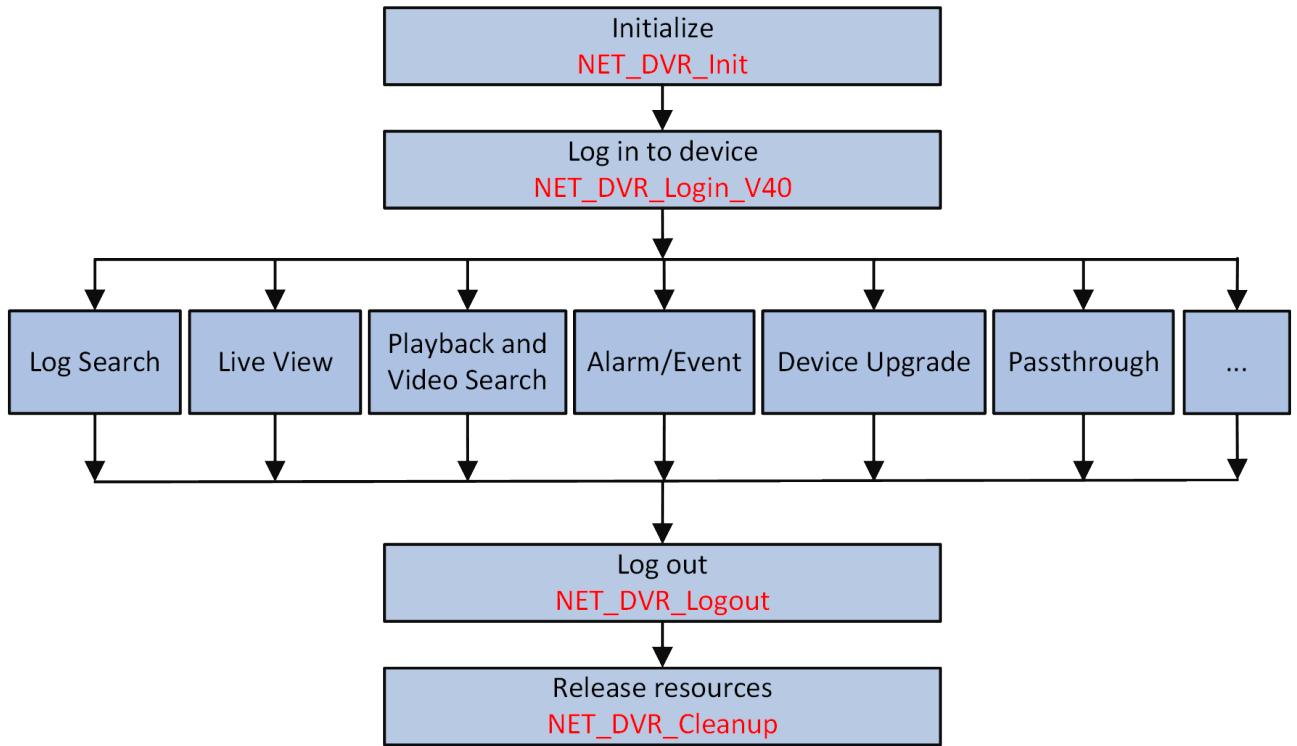


Figure 1-2 Overall Progress of Integrating Based on Text Protocol



- Before and after integrating each application, the initialization, login, logout, and releasing resources are required.
- For the integration progresses of supported applications based on text protocol, refer to [Start Live View Based on ISAPI Protocol](#), [Start Playback](#), [Search Video Files by Time](#), [Receive Alarm/Event in Arming Mode](#), [Receive Alarm/Event in Listening Mode](#), [Remotely Upgrade Device](#), [Search for Logs](#), and [Integrate by Transmitting Text Protocol](#).

1.2 System Requirements

To integrate functions by HCNetSDK, both the Windows and Linux operating systems are available. The followings show the requirement details.

For Windows Operating System with 32-bit

Windows8/Windows7/Vista/XP/2000, Windows Server 2012/2008/2003

For Windows Operating System with 64-bit

Windows10/Windows8/Windows7, Windows Server 2012/2008

For Linux Operating System with 32-bit

GCC Version 4.1.2 or Above

OS: CentOS 5.4, Redhat AS5.4, Ubuntu 9.10, Fedora12 or Above

For Linux Operating System with 64-bit

GCC Version 4.1.2 or Above

OS: CentOS 5.4, Redhat AS5.4, Ubuntu 9.10, SUSE10 or Above

1.3 SDK Component

The function implementations dependent on the components provided by SDK, here lists the corresponding files and notices of different components for reference.

SDK for Windows Operating System

Main Component	Sub Component	File Name and Type
Network Communication Library	External API File	HCNetSDK.h
		HCNetSDK.lib
		HCNetSDK.dll
	Core File	HCCore.lib
		HCCore.dll
Module Library (Folder Name: HCNetSDKCom)	Device Configuration Module File	HCCoreDevCfg.dll
	Live View Module File	HCPreview.dll
	Playback Module File	HCPlayBack.dll
	Audio Module File	HCVoiceTalk.dll
	Alarm Module File	HCAlarm.dll
	Display Module File	HCDisplay.dll
	Application Management and Configuration Module File	HCIndustry.dll
	Maintenance, Management, and Configuration Module File	HCGeneralCfgMgr.dll
RTSP Communication Library (Folder Name: HCNetSDKCom)	None.	StreamTransClient.dll

Main Component	Sub Component	File Name and Type
Container Format Conversion Library (Folder Name: HCNetSDKCom)	None.	SystemTransform.dll
Character Conversion Library (Folder Name: HCNetSDKCom)	None.	libiconv2.dll
Analog Capability Set	None.	LocalXml.zip
Stream Analysis Library (Folder Name: HCNetSDKCom)	None.	AnalyzeData.dll
Two-Way Audio Library (Folder Name: HCNetSDKCom)	None.	AudioIntercom.dll
Playing Library	Core File	PlayM4.h WindowsPlayM4.h PlayCtrl.lib PlayCtrl.dll
	Video Rendering Library	SuperRender.dll D3DX9_43.dll
	Audio Rendering Library	AudioRender.dll OpenAL32.dll
	Panoramic Camera Library	EagleEyeRender.dll
	Hardware Decoding Library	HWDecode.dll
	Fisheye Library	MP_Render.dll
	Image Enforcement Library (only supported by 32-bit Windows operating system)	MP_VIE.dll
	Capture with Temperature Info Library	YUVProcess.dll
	Fisheye and PanoVu Camera Dependent Library	D3DCompiler_43.dll D3DX9_43.dll

SDK for Linux Operating System

Main Component	Sub Component	File Name and Type
Network Communication Library	External API File	HCNetSDK.h libhcnetsdk.so
	Core File	libHCCore.so
Module Library (Folder Name: HCNetSDKCom)	Device Configuration Module File	libHCCoreDevCfg.so
	Live View Module File	libHCPreview.so
	Playback Module File	libHCPplayBack.so
	Audio Module File	libHCVoiceTalk.so
	Alarm Module File	libHCAlarm.so
	Display Module File	libHCDisplay.so
	Application Management and Configuration Module File	libHCIndustry.so
	Maintenance, Management, and Configuration Module File	libHCGeneralCfgMgr.so
HPR Library	None.	libhpr.so
RTSP Communication Library (Folder Name: HCNetSDKCom)	None.	libStreamTransClient.so
Container Format Conversion Library (Folder Name: HCNetSDKCom)	None.	libSystemTransform.so
Character Conversion Library (Folder Name: HCNetSDKCom)	None.	libiconv2.so
Stream Analysis Library (Folder Name: HCNetSDKCom)	None.	libanalyzedata.so
OpenSSL	None.	libssl.so
	Dependent Library	libcrypto.so
Playing Library	Core File	PlayM4.h
		LinuxPlayM4.h

Main Component	Sub Component	File Name and Type
		libPlayCtrl.so
	Video Rendering Library	libSuperRender.so
	Audio Rendering Library	libAudioRender.so

Remarks

- The component files, HCNetSDK.dll and HCCore.dll must be loaded at any time; and when updating the development kit, the HCNetSDK.dll, HCCore.dll, and the files in HCNetSDKCom folder must be loaded again, and the folder name HCNetSDKCom cannot be edited.
- The dependent library files D3DX9_43.dll and D3DCompiler_43.dll are not directly provided by SDK, you should go to official website to download these files via [DirectX](#) and load to the development environment by yourself.
- Network Communication Library: The main body of SDK, which is used for the communication and interaction between systems/clients and devices. It provides the functions of remote control, remote configuration, getting stream, and processing stream.
- HPR Library: The dependent library of network communication library, it should be loaded together with the network communication library for SDK in Linux operating system.
- RTSP Communication Library: Network library that supports RTSP transmission, it is required to be loaded when the device supports RTSP streaming.
- Container Format Conversion Library: Convert the standard stream format to other container formats, such as 3GPP, PS, and so on. It is required to be loaded when the device supports RTSP streaming.
- Character Conversion Library: It is required when the character sets of computer and device mismatches. You can also set transcoding callback function to convert the character if you want to use your own API.
- Analog Capability Set: It is required to be loaded if you want to get the analog capability set. And the component file must be in the same directory of network communication library.
- Two-Way Audio Library: It is used to collect data via sound card, encode stream to a specific encoding format, and decode and display the audio data (stream data without container format). The dependent library OpenAL32.dll is required in two-way audio mode.
Audio encoding and decoding are only supported by SDK in 32-bit Windows operating system; and for Linux operating system, only audio forwarding is supported.
- Stream Analysis Library: It is required to be loaded for video and audio data analysis by setting raw stream callback function.
- OpenSSL: It is required to be loaded for login and sensitive information encryption.
- Playing Library: It is mainly used to decode and display the real-time stream data, and play back the video files. It is only required to be loaded when you play stream via SDK.

1.4 Update History

1.4.1 Version 6.1.X.X

Summary of Changes in Version 6.1.7.15_Aug., 2021

Extended the structure about RS-485 parameters of the security control panel

NET_DVR_ALARM_RS485CFG (related APIs: **NET_DVR_GetDVRConfig** with command

1188—"NET_DVR_SET_ALARM_RS485CFG" and **NET_DVR_SetDVRConfig** with command

1189—"NET_DVR_GET_ALARM_RS485CFG");

added a member **byStairsOutputDataType** (output data type of the elevator controller) by one byte.

Summary of Changes in Version 6.1.7.X_July, 2021

1. Extended the message about capability of device information **XML_Cap_DeviceInfo** and the message about device information **XML_DeviceInfo** (related URLs: **/ISAPI/System/deviceInfo/capabilities** and **/ISAPI/System/deviceInfo** ; related API: **NET_DVR_STDXMLConfig**):
added 3 language types to the node **languageType**: "dutch" (Dutch), "hungarian" (Hungarian), and "irish" (Irish).
2. Extended the message about the configuration capability of accessing servers via ISUP **XML_Cap_EHome**, message about the parameters of accessing a server via ISUP **XML_Ehome**, and message about the parameters of accessing all servers via ISUP **XML_EhomeList** (related URLs: **/ISAPI/System/Network/Ehome/capabilities** , **/ISAPI/System/Network/Ehome/<ISUPID>** , and **/ISAPI/System/Network/EhomeList?centerID=** ; related API: **NET_DVR_STDXMLConfig**):
added 2 nodes: **periodicTestEnabled** (whether to enable periodic test) and **periodicTestTime** (periodic test interval).
3. Extended the message about the EZVIZ access configuration capability **XML_Cap_EZVIZ** and message about the EZVIZ access configuration parameters **XML_EZVIZ** (related URLs: **/ISAPI/System/Network/EZVIZ/capabilities** and **/ISAPI/System/Network/EZVIZ** ; related API: **NET_DVR_STDXMLConfig**):
added 2 nodes: **periodicTestEnabled** (whether to enable periodic test) and **periodicTestTime** (periodic test interval).
4. Extended the message about the configuration capability of all users' permission **XML_Cap_UserPermissionList**, message about a specific user's permission **XML_UserPermission**, message about the capability of managing the user permission **XML_UserPermissionCap**, and message about all users' permission **XML_UserPermissionList** (related URLs: **/ISAPI/Security/UserPermission/capabilities** , **/ISAPI/Security/UserPermission/<ID>** , **/ISAPI/Security/UserPermission/adminCap** , and **/ISAPI/Security/UserPermission** ; related API: **NET_DVR_STDXMLConfig**):
added 2 sub nodes to the node **remotePermission**: **restoreTamper** (whether to restore tampering) and **restoreConfirmedAlarm** (whether to restore alarm acknowledgment).

5. Extended the message about the capability of managing cloud users
JSON_CloudUserManageCap, message about the cloud user information
JSON_CloudUserManage, and message about information of multiple cloud users
JSON_CloudUserManageList (related URIs: [/ISAPI/Security/CloudUserManage/users/capabilities?format=json](#) , [/ISAPI/Security/CloudUserManage/users/<ID>?format=json](#) , [/ISAPI/Security/CloudUserManage/users?format=json](#) , and [/ISAPI/Security/CloudUserManage/users/byType?format=json](#) ; related API: NET_DVR_STDXMLConfig):
added 2 nodes: **adminType** (admin type) and **installerType** (installer type).
6. Extended the message about user configuration capability XML_Cap_User, message about a specific user's configuration XML_User, and message about configuration of all users
XML_UserList (related URIs: [/ISAPI/Security/users/<ID>/capabilities](#) , [/ISAPI/Security/users/<ID>](#) , and [/ISAPI/Security/users](#) ; related API: NET_DVR_STDXMLConfig):
added 2 nodes: **adminType** (admin type) and **installerType** (installer type).

Summary of Changes in Version 6.1.7.15_May, 2021

Added one function of getting all tasks of manually copying back videos:
related API NET_DVR_StartRemoteConfig with command
"NET_DVR_GET_ALL_RECORD_PASSBACK_TASK_MANUAL" (command No.: 6235).

Summary of Changes in Version 6.1.6_Jan., 2021

1. Extended the enumeration about files to be downloaded NET_SDK_DOWNLOAD_TYPE :
added one enumeration type **NET_SDK_DOWNLOAD_LENS_PARAM_FILE** (lens parameters file).
2. Extended the enumeration about files to be uploaded NET_SDK_UPLOAD_TYPE :
added one enumeration type **UPLOAD_LENS_PARAM_FILE** (lens parameters file).

Summary of Changes in Version 6.1.6.151_Dec., 2020

Added one API of getting the error code categorized by functional modules:
NET_DVR_GetLastErrorModelCode .

Summary of Changes in Version 6.1.6_Aug., 2020

1. Extended the file search condition structure NET_DVR_FILECOND_V50 (related API: NET_DVR_FindFile_V50):
added a member **dwTimeout** (timeout time of searching for files) by 4 bytes.
2. Added an API of getting the IP address and port information of the stream sender and recipient:
NET_DVR_GetLinkAddr .
3. Extended the structure about alarm search condition parameters
NET_DVR_ALARM_SEARCH_COND (related API: NET_DVR_StartRemoteConfig with command 4193-"NET_DVR_GET_ALARM_INFO"):
added a member **byNoBoundary** (whether form data is contained in the returned alarm information) by 1 bytes.
4. Added a remark to the API NET_DVR_StartVoiceCom_V30 :"This API is supported for programming in 64-bit Windows/Linux operating system and 32-bit Windows operating system".

Summary of Changes in Version 6.1.5.26_Nov., 2020

1. Extended the channel attributes message [XML_ChannelInfo](#) (related API: [NET_DVR_STDXMLConfig](#) ; related URI: [/ISAPI/AUXInfo/attributes/Channels/<ID>](#)) :
added one node: <TemperatureIntervalMeasurement> (interval temperature measurement).
2. Extended recording schedule configuration capability [XML_Cap_Track](#) (related API: [NET_DVR_STDXMLConfig](#) ; related URI: [/ISAPI/ContentMgmt/record/tracks/<ID>/capabilities](#)) :
added one record type "temperatureIntervalMeasurement" (interval temperature measurement) to nodes <ActionRecordingMode> and <DefaultRecordingMode>.
3. Extended recording schedule configuration parameters [XML_Track](#) (related API: [NET_DVR_STDXMLConfig](#) ; related URI: [/ISAPI/ContentMgmt/record/tracks](#)) :
added one record type "temperatureIntervalMeasurement" (interval temperature measurement) to node <ActionRecordingMode>.
4. Extended search conditions of video and picture [XML_VideoPic_CMSearchDescription](#) (related API: [NET_DVR_STDXMLConfig](#) ; related URI: [/ISAPI/ContentMgmt/search](#)) :
added one search condition "temperatureIntervalMeasurement" (interval temperature measurement) to node <metadataDescriptor>.
5. Extended device storage capability [XML_RacmCap](#) (related API: [NET_DVR_STDXMLConfig](#) ; related URI: [/ISAPI/ContentMgmt/capabilities](#)) :
added one search condition "temperatureIntervalMeasurement" (interval temperature measurement) to nodes <pictureSearchType> and <recordSearchType>.
6. Extended the event capability of a single channel [XML_ChannelEventCap](#) (related API: [NET_DVR_STDXMLConfig](#) ; related URI: [/ISAPI/Event/channels/<ID>/capabilities](#)) :
added one event type "temperatureIntervalMeasurement" (interval temperature measurement) to node <eventType>.
7. Extended device event capability [XML_EventCap](#) (related API: [NET_DVR_STDXMLConfig](#) ; related URI: [/ISAPI/Event/capabilities](#)) :
added one node <isSupportTemperatureIntervalMeasurement> (whether the device supports interval temperature measurement).

Summary of Changes in Version 6.1.5.25_Nov., 2020

Extended the message about capability of device information [XML_Cap_DeviceInfo](#) and message about device information [XML_DeviceInfo](#) (related URIs: [/ISAPI/System/deviceInfo/capabilities](#) and [/ISAPI/System/deviceInfo](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added 3 nodes: <ZigBeeVersion> (ZigBee module version), <R3Version> (R3 module version), and <RxVersion> (Rx module version).

Summary of Changes in Version 6.1.5.25_Sept., 2020

1. Extended the device information capability [XML_Cap_DeviceInfo](#) and device information [XML_DeviceInfo](#) (related URLs: [/ISAPI/System/deviceInfo/capabilities](#) and [/ISAPI/System/deviceInfo](#) ; related API: [NET_DVR_STDXMLConfig](#)):

- added one node <**bootTime**> (system boot time).
2. Extended focus configuration capability [XML_Cap_FocusConfiguration](#) (related URI: [/ISAPI/Image/channels/<ID>/focusConfiguration/capabilities](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added two nodes <**focusStatus**> (focus status) and <**motorMovementType**> (motor movement type).
 3. Added one URI for getting focus status: GET [/ISAPI/Image/channels/<ID>/focusConfiguration/status?format=json](#) (related API: [NET_DVR_STDXMLConfig](#)).

Summary of Changes in Version 6.1.5.25_Aug., 2020

1. Extended audible warning configuration capability [JSON_AudioAlarmCap](#) (related URI: [/ISAPI/Event/triggers/notifications/AudioAlarm/capabilities?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added one audio type 16 (AI custom audio file) to node **audioTypeListCap**.
2. Extended information of custom alarm audio file [JSON_GetCustomAudio](#) (related URI: [/ISAPI/Event/triggers/notifications/AudioAlarm/customAudioInfo?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added one node **AudioType** (audio file type).
3. Extended the information of custom alarm audio file to be imported [JSON_ImportCustomAudio](#) (related URI: [/ISAPI/Event/triggers/notifications/AudioAlarm/customAudio?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added two nodes **AudioType** (audio file type) and **isCover** (whether to cover the audio file with same name).
4. Extended the device capability [XML_DeviceCap](#) (related URI: [/ISAPI/System/capabilities](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added two nodes <**isSupportAIOpenPlatform**> (whether the device supports AI open platform capabilities) and <**isSupportPictureDownloadError**> (whether the device supports reporting picture download failure).

Summary of Changes in Version 6.1.5.20_Sept, 2020

1. Extended the device capability [XML_DeviceCap](#) (related URI: [/ISAPI/System/capabilities](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added one node <**characteristicCode**> (device attribute code).
2. Added one URI for getting the device attribute code (related API: [NET_DVR_STDXMLConfig](#)):
GET [/ISAPI/System/deviceInfo/characteristicCode?format=json](#) .
3. Extended the capability of exporting diagnosis information [XML_Cap_DiagnosedDataParameter](#) (related URI: [/ISAPI/System/diagnosedData/parameter/capabilities](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added one node <**isSupportServer**> (whether the device supports diagnostic server configuration).
4. Added URIs for diagnostic server configuration (related API: [NET_DVR_STDXMLConfig](#)):
Get the diagnostic server capability: GET [/ISAPI/System/diagnosedData/server/capabilities?format=json](#) ;

Get or set diagnostic server parameters: GET or PUT [/ISAPI/System/diagnosedData/server?format=json](#).

5. Added one URI for testing diagnostic server (related API: [NET_DVR_STDXMLConfig](#)): POST [/ISAPI/System/diagnosedData/server/test?format=json](#).

Summary of Changes in Version 6.1.5.15_Aug., 2020

1. Extended the network capability message [XML_NetworkCap](#) (related URI: [/ISAPI/System/Network/capabilities](#); related API: [NET_DVR_STDXMLConfig](#)):
added two nodes <isSupportEZVIZUnbind> (whether it supports unbinding devices from the Guarding Vision account) and <isSupportEZVIZQRcode> (whether it supports getting the Guarding Vision QR code).
2. Added a URI of getting the Guarding Vision QR code of the device (related API: [NET_DVR_STDXMLConfig](#)): GET [/ISAPI/System/Network/EZVIZ/QRCode](#).

Summary of Changes in Version 6.1.5.5_July, 2020

1. Extended wireless dial-up capability message [XML_Cap_Dial](#) and wireless dial-up parameter message [XML_Dial](#) (related URIs: [/ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial/capabilities](#) and [/ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial](#); related API: [NET_DVR_STDXMLConfig](#)):
added four nodes: <ISPName> (ISP name), <IMEINo> (IMEI code), <ICCID> (ICCID code), and <netType> (network type).
2. Extended condition message of searching for specific resources [XML_VideoPic_CMSearchDescription](#) (related URI: [/ISAPI/ContentMgmt/search](#); related API: [NET_DVR_STDXMLConfig](#)):
added a sub node **pircamID** (pircam search ID) to the node **contentTypeList**.
3. Extended event linkage notification message [XML_EventTriggerNotification](#) (related URI: [/ISAPI/Event/triggers/<eventType>-<channelID>/notifications](#); related API: [NET_DVR_STDXMLConfig](#)):
added two linkage actions "siren" (siren linkage) and "output" (relay linkage) to the sub node **EventTriggerNotification** of the node **EventTriggerNotificationList**.
4. Extended alarm linkage capability message [XML_EventTriggersCap](#) (related URI: [/ISAPI/Event/triggersCap](#); related API: [NET_DVR_STDXMLConfig](#)):
added two sub nodes to the node **DiskfullTriggerCap**, i.e., **isSupportSiren** (whether it supports siren linkage) and **isSupportOutput** (whether it supports relay linkage).
5. Added 8 sub status codes to status code 4 (Invalid Operation) in [Response Codes of Text Protocol](#):
0x40008038-"overAudioFileNumLimit" (The number of audio files exceeds the limit),
0x40008039-"audioFileNameIsLong" (The audio file name is too long),
0x4000803a-"audioFormatIsWrong" (The audio file format is invalid),
0x4000803b-"audioFileIsLarge" (The size of the audio file exceeds the limit),
0x4000803c-"pircamCapTimeOut" (Capturing of pircam timed out),
0x4000803d-"pircamCapFail" (Capturing of pircam failed), 0x4000803e-"pircamIsCaping" (The pircam is capturing), and 0x4000803f-"audioFileHasExisted" (The audio file already exists).

Summary of Changes in Version 6.1.4.40_July, 2020

1. Extended the device parameter structure (V30) **NET_DVR_DEVICEINFO_V30** :
added one value bySupport4&0x02 (whether the device supports transmitting form format data) to the member **bySupport4**.
2. Extended the capability of digital video channel proxy **XML_Cap_StreamingChannel** (related URI: [/ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/capabilities](#) ; related API: **NET_DVR_STDXMLConfig**):
added one node <isSupportPictureByUrl> (whether the device supports capture picture in URL format).
3. Added URIs for capturing the picture in URL format (related API: **NET_DVR_STDXMLConfig**):
Get capability: GET [/ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/PictureByUrl/capabilities?format=json](#) ;
Capture the picture in URL format: POST [/ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/PictureByUrl?format=json](#) .
4. Extended the Guarding Vision access configuration capability **XML_Cap_EZVIZ** (related URI: [/ISAPI/System/Network/EZVIZ/capabilities](#) ; related API: **NET_DVR_STDXMLConfig**):
added one node <isSupportEZVIZQRCode> (whether the device supports getting the Guarding Vision QR code).
5. Added one URI for getting Guarding Vision QR code (related API: **NET_DVR_STDXMLConfig**):
GET [/ISAPI/System/Network/EZVIZ/QRCode?format=json](#) .
6. Extended the HDD parameters **XML_hdd** and HDD list parameters **XML_hddList** (related URI: [/ISAPI/ContentMgmt/Storage/hdd](#) and [/ISAPI/ContentMgmt/Storage/hdd/<ID>](#) ; related API: **NET_DVR_STDXMLConfig**):
added three values to node <status> (HDD status): "synching" (synchronizing), "syncError" (synchronization exception), and "unloaded".
7. Added the URI for getting the HDD data sync status (related API: **NET_DVR_STDXMLConfig**):
GET [/ISAPI/ContentMgmt/Storage/hdd/<ID>/syncStatus?format=json](#) .
8. Extended the device storage capability **XML_RacmCap** (related URI: [/ISAPI/ContentMgmt/capabilities](#) ; related API: **NET_DVR_STDXMLConfig**):
added one node <isSupportSSDSMARTTest> (whether the device supports SSD S.M.A.R.T detection).
9. Extended the device capability **XML_DeviceCap** (related URI: [/ISAPI/System/capabilities](#) ; related API: **NET_DVR_STDXMLConfig**):
added three nodes: <isSupportSSDFileSystemUpgrade> (whether the device supports SSD file system upgrade), <isSupportSSDFileSystemFormat> (whether the device supports SSD file system formatting), and <isSupportSSDFileSystemCapacity> (whether the device supports getting space distribution information of SSD file system).
10. Added URIs for SSD management, see **SSD Management** for details.
11. Extended the log types **HCNetSDK Log Types** :
added one minor exception log type "MINOR_EZVIZ_UPGRADE_EXCEPTION" (Guarding Vision upgrade exception);

added four minor operation log types: "MINOR_LOCAL SSD OPERATE_START" (Local SSD operation (firmware operations) started), "MINOR_LOCAL SSD OPERATE_STOP" (Local SSD operation (firmware operations) ended), "MINOR_REMOTE SSD OPERATE_START" (Remote SSD operation (firmware operations) started), "MINOR_REMOTE SSD OPERATE_STOP" (Remote SSD operation (firmware operations) ended).

Summary of Changes in Version 6.1.4.35_July, 2020

1. Extended the enumeration about file types to be uploaded [NET_SDK_UPLOAD_TYPE](#) (related API: [NET_DVR_UploadFile_V40](#)):
added a file type to be uploaded "UPLOAD_SECURITY_CFG_FILE" (configuration file to be securely imported).
2. Extended the enumeration about file types to be downloaded [NET_SDK_DOWNLOAD_TYPE](#) (related API: [NET_DVR_StartDownload](#)):
added a file type to be downloaded "NET_SDK_DOWNLOAD_SECURITY_CFG_FILE" (configuration file to be securely exported).
3. Extended the log types in [HCNetSDK Log Types](#):
added two minor types, i.e., 0x43a-"MINOR_DSP_START_FAILED" (Starting up DSP failed) and 0x43b-"MINOR_SMART_REGULATION_NOT_ALLOWED" (Intelligent rule is not supported) to major type 0x2 (exception log).

Summary of Changes in Version 6.1.4.20_June, 2020

1. Extended picture search result structure [NET_DVR_FIND_PICTURE_V50](#) (related API: [NET_DVR_FindPicture](#)):
added two members **byThermometryUnit** (temperature unit) and **fFaceSnapTemperature** (temperature information of face in captured face picture).
2. Extended the channel attributes [XML_ChannelInfo](#) (related API: [NET_DVR_STDXMLConfig](#); related URI: GET [/ISAPI/AUXInfo/attributes/Channels/<ID>](#)):
added one sub node <FaceSnapThermometry> (upload the captured face picture with temperature information) to node <Thermal>.
3. Extended the audible warning configuration capability [JSON_AudioAlarmCap](#) (related API: [NET_DVR_STDXMLConfig](#); related URI: GET [/ISAPI/Event/triggers/notifications/AudioAlarm/capabilities?format=json](#)) and audible warning configurations [JSON_AudioAlarm](#) (related API: [NET_DVR_STDXMLConfig](#); related URI: [/ISAPI/Event/triggers/notifications/AudioAlarm?format=json](#)):
added one alarm type "noMaskDetection" (no wearing mask detection) to node **alarmType**.
4. Extended the device event capability [XML_EventCap](#) (related API: [NET_DVR_STDXMLConfig](#); related URI: GET [/ISAPI/Event/capabilities](#)):
added two nodes <isSupportNoMaskDetection> (whether device supports no wearing mask detection) and <isSupportTMPC> (whether device supports temperature measurement pre-alarm).

Summary of Changes in Version 6.1.4.20_May, 2020

1. Extended device capability **XML_DeviceCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: GET **/ISAPI/System/capabilities**):
added five nodes: <isSupportPersonArmingTrack> (whether device supports auto person arming), <isSupportManualPersonArmingTrack> (whether device supports manual person arming), <isSupportGPSCalibrationMode> (whether device supports GPS calibration), <isSupportGPSVerification> (whether device supports GPS verification), and <isSupportHBDLib> (whether device supports human body picture library).
2. Extended message of event capability supported by a single channel **XML_ChannelEventCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: GET **/ISAPI/Event/channels/<ID>/capabilities**):
added two event types "personArmingTrack" (person arming) and "manualPersonArmingTrack" (manual person arming) to the node <eventType>.
3. Extended the linkage capabilities of different alarm categories **XML_EventTriggersCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/Event/triggersCap**):
added one node <HBDLibTriggerCap> (alarm linkage capability of human body picture library).
4. Extended the smart linkage capability **XML_MasterSlaveTrackingCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/MasterSlaveTracking/capabilities**):
added one node <isSupportLinkedTracking> (whether device supports advanced parameters configuration of linked tracking capture).
5. Added a function of configuring advanced parameters of linked tracking capture (related API: **NET_DVR_STDXMLConfig**):
Get capability: GET **/ISAPI/MasterSlaveTracking/linkedTracking/capabilities?format=json** ;
Get or set advanced parameters of linked tracking capture: GET or PUT **/ISAPI/MasterSlaveTracking/linkedTracking?format=json** .
6. Extended the capability of exporting device diagnose information **XML_Cap_DiagnosedDataParameter** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/System/diagnosedData/parameter/capabilities**) and device diagnose information exporting parameters **XML_DiagnosedDataParameter** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/System/diagnosedData/parameter**):
added three nodes <HardwareInfo> (whether to enable exporting hardware information), <SoftwareInfo> (whether to enable exporting software information), and <logInfo> (whether to enable exporting startup log).
7. Added the URIs for verifying GPS calibration, refer to **GPS Calibration** for details.
8. Extended the device protocol capability **XML_Cap_AdminAccessProtocolList** (related API: **NET_DVR_STDXMLConfig** ; related URI: GET **/ISAPI/Security/adminAccesses/capabilities**) and device supported protocol parameters **XML_AdminAccessProtocol** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/Security/adminAccesses**):
added one protocol "Bonjour" to node <protocol>.
9. Extended the device security capability **XML_SecurityCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/Security/capabilities**) and

XML_IllegalLoginLock (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/Security/illegalLoginLock**):

added one node <maxIllegalLoginLockTime> (lock duration when maximum illegal login attempts reached the upper limit).

10. Extended the PTZ channel capability **XML_PTZChanelCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/PTZCtrl/channels/<ID>/capabilities**):
added one nodes <isSupportGyroVerify> (whether device supports gyroscope attitude calibration).
11. Added URLs for calibrating gyroscope attitude (related API: **NET_DVR_STDXMLConfig**):
Get capability: GET **/ISAPI/PTZCtrl/channels/<ID>/gyroVerify/capabilities?format=json**
Set gyroscope attitude calibration parameters: PUT **/ISAPI/PTZCtrl/channels/<ID>/gyroVerify?format=json**
12. Added two event types "personArmingTrack" (auto person arming) and "manualPersonArmingTrack" (manual person arming) to **Supported Alarm/Event Types**.
13. Added one error code 3504—"NET_DVR_UPLOAD_HBDLIBID_ERROR" (incorrect ID of human body picture library) to **Device Network SDK Errors**.

Summary of Changes in Version 6.1.4.15_June, 2020

1. Extended configuration capability message of the device information **XML_Cap_DeviceInfo** (related URI: **/ISAPI/System/deviceInfo/capabilities** ; related API: **NET_DVR_STDXMLConfig**):
added a node <isSupportNewVersionDevLanguageSwitch> (whether it supports switching language by new version of the protocol).
2. Extended configuration capability message of the device language **XML_Cap_DeviceLanguage** (related URI: **/ISAPI/System/DeviceLanguage/capabilities** ; related API: **NET_DVR_STDXMLConfig**):
added a node <upgradeFirmWareEnabled> (whether to upgrade the firmware).

Summary of Changes in Version 6.1.4.15_May, 2020

Added a function of configuring device USB parameters (related API: **NET_DVR_STDXMLConfig**):
Get USB parameters configuration capability: GET **/ISAPI/System/usb/capabilities?format=json** ;
Get or set USB parameters: GET or PUT **/ISAPI/System/usb?format=json** .

Summary of Changes in Version V6.1.4.15_April, 2020

1. Extended the supplement light configuration capability **XML_Cap_SupplementLight** and supplement light configuration parameters **XML_SupplementLight** (related API: **NET_DVR_STDXMLConfig** ; related URLs: **/ISAPI/Image/channels/<ID>/supplementLight/capabilities** and **/ISAPI/Image/channels/<ID>/supplementLight**):
added two nodes: <irLightbrightnessLimit> (IR light brightness limit) and <whiteLightbrightnessLimit> (white light brightness limit).
2. Extended the structure about supplement light configuration parameters **NET_DVR_BUILTIN_SUPPLEMENTLIGHT** (related API: **NET_DVR_GetSTDConfig** with command 3728—"NET_DVR_GET_SUPPLEMENTLIGHT", and **NET_DVR_SetSTDConfig** with command 3729—"NET_DVR_SET_SUPPLEMENTLIGHT"):

added two members **byIrLightbrightnessLimit** (IR light brightness limit) and **byWhiteLightbrightnessLimit** (white light brightness limit) with two reserved bytes.

Summary of Changes in Version 6.1.4.15_Mar., 2020

1. Added function of configuring vibration detection alarm, see [Configure Vibration Detection Alarm](#) for details.
2. Extended device capability message [XML_DeviceCap](#) (related API: [NET_DVR_STDXMLConfig](#); related URI: [/ISAPI/System/capabilities](#)):
added one node <**isSupportVibrationDetection**> (whether it supports vibration detection).
3. Extended message about event capability of all channels [XML_ChannelEventCapList](#) and message about event capability of a single channel [XML_ChannelEventCap](#) (related API: [NET_DVR_STDXMLConfig](#); related URIs: [/ISAPI/Event/channels/capabilities](#) and [/ISAPI/Event/channels/<ID>/capabilities](#)):
added one event type "vibrationDetection" (vibration detection) to the node <**eventType**>.
4. Extended message about linkage capabilities of different alarm categories [XML_EventTriggersCap](#) (related API: [NET_DVR_STDXMLConfig](#); related URI: [/ISAPI/Event/triggersCap](#)):
added one node <**VibrationDetectionTriggerCap**> (alarm linkage capability of vibration detection).
5. Extended linkage parameter message [XML_EventTrigger](#) (related API: [NET_DVR_STDXMLConfig](#); related URI: [/ISAPI/Event/triggers/<eventType>-<channelID>](#)):
added one event type "vibrationDetection" (vibration detection) to the node <**eventType**>.
6. Extended capability message [XML_Cap_VCAResource](#) and configuration message [XML_VCAResource](#) about intelligent resources switch (related API: [NET_DVR_STDXMLConfig](#); related URIs: [/ISAPI/System/Video/inputs/channels/<ID>/VCAResource/capabilities](#) and [/ISAPI/System/Video/inputs/channels/<ID>/VCAResource](#)):
added one intelligent resource type "faceCounting" (face counting) to the node <**type**>.
7. Extended picture search condition structure [NET_DVR_FIND_PICTURE_PARAM](#) (related API: [NET_DVR_FindPicture](#)):
added one picture capture type 0x58 (capture based on vibration detection) to the member **byFileType**.
8. Extended picture capture capability message [XML_JpegCaptureAbility](#) (related API: [NET_DVR_GetDeviceAbility](#); related Command: DEVICE_JPEG_CAP_ABILITY; **pInBuf**: [XML_Desc_JpegCaptureAbility](#)):
added one supported capture type 0x58-"vibrationDetection" (vibration detection) to the sub node <**supportFileType**> of <**FindPicInfo**>.
9. Extended the log types in [HCNetSDK Log Types](#):
added two minor types 0x132-"MINOR_VIBRATION_DETECTION_ALARM_BEGIN" (Vibration detection alarm started) and 0x133-"MINOR_VIBRATION_DETECTION_ALARM_END" (Vibration detection alarm ended) to major type 0x1 (alarm logs).

Summary of Changes in Version 6.1.4_Nov., 2019

1. Added a local SDK configuration type "NET_SDK_LOCAL_CFG_ASYNC" (asynchronous configuration parameters) to the enumeration **NET_SDK_LOCAL_CFG_TYPE** (related APIs: **NET_DVR_GetSDKLocalCfg** and **NET_DVR_SetSDKLocalCfg**);
2. Extended the API (**NET_DVR_InitG711Decoder**) about initializing resources for decoding audio data encoded by G.711 (only for Linux operating system) :
added a parameter **pDecHandle** (handle for audio decoding).
3. Extended the playback or downloading information structure **NET_DVR_PLAYCOND** (related API: **NET_DVR_GetFileByTime_V40**):
added three members, i.e., **byDownload** (whether to download), **byOptimalStreamType** (whether to play back the stream with optimal type), and **byVODFileType** (stream type of downloaded video file) by 3 bytes.
4. Added two initialization parameter types (related API: **NET_DVR_SetSDKInitCfg**):
"NET_SDK_INIT_CFG_LIBEAY_PATH"-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
"NET_SDK_INIT_CFG_SSLEAY_PATH"-set path (including library name) for ssleay32.dll (Windows), libssl.so (Linux), libssl.dylib (Mac) of OpenSSL in version 1.1.1 and 1.0.2
5. Added the function of log file compression by extended the value of parameter **bAutoDel** in the API **NET_DVR_SetLogFile** : "FALSE"-compress the log files when the number of files reaches 50.
6. Added the support of 64-bit Windows operating system to the audio encoding and decoding APIs: **NET_DVR_EncodeG711Frame**, **NET_DVR_DecodeG711Frame**, **NET_DVR_InitG722Encoder**, **NET_DVR_EncodeG722Frame**, **NET_DVR_InitG722Decoder**, and **NET_DVR_DecodeG722Frame**.
7. Added an exception message type to the callback function **fExceptionCallBack** (related API: **NET_DVR_SetExceptionCallBack_V30**):
0x804b-"EXCEPTION_VIDEO_DOWNLOAD" (video downloading exception).
8. Added an error to **Device Network SDK Errors** : 186-"NET_SDK_ERR_FUNCTION_INVALID"
(Invalid function. The asynchronous mode is enabled).

Summary of Changes in Version 6.1.4.10_Feb., 2020

Added URIs of configuring laser optical axis (related API: **NET_DVR_STDXMLConfig**):
get capability: GET **/ISAPI/System/laserOpticalAxis/capabilities?format=json** ;
get or set laser optical axis parameters: GET or PUT **/ISAPI/System/laserOpticalAxis?format=json** ;
adjust laser optical axis: PUT **/ISAPI/System/laserOpticalAxis/goto?format=json** .

Summary of Changes in Version 6.1.3.40_Feb., 2020

1. Extended configuration capability message **XML_Cap_DeviceLanguage** and parameter message **XML_DeviceLanguage** of device language (related URIs: **/ISAPI/System/DeviceLanguage/capabilities** and **/ISAPI/System/DeviceLanguage** ; related API: **NET_DVR_STDXMLConfig**):
added a language supported by the device "Ukrainian" to the node <language>.
2. 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 6.1.3.35_Dec., 2019

1. Extended the arming parameter structures, i.e., **NET_DVR_SETUPALARM_PARAM** and **NET_DVR_SETUPALARM_PARAM_V50** :
added a member **bySubSCRIPTION** (subscription parameters) via one reserved byte.
2. Extended the constant alarm information structure **NET_DVR_ALRAM_FIXED_HEADER** :
added four members, i.e., **dwPicLen** (alarm picture size), **byPicURL** (picture data format),
byTarget (detection target type), and **pDataBuff** (alarm picture data or URL), to the sub
structure **struAlarmChannel** (alarm channel parameter structure) of the union **uStruAlarm**
(alarm information union).

Summary of Changes in Version 6.1.3.30_Jan., 2020

Extended message about version information of algorithm library **XML_AlgorithmsVersion** (related API: **NET_DVR_STDXMLConfig**, related URI: [/ISAPI/System/algorithmsVersion](#)):
added two algorithm library names "streetBehavior" (street behavior) and "workBehavior" (work
behavior) to the node <name> of <AlgorithmsVersionInfo>.

Summary of Changes in Version 6.1.3.25_Dec., 2019

1. Extended serial port capability **XML_SerialCap** (related API: **NET_DVR_STDXMLConfig** ; related
URI: [/ISAPI/System/Serial/capabilities](#)):
added one node <isSupportAuthenticationService> (whether the device supports serial port
authentication).
2. Added URIs of serial authentication service (related API: **NET_DVR_STDXMLConfig**):

- Get capability of serial authentication service: GET [/ISAPI/System/Serial/authentication/capabilities?format=json](#) .
 - Get or set parameters of serial authentication service: GET or PUT [/ISAPI/System/Serial/authentication?format=json](#) .
3. Extended device parameter structure [NET_DVR_DEVICEINFO_V40](#) :
added one member **byPassWordResetLevel** (whether to prompt the non-admin user to change the password) via one byte.
4. Extended the batch configuration of SNMP [XML_SNMP](#) (related API: [NET_DVR_STDXMLConfig](#) ; related URI: [/ISAPI/System/Network/SNMP](#)):
added one node <SNMPAdvanced> (advanced parameters).
5. Extended advanced parameters of SNMP [XML_SNMPAdvanced](#) (related API: [NET_DVR_STDXMLConfig](#) ; related URI: [/ISAPI/System/Network/SNMP/advanced](#)):
added one node <SNMPTrapReceiverList> (all trap address information of SNMP in version 3).
6. Add URIs of trap address configuration of SNMP in version 3, see [SNMP Access](#) for details.
7. Extended capability [XML_RTSPCertificateCap](#) and parameters [XML_RTSPCertificate](#) of device's RTSP authentication mode, and parameters about certificate types of the web service [XML_WebCertificate](#) (related API: [NET_DVR_STDXMLConfig](#) ; related URIs: [/ISAPI/Security/RTSPCertificate/capabilities](#) , [/ISAPI/Security/RTSPCertificate](#) and [/ISAPI/Security/webCertificate](#)):
added one node <SecurityAlgorithm> (authentication algorithm), which contains a sub node <algorithmType> (algorithm type).
8. Extended security capability of the device [XML_SecurityCap](#) (related API: [NET_DVR_STDXMLConfig](#) ; related URI: [/ISAPI/Security/capabilities](#)):
added one sub node <SecurityAlgorithm> (authentication algorithm) to the node <WebCertificateCap>.
9. Added URIs to configure log storage mode, see [Configure Log Storage Mode](#) for details.
10. Extended the log types in [HCNetSDK Log Types](#) :
added two minor types 0x2671-"MINOR_LOCAL_EZVIZ_OPERATION" (local EZVIZ operations) and 0x2672-"MINOR_REMOTE_EZVIZ_OPERATION" (remote EZVIZ operations) to operation logs (MAJOR_OPERATION-0x3).

Summary of Changes in Version 6.1.3.25_Dec., 2019

1. Added the function of searching for alarms or events, refer to [Search for Alarms or Events](#) .
2. Extended event capability message [XML_EventAbility](#) (related API: [NET_DVR_GetDeviceAbility](#) ; capability type: 0x011-"DEVICE_ABILITY_INFO"):
added a node <AlarmSearch> (capability of searching for alarm or event information).

Summary of Changes in Version 6.1.3.25_Dec., 2019

1. Added URIs to be transmitted by [NET_DVR_STDXMLConfig](#) to set clearing parameters of HDD storage space, see [HDD Clearing](#) for details.
2. Extended the condition and result of log search condition and result message (related API: [NET_DVR_STDXMLConfig](#) ; related URI: [/ISAPI/ContentMgmt/logSearch?format=json](#)):
added a node "operationType" (operation types) to [JSON_SearchCondition](#) .

- added two nodes "**operationType**" (operation types) and "**remarks**" (operation remarks) to [**JSON MatchList**](#).
3. Extended the capability ([**XML LockCap**](#)) and parameters ([**XML LockByName**](#)) of video file locking and unlocking (related API: [**NET DVR STDXMLConfig**](#); related URIs: [**/ISAPI/ContentMgmt/record/control/locks/capabilities**](#) and [**/ISAPI/ContentMgmt/record/control/locks/name**](#)):
added query parameters **security** and **iv** to encrypt the nodes of <recorderCode> and <policeCode>.
4. Extended the file search result ([**XML VideoPic CMSearchResult**](#); related API: [**NET DVR STDXMLConfig**](#); related URI: [**/ISAPI/ContentMgmt/search**](#)):
added three nodes to the sub node <mediaSegmentDescriptor>, i.e., <shootingTime> (captured or recorded time), <FileTime> (time duration of the video files), and <remarks> (file remarks).
5. Extended the capability ([**JSON Cap RemarkList**](#)) and parameters ([**JSON RemarkList**](#)) of adding remarks to video files by file name (related API: [**NET DVR STDXMLConfig**](#); related URIs: [**/ISAPI/ContentMgmt/record/control/remark/capabilities?format=json**](#) and [**/ISAPI/ContentMgmt/record/control/remark?format=json**](#)):
added three nodes, i.e., "recordName" (video file name), "recorderCode" (ID of device that collects data), and "policeCode" (ID of police that collects the data).

Summary of Changes in Version 6.1.3.20_Nov., 2019

1. Added the function of searching for recording start and end time, refer to [**Search for Recording Start and End Time by Channel**](#) for details.
2. Extended device storage capability message [**XML RacmCap**](#) (related API: [**NET DVR STDXMLConfig**](#); related URI: [**/ISAPI/ContentMgmt/capabilities**](#)):
Added a node <isSupportTimeSearch> (whether it supports searching for recording start and end time by channel).
3. Extended the log types in [**HCNetSDK Log Types**](#):
added two minor types 0x320-"MINOR_802_1X_AUTH_SUCC" (802.1x authentication succeeded) and 0x321-"MINOR_802_1X_AUTH_FAIL" (802.1x authentication failed) to major type 0x4 (additional information log).
4. Extended structure of single physical disk parameter [**NET DVR PHY DISK INFO**](#):
added one disk status type 11-"SMR disk does not support RAID" to the member **byStatus**.
5. Extended error codes in [**Device Network SDK Errors**](#):
added one error code 2269-NET_DVR_SMRDISK_NOT_SUPPORT_RAID (SMR disk does not support RAID).

Summary of Changes in Version 6.1.3.15_March, 2020

1. Extended the camera parameters capability [**XML CAMERAPARA**](#):
added seven modes to node <fusionMode> (fusion mode): "Visible" (visible light mode), "fusionB/W" (white and black mode), "city" (city mode), "jungle" (jungle mode), "deset" (desert mode), "sea" (sea mode), "snow" (snow mode);

- added two nodes <isSupportGPSControl> (whether device supports GPS control capability) and <gearRange> (number of ranges supported by the device).
2. Extended the structure of camera configuration parameters **NET_DVR_CAMERAPARAMCFG_EX** (related API: **NET_DVR_GetDVRConfig** with command 3368-"NET_DVR_GET_CCDPARAMCFG_EX", and **NET_DVR_SetDVRConfig** with command 3369-"NET_DVR_SET_CCDPARAMCFG_EX")
added seven modes to member **byFusionMode**: 3-visible light mode, 4-black and white fusion mode, 5-jungle (fusion color mode), 6-desert (fusion color mode), 7-snow (fusion color mode), 8-sea (fusion color mode), 9-city (fusion color mode);
added one member **byGPSEnabled** (whether to enable GPS) with one reserved byte.

Summary of Changes in Version 6.1.3.10_Jan., 2020

Extended device capability message **XML_DeviceCap** (related URI: [/ISAPI/System/capabilities](#) ; related API: **NET_DVR_STDXMLConfig**):
added two nodes: <isSupportCertificateCaptureEvent>> (whether it supports certificate capture and comparison events) and <isSupportAlgorithmsInfo> (whether it supports getting the algorithm library version information).

Summary of Changes in Version 6.1.3.10_Nov., 2019

1. Extended event capability supported by the channel **XML_ChannelEventCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: [/ISAPI/Event/channels/<ID>/capabilities](#)):
added one event type "dataPreAlarm" (traffic pre-alarm) to the node <eventType>.
2. Extended traffic monitoring configuration capability **JSON_TrafficMonitorCap** and traffic monitoring parameters **JSON_TrafficMonitor** (related API: **NET_DVR_STDXMLConfig** ; related URIs: [/ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor/capabilities?format=json](#) and [/ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor?format=json](#)):
 - added one data plan type "year" (annual plan) to the node **packageType**.
 - added one notification method "center" (upload to center) to the node **alertType**.
3. Extended network traffic monitoring status **JSON_TrafficMonitorStatus** (related API: **NET_DVR_STDXMLConfig** ; related URI: GET [/ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor/status?format=json](#)):
added 11 nodes: **todayUsedUpstreamTrafficValue** (today's upstream data usage), **todayUsedDownstreamTrafficValue** (today's downstream data usage), **monthUsedTrafficValue** (data usage of this month), **monthUsedUpstreamTrafficValue** (upstream data usage of this month), **monthUsedDownstreamTrafficValue** (downstream data usage of this month), **yearUsedTrafficValue** (data usage of this year), **yearUsedUpstreamTrafficValue** (upstream data usage of this year), **yearUsedDownstreamTrafficValue** (downstream data usage of this year), **enabled** (whether traffic monitoring is enabled), **packageType** (data plan type), and **alertView** (alarm value).
4. Extended network capability **XML_NetworkCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: GET [/ISAPI/System/Network/capabilities](#)):

- added one node <**isSupportWiredandWirelessTrafficMonitor**> (whether it supports monitoring traffic of wired NIC and wireless NIC).
5. Extended country index enumeration **COUNTRY_INDEX** :
added a member **COUNTRY_INVALID = 0xfd** (**byCountry** is invalid and you should use **CRIndex**).
 6. Added an enumeration about country or region index **CR_INDEX** .
 7. Extended log types in **HCNetSDK Log Types** :
added one minor type "MINOR_DATA_PREALARM_ALARM"- 0x127 (network traffic pre-alarm) to alarm logs.

Summary of Changes in Version 6.1.3.5_Nov., 2019

Extended PTZ control capability message **XML_PTZChanelCap** (related URI: [/ISAPI/PTZCtrl/channels/<ID>/capabilities](#) ; related API: **NET_DVR_STDXMLConfig**):

added a node <**isSupportPTZCtrlStatus**> (whether it supports getting the absolute PTZ position of the current channel).

1.5 Notices

This chapter shows the notices and additional information that may be used during the SDK integration.

- The design of text protocol adopts the RESTful style and is based on HTTP (Hyper Text Transport Protocol) or HTTPs. It can be transmitted on the application layer without packaging and processing, and the API provided by the SDK is **NET_DVR_STDXMLConfig** .
- For private protocol, the default service port No. for third-party platform to log in to the devices is 8000; for ISAPI Protocol, the default port No. for logging in to the devices is 80 (HTTP port) or 443 (HTTPs port). But for most of products (except some specific products, e.g., Face Recognition Server, etc.), login via private protocol is supported, so this login mode is recommended before the further integrations.

Chapter 2 Security Service

This part provides some programming guidelines to enhance the security of stream, accounts, and so on, during the integration.

Get Device Security Questions

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Security/questionConfiguration](#).

The question parameters are returned in the message [XML_SecurityQuestion](#) by the output parameter [IpOutputParam](#).

Set Device Security Questions

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/Security/questionConfiguration](#) and set the input parameter [IpInputParam](#) to the message [XML_SecurityQuestion](#).

Get Certificate Type Parameters of Web Service

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Security/webCertificate](#).

The certificate type parameters are returned in the message [XML_WebCertificate](#) by the output parameter [IpOutputParam](#).

Set Certificate Type Parameters of Web Service

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/Security/webCertificate](#) and set the input parameter [IpInputParam](#) to the message [XML_WebCertificate](#).

2.1 Configure Double Verification

Double verification helps to protect the critical video files of NVR/DVR by limiting playback and download. The basic concept is that two users should always be required to start playback and download. For example, when a normal user A (operator or guest) wants to play back the video of a channel which requires double verification, he/she should ask a double verification user to enter the correct user name and password for double verification.

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 2-1 Programming Flow of Configuring Double Verification



- Note**
- Only the admin user can configure double verification.
 - For admin user, double verification is not required.
 - The double verification user name and password is only for double verification, and cannot be used for login.
-

1. Optional: Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Security/capabilities](#) for getting the device security capability to check whether the double verification function is supported.

The device security capability is returned by the output parameter pointer **IpOutBuffer** in message [XML_SecurityCap](#).

2. Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/Security/doubleVerification?format=json](#) for enabling the double verification.



Before enabling double verification, you can check whether double verification is enabled by calling [NET_DVR_STDXMLConfig](#) to pass through the URL: GET [/ISAPI/Security/doubleVerification?format=json](#).

3. Optional: Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Security/doubleVerification/users/capabilities?format=json](#) for getting the capability of double verification user configuration.

4. Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: POST [/ISAPI/Security/doubleVerification/users?format=json](#) for adding a double verification user.



Before adding a double verification user, you can get all verification users by calling the [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Security/doubleVerification/users?format=json](#).

The ID of added double verification user is returned in [JSON_id](#).

5. Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/Security/doubleVerification/users/<ID>?format=json](#) for editing a specified double verification user.



Before editing the double verification user, you can get the user information by calling [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Security/doubleVerification/users/<ID>?format=json](#).

6. Optional: Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: DELETE [/ISAPI/Security/doubleVerification/users/<ID>?format=json](#) for deleting a specified double verification user.

7. Optional: Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Security/doubleVerification/UsersPermission/capabilities?format=json](#) for getting the capability of permission configuration for double verification users.

8. Call **NET_DVR_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/Security/doubleVerification/UsersPermission/<ID>?format=json** for setting the permission of a specified double verification user.
-



Note

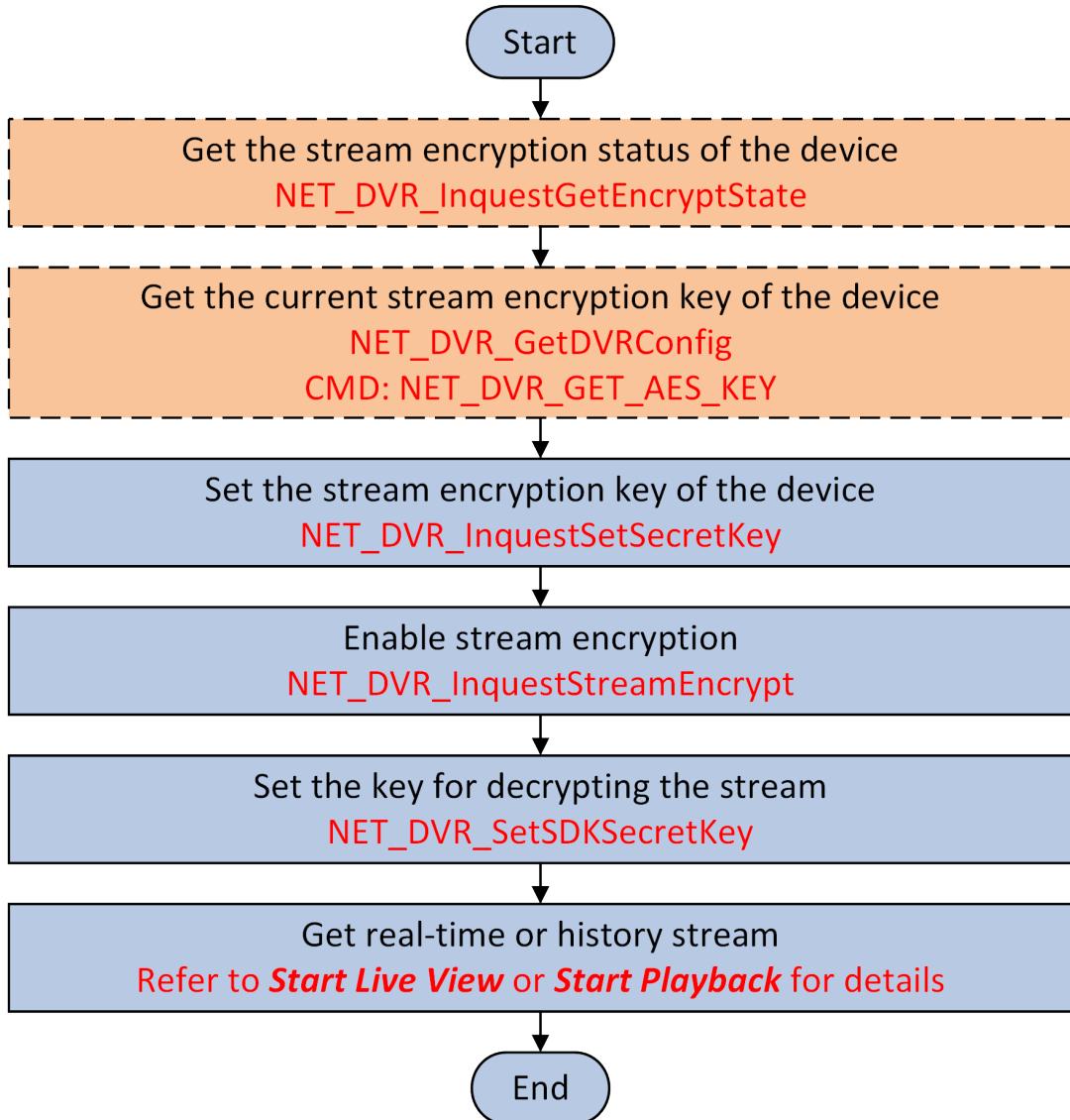
Before setting the permission of a double verification user, you can get the permission parameters by calling **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Security/doubleVerification/UsersPermission/<ID>?format=json**.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out off the device and release the resources.

2.2 Encrypt Stream

To improve the security of real-time or history stream from devices and comply with the laws and regulations of some parts of international markets, we have provided the stream encryption function based on AES algorithm to encrypt the stream.

Steps**Figure 2-2 API Calling Flow of Encrypting Stream**

If the stream is encrypted but the decryption key is not configured or an incorrect decryption key is configured, the display image will be black.

1. **Optional:** Call **NET_DVR_InquestGetEncryptState** to get the steam encryption status of the device for reference.
2. **Optional:** Call **NET_DVR_GetDVRConfig** with the command "NET_DVR_GET_AES_KEY" (command No.: 6113) to get the current stream encryption key of AES algorithm of the device for reference.



Note

The encryption key information is returned in the structure [NET_DVR_AES_KEY_INFO](#) by `IpOutBuffer`.

3. Call [NET_DVR_InquestSetSecretKey](#) to set the stream encryption key of the device.
4. Call [NET_DVR_InquestStreamEncrypt](#) to enable stream encryption.
5. Call [NET_DVR_SetSDKSecretKey](#) to set the key for decrypting the stream.
6. Get real-time or history stream. Refer to [Start Live View](#) or [Start Playback](#) for details.

2.3 Encrypt Stream Transmission Link

To improve the security of stream transmission between device and server, prevent man-in-the-middle attack, and comply with the laws and regulations of some parts of oversea markets, we have provided a encryption solution for different transmission links without directly encrypting stream.

2.3.1 (Optional) Verify Device

To meet different security requirements, before encrypting the transmission links, you can verify the logged device via the device root certificate or user-purchased certificate by calling HCNetSDK APIs to make sure the device is available or valid.



Note

- The device root certificate is for communication only. You can call [NET_DVR_RemoteControl](#) with [NET_DVR_CREATE_CERT](#) (command No.: 6138) to generate a custom certificate by device. The generated certificate is neutral with required information only.
- For the first time development, as there is no certificate provided during initialization, you'd better not verify the device, and the certificate verification callback function will directly return "TRUE" (verified).

Verify via Root Certificate

1. Call [NET_DVR_Login_V40](#) to log in to device.
2. Call [NET_DVR_StartDownload](#) with [NET_SDK_DOWNLOAD_CERT](#) (command No.: 0) to download the root certificate from device.
3. Call [NET_DVR_SetSDKLocalCfg](#) and set related parameter structure ([NET_DVR_LOCAL_CERTIFICATION](#)) to transmit the root certificate to the development program for verification.



Note

The verification results is returned in the structure [NET_DVR_CETTIFICATE_INFO](#) by the certificate verification callback function of structure [NET_DVR_LOCAL_CERTIFICATION](#).

Verify via User-Purchased Certificate

1. Call [NET_DVR_Login_V40](#) to log in to device.
 2. Call [NET_DVR_UploadFile_V40](#) with
UPLOAD_CERTIFICATE
(command No.: 1) to import the user-purchased certificate to the device.
 3. Call [NET_DVR_StartDownload](#) with
NET_SDK_DOWNLOAD_CERT
(command No.: 0) to download the root certificate from device.
 4. Call [NET_DVR_SetSDKLocalCfg](#) and set related parameter structure
([NET_DVR_LOCAL_CERTIFICATION](#)) to transmit the root certificate to the development program for verification.
-



Note

The verification results is returned in the structure [NET_DVR_CETTIFICATE_INFO](#) by the certificate verification callback function of structure [NET_DVR_LOCAL_CERTIFICATION](#).

2.3.2 Encrypt HTTPS Link

When the streaming mode is RTP over HTTPS, that is, the stream with RTP container format is transmitted on the link based on HTTPS, you can encrypt the HTTPS link to ensure the security of stream transmission.

Steps

1. Call [NET_DVR_GetDeviceAbility](#) and set the capability type to "DEVICE_SOFTHARDWARE_ABILITY" (macro definition value: 0x001) for getting the device software and hardware capability ([XML_BasicCapability](#)) to check if encrypting HTTPS link is supported by device.
If supported, the node <RtsپoverHttpsSupport> will be returned in the capability and its value is "1".
2. Enable HTTPS link encryption for live view or playback.
 - Call [NET_DVR_RealPlay_V40](#) and set **dwLinkMode** and **byProtoType** of live view parameter structure ([NET_DVR_PREVIEWINFO](#)) to "7" and "1", respectively, for enabling HTTPS link encryption during live view.
 - Call [NET_DVR_PlayBackByTime_V50](#) or [NET_DVR_PlayBackByName](#) to automatically enable HTTPS link encryption without setting any parameters during playback.
3. Call [NET_DVR_RealPlay_V40](#) or [NET_DVR_PlayBackByTime_V50](#) / [NET_DVR_PlayBackByName](#) to automatically enable HTTPS link encryption without setting any parameters during live view or playback.

2.3.3 Encrypt TLS Link

When the streaming mode is SDK over TLS, that is, the stream of private protocol communication is transmitted on the link based on TLS, you can encrypt the TLS link to ensure the security of stream transmission.

Steps

1. Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Security/adminAccesses/capabilities](#) for getting the device protocol capability ([XML_Cap_AdminAccessProtocolList](#)) to check if encrypting TLS link is supported by device.
If supported, the node <streamOverTls> is returned in the capability and its value is "true".
2. Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/Security/adminAccesses/<ID>](#) or [/ISAPI/Security/adminAccesses](#) and set the <streamOverTls> value to "true" for enabling TLS link encryption during live view of a device or multiple devices.
3. Call [NET_DVR_RealPlay_V40](#) or [NET_DVR_PlayBackByTime_V50](#) / [NET_DVR_PlayBackByName](#) to automatically enable TLS link encryption without setting any parameters during live view or playback.

Chapter 3 User Management

You can configure the user parameters, such as user name, password, local permissions, remote permissions, and so on.

Get User Management Capability

Call `NET_DVR_GetDeviceAbility`, set the capability type (`dwAbilityType`) to "DEVICE_USER_ABILITY" (macro definition value: 0x00c), and set the input pointer (`pInbuf`) to the message `XML_Desc_UserAbility`.

And the user configuration capability is returned in the message `XML_UserAbility` by the output pointer (`pOutBuf`).

Get User Parameters

Call `NET_DVR_GetDVRConfig` with the command of

`NET_DVR_GET_USERCFG_V52`

(command No.: 4194), the user information are returned in the structure by output buffer (`IpOutBuffer`).



The password will not be returned in the user information.

Set User Parameters

Call `NET_DVR_SetDVRConfig` with the command of

`NET_DVR_SET_USERCFG_V52`

(command No.: 4195), and set the input buffer (`IpInBuffer`) to .



If the password field is not configured, it indicates that the password will not be changed.

Chapter 4 Device Management and Maintenance

This chapter provides the management and maintenance description of device language, device upgrade, log search, password resetting, recovery email configuration, IoT device access, and so on.

4.1 Reset Password by Setting Recovery Email

A recovery email is added or configured for resetting the password as required. The admin user can set a recovery email after activating the device, and then receive the verification code from the manufacturer via the recovery email to reset the device password.

Before You Start

- Make sure you have called **NET_DVR_Init** to initialize the integration resources.
- Make sure you have called **NET_DVR_Login_V40** to log in to the device.

Steps

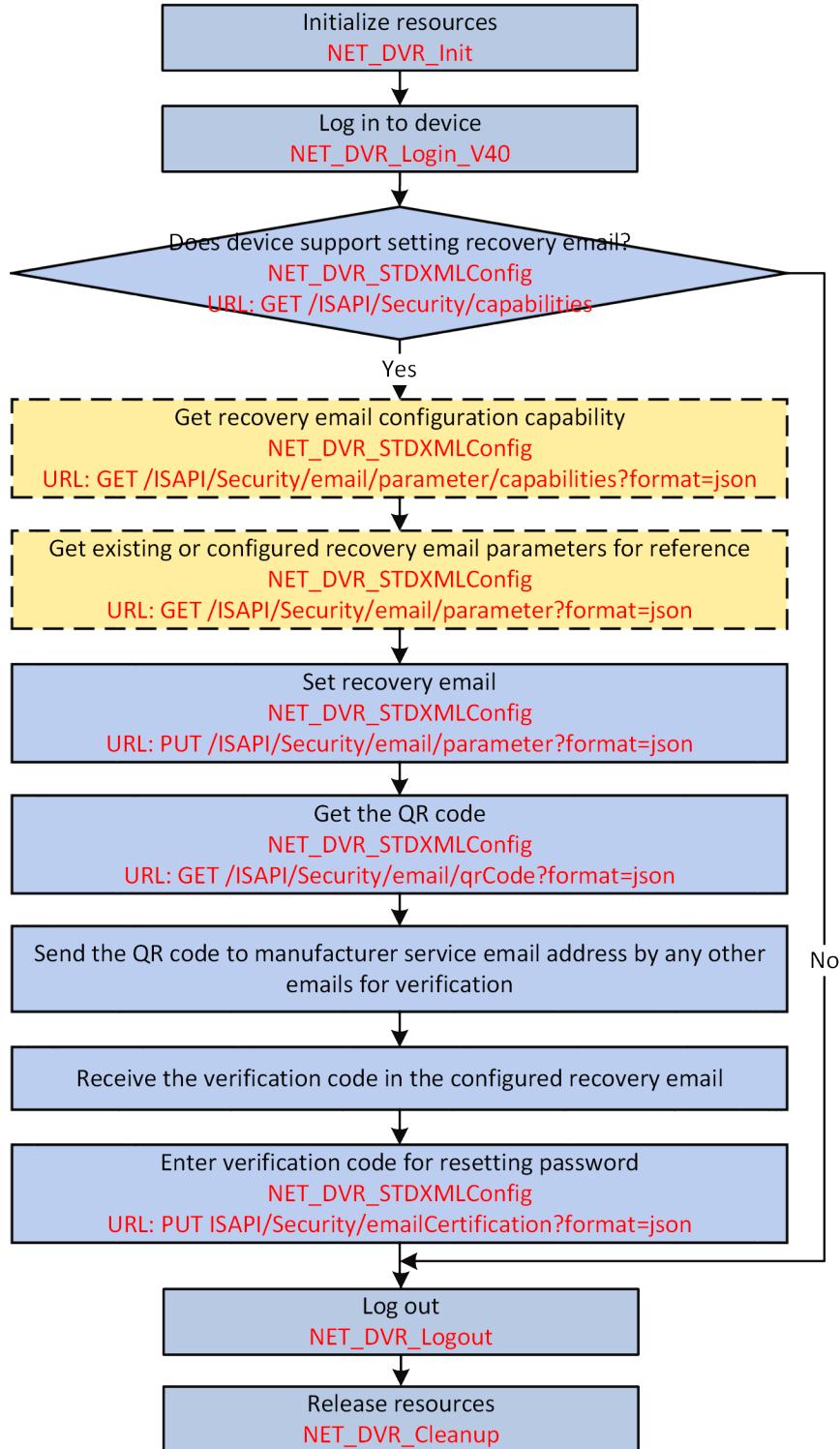


Figure 4-1 Programming Flow of Resetting Password by Setting Recovery Email



Note

You can reset password by other methods, such as answering security question, importing GUID files, and so on. But all methods should be supported by device, so you should call

NET_DVR_STDXMLConfig to pass through the request URL: GET **/ISAPI/Security/extern/capabilities** for getting the other security capability (**XML_externSecurityCap**) before choosing method to reset password.

1. Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Security/extern/capabilities** for checking if the device supports setting recovery email.

The security capability is returned in the message **XML_SecurityCap** by the output parameter pointer (**IpOutputParam**).

If the node <isSupportSecurityEmail> exists in the returned message and its value is "true", it indicates that setting recovery email is supported by the device, and you can continue to perform the following steps; otherwise, end this task.

2. **Optional:** Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Security/email/parameter/capabilities?format=json** for getting the recovery email configuration capability.

The recovery email configuration capability is returned in the message **JSON_SecurityEmailCap** by the output parameter pointer (**IpOutputParam**).

3. **Optional:** Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Security/email/parameter?format=json** for getting the existing or configured recovery email parameters for reference.

4. Call **NET_DVR_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/Security/email/parameter?format=json** and set the input parameter pointer (**IpInputParam**) to the message **JSON_SecurityEmail** for setting recovery email parameters.

5. Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Security/email/qrCode?format=json** for getting the QR code of the configured recovery email.

6. Send the QR code to manufacturer service email address by any other emails for verification.

7. Receive the verification code in the configured recovery email.

8. Call **NET_DVR_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/Security/emailCertification?format=json** and set the input parameter pointer (**IpInputParam**) to the message **JSON_EmailCertification** (with the received verification code entered) for resetting the device password.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out off the device and release the resources.

4.2 Remotely Upgrade Device

You can remotely upgrade device firmware and get the upgrade progress. The integration of device upgrade is supported both by private protocol and text protocol.

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

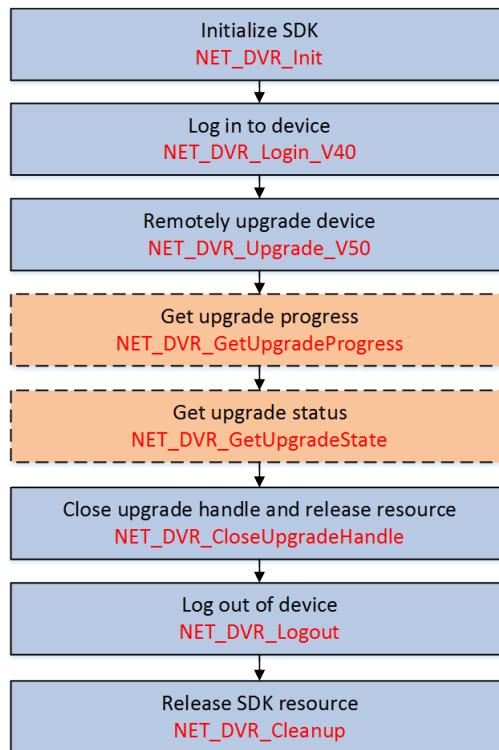


Figure 4-2 Programming Flow for Remotely Upgrading Device

1. Call [NET_DVR_Upgrade_V50](#) to upgrade the device remotely.
2. Optional: Call [NET_DVR_GetUpgradeProgress](#) to get the upgrade progress.
3. Optional: Call [NET_DVR_GetUpgradeState](#) to get the upgrade status.
4. Call [NET_DVR_CloseUpgradeHandle](#) to close the remote upgrade handle and release resources.

What to do next

Call [NET_DVR_Logout](#) and [NET_DVR_Cleanup](#) to log out of the device and release all resources.

4.3 Search for Logs

You can search for the device logs (including alarm, exception, operation, event, and so on) by time or type, and get the searched log information.

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

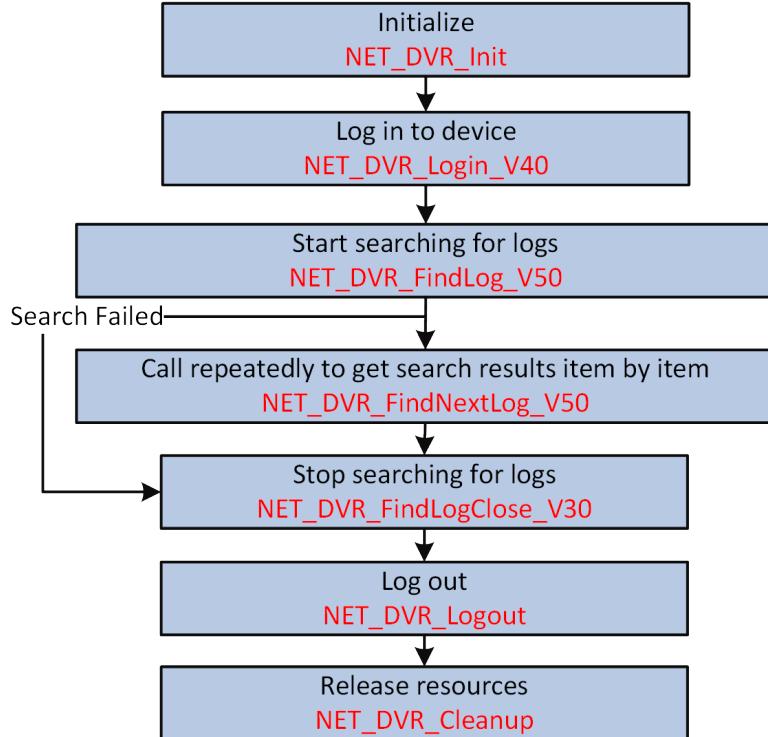


Figure 4-3 Programming Flow of Searching for Logs

1. Call **NET_DVR_FindDVRLog_V50** to start searching for logs.



Note

You can also call **NET_DVR_STDXMLConfig** to transmit **/ISAPI/ContentMgmt/logSearch?format=json** by POST method and set **lpInputParam** to **JSON_SearchCondition** to search for log files. The search results are returned in the message **JSON_MatchList** by **lpOutputParam**.

2. Call **NET_DVR_FindNextLog_V50** to get searched log information item by item.
3. Call **NET_DVR_FindLogClose_V30** to stop searching and release log search resource.

Example

Sample Code of Searching for Logs

```

#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"); //IP address
struLoginInfo.wPort = 80; //HTTP port
strcpy(struLoginInfo.sUserName, "admin"); //User name
strcpy(struLoginInfo.sPassword, "abcd1234"); //Password

//Device information, output parameters
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;
}

NET_DVR_FIND_LOG_COND struFindLogCond = { 0 };
struFindLogCond.dwMainType = 0;
struFindLogCond.dwSubType = 0;
struFindLogCond.dwSelectMode = 0;
struFindLogCond.bOnlySmart = FALSE;

struFindLogCond.struStartTime.wYear = 2019;
struFindLogCond.struStartTime.byMonth = 11;
struFindLogCond.struStartTime.byDay = 2;
struFindLogCond.struStartTime.byHour = 9;
struFindLogCond.struStartTime.byMinute = 0;
struFindLogCond.struStartTime.bySecond = 0;
//struFindLogCond.struStartTime.byISO8601 = 0;
//struFindLogCond.struStartTime.cTimeDifferenceH = 0;
//struFindLogCond.struStartTime.cTimeDifferenceM = 0;

struFindLogCond.struEndTime.wYear = 2019;
struFindLogCond.struEndTime.byMonth = 11;
struFindLogCond.struEndTime.byDay = 3;
struFindLogCond.struEndTime.byHour = 9;
struFindLogCond.struEndTime.byMinute = 0;
struFindLogCond.struEndTime.bySecond = 0;
//struFindLogCond.struEndTime.byISO8601 = 0;
//struFindLogCond.struEndTime.cTimeDifferenceH = 0;
//struFindLogCond.struEndTime.cTimeDifferenceM = 0;

//Search for logs
LONG lLogHandle = NET_DVR_FindDVRLog_V50(lUserID, &struFindLogCond);
```

```
if (!LogHandle < 0)
{
    printf("find log fail, last error %d \n", NET_DVR_GetLastError());
    return;
}

NET_DVR_LOG_V50 struLogInfo = {0};
while(true)
{
    LONG IRet = NET_DVR_FindNextLog_V50(LogHandle, &struLogInfo);
    if (IRet == NET_DVR_FILE_SUCCESS)
    {
        printf("log:%04d-%02d-%02d %02d:%02d:%02d\n", struLogInfo.struLogTime.wYear,
struLogInfo.struLogTime.byMonth, struLogInfo.struLogTime.byDay,\n
                                         struLogInfo.struLogTime.byHour, struLogInfo.struLogTime.byMinute,
struLogInfo.struLogTime.bySecond,);
    }
    else if (IRet == NET_DVR_ISFINDING)
    {
        printf("finding \n");
        Sleep(5);
        continue;
    }
    else if ((IRet == NET_DVR_NOMOREFILE) || (IRet == NET_DVR_FILE_NOFIND))
    {
        printf("find ending \n");
        break;
    }
    else
    {
        printf("find log fail for illegal get file state \n");
        break;
    }
}

//Stop searching for logs
if (LogHandle >= 0)
{
    NET_DVR_FindLogClose_V30(LogHandle);
}

//Log out
NET_DVR_Logout(userID);

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

What to do next

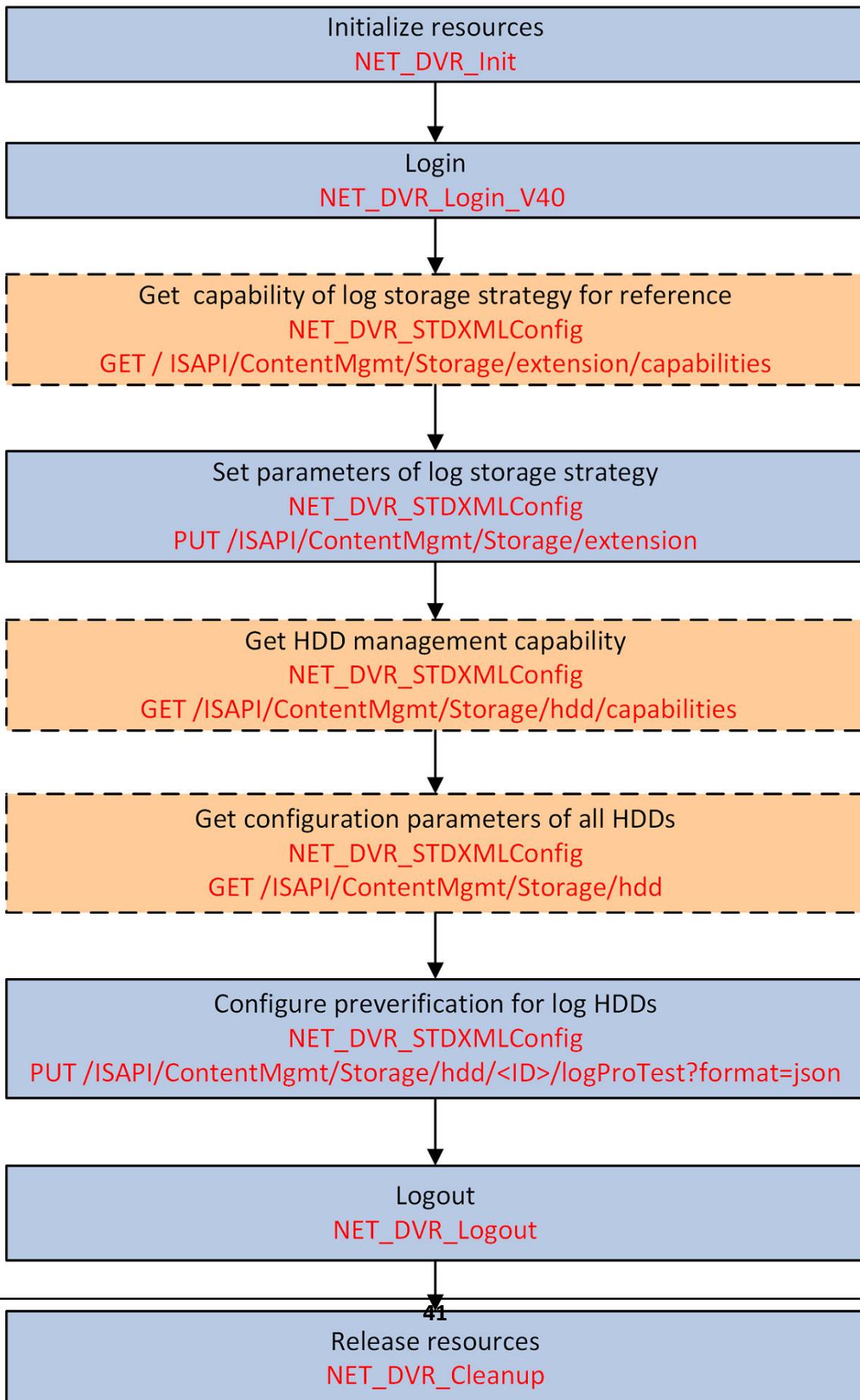
Call [NET_DVR_Logout](#) and [NET_DVR_Cleanup](#) to log out and release all resources.

4.4 Configure Log Storage Mode

You can save the log files in default mode or custom mode. In the default mode, each HDD can save up to 64 MB log files, and the storage space will automatically be overwritten when the space usage reaches 64 MB. In the custom mode, it is available to specify HDDs to save log files.

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 4-4 Programming Flow of Configuring Log Storage Mode**

- 1. Optional:** Call [NET_DVR_STDXMLConfig](#) to transmit the request URI: /ISAPI/ContentMgmt/Storage/extension/capabilities by GET method to get the capability of log storage strategy for reference.

The capability is returned in the message [XML_Cap_storageExtension](#) by IpOutputParam.

- 2.** Call [NET_DVR_STDXMLConfig](#) to transmit the request URI /ISAPI/ContentMgmt/Storage/extension by PUT method, and set the node <logStorageMode> as "system" (system default mode) or "custom" (custom mode) in the message [XML_storageExtension](#).

- 3. Optional:** Call [NET_DVR_STDXMLConfig](#) to transmit the request URI: /ISAPI/ContentMgmt/Storage/hdd/capabilities by GET method to get the HDD management capability.

The capability is returned in the message [XML_Cap_hddList](#) by IpOutputParam.

- 4. Optional:** Call [NET_DVR_STDXMLConfig](#) to transmit the request URI: /ISAPI/ContentMgmt/Storage/hdd by GET method to get configuration parameters of all HDDs.

The parameters are returned in the message [XML_hddList](#) by IpOutputParam.

- 5.** Call [NET_DVR_STDXMLConfig](#) to transmit the request URI: /ISAPI/ContentMgmt/Storage/hdd/<ID>/logProTest?format=json by PUT method to configure preverification for log HDDs.

What to do next

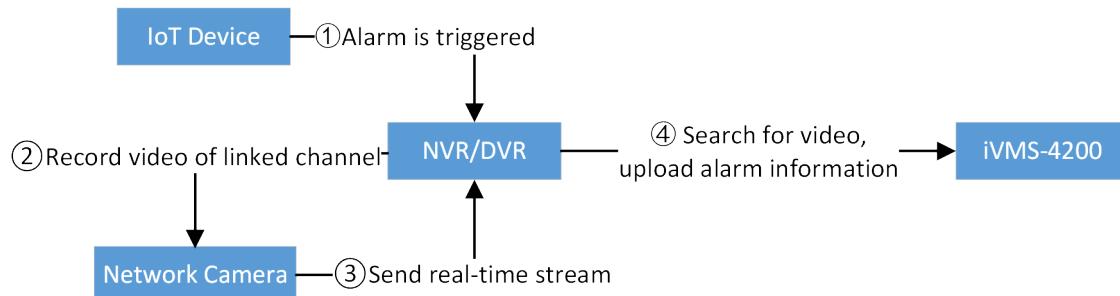
Call [NET_DVR_Logout](#) and [NET_DVR_Cleanup](#) to log out of the device and release all resources.

4.5 IoT Devices Access

The IoT devices here include access control terminal, security control panel, video intercom, analog camera RF, etc. The NVR/DVR supports accessing IoT devices for device management, status search, event/alarm linkage, and alarm receiving. When the IoT devices are added to NVR/DVR, the NVR/DVR can receive the alarm of IoT device, link the channel for recording, and receive the real-time stream when alarm is triggered.

This chapter mainly introduces the methods of adding IoT devices, and configuring event/alarm of IoT devices.

Application Scenario



4.5.1 Add IoT Device to NVR/DVR

For convenient management of IoT device, such as status searching, event/alarm linkage, alarm receiving, and so on, you should add the IoT device to NVR/DVR first.

Before You Start

- Make sure you have called [NET_DVR_Init](#) to initialize the integration resources.
- Make sure you have called [NET_DVR_Login_V40](#) to log in to the device.

Steps

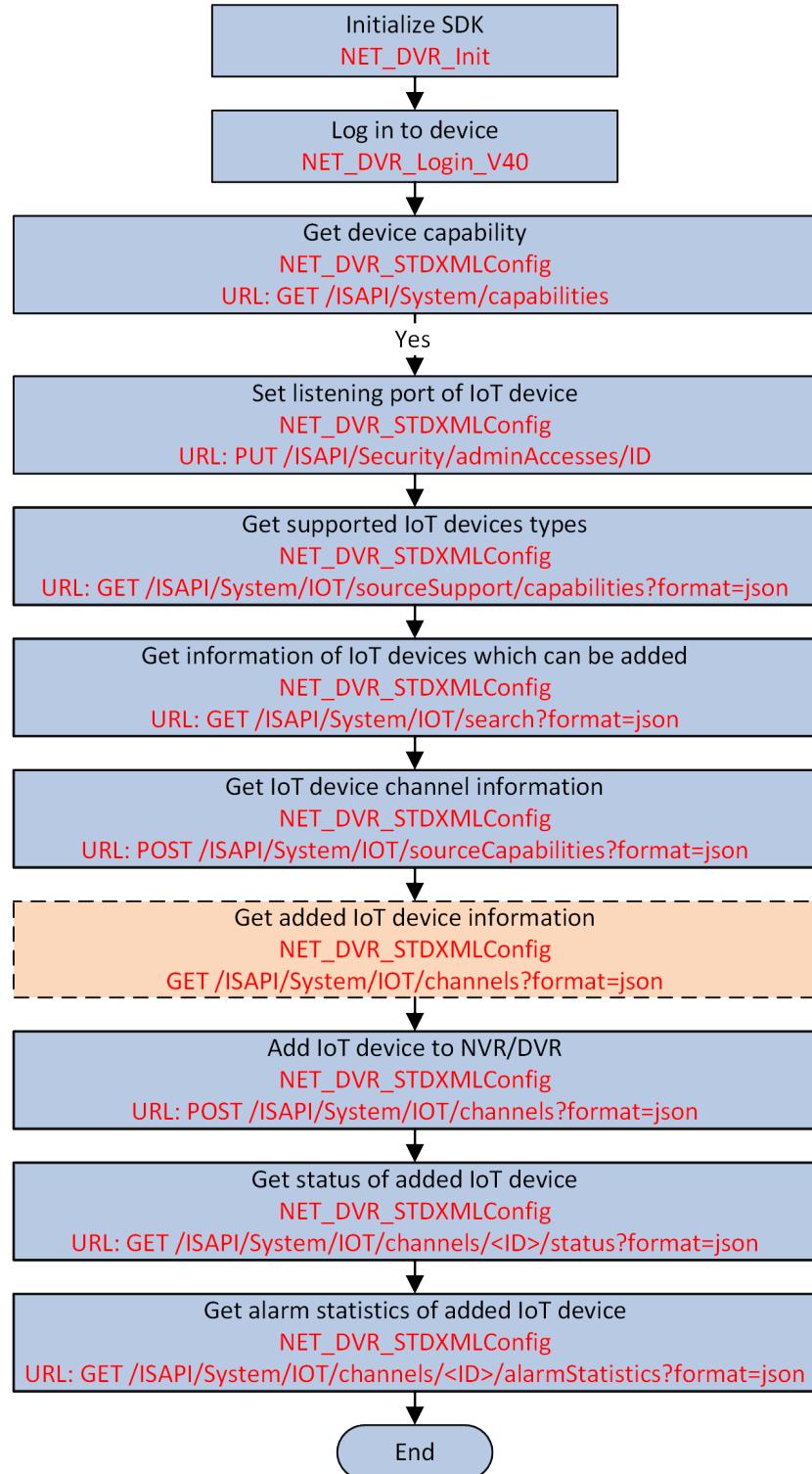


Figure 4-5 Programming Flow of Adding IoT Device to NVR/DVR

1. Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/capabilities** for getting the device capability to check if the IoT device access function is supported via the node <IOTCap>.

The device capability is returned in the message of **XML_DeviceCap**.

2. Call **NET_DVR_STDXMLConfig** to pass through the URL: PUT **/ISAPI/Security/adminAccesses/<ID>**, and set the value of <protocol> in **XML_AdminAccessProtocol** to "IOT" for configuring listening port before adding the OPTEX and Luminite IoT devices.
3. Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/IOT/sourceSupport/capabilities?format=json** for getting the supported IoT devices types of device.

The supported IoT devices types are returned in **JSON_IOTSourceSupport**.

4. Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/IOT/search?format=json** for getting the information of IoT devices which can be added.

The IoT devices which can be added is returned in **JSON_IOTSourceList**.

5. Call **NET_DVR_STDXMLConfig** to pass through the URL: POST **/ISAPI/System/IOT/sourceCapabilities?format=json** for getting the number of IoT device channels.

The supported source capability is returned in **JSON_SourceCapabilities**.

6. **Optional:** Get the information of added IoT devices.

- Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/IOT/channels?format=json** for getting the information of added IoT devices.
- Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/IOT/channels?format=json&deviceInductiveType=** for getting the information of added IoT devices according to inductive type.

7. Call **NET_DVR_STDXMLConfig** to pass through the URL: POST **/ISAPI/System/IOT/channels?format=json** for adding the IoT device to NVR/DVR.

8. Call **NET_DVR_STDXMLConfig** to pass through the URL: PUT **/ISAPI/System/IOT/channels/<ID>/basicParam?format=json** for setting the basic parameters of added IoT device according to channel ID.

9. **Optional:** Call **NET_DVR_STDXMLConfig** to pass through the URL: PUT **/ISAPI/System/IOT/channels/<ID>?format=json** for setting the specified added IoT device according to channel ID.



Note

Before setting the specified IoT device, you can get the IoT device by the URL: GET **/ISAPI/System/IOT/channels/<ID>?format=json**.

10. **Optional:** Delete the added IoT device.

- Call **NET_DVR_STDXMLConfig** to pass through the URL: DELETE **/ISAPI/System/IOT/channels/<ID>?format=json** for deleting the specified IoT channel.
- Call **NET_DVR_STDXMLConfig** to pass through the URL: DELETE **/ISAPI/System/IOT/channels/<ID>/all?format=json** for deleting all channels (video channel and IoT channel) of added IoT device.

11. Get the status of added IoT device.

- Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/IOT/channels/status?format=json** for getting the status of added IoT device.

- Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/IOT/channels/status?format=json&deviceInductiveType=** for getting the status of added IoT device according to inductive type.
 - Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/IOT/channels/<ID>/status?format=json** for getting the status of specified channel of added IoT device.
12. Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/IOT/channels/<ID>/alarmStatistics?format=json** for getting the alarm statistics of added IoT device according to channel ID.
13. **Optional:** Call **NET_DVR_STDXMLConfig** to pass through the URL: POST **/ISAPI/System/accessDevice/associatedChannel?format=json** by getting the linked channel information of added IoT device.
14. **Optional:** Import or export the list of added IoT devices.

- Import** Call **NET_DVR_STDXMLConfig** to pass through the URL: PUT **/ISAPI/System/IOT/channelConfig?format=json** for importing the list of added IoT devices.
- Export** Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/IOT/channelConfig?format=json** for exporting the list of added IoT devices.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release the resources.

4.5.2 Configure Alarm of IoT Device

When the IoT device is added to NVR/DVR, you can configure the alarm parameters such as arming schedule, linkage method, OSD, and so on for receiving the alarm of IoT device.

Before You Start

- Make sure you have called **NET_DVR_Init** to initialize the integration resources.
- Make sure you have called **NET_DVR_Login_V40** to log in to the device.
- Make sure the IoT device is added to NVR/DVR.

Steps

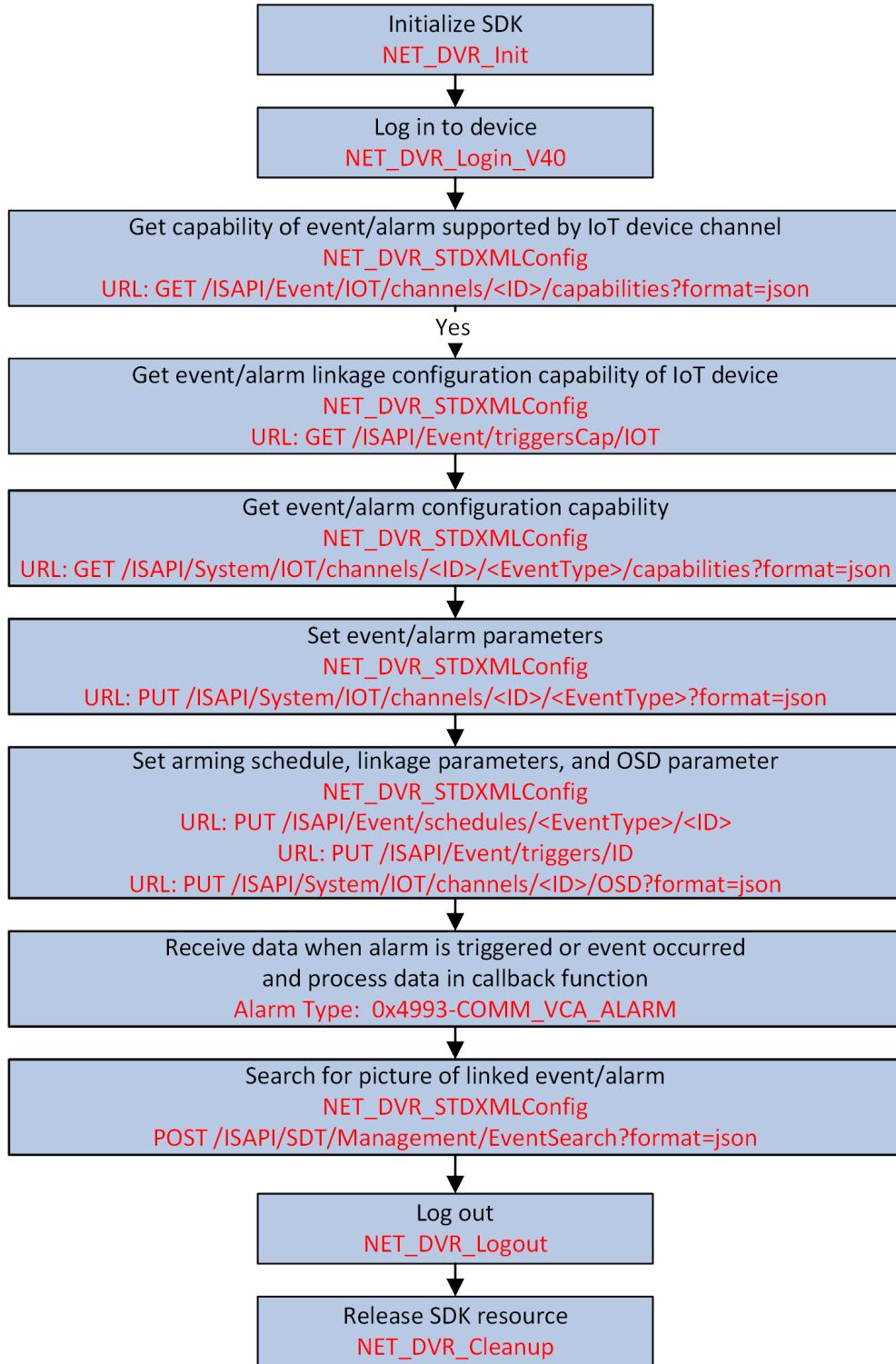


Figure 4-6 Programming Flow of Configuring Alarm of IoT Device

1. Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/Event/IOT/channels/<ID>/capabilities?format=json** for getting the IoT device channel capability.
The event capability supported by IoT device channel is returned in **JSON_IOTChannelEventCap**.
 2. Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/Event/triggersCap/IOT** for getting the event/alarm linkage configuration capability of IoT device.
The capability is returned in **XML_IOTTriggersCap**.
 3. Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/IOT/channels/<ID>/<EventType>/capabilities?format=json** for getting the event configuration capability.
The event configuration capability is returned in **JSON_XXCap**.
 4. Call **NET_DVR_STDXMLConfig** to pass through the URL: PUT **/ISAPI/System/IOT/channels/<ID>/<EventType>?format=json** for setting the event/alarm parameters.
 5. Call **NET_DVR_STDXMLConfig** to pass through the URL: PUT **/ISAPI/Event/schedules/<EventType>/<ID>** for setting the arming schedule.
 6. Call **NET_DVR_STDXMLConfig** to pass through the URL: PUT **/ISAPI/Event/triggers/<eventType>-<channelID>** for setting the event/alarm linkage parameters.
 7. Configure OSD.
 - 1) Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/IOT/channels/<ID>/OSD/capabilities?format=json** for getting the OSD configuration capability.
 - 2) Call **NET_DVR_STDXMLConfig** to pass through the URL: GET **/ISAPI/System/IOT/channels/<ID>/OSD?format=json** for getting the OSD parameters.
 - 3) Call **NET_DVR_STDXMLConfig** to pass through the URL: PUT **/ISAPI/System/IOT/channels/<ID>/OSD?format=json** for setting the OSD parameters.
 8. Receive event/alarm information, and the alarm type is "0x4993-COMM_VCA_ALARM".
 - Receive event/alarm in arming mode, see details in **Receive Alarm/Event in Arming Mode**.
 - Receive event/alarm in listening mode, see details in **Receive Alarm/Event in Listening Mode**.
 9. Call **NET_DVR_STDXMLConfig** to pass through the URL: POST **/ISAPI/SDT/Management/EventSearch?format=json** for searching for the picture of linked event/alarm.
-



Before searching for event, you can get the event search capability by the URL: GET **/ISAPI/SDT/Management/EventSearch/capabilities?format=json**.

10. **Optional:** Call **NET_DVR_STDXMLConfig** to pass through the URL: POST **/ISAPI/System/IOT/linkageChannels?format=json** for getting linked channel of searched event.
-

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release the resources.

4.6 Online Upgrade Device via Guarding Vision

For the devices that can connect to Guarding Vision, you can online upgrade their firmware via Guarding Vision, and get the upgrade progress. Besides, in the condition of bad network, you can

also enable automatic download of upgrade package in the background to improve the upgrade speed.

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

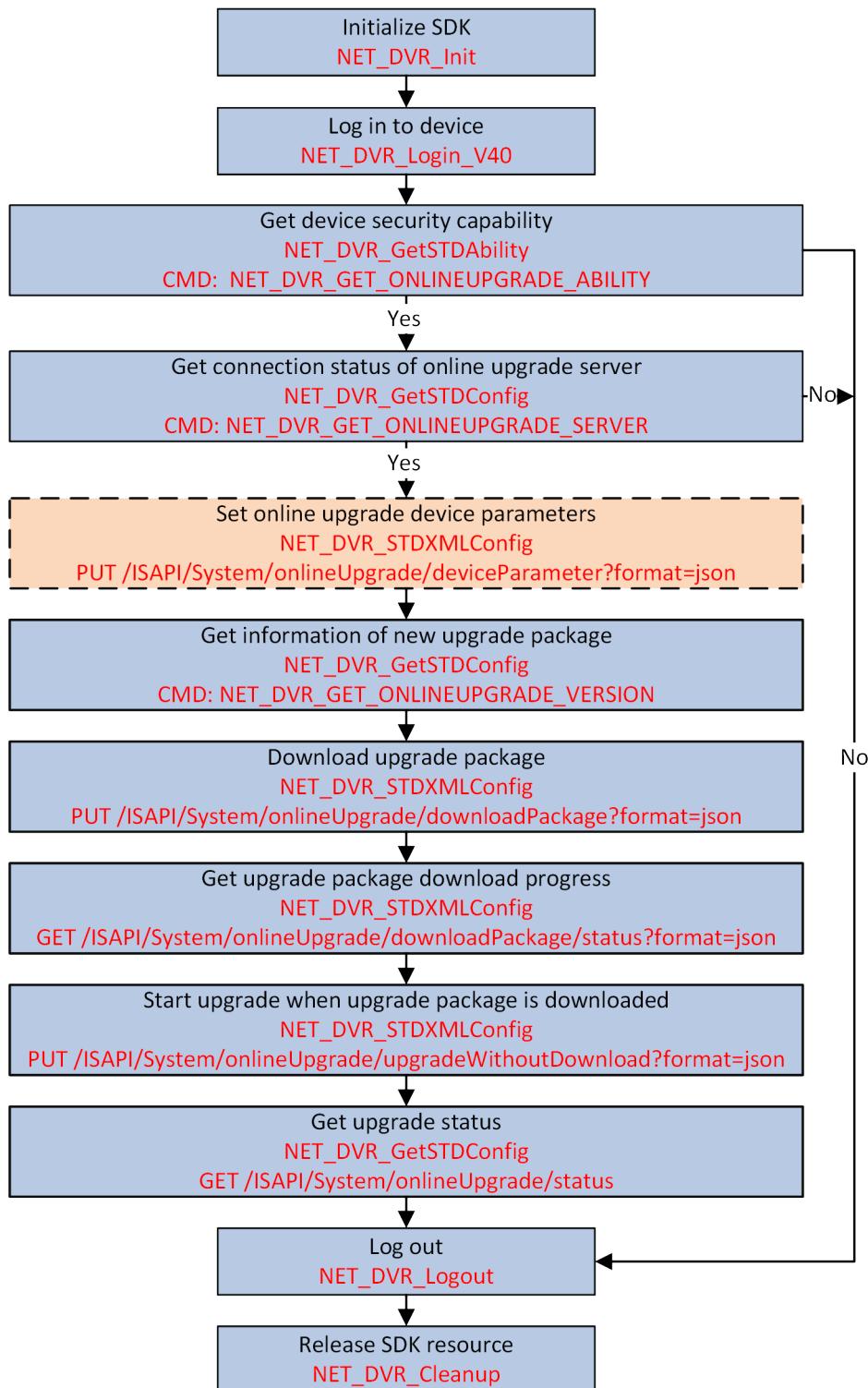


Figure 4-7 Programming Flow of Online Upgrading Device via Guarding Vision

1. Call **NET_DVR_GetSTDAbility** with the command of

NET_DVR_GET_ONLINEUPGRADE_ABILITY

(command No.: 9309) to get the device support online upgrade capability set, and check if the device support this function.

The online upgrade capability set is returned in **XML_OnlineUpgradeCap** by output buffer **IpOutBuffer** of structure .

2. Call **NET_DVR_GetSTDConfig** with the command of

NET_DVR_GET_ONLINEUPGRADE_SERVER

(command No.: 9304) and set the condition parameter **IpCondBuffer** of structure to "Null" for getting the connection status of online upgrade server.



Note

Only when the online upgrade server is connected, you can go on for next step. Otherwise, you should end this task.

The upgrade server connection status is returned in structure by output buffer **IpOutBuffer** of structure .

3. **Optional:** Call **NET_DVR_STDXMLConfig** to pass through the request URL: **PUT /ISAPI/System/onlineUpgrade/deviceParameter?format=json** , and set the input parameter pointer **IpInParam** to **JSON_OnlineUpgradeParameter** for setting the device online upgrade parameters.



Note

Before setting the device online upgrade parameters, you can get these parameters via calling **NET_DVR_STDXMLConfig** to pass through the request URL: **GET /ISAPI/System/onlineUpgrade/deviceParameter?format=json** .

4. Call **NET_DVR_GetSTDConfig** with the command of

NET_DVR_ONLINEUPGRADE_VERSION_COND

(command No.: 9305) and set the input buffer **IpInBuffer** of structure to structure for getting the new upgrade package information.

The new upgrade package information is returned in the structure by output buffer **IpOutBuffer** of structure .

5. Call **NET_DVR_STDXMLConfig** to pass through the request URL: **PUT /ISAPI/System/onlineUpgrade/downloadPackage?format=json** for starting to download upgrade package to device.

6. **Optional:** During downloading the upgrade package, you can perform the following operations.

Pause	Call <u>NET_DVR_STDXMLConfig</u> to pass through the request URL: PUT /ISAPI/System/onlineUpgrade/downloadPackage/pause?format=json
Download	<u>ISAPI/System/onlineUpgrade/downloadPackage/pause?format=json</u>
Resume	Call <u>NET_DVR_STDXMLConfig</u> to pass through the request URL: PUT /ISAPI/System/onlineUpgrade/downloadPackage/resume?format=json
Download	<u>ISAPI/System/onlineUpgrade/downloadPackage/resume?format=json</u>
Cancel	Call <u>NET_DVR_STDXMLConfig</u> to pass through the request URL: DELETE /ISAPI/System/onlineUpgrade/downloadPackage?format=json
Download	<u>ISAPI/System/onlineUpgrade/downloadPackage?format=json</u>

7. Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/System/onlineUpgrade/downloadPackage/status?format=json** for getting the upgrade package download progress.
The upgrade package download progress is returned in **JSON_DownloadPackageStatus** by output parameter pointer **IpOutputParam**.
8. Call **NET_DVR_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/System/onlineUpgrade/upgradeWithoutDownload?format=json** for starting upgrade when the upgrade package is downloaded.
9. Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/System/onlineUpgrade/status** for getting the upgrade status.

4.7 Configuration and Maintenance

4.7.1 Automatic Maintenance

Function	API
Get configuration capability of automatic maintenance	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/autoMaintenance/capabilities?format=json</u> with GET method.</p> <p>The configuration capability is returned in the message <u>JSON_AutoMaintenanceCap</u> by IpOutputParam.</p> <p> Note</p> <p>To check whether the device supports automatic maintenance, you can call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/capabilities</u> with GET method to get the device capability (<u>XML_DeviceCap</u>). If supports, the node <isSupportAutoMaintenance> will be returned and its value is true.</p>
Get automatic maintenance parameter	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/autoMaintenance?format=json</u> with GET method.</p> <p>The configuration parameter is returned in the message <u>JSON_AutoMaintenance</u> by IpOutputParam.</p>
Set automatic maintenance parameters	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/autoMaintenance?format=json</u> with PUT method and set IpInputParam to the message <u>JSON_AutoMaintenance</u>.</p>

4.7.2 Basic Maintenance

Function	API
Get device working status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URL: GET <u>/ISAPI/System/workingstatus?format=json</u> . And the device status is returned in the message <u>JSON_WorkingStatus</u> by output parameter pointer (IpOutputParam).
Enable/Disable device status checking	Enable: <u>NET_DVR_StartGetDevState</u> Disable: <u>NET_DVR_StopGetDevState</u>
Receive device status changes	Refer to <u>Receive Alarm/Event in Arming Mode</u> for the integration process of uploading status changes, and refer to <u>Subscribe Alarm/Event in Arming Mode</u> for the integration process of uploading capability changes. And then the command (ICommand) to upload status changes in the API <u>NET_DVR_SetDVRMessageCallBack_V50</u> should be set to COMM_DEV_STATUS_CHANGED (command No.: 0x7000).  Note To check if uploading status changes is supported by device, you can call <u>NET_DVR_GetSTDAbility</u> with capability type "NET_DVR_GET_SYSTEM_CAPABILITIES" to get the device capability (<u>XML_DeviceCap</u>). If supports, the node <isSupportChangedUpload> is returned.
Restore to factory settings	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/factoryReset?mode=</u> with PUT method.
Reboot the device	<u>NET_DVR_RebootDVR</u>
Get device algorithm version	<ul style="list-style-type: none"> Option 1: Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/algorithmsVersion</u> with GET method. The algorithm version information is returned in the message <u>XML_AlgorithmsVersion</u> by IpOutputParam. Option 2: Call <u>NET_DVR_GetSTDConfig</u> with "NET_DVR_GET_VCA_VERSION_LIST" (command No.: 3973) and set IpCondBuffer in the structure <u>NET_DVR_STD_CONFIG</u> to NULL.

Function	API
	The algorithm version information (<u>NET_DVR_VCA_VERSION_LIST</u>) is returned by IpOutBuffer in the structure <u>NET_DVR_STD_CONFIG</u> .
Get device information	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/deviceInfo</u> with GET method. The device information is returned in the message <u>XML_DeviceInfo</u> by IpOutputParam .
Get capability of exporting device diagnose information	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/diagnosedData/parameter/capabilities</u> with GET method. The capability is returned in the message <u>XML_Cap_DiagnosedDataParameter</u> by IpOutputParam .
Get parameters of exporting device diagnose information	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/diagnosedData/parameter</u> with GET method. The parameters are returned in the message <u>XML_DiagnosedDataParameter</u> by IpOutputParam .
Set parameters of exporting device diagnose information	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/diagnosedData/parameter</u> with PUT method and set IpInputParam to the message <u>XML_DiagnosedDataParameter</u> .
Get diagnostic server capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/diagnosedData/server/capabilities?format=json</u> with GET method. The capability is returned in the message <u>JSON_DiagnosedDataServerCap</u> by IpOutputParam .
Get diagnostic server parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/diagnosedData/server?format=json</u> with GET method. The parameters are returned in the message <u>JSON_DiagnosedDataServerList</u> by IpOutputParam .
Set diagnostic server parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/diagnosedData/server?format=json</u> with PUT method and set IpInputParam to the message <u>JSON_DiagnosedDataServerList</u> .
Test diagnostic server	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/diagnosedData/server/test?format=json</u> with POST method.

4.7.3 Basic Settings

Table 4-1 Device Basic Parameters

Function	API
Get device basic parameters	Call <u>NET_DVR_GetDVRConfig</u> with "NET_DVR_GET_DEVICECFG_V50" (command No.: 3801). The device basic parameters are returned in the structure <u>NET_DVR_DEVICECFG_V50</u> by lpOutBuffer .
Set device basic parameters	Call <u>NET_DVR_SetDVRConfig</u> with "NET_DVR_SET_DEVICECFG_V50" (command No.: 3802) and set lpInBuffer to the structure <u>NET_DVR_DEVICECFG_V50</u> .

Table 4-2 Device Language

Function	API
Get device language configuration capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/DeviceLanguage/capabilities</u> with GET method. The capability is returned in the message <u>XML_Cap_DeviceLanguage</u> by the output parameter lpOutputParam .
Get device language	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/DeviceLanguage</u> with GET method.
Set device language	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/DeviceLanguage</u> with PUT method and set lpInputParam to the message <u>XML_DeviceLanguage</u> .

Table 4-3 Configuration Files

Function	API
Securely import configuration files to device	Call <u>NET_DVR_UploadFile_V40</u> , set the file type dwUploadType to "UPLOAD_SECURITY_CFG_FILE", and set lpInBuffer to the structure <u>NET_DVR_SECURITY_CFG_FILE_COND</u> . The device basic parameters are returned in the structure <u>NET_DVR_DEVICECFG_V50</u> by lpOutBuffer .

Function	API
	 Note You can check whether the device supports importing calibration file by calling API <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/capabilities</u> with GET method. When the node isSupportCalibrationFile exists and its value is true in the returned device capability (<u>XML_DeviceCap</u>), it indicates the function is supported.
Get configuration files importing progress	Call <u>NET_DVR_GetUploadState</u> .
Stop importing configuration files	Call <u>NET_DVR_UploadClose</u> .

Table 4-4 Camera Software Service

Function	API
Get configuration capability of software service	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/Software/channels/<ID>/capabilities</u> with GET method. The configuration capability is returned in the message <u>XML_Cap_SoftwareService</u> by IpOutputParam .
Get software service parameter	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/Software/channels/<ID></u> with GET method. The configuration parameter is returned in the message <u>XML_SoftwareService</u> by IpOutputParam .
Set software service parameter	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/Software/channels/<ID></u> with PUT method and set IpInputParam to the message <u>XML_SoftwareService</u> .

Table 4-5 Device Attribute Code

Function	API
Get device attribute code	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/deviceInfo/characteristicCode?format=json</u> with GET method. The parameters are returned in the message <u>JSON_Characteristic</u> by IpOutputParam .

4.7.4 Device Calibration

Table 4-6 Accelerometer Zero Bias Calibration

Function	API
Get device capability to check whether device supports accelerometer zero bias calibration	<p>Call <i>NET_DVR_STDXMLConfig</i> to transmit <i>/ISAPI/System/capabilities</i> with GET method.</p> <p>The device capability is returned in the message <i>XML_DeviceCap</i> by <i>lpOutputParam</i>.</p> <p>If supports, the node <i><isSupportZeroBiasCalibration></i> is returned in the message <i>XML_DeviceCap</i> and its value is true.</p>
Calibrate accelerometer zero bias	Call <i>NET_DVR_STDXMLConfig</i> to transmit <i>/ISAPI/System/zeroBiasCalibration/channels/<ID>?format=json</i> by PUT method.

Table 4-7 Installation Angle Calibration

Function	API
Get device capability to check whether device supports installation angle calibration	<p>Call <i>NET_DVR_STDXMLConfig</i> to transmit <i>/ISAPI/System/capabilities</i> with GET method.</p> <p>The device capability is returned in the message <i>XML_DeviceCap</i> by <i>lpOutputParam</i>.</p> <p>If supports, the node <i><isSupportInstallationAngleCalibration></i> is returned in the message <i>XML_DeviceCap</i> and its value is true.</p>
Get capability of installation angle calibration	Call <i>NET_DVR_STDXMLConfig</i> to transmit <i>/ISAPI/System/installationAngleCalibration/channels/<ID>/capabilities?format=json</i> with GET method.
Get installation angle calibration status or calibrate installation angle	Call <i>NET_DVR_STDXMLConfig</i> to transmit <i>/ISAPI/System/installationAngleCalibration/channels/<ID>?format=json</i> with GET or PUT method.

Table 4-8 Laser Optical Axis

Function	API
Get configuration capability of laser optical axis	Call <i>NET_DVR_STDXMLConfig</i> to transmit <i>/ISAPI/System/laserOpticalAxis/capabilities?format=json</i> with GET method.

Function	API
	The device capability is returned in the message <u>JSON_LaserOpticalAxisCap</u> by IpOutputParam .
Get laser optical axis parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/laserOpticalAxis?format=json</u> with GET method. The parameters are returned in the message <u>JSON_LaserOpticalAxis</u> by IpOutputParam .
Set laser optical axis parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/laserOpticalAxis?format=json</u> with GET or PUT methd and set IpInputParam to the message <u>JSON_LaserOpticalAxis</u> .
Adjust laser optical axis	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/laserOpticalAxis/goto?format=json</u> with PUT method and set IpInputParam to the message <u>JSON_LaserOpticalAxis_direction</u> .

4.7.5 Configure Camera Correction

If the image distortion occurs, you can configure camera parameters (such as focal length, camera position, lens distortion coefficient, etc.) to correct the lens distortion for getting the normal images.

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

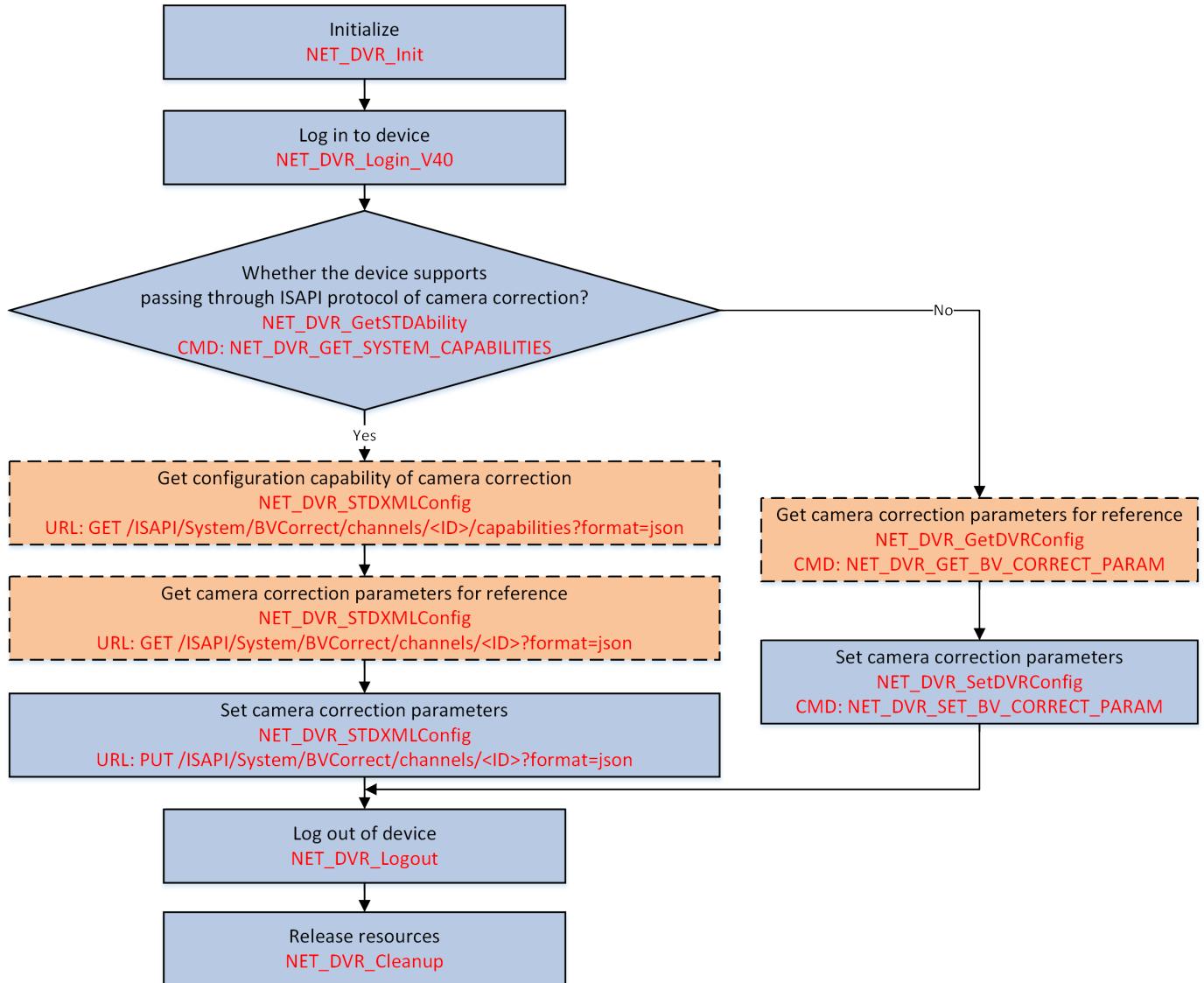


Figure 4-8 Programming Flow of Configuring Camera Correction

1. Call **NET_DVR_GetSTDAbility**, and set the capability type (**dwAbilityType**) to "NET_DVR_GET_SYSTEM_CAPABILITIES" (value: 8100) for getting the device capability to check if the device supports passing through ISAPI protocol of camera correction.
 - The node **<isSupportBVCorrect>** is returned in the capability message **XML_DeviceCap** and its value is "true": the device supports passing through ISAPI protocol of camera correction, and continue to perform the steps below.
 - a. Call **NET_DVR_STDXMLConfig** to pass through the request URL: **GET /ISAPI/System/BVCorrect/channels/<ID>/capabilities?format=json** to get the configuration capability of camera correction and know the supported parameters.

The configuration capability is returned in the message **JSON_BVCorrectParamCap** by **IpOutputParam**.

- b. Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/System/BVCorrect/channels/<ID>?format=json** to get the default or configured camera correction parameters for reference.
The camera correction parameters are returned in the message **JSON_BVCorrectParam** by **IpOutBuffer** of **IpOutputParam**.
 - c. Call **NET_DVR_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/System/BVCorrect/channels/<ID>?format=json** and set **IpInBuffer** of **IpInputParam** to the message **JSON_BVCorrectParam** to configure the camera correction parameters.
- The node <isSupportBVCorrect> is not returned in the capability message **XML_DeviceCap** : perform the steps below.
- a. Call **NET_DVR_GetDVRConfig** with command "NET_DVR_GET_BV_CORRECT_PARAM" (command No.: 3240) to get the default or configured camera correction parameters for reference.
The camera correction parameters are returned in the structure **NET_DVR_BV_CORRECT_PARAM** by the output buffer (**IpOutBuffer**).
 - b. Call **NET_DVR_SetDVRConfig** with command "NET_DVR_SET_BV_CORRECT_PARAM" (command No.: 3241) and set the input buffer (**IpInBuffer**) to the structure **NET_DVR_BV_CORRECT_PARAM** for configuring the camera correction parameters.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out of the device and release resources.

Chapter 5 Video and Audio



Note

During live view, playback, two-way audio, or audio forward, in order to record the information including stream data, packet loss rate, etc, you can call the API [**NET_DVR_GetLinkAddr**](#) to get the IP address and port information of the stream sender and recipient.

5.1 Start Live View

You can set stream parameters and remotely get the live stream for decoding and live view by calling SDK API based on private protocol. And forcing to generate I frame or refresh frame is also available for improving the stream delay problem.

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

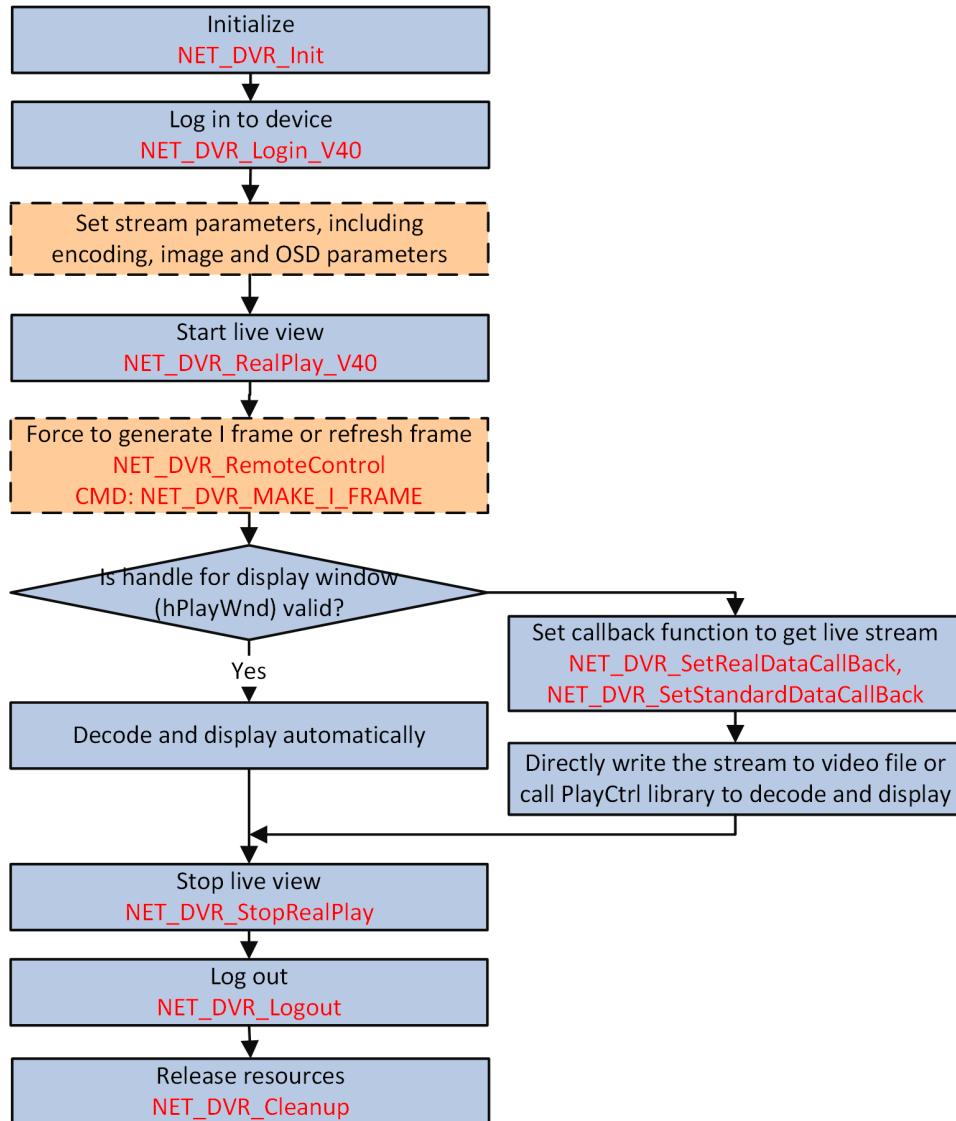


Figure 5-1 Programming Flow of Starting Live View

- 1. Optional:** Perform the following operation(s) to set the stream parameters (i.e., encoding parameters, image parameter, and OSD parameters).

- | | |
|--------------------------------|---|
| Set Encoding Parameters | a. Call <u>NET_DVR_GetDeviceConfig</u> with the command of NET_DVR_GET_MULTI_STREAM_COMPRESSIONCFG (command No.: 3216) to get the existing or configured encoding parameters for reference.
b. Call <u>NET_DVR_SetDeviceConfig</u> with the command of NET_DVR_SET_MULTI_STREAM_COMPRESSIONCFG (command No: 3217) to set the resolution, frame rate, stream type (video & audio, video), encoding type, and so on. |
|--------------------------------|---|



Note

The input parameter **IpInBuffer/ IpInParamBuffer** and output parameter **IpOutBuffer** of encoding configurations refer to the structure **NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND**.

Set Image Parameters

- a. Call **NET_DVR_GetDVRConfig** with the command of NET_DVR_GET_PICCFG_V40 (command No.: 6179) to get the existing or configured image parameters for reference.
 - b. Call **NET_DVR_SetDVRConfig** with the command of NET_DVR_SET_PICCFG_V40 (command No.: 6180) to set the video channel name, time display parameters, and so on.
-



Note

The input parameter **IpInBuffer** and output parameter **IpOutBuffer** of image configurations refer to the structure **NET_DVR_PICCFG_V40**.

Set OSD parameters

- a. Call **NET_DVR_GetDVRConfig** with the command of NET_DVR_GET_SHOWSTRING_V30 (command No.: 1030) to get the existing or configured OSD parameters for reference.
 - b. Call **NET_DVR_SetDVRConfig** with the command of NET_DVR_SET_SHOWSTRING_V30 (command No.: 1031) to set OSD parameters for live view.
-



Note

The input parameter **IpInBuffer** and output parameter **IpOutBuffer** of OSD configurations refer to the structure **NET_DVR_SHOWSTRING_V30**.

2. Call **NET_DVR_RealPlay_V40** to start live view.

3. **Optional:** Perform one of the following operations according to the stream encoding types to force to generate I frame or refresh frame.

- For stream with Smart264 encoding type, call **NET_DVR_STDCONTROL** with the command of NET_DVR_STREAMING_REFRESH_FRAME (command No.: 3714), : structure) to force the device to generate a deep P frame.
-



The control parameter **IpCondBuffer** refers to the structure **NET_DVR_STREAMING_COND**.

- For stream with other encoding types, call **NET_DVR_RemoteControl** with the command of NET_DVR_MAKE_I_FRAME (command No.: 3402) to force the device stream to generate an I frame for improving the stream delay problem.
-



The input parameter **IpInBuffer** refers to the structure **NET_DVR_I_FRAME**.

4. Check the validity of handle for display window (**hPlayWnd**) and perform one of the following operations according to the checking result to start decoding and displaying.
- For valid **hPlayWnd**, the decoding and displaying starts automatically.
 - For unconfigured **hPlayWnd**, perform the following steps:
 - a. Call **NET_DVR_SetRealDataCallBack** or **NET_DVR_SetStandardDataCallBack** to set callback functions for getting stream in PS or RTP container format.
 - b. Write the obtained stream to file or call PlayCtrl library to decode and display.
-



Note

For details about calling PlayCtrl library, refer to *User Manual of Cross-Platform PlayCtrl Library SDK*.

5. Call **NET_DVR_StopRealPlay** to stop live view.
-

Example

Sample Code of Starting Live View

- Decode and Display via SDK

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

typedef HWND (WINAPI *PROCGETCONSOLEWINDOW)();
PROCGETCONSOLEWINDOW GetConsoleWindowAPI;

void CALLBACK g_ExceptionCallBack(DWORD dwType, LONG lUserID, LONG lHandle, void *pUser)
{
    char tempbuf[256] = {0};
    switch(dwType)
    {
        case EXCEPTION_RECONNECT: //Reconnect during live view
            printf("-----reconnect-----%d\n", time(NULL));
            break;
        default:
            break;
    }
}

void main() {

//-----
// Initialize
NET_DVR_Init();
//Set connection time and reconnection time
NET_DVR_SetConnectTime(2000, 1);
NET_DVR_SetReconnect(10000, true);

//-----
```

```
//Set callback function for exceptional message
NET_DVR_SetExceptionCallBack_V30(0, NULL,g_ExceptionCallBack, NULL);

//-----
// Get the window handle
HMODULE hKernel32 = GetModuleHandle("kernel32");
GetConsoleWindowAPI = (PROCGETCONSOLEWINDOW)GetProcAddress(hKernel32,"GetConsoleWindow");

//-----
// 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, "10.13.35.229"); //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;
}

//-----
//Start live view and set stream callback function
LONG lRealPlayHandle;
HWND hWnd = GetConsoleWindowAPI(); //Get window handle
NET_DVR_PREVIEWINFO struPlayInfo = {0};
struPlayInfo.hPlayWnd = hWnd; //Set the handle as valid for SDK decoding; set the handle as null for
streaming only
struPlayInfo.iChannel = 1; //Live view channel No.
struPlayInfo.dwStreamType = 0; //0-Main Stream, 1-Sub Stream, 2-Stream 3, 3-Stream 4, and so on
struPlayInfo.dwLinkMode = 0; //0- TCP Mode, 1- UDP Mode, 2- Multicast Mode, 3- RTP Mode, 4-RTP/RTSP, 5-
RSTP/HTTP
struPlayInfo.bBlocked = 1; //0- Non-blocking Streaming, 1- Blocking Streaming

lRealPlayHandle = NET_DVR_RealPlay_V40(lUserID, &struPlayInfo, NULL, NULL);
if (lRealPlayHandle < 0)
{
    printf("NET_DVR_RealPlay_V40 error\n");
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}
```

```

Sleep(10000);
//-----
//Stop live view
NET_DVR_StopRealPlay(lRealPlayHandle);
//Log out
NET_DVR_Logout(lUserID);
//Release SDK resource
NET_DVR_Cleanup();

return;
}

```

- Decode and Display via PlayCtrl Library (for API named by the prefix "PlayM4_", refer to *User Manual of Cross-Platform PlayCtrl Library SDK*)

```

#include <stdio.h>
#include <iostream>
#include "Windows.h"
#include "HCNetSDK.h"
#include "plaympeg4.h"
#include <time.h>
using namespace std;

typedef HWND (WINAPI *PROCGETCONSOLEWINDOW)();
PROCGETCONSOLEWINDOW GetConsoleWindowAPI;

LONG lPort; //Port No. of global player SDK

void CALLBACK g_RealDataCallBack_V30(LONG lRealHandle, DWORD dwDataType, BYTE *pBuffer,DWORD dwBufSize,void* dwUser)
{
    HWND hWnd=GetConsoleWindowAPI();

    switch (dwDataType)
    {
        case NET_DVR_SYSHEAD: //System header
            if (m_lPort >= 0)
            {
                break; //The handle of this channel is already obtained before streaming. There is no need to call it in the following.
            }

            if (!PlayM4_GetPort(&lPort)) //Get unused channel No. of player
            {
                break;
            }
            //m_lPort = lPort; //The firstly called back is system header. The obtained player SDK port No. will be assigned to global port No. for next time play.
            if (dwBufSize > 0)
            {
                if (!PlayM4_SetStreamOpenMode(lPort, STREAME_REALTIME)) //Set play mode of real-time stream
            }
    }
}

```

```

        break;
    }

    if (!PlayM4_OpenStream(lPort, pBuffer, dwBufSize, 1024*1024)) //Open stream API
    {
        break;
    }

    if (!PlayM4_Play(lPort, hWnd)) //Start playing
    {
        break;
    }
}

break;

case NET_DVR_STREAMDATA: //Stream data
if (dwBufSize > 0 && lPort != -1)
{
    if (!PlayM4_InputData(lPort, pBuffer, dwBufSize))
    {
        break;
    }
}
break;

default: //Other data
if (dwBufSize > 0 && lPort != -1)
{
    if (!PlayM4_InputData(lPort, pBuffer, dwBufSize))
    {
        break;
    }
}
break;
}

void CALLBACK g_ExceptionCallBack(DWORD dwType, LONG lUserID, LONG lHandle, void *pUser)
{
    char tempbuf[256] = {0};
    switch(dwType)
    {
        case EXCEPTION_RECONNECT: //Reconnect during live view
            printf("-----reconnect-----%d\n", time(NULL));
            break;
        default:
            break;
    }
}

void main() {

//-----
// Initialize

```

```
NET_DVR_Init();
//Set connection time and reconnection time
NET_DVR_SetConnectTime(2000, 1);
NET_DVR_SetReconnect(10000, true);

//-----
//Set callback function for exceptional message
NET_DVR_SetExceptionCallBack_V30(0, NULL, g_ExceptionCallBack, NULL);

//-----
// Get the window handle
HMODULE hKernel32 = GetModuleHandle("kernel32");
GetConsoleWindowAPI = (PROCGETCONSOLEWINDOW)GetProcAddress(hKernel32, "GetConsoleWindow");

//-----
// 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, "10.13.35.229"); //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;
}

//-----
//Start live view and set stream callback function
LONG lRealPlayHandle;
NET_DVR_PREVIEWINFO struPlayInfo = {0};
struPlayInfo.hPlayWnd = NULL; //Set the handle as valid for SDK decoding; set the handle as null for streaming only
struPlayInfo.lChannel = 1; //Live view channel No.
struPlayInfo.dwStreamType = 0; //0-Main Stream, 1-Sub Stream, 2-Stream 3, 3-Stream 4, and so on
struPlayInfo.dwLinkMode = 0; //0- TCP Mode, 1- UDP Mode, 2- Multicast Mode, 3- RTP Mode, 4-RTP/RTSP, 5-RSTP/HTTP
struPlayInfo.bBlocked = 1; //0- Non-blocking Streaming, 1- Blocking Streaming

lRealPlayHandle = NET_DVR_RealPlay_V40(lUserID, &struPlayInfo, g_RealDataCallBack_V30, NULL);
if (lRealPlayHandle < 0)
{
```

```
printf("NET_DVR_RealPlay_V40 error, %d\n", NET_DVR_GetLastError());
NET_DVR_Logout(lUserID);
NET_DVR_Cleanup();
return;
}

Sleep(10000);
//-----
//Stop live view
NET_DVR_StopRealPlay(lRealPlayHandle);

//Release player SDK resources
PlayM4_Stop(lPort);
PlayM4_CloseStream(lPort);
PlayM4_FreePort(lPort);

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

return;
}
```

What to do next

Call [NET_DVR_Logout](#) and [NET_DVR_Cleanup](#) to log out and release resources.

5.2 Start Live View Based on ISAPI Protocol

For some specific devices, e.g., face recognition server, you can remotely get the live stream via the URL returned by device for live view by calling HCNetSDK API based on ISAPI protocol communication.

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

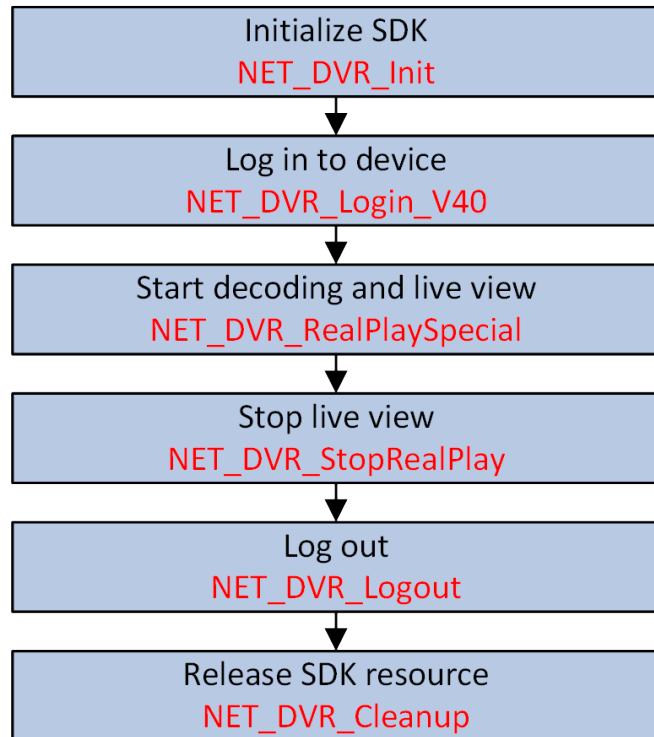


Figure 5-2 Programming Flow of Starting Live View Based on ISAPI Protocol

1. Call `NET_DVR_RealPlaySpecial` to start decoding and live view.
2. Call `NET_DVR_StopRealPlay` to stop live view.

What to do next

Call `NET_DVR_Logout` and `NET_DVR_Cleanup` to log out and release resources.

5.3 Perform PTZ Control

Perform PTZ control to realize the pan, tilt, and zoom functions of cameras, set the presets for specific monitoring areas to fast switch the camera as needed, and group the user-defined presets as a scanning track for patrol.

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

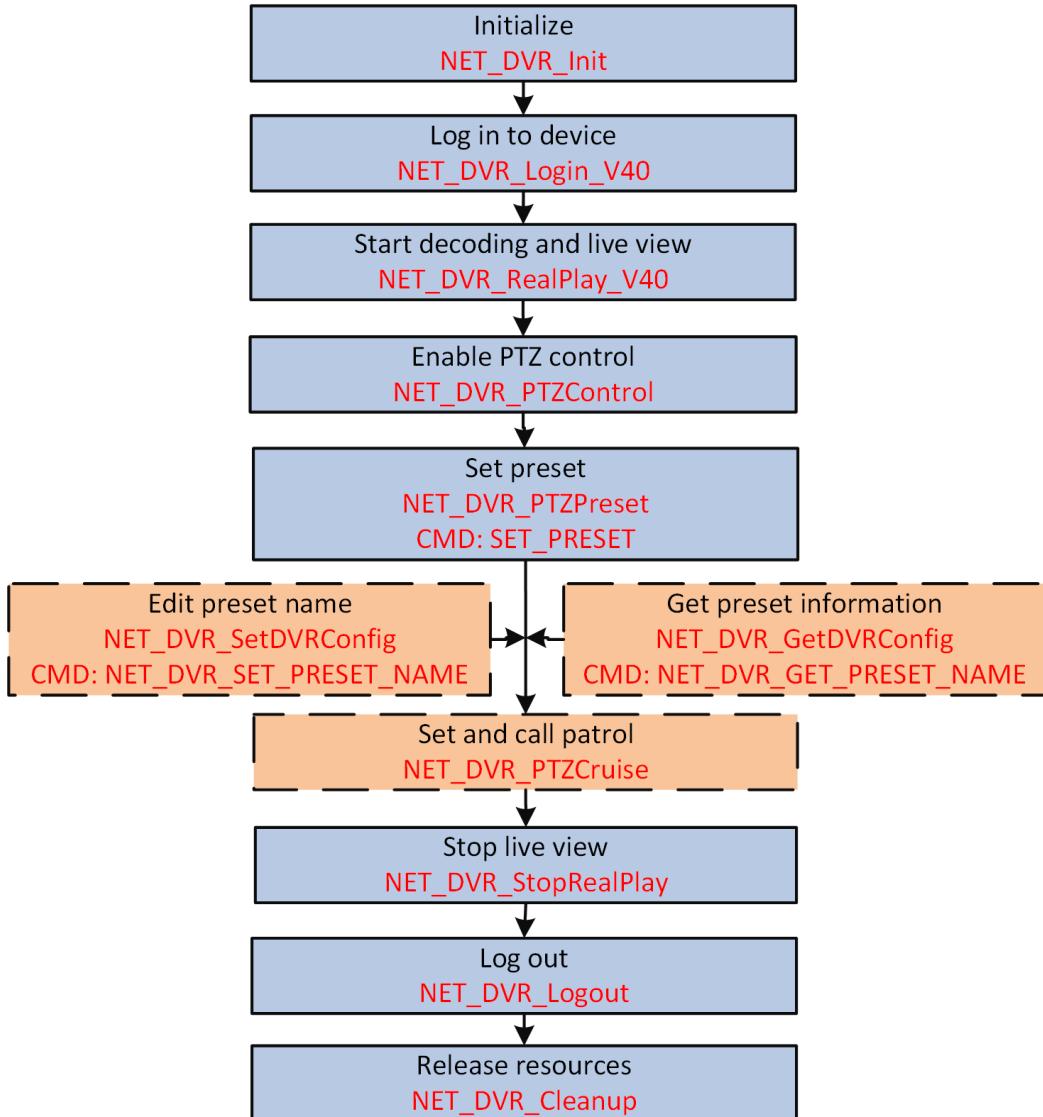


Figure 5-3 Calling Process of Performing PTZ Control

1. Call [NET_DVR_RealPlay_V40](#) to start live view.



For more about starting live view, refer to [Start Live View](#).

2. Call [NET_DVR_PTZControl](#) to enable PTZ control.

3. Set the preset(s).

- 1) Control the camera to move to a specific position.
- 2) Call [NET_DVR_PTZPreset](#) with the command of **SET_PRESET** (command No.: 8) to set the position as a preset.

- 3) **Optional:** Repeatedly call **NET_DVR_PTZPreset** with the command of **SET_PRESET** (command No.: 8) to set multiple presets.
4. **Optional:** Perform the following operation(s) after setting the presets.
- | | |
|-------------------------------|--|
| Call Preset | Call <u>NET_DVR_PTZPreset</u> with
<u>GOTO_PREST</u>
(command No.: 39) to call the configured preset. |
| Delete Preset | Call <u>NET_DVR_PTZPreset</u> with
<u>CLE_PREST</u>
(command No.: 9) to delete the preset. |
| Get Preset Information | Call <u>NET_DVR_GetDVRConfig</u> with
<u>NET_DVR_GET_PRESET_NAME</u>
(command No.: 3383) to get the preset name and predefined PTZ information, which is returned by IpOutBuffer in . |
| Edit Preset Name | Call <u>NET_DVR_SetDVRConfig</u> with
<u>NET_DVR_SET_PRESET_NAME</u>
(command No.: 3382) and set IpInBuffer to for editing the preset name. |

5. **Optional:** Call **NET_DVR_PTZCruise** with different commands to add the configured presets to the patrol, and set the scanning speed between two presets and the dwell time at a preset.
-



At least two presets are required to set the patrol.

6. **Optional:** Call **NET_DVR_PTZCruise** with
RUN_SFQ
(command No.: 37) or
STOP_SEQ
(command No.: 38) to start and stop the configured patrol.
7. Call **NET_DVR_StopRealPlay** to stop the live view.

The PTZ control is disabled.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release the resources.

5.3.1 PTZ Configuration

Basic PTZ Parameters

Function	API
Get PTZ control capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/PTZCtrl/channels/<ID>/capabilities</i> , and the capability is returned in the message <i>XML_PTZChanelCap</i> by the output parameter pointer (<i>lpOutputParam</i>).
Get zoom/focus coordinate parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/PTZCtrl/channels/<ID>/zoomFocus</i> , and the zoom/focus coordinates parameters are returned in the message <i>XML_ZoomFocus</i> by the output parameter pointer (<i>lpOutputParam</i>).
Set zoom/focus coordinate parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/PTZCtrl/channels/<ID>/zoomFocus</i> and set the input parameter pointer (<i>lpInputParam</i>) to the message <i>XML_ZoomFocus</i> .
Maximum Tilt Angle	<ul style="list-style-type: none"> Get configuration capability of maximum tilt angle: Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET /<i>ISAPI/PTZCtrl/channels/<ID>/maxelevation/capabilities</i>, and the capability is returned in the message <i>XML_Cap_MaxElevation</i> by the output parameter pointer (<i>lpOutputParam</i>). Get parameters of maximum tilt angle: Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET /<i>ISAPI/PTZCtrl/channels/<ID>/maxelevation</i>, and the max. tilt-angle parameters are returned in the message <i>XML_MaxElevation</i> by the output parameter pointer (<i>lpOutputParam</i>). Set parameters of maximum tilt angle: Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT /<i>ISAPI/PTZCtrl/channels/<ID>/maxelevation</i> and set the input parameter pointer (<i>lpInputParam</i>) to the message <i>XML_MaxElevation</i>.

Other PTZ Parameters

Function	API
Get PTZ OSD parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/PTZCtrl/channels/<ID>/PTZOSDDisplay</i> , and the on-

Function	API
	screen display parameters of the PTZ status are returned in the message <u>XML_PTZOSDDisplay</u> by the output parameter pointer (lpOutputParam).
Set PTZ OSD	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/PTZCtrl/channels/<ID>/PTZOSDDisplay</u> and set the input parameter pointer (lpInputParam) to the message <u>XML_PTZOSDDisplay</u> .

PTZ Auxiliaries Control

Function	API
Get single PTZ auxiliary status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/PTZCtrl/channels/<ID>/auxcontrols/<ID></u> , and the auxiliary status information is returned in the message <u>XML_PTZAux</u> by the output parameter pointer (lpOutputParam).
Set single PTZ auxiliary status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/PTZCtrl/channels/<ID>/auxcontrols/<ID></u> and set the input parameter pointer (lpInputParam) to the message <u>XML_PTZAux</u> .

E-PTZ Parameters

Function	API
Get e-PTZ parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/Image/channels/<ID>/EPTZ</u> . The parameters are returned in the message <u>XML_EPTZ</u> by the output parameter lpOutputParam .
Set or set e-PTZ parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/Image/channels/<ID>/EPTZ</u> and set the input parameter lpInputParam to the message <u>XML_EPTZ</u> .
Get capability of switching e-PTZ mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/Image/channels/<ID>/EPTZ/mode/capabilities?format=json</u> . The capability is returned in the message <u>JSON_EPTZModeCap</u> by the output parameter lpOutputParam .

Function	API
Get parameters of switching e-PTZ mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET /ISAPI/Image/channels/<ID>/EPTZ mode?format=json . The parameters are returned in the message <u>JSON_EPTZMode</u> by the output parameter lpOutputParam .
Set parameters of switching e-PTZ mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT /ISAPI/Image/channels/<ID>/EPTZ mode?format=json and set lpInputParam to the message <u>JSON_EPTZMode</u> .

Moving Object Tracking

Function	API
Get moving object tracking status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET /ISAPI/PTZCtrl/channels/<ID>/moveAutoTracking?format=json . The status is returned in the message <u>JSON_MoveAutoTracking</u> by lpOutputParam .
Set moving object tracking status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT /ISAPI/PTZCtrl/channels/<ID>/moveAutoTracking?format=json and set lpInputParam to the message <u>JSON_MoveAutoTracking</u> .

Linked Tracking Capture

Function	API
Get capability of linked tracking capture	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET /ISAPI/MasterSlaveTracking/linkedTracking/capabilities?format=json .
Get linked tracking capture parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET /ISAPI/MasterSlaveTracking/linkedTracking?format=json . The parameters are returned in the message <u>JSON_LinkedTracking</u> by lpOutputParam .
Set linked tracking capture parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT /ISAPI/MasterSlaveTracking/linkedTracking?format=json and set lpInputParam to the message <u>JSON_LinkedTracking</u> .

Gyroscope Calibration

Function	API
Get gyroscope attitude calibration	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/PTZCtrl/channels/<ID>/gyroVerify/capabilities?format=json</u>
Set gyroscope attitude calibration parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/PTZCtrl/channels/<ID>/gyroVerify?format=json</u>

GPS Calibration

Function	API
Get capability of verifying GPS calibration	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/System/GPSVerification/channels/<ID>/points/capabilities?format=json</u>
Get GPS calibration configuration capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/System/GPSCalibration/channels/<ID>/capabilities?format=json</u>
Configure a single GPS calibration point.	<ul style="list-style-type: none"> Get parameters of a single GPS calibration point: Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>GET /ISAPI/System/GPSCalibration/channels/<ID>/points/<PID>?format=json</u> Set a single GPS calibration point: Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>PUT /ISAPI/System/GPSCalibration/channels/<ID>/points/<PID>?format=json</u> Delete a single GPS calibration point: Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>DELETE /ISAPI/System/GPSCalibration/channels/<ID>/points/<PID>?format=json</u>
Configure multiple GPS calibration points	<ul style="list-style-type: none"> Get parameters of multiple GPS calibration points in a batch: Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>GET /ISAPI/System/GPSCalibration/channels/<ID>/points?format=json</u> Set multiple GPS calibration points in a batch:

Function	API
	<p>Call <code>NET_DVR_STDXMLConfig</code> to transmit the request URI: PUT /ISAPI/System/GPSCalibration/channels/<ID>/points?format=json</p> <ul style="list-style-type: none"> Delete multiple GPS calibration points in a batch <p>Call <code>NET_DVR_STDXMLConfig</code> to transmit the request URI: DELETE /ISAPI/System/GPSCalibration/channels/<ID>/points?format=json</p>
Enable GPS calibration	Call <code>NET_DVR_STDXMLConfig</code> to transmit the request URI: PUT /ISAPI/System/GPSPTZCalibration/channels/<ID>/calibration?format=json
Get capability of verifying GPS calibration	Call <code>NET_DVR_STDXMLConfig</code> to transmit the request URI: GET /ISAPI/System/GPSVerification/channels/<ID>/points/capabilities?format=json
Configure multiple verification points	<ul style="list-style-type: none"> Add a verification points: <p>Call <code>NET_DVR_STDXMLConfig</code> to transmit the request URI: POST /ISAPI/System/GPSVerification/channels/<ID>/points?format=json</p> <ul style="list-style-type: none"> Get information of multiple verification points: <p>Call <code>NET_DVR_STDXMLConfig</code> to transmit the request URI: GET /ISAPI/System/GPSVerification/channels/<ID>/points?format=json</p> <ul style="list-style-type: none"> Set multiple verification points: <p>Call <code>NET_DVR_STDXMLConfig</code> to transmit the request URI: PUT /ISAPI/System/GPSVerification/channels/<ID>/points?format=json</p> <ul style="list-style-type: none"> Delete multiple verification points: <p>Call <code>NET_DVR_STDXMLConfig</code> to transmit the request URI: DELETE /ISAPI/System/GPSVerification/channels/<ID>/points?format=json</p>
Configure one specified verification point	<ul style="list-style-type: none"> Get information of one specified verification point: <p>Call <code>NET_DVR_STDXMLConfig</code> to transmit the request URI: GET /ISAPI/System/GPSVerification/channels/<ID>/points/<ID>?format=json</p> <ul style="list-style-type: none"> Set information of one specified verification point: <p>Call <code>NET_DVR_STDXMLConfig</code> to transmit the request URI: PUT /ISAPI/System/GPSVerification/channels/<ID>/points/<ID>?format=json</p> <ul style="list-style-type: none"> Delete one specified verification point:

Function	API
	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: DELETE <u>/ISAPI/System/GPSVerification/channels/<ID>/points/<ID>?format=json</u>
Get GPS calibration verification results in a batch	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/System/GPSVerification/channels/<ID>/points/resultInfo?format=json</u>
Get or set parameters of displaying GPS calibration verification result	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET or PUT <u>/ISAPI/System/GPSVerification/channels/<ID>/display?format=json</u>

5.4 Search for Recording Start and End Time by Channel

You can check whether the channel has recorded videos and get the recording start and end 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

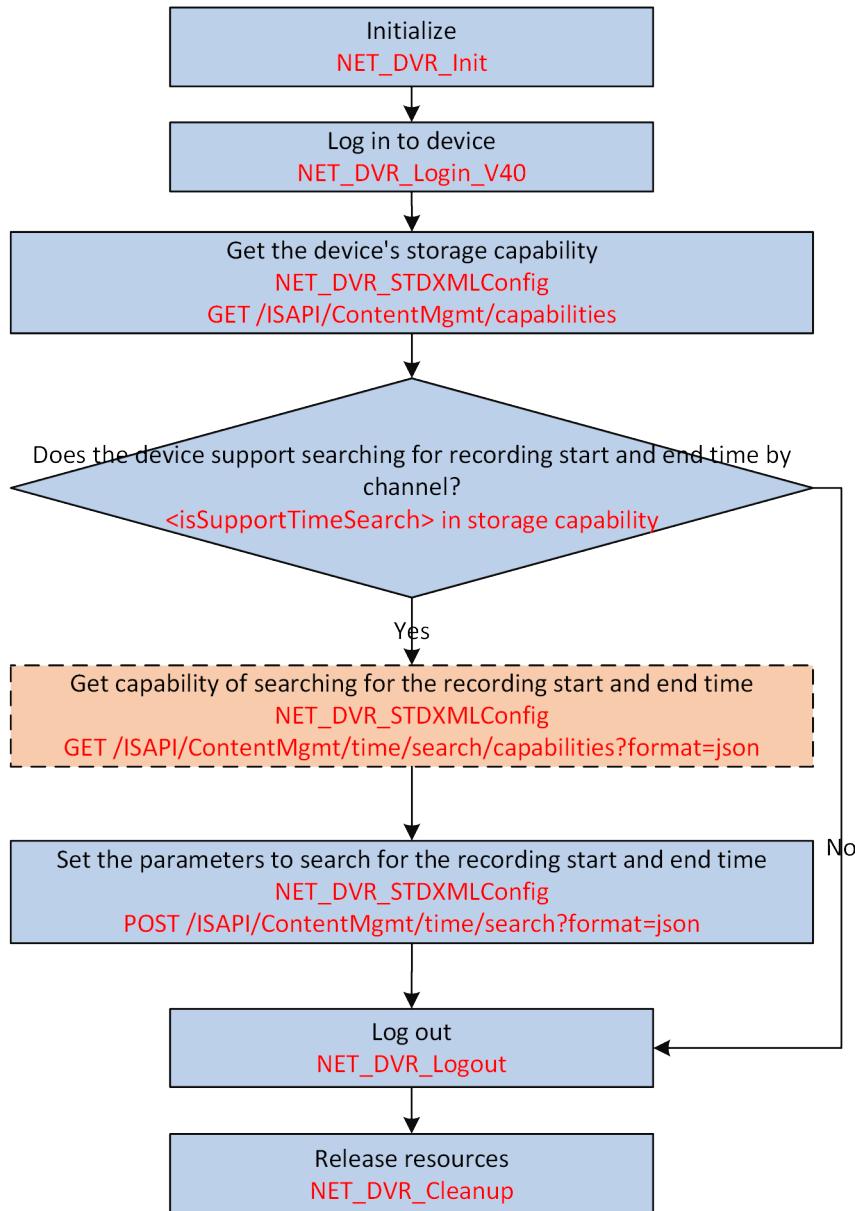


Figure 5-4 API Calling Flow of Searching for Recording Start and End Time by Channel

1. Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/ContentMgmt/capabilities** for getting the device's storage capability to check whether it supports searching for recording start and end time.

The device's storage capability is returned by the output parameter pointer **IpOutputParam** in the message **XML_RacmCap**.

If the node <isSupportTimeSearch> is returned in the message and its value is "true", it indicates that searching for recording start and end time is supported and you can continue to perform the following steps.

2. **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/ContentMgmt/time/search/capabilities?format=json** for getting the capability of searching for the recording start and end time for reference.

The capability is returned by the output parameter pointer **IpOutputParam** in the message **JSON_Cap_TimeSearchCond**.

3. Call **NET_DVR_STDXMLConfig** to transmit the request URI: POST **/ISAPI/ContentMgmt/time/search?format=json** to set the parameters to search for the recording start and end time.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out off the device and release the resources.

5.5 Start Playback

After searching the video files (optional), you can remotely play the video files stored in the device by time or by file to view the previously occurred events as needed.

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

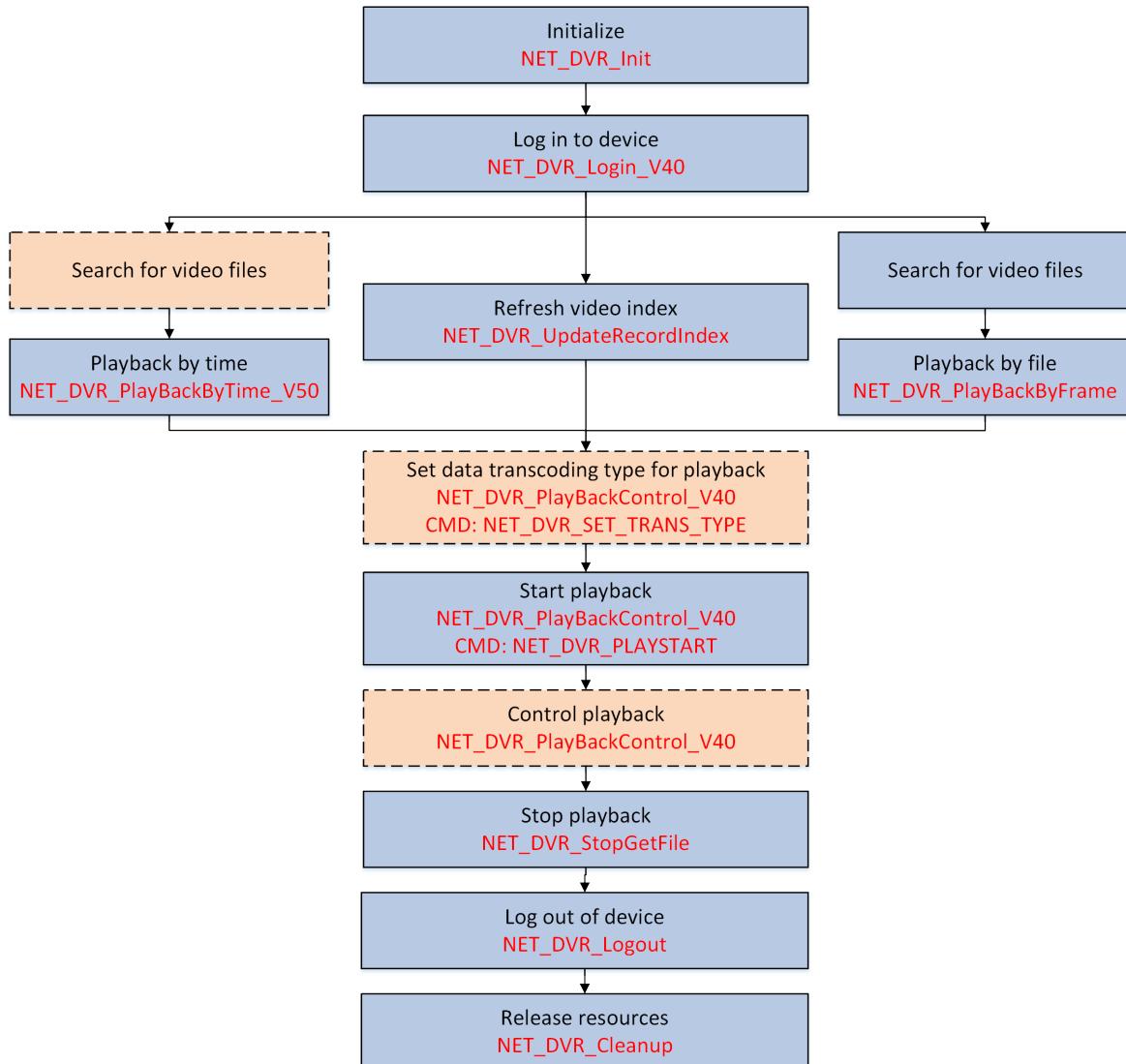


Figure 5-5 Programming Flow of Starting Playback



Playback by file is not supported when login via ISAPI protocol.

1. Perform one of the following operations to start playback by time or by file.

- a. Search video files if there is no video recorded in the specified time period.
- b. Call **NET_DVR_PlayBackByTime_V50** to specify the start time and end time to play all video files in the time period.
- a. Search video files to get the file name.
- b. Call **NET_DVR_PlayBackByName** to specify files for playback according to the obtained file name.

- Call **NET_DVR_UpdateRecordIndex** to refresh the video index for instant playback.
-



Note

The device refreshes the video index every two minutes by default.

2. Call **NET_DVR_PlayBackControl_V40** with "NET_DVR_SET_TRANS_TYPE" (command No.: 32) and set input parameter pointer (**IplInBuffer**) to a 4-byte integer container format value (i.e., "1"-PS, "2"-TS, "3"-RTP, "5"-MP4 or 3GPP (supported only when playback by file)) for setting the data transcoding type for playback.
 3. Call **NET_DVR_PlayBackControl_V40** with "NET_DVR_PLAYSTART" (command No.: 1) and set input parameter pointer (**IplInBuffer**) to a 4-byte integer offset (for ANR) to start playback.
 4. **Optional:** Call **NET_DVR_PlayBackControl_V40** with different commands to control the playback, such as fast forwarding, slow forwarding, pausing playback, and so on.
-



- When playback by time, getting and setting playback progress is not supported.
- If login via ISAPI protocol, only the following playback control commands are supported: "NET_DVR_PLAYSTART" (start playing), "NET_DVR_PLAYPAUSE" (pausing playing), "NET_DVR_PLAYRESTART" (resume playing), "NET_DVR_PLAYFAST" (fast forward), "NET_DVR_PLAYSLOW" (slow forward), "NET_DVR_PLAYNORMAL" (play in 1x speed).

5. Call **NET_DVR_StopPlayBack** to stop the playback.
-

Example

Sample Code of Playback by Time

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

typedef HWND (WINAPI *PROCGETCONSOLEWINDOW)();
PROCGETCONSOLEWINDOW GetConsoleWindow;

void main() {

    //-----
    // Initialize
    NET_DVR_Init();
    //Set connection time and reconnection time
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

    //-----
    // Get window handle of control center
    HMODULE hKernel32 = GetModuleHandle("kernel32");
    GetConsoleWindow = (PROCGETCONSOLEWINDOW)GetProcAddress(hKernel32,"GetConsoleWindow");

    //-----
```

```
// 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"); //IP address
struLoginInfo.wPort = 8000; //Service port
strcpy(struLoginInfo.sUserName, "admin"); //User name
strcpy(struLoginInfo.sPassword, "abcd1234"); //Password

//Device information, output parameters
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;
}

HWND hWnd = GetConsoleWindow(); //Get window handle

NET_DVR_VOD_PARA_V50 struVodPara = {0};
struVodPara.dwSize = sizeof(struVodPara);
struVodPara.struIDInfo.dwSize = sizeof(NET_DVR_STREAM_INFO);
struVodPara.struIDInfo.dwChannel = 1;
struVodPara.hWnd = hWnd;
struVodPara.struBeginTime.wYear = 2019;
struVodPara.struBeginTime.byMonth = 6;
struVodPara.struBeginTime.byDay = 14;
struVodPara.struBeginTime.byHour = 9;
struVodPara.struBeginTime.byMinute = 0;
struVodPara.struBeginTime.bySecond = 0;
struVodPara.struBeginTime.byISO8601 = 0;
struVodPara.struBeginTime.cTimeDifferenceH = 0;
struVodPara.struBeginTime.cTimeDifferenceM = 0;
struVodPara.struEndTime.wYear = 2019;
struVodPara.struEndTime.byMonth = 7;
struVodPara.struEndTime.byDay = 14;
struVodPara.struEndTime.byHour = 9;
struVodPara.struEndTime.byMinute = 0;
struVodPara.struEndTime.bySecond = 0;
struVodPara.struEndTime.byISO8601 = 0;
struVodPara.struEndTime.cTimeDifferenceH = 0;
struVodPara.struEndTime.cTimeDifferenceM = 0;
struVodPara.byVolumeType = 0;
struVodPara.byDrawFrame = 0;
struVodPara.byStreamType = 0;
struVodPara.byLinkMode = 0;
struVodPara.byCourseFile = 0;
```

```
struVodPara.byOptimalStreamType = 0;
struVodPara.byDownload = 0;
struVodPara.byDisplayBufNum = 0;
sprintf((char *)struVodPara.sUserName, "%s", "admin");
sprintf((char *)struVodPara.sPassword, "%s", "12345");
struVodPara.byRemoteFile = 1;
struVodPara.byPlayMode = 0;

//-----
//Playback by time
LONG IPlayHandle = NET_DVR_PlayBackByTime_V50(IUserID, &struVodPara);
if (IPlayHandle < 0)
{
    printf("NET_DVR_PlayBackByTime_V50 fail, last error %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}
//-----
//Start playback
if(!NET_DVR_PlayBackControl_V40(hPlayback, NET_DVR_PLAYSTART,NULL, 0, NULL,NULL))
{
    printf("play back control failed [%d]\n",NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

Sleep(15000); //millisecond
if(!NET_DVR_StopPlayBack(hPlayback))
{
    printf("failed to stop file [%d]\n",NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

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

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release resources.

5.6 Two-Way Audio

The two-way audio application provides the functions of voice talk between control center and device, audio forwarding, encoding and decoding audio, and audio broadcast.



- Note**
- When developing under Windows (64-bit) or Linux system, only the audio forwarding application is supported.
 - When developing under Windows 7 system, the APIs used to implement two-way audio and audio broadcast application will return "FALSE" if there is no external audio device.
-

5.6.1 Start Two-Way Audio

The two-way audio implements the function of audio sending or receiving between monitoring center and specific cameras. It is applied to the following situation: when there are multiple entrances and exits in a park or campus, and the booth of each entrance and exit has mounted with cameras, if the exception occurs in an entrance and exit, the monitoring center can talk with the corresponding booth to ask for information via the two-way audio.

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

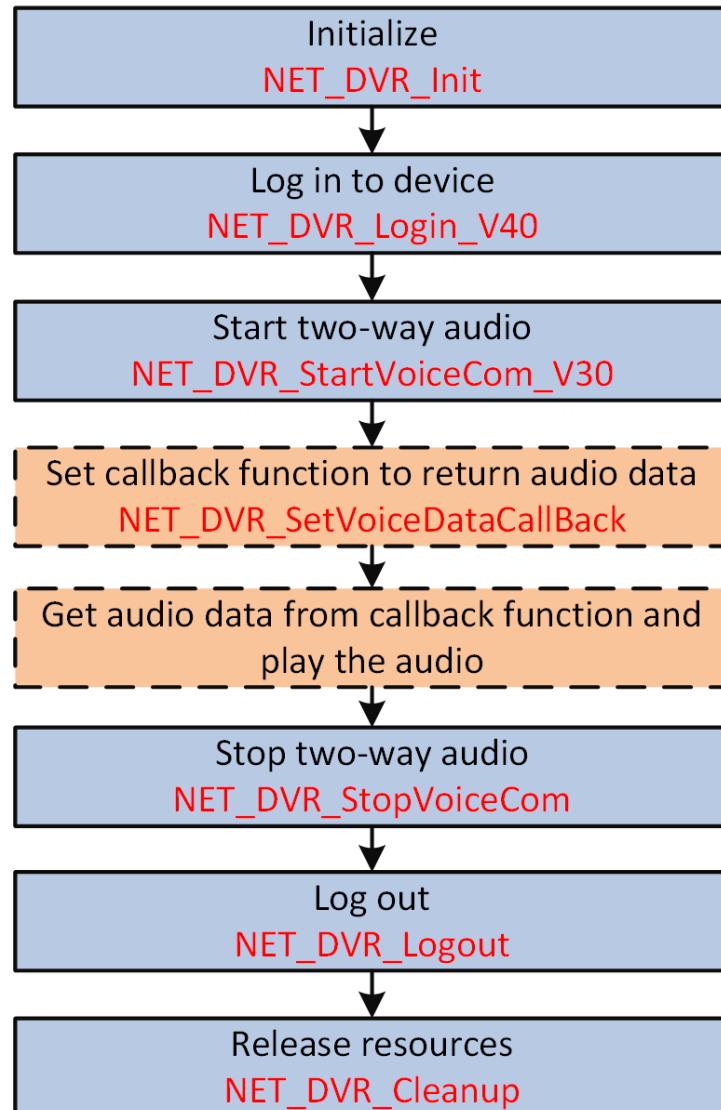


Figure 5-6 Programming Flow of Starting Two-Way Audio

1. Call [NET_DVR_StartVoiceCom_V30](#) to start the two-way audio.
2. **Optional:** Call [NET_DVR_SetVoiceDataCallBack](#) to set the callback function for returning the audio data.
3. **Optional:** Get the audio data from the configured callback function and play the audio.
4. Call [NET_DVR_StopVoiceCom](#) to stop the two-way audio.

Example

Sample Code of Starting Two-way Audio

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

```
#include "HCNetSDK.h"
using namespace std;

void CALLBACK fVoiceDataCallBack(LONG lVoiceComHandle, char *pRecvDataBuffer, DWORD dwBufSize, BYTE
byAudioFlag, void* pUser)
{
    printf("receive voice data, %d\n", dwBufSize);
}

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"); //IP address
struLoginInfo.wPort = 8000; //Service port
strcpy(struLoginInfo.sUserName, "admin"); //User name
strcpy(struLoginInfo.sPassword, "abcd1234"); //Password

//Device information, output parameters
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;
}

//Start two-way audio
LONG lVoiceHanle;
lVoiceHanle = NET_DVR_StartVoiceCom_V30(lUserID, 1,0, fVoiceDataCallBack, NULL);
if (lVoiceHanle < 0)
{
    printf("NET_DVR_StartVoiceCom_V30 error, %d!\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}
```

```
    Sleep(5000); //millisecond
    //Stop two-way audio
    if (!NET_DVR_StopVoiceCom(lVoiceHandle))
    {
        printf("NET_DVR_StopVoiceCom 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 resources.

5.6.2 Forward Audio

When there are multiple entrances and exits in a park or campus, and the booth of each entrance and exit has been mounted with cameras, the monitoring center can send audio information to a specific camera as needed via the audio forwarding.

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 prepared source audio with specific format for forwarding.

Steps

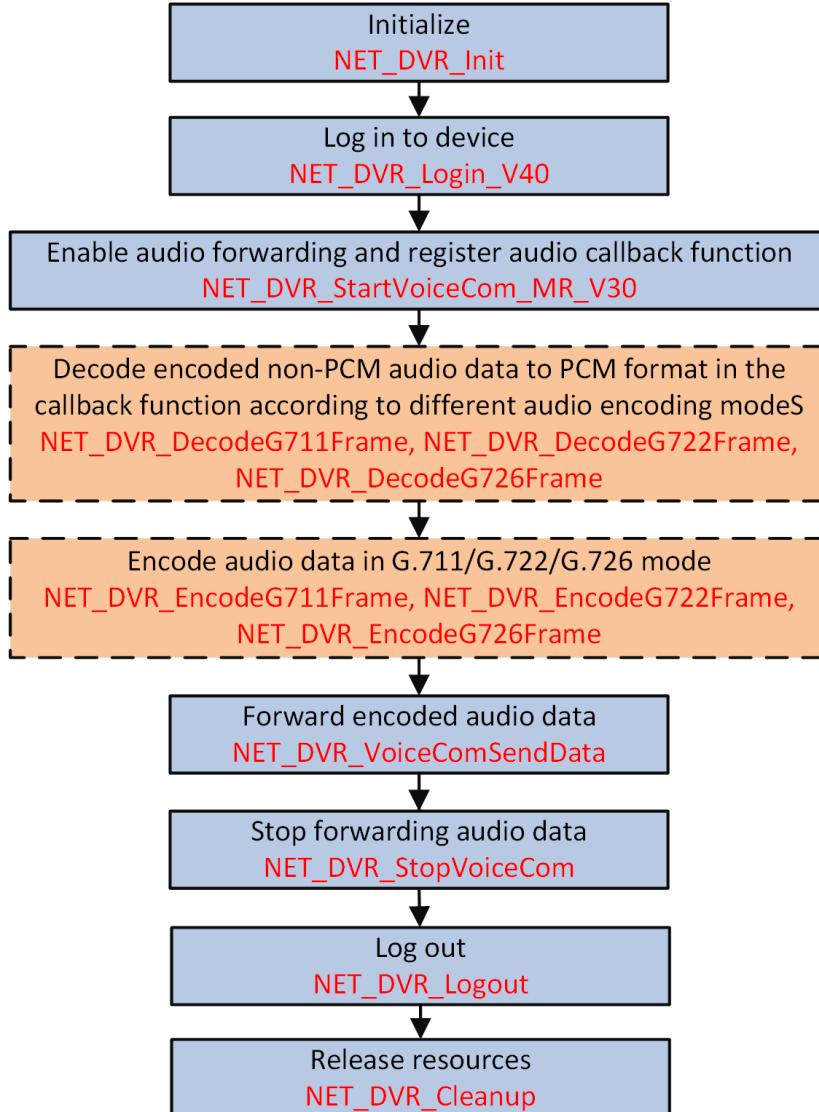


Figure 5-7 Programming Flow of Forwarding Audio

1. Call `NET_DVR_StartVoiceCom_MR_V30` to enable audio forwarding and register the callback function (`fVoiceDataCallBack`) for processing audio data.
The connection between center and device is built to wait for sending data.
2. **Optional:** Perform one of the following operations in the callback function to decode the encoded non-PCM audio data to PCM format according to different audio encoding modes.
 - a. Call to initialize the decoding resources.



Note

This API is only available for Linux operating system.

- b. Call to start decoding the encoded non-PCM audio data to PCM format.
- c. Call to release the decoding resources.



Note

This API is only available for Linux operating system.

- a. Call to initialize the decoding resources.
- b. Call to start decoding the encoded non-PCM audio data to PCM format.
- c. Call to release the decoding resources.
- a. Call to initialize the decoding resources.
- b. Call to start decoding the encoded non-PCM audio data to PCM format.
- c. Call to release the decoding resources.



Note

- The source audio data is collected from the sound card of PC or read from the file.
- The decoding APIs must correspond to the following audio encoding mode. For example, if the encoding mode is G.711, the decoding APIs should be , , and .

3. Optional: Perform one of the following operations to encode the decoded audio data or the original audio data with PCM format in G.711, G.722, or G.726 mode.

- a. Call **NET_DVR_InitG711Encoder** to initialize the encoding resources.



Note

This API is only available for Linux operating system.

- b. Call **NET_DVR_EncodeG711Frame** to start encoding the audio data in G.711 mode.
- c. Call **NET_DVR_ReleaseG711Encoder** to release the encoding resources.



Note

This API is only available for Linux operating system.

- a. Call **NET_DVR_InitG722Encoder** to initialize the encoding resources.
- b. Call **NET_DVR_EncodeG722Frame** to start encoding the audio data in G.722 mode.
- c. Call **NET_DVR_ReleaseG722Encoder** to release the encoding resources.
- a. Call **NET_DVR_InitG726Encoder** to initialize the encoding resources.
- b. Call **NET_DVR_EncodeG726Frame** to start encoding audio data in G.726 mode.
- c. Call **NET_DVR_ReleaseG726Encoder** to release the encoding resources.

4. Call **NET_DVR_VoiceComSendData to start forwarding the encoded audio data to device.**

5. Call **NET_DVR_StopVoiceCom to stop forwarding audio.**

Example

Forward Audio (For Windows Operating System)

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

```
#include "HCNetSDK.h"
using namespace std;
void CALLBACK fVoiceDataCallBack(LONG lVoiceComHandle, char *pRecvDataBuffer, DWORD dwBufSize, BYTE byAudioFlag, void* pUser)
{
    //The audio data here can be the data encoded and sent by device, or the locally collected and encoded data.
    NET_DVR_VoiceComSendData(lVoiceComHandle, pRecvDataBuffer, 80);
}

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"); //IP address
struLoginInfo.wPort = 8000; //IP address
strcpy(struLoginInfo.sUserName, "admin"); //User name
strcpy(struLoginInfo.sPassword, "abcd1234"); //Password

//Device information, output parameters
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;
}

//Forward audio
LONG lVoiceHandle;
lVoiceHandle = NET_DVR_StartVoiceCom_MR_V30(lUserID, 1, fVoiceDataCallBack, NULL);
if (lVoiceHandle < 0)
{
    printf("NET_DVR_StartVoiceCom_MR_V30 error, %d!\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}
```

```
    Sleep(5000); //millisecond
    //Stop forwarding audio
    if (!NET_DVR_StopVoiceCom(lVoiceHandle))
    {
        printf("NET_DVR_StopVoiceCom 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 resources.

5.6.3 Broadcast Audio

When there are multiple entrances and exits in a park or campus, and the booth of each entrance and exit has mounted with cameras, the monitoring center can send audio information to multiple or all cameras at same time as needed via the audio broadcast.

Before You Start

Make sure you have called [NET_DVR_Init](#) to initialize the development environment.

Steps

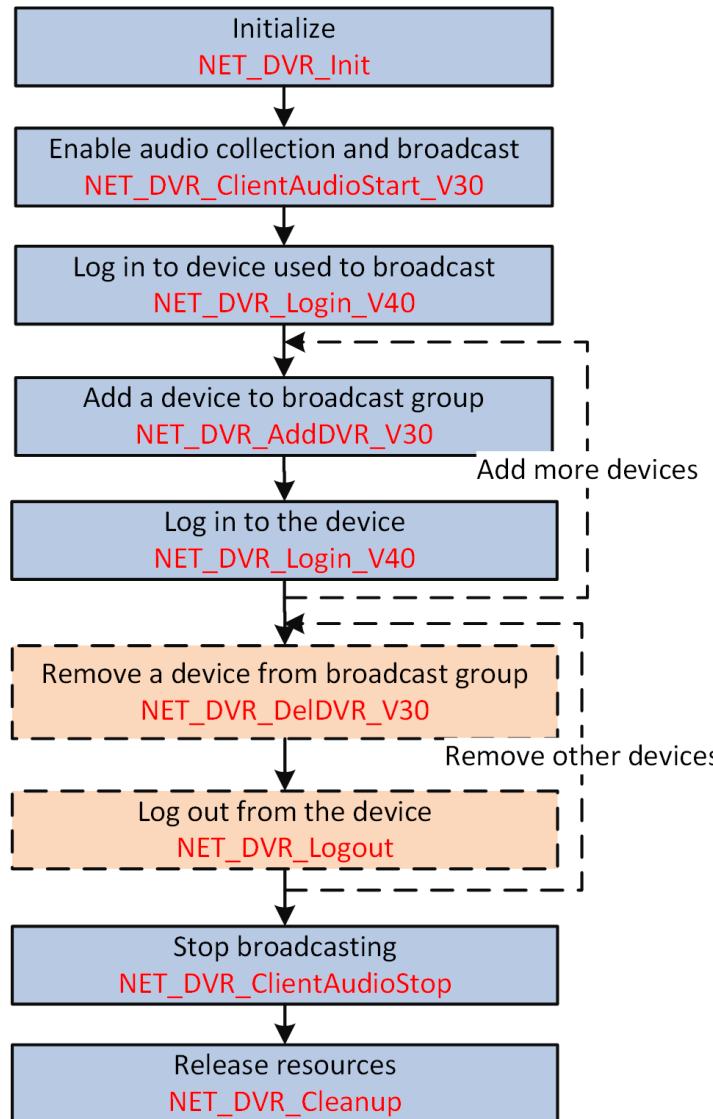


Figure 5-8 Programming Flow of Audio Broadcast

1. Call [NET_DVR_ClientAudioStart_V30](#) to enable the audio collection and broadcast.
2. Call [NET_DVR_Login_V40](#) to log in to the device that used to broadcast.
3. Add devices to the broadcast group for receiving audio.
 - 1) Call [NET_DVR_AddDVR_V30](#) to add a device to the broadcast group.
 - 2) Call [NET_DVR_Login_V40](#) to log in to the added device.
 - 3) **Optional:** Repeat the above two steps to add more devices to the broadcast group if you want to broadcast the audio to multiple devices.
4. **Optional:** Remove devices from the broadcast group.
 - 1) Call [NET_DVR_DelDVR_V30](#) to remove a device from the broadcast group.
 - 2) Call [NET_DVR_Logout](#) to log out from the removed device.

- 3) **Optional:** Repeat the above two steps to remove other devices from the broadcast group.
5. Call [NET_DVR_ClientAudioStop](#) to stop collecting and broadcasting audio.

What to do next

Call [NET_DVR_Cleanup](#) to release resources.

5.6.4 Audio Settings

Get Audio Parameters of All Two-Way Audio Channels

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/System/TwoWayAudio/channels](#).

The parameters are returned in the message [XML_TwoWayAudioChannelList](#) by **IpOutputParam**.

Set Audio Parameters of All Two-Way Audio Channels

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/System/TwoWayAudio/channels](#) and set **IpInputParam** to [XML_TwoWayAudioChannelList](#).

Get Audio Parameters of a Two-Way Audio Channel

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/System/TwoWayAudio/channels/<ID>](#).

The parameters are returned in the message [XML_TwoWayAudioChannel](#) by **IpOutputParam**.

Set Audio Parameters of a Two-Way Audio Channel

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/System/TwoWayAudio/channels/<ID>](#) and set **IpInputParam** to [XML_TwoWayAudioChannel](#).

5.7 Stream Encoding

The following APIs and request URLs are used for configuring encoding parameters of video and audio stream.



The whole stream encoding capability ([XML_Cap_AudioVideoCompressInfo](#)) of device should be obtained by calling [NET_DVR_GetDeviceAbility](#), setting capability type (**dwAbilityType**) to "DEVICE_ENCODE_ALL_ABILITY_V20" (macro definition value: 0x008), and setting input buffer (**pInBuf**) to [XML_Desc_AudioVideoCompressInfo](#).

Multi-Stream Encoding Parameter

Get multi-stream encoding parameters

Call [NET_DVR_GetDeviceConfig](#) with
[NET_DVR_GET_MULTI_STREAM_COMPRESSIONCFG](#)

(command No.: 3216) and set condition buffer (**lpInBuffer**) to the structure **NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND**. The parameters are returned in the structure **NET_DVR_MULTI_STREAM_COMPRESSIONCFG** by output buffer (**lpOutBuffer**).

Set multi-stream encoding parameters

Call **NET_DVR_SetDeviceConfig** with

NET_DVR_SET_MULTI_STREAM_COMPRESSIONCFG

(command No.: 3217), set condition buffer (**lpInBuffer**) to the structure

NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND, and set input parameter buffer (**lpInParamBuffer**) to the structure **NET_DVR_MULTI_STREAM_COMPRESSIONCFG**.

Channel Encoding Parameter

Get stream encoding capability by channel

Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET </ISAPI/Streaming/channels/<ID>/capabilities>, set the input parameter pointer **lpInBuffer** to NULL. And the capability is returned by output parameter pointer **lpOutBuffer** in the message **XML_Cap_StreamingChannel**.

Get encoding parameters of one channel

Call **NET_DVR_GetDVRConfig** with

NET_DVR_GET_COMPRESSCFG_V30

(command No.: 1040). The encoding parameters are returned in the structure

NET_DVR_COMPRESSIONCFG_V30 by the output buffer (**lpOutBuffer**).

Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET </ISAPI/Streaming/channels/<ID>>, set the input parameter pointer **lpInBuffer** to NULL. And the encoding parameters are returned by output parameter pointer **lpOutBuffer** in the message **XML_StreamingChannel**.

Set encoding parameters of one channel

Call **NET_DVR_SetDVRConfig** with

NET_DVR_SET_COMPRESSCFG_V30

(command No.: 1041) and set the input buffer (**lpInBuffer**).

Call **NET_DVR_STDXMLConfig** to pass through the request URL: PUT </ISAPI/Streaming/channels/<ID>>, and set the input parameter pointer **lpInBuffer** to **XML_StreamingChannel**.

Audio Encoding Parameter

Get audio encoding parameters

Call **NET_DVR_GetDVRConfig** with

NET_DVR_GET_COMPRESSCFG_AUD

(command No.: 1058). The parameters are returned in the structure

NET_DVR_COMPRESSION_AUDIO by output buffer (**lpOutBuffer**).

Set audio encoding parameters

Call [NET_DVR_SetDVRConfig](#) with
NET_DVR_SET_COMPRESSCFG_AUD
(command No.: 1059) and set input buffer (**IplInBuffer**) to the structure
NET_DVR_COMPRESSION_AUDIO.

Get current audio encoding progress

Call [NET_DVR_GetCurrentAudioCompress_V50](#).

Chapter 6 Capture and Recording

For some remarkable views in the live view, you can record the video segments and save the captured pictures or videos to the configured storage. You can also configure recording schedule to record videos continuously or based on command or event.

Capture

Function	API
Get device's capture parameters	Call <u>NET_DVR_GetDVRConfig</u> with "NET_DVR_GET_JPEG_CAPTURE_CFG" (command No.: 1280) and set IChannel to the channel No. The parameters are returned in the structure <u>NET_DVR_JPEG_CAPTURE_CFG_V40</u> by IpOutBuffer .
Set device's capture parameters	Call <u>NET_DVR_SetDVRConfig</u> with "NET_DVR_SET_JPEG_CAPTURE_CFG" (command No.: 1281), set IChannel to the channel No., and set IpInBuffer to the structure <u>NET_DVR_JPEG_CAPTURE_CFG_V40</u> .
Get capability of digital video channel proxy	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/capabilities</u> by GET method. The capability is returned in the message <u>XML_Cap_StreamingChannel</u> by IpOutputParam
Get capability of capturing picture in URL format	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/PictureByUrl/capabilities?format=json</u> by GET method. The capability is returned in the message <u>JSON_Cap_CaputreDescription</u> by IpOutputParam
Capture picture in URL format	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/PictureByUrl?format=json</u> by POST method and set IpInputParam to <u>JSON_CaputreResult</u> .

Recording

Function	API
Get configuration capability of recording schedule	Call NET_DVR_STDXMLConfig to pass through the request URI: GET /ISAPI/ContentMgmt/record/tracks/<ID>/capabilities . The capability is returned in the message XML_Cap_Track by the output parameter IpOutputParam .
Get parameters of all recording schedules	Call NET_DVR_STDXMLConfig to pass through the request URI: GET /ISAPI/ContentMgmt/record/tracks . The parameters are returned in the message XML_TrackList by the output parameter IpOutputParam .
Set parameters of all recording schedules	Call NET_DVR_STDXMLConfig to pass through the request URI: PUT /ISAPI/ContentMgmt/record/tracks and set IpInBuffer of IpInputParam to the message XML_TrackList .
Get parameters of a specific recording schedule	Call NET_DVR_STDXMLConfig to pass through the request URI: GET /ISAPI/ContentMgmt/record/tracks/<ID> . The parameters are returned in the message XML_Track by the output parameter IpOutputParam .
Set parameters of a specific recording schedule	Call NET_DVR_STDXMLConfig to pass through the request URI: PUT /ISAPI/ContentMgmt/record/tracks/<ID> and set IpInBuffer of IpInputParam to the message XML_Track .

Chapter 7 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.

7.1 Supported Alarm/Event Types

This part lists the alarm or event types that can be received or subscribed via HCNetSDK and the corresponding commands for getting the details.

Alarm/Event Type	Value of <type>in Subscription Capability	ICommand (Command No.)	Alarm/Event Details
Sensor Alarm	IO	COMM_ALARM_V40 (0x4007)	<u>NET_DVR_ALARMINFO_V40</u>
Motion Detection Alarm	VMD		
Video Loss Alarm	videoloss		
Tampering Alarm	shelteralarm		
Defocus Alarm	defocus	COMM_ALARM_DEFOCUS (0x1151)	<u>NET_DVR_DEFOCUS_ALARM</u>

Alarm/Event Type	Value of <type>in Subscription Capability	ICommand (Command No.)	Alarm/Event Details
Audio Exception Alarm	audioexception	COMM_ALARM_AUDIOEXCEPTION (0x1150)	<u>NET_DVR_AUDIOEXCEPTION_ALARM</u>
Scene Changed Alarm	scenechangedetection	COMM_SCENECHANGE_DETECTION_UPLOAD (0x1130)	<u>NET_DVR_SCENECHANGE_DETECTION_RESULT</u>
Intrusion Detection	fielddetection	COMM_ALARM_RULE (0x1102)	Refer to <i>Device Network SDK (Behavior Analysis)_Developer Guide</i> for details.
Line Crossing Detection	linedetection		
Region Entrance Detection	regionEntrance		
Region Exiting Detection	regionExiting		
Loitering Detection	loitering		
People Gathering Detection	group		
Fast Moving Detection	rapidMove		
Parking Detection	parking		
Unattended Baggage Detection	unattendedBaggage		
Object Removal Detection	attendedBaggage		
Absence Alarm	leavePosition		

Alarm/Event Type	Value of <type>in Subscription Capability	ICommand (Command No.)	Alarm/Event Details
People Number Changed Alarm	peopleNumChange		
Fast Moving Alarm	running		
Violent Motion Alarm	violentMotion		
People Falling Down Alarm	failDown		
Overstay Alarm	retention		
PIR Alarm	PIR	COMM_IPC_AUXALARM_RESULT (0x2820)	<u>NET_IPC_AUXALARM_RESULT</u>
People Counting Alarm	PeopleCounting	COMM_ALARM_PDC (0x1103)	Refer to <i>Device Network SDK (People Counting)_Developer Guide</i>
People Counting Event of Single Frame	framePeopleCounting	COMM_FRAMES_PEOPLE_COUNTING_ALARM (0x6069)	
Temperature Alarm	thermometry	COMM_THERMOMETRY_ALARM (0x5212)	Refer to <i>Device Network SDK (Thermal Application)_Developer Guide</i> for details
Temperature Difference Alarm	temperature	COMM_THERMOMETRY_DIFF_ALARM (0x5211)	
Fire Source Detection	fireDetection	COMM_FIREDETECTION_ALARM (0x4991)	
Smoke Detection	smokeDetection		
Fire and Smoke Detection	smokeAndFireDetection		
Ship Detection	shipsDetection	COMM_ALARM_SHIPSDETECTION (0x4521)	

Alarm/Event Type	Value of <type>in Subscription Capability	ICommand (Command No.)	Alarm/Event Details
Face Thermometry Event	faceThermometry	COMM_FACE_THERMOMETRY_ALARM (0x4994)	
HDD Full	diskfull	COMM_ALARM_V40 (0x4007)	<u>NET_DVR_ALARMINFO_V40</u>
HDD Error	diskerror		
Network Disconnected	nicbroken		
IP Address Conflicted	ipconflict		
Illegal Login Alarm	illaccess		
PoE Power Exception Alarm	poe		
Video Exception Alarm	videoException		
Recording Exception Alarm	recordException		
HDD Unformatted Alarm	diskUnformat		
Supply Voltage Exception Alarm	voltageinstable		
Face Detection Alarm	facedetection	COMM_ALARM_FACE_DETECTION (0x4010)	Refer to <i>Device Network SDK (Facial Application)_Developer Guide</i> for details.
Face Capture Alarm	faceCapture	COMM_UPLOAD_FACESNAP_RESULT (0x1112)	

Alarm/Event Type	Value of <type>in Subscription Capability	ICommand (Command No.)	Alarm/Event Details
Face Picture Comparison Result	alarmResult	COMM_SNAP_MATCH_ALARM (0x2902)	
Stranger Detection Alarm	whiteFaceContrast		
High Frequently Person Detection	HFPD	COMM_HFPD_ALARM (0x6121)	
Waiting Time Detection Alarm	personQueueTime	COMM_VCA_ALARM (0x4993)	Refer to <i>Device Network SDK (Queue Management)_Developer Guide</i> for details
People Queuing-Up Alarm	personQueueCounting		
Real-Time Queue Management Alarm	personQueueRealTime		
Multi-Target-Type Detection Alarm	mixedTargetDetection		Refer to <i>Multi-Target-Type Detection Application_HCNetSDK User Manual</i> for details
Low Frequency Person Detection	LFPD		Refer to <i>Device Network SDK (Facial Application)_Developer Guide</i> for details
Hard Hat Detection	safetyHelmetDetection		Refer to <i>Device Network SDK (Behavior Analysis)_Developer Guide</i> for details
HDD High Temperature Detection	highHDTemperature		<u>JSON_EventNotificationAlert_HDDHighTemperatureEventMsg</u>

Alarm/Event Type	Value of <type> in Subscription Capability	ICommand (Command No.)	Alarm/Event Details
HDD Low Temperature Detection	lowHDTemperature		<u>JSON_EventNotificationAlert_HDDLowTemperatureEventMsg</u>
HDD Impact Detection	hdImpact		<u>JSON_EventNotificationAlert_HDDImpactEventMsg</u>
HDD Bad Sector Detection	hdBadBlock		<u>JSON_EventNotificationAlert_HDBadSectorEventMsg</u>
HDD Severe Fault Detection	severeHDFailure		<u>JSON_EventNotificationAlert_HDSevereFaultEventMsg</u>
Certificate Expiry Alarm	certificateRevocation		<u>JSON_EventNotificationAlert_CertificateExpiryAlarmMsg</u>
Auto Person Arming	personArmingTrack		Refer to <i>Intelligent Security API (Person Arming) _Developer Guide</i> for details
Manual Person Arming	manualPersonArmingTrack		
ANPR Alarm	ANPR	COMM_ITS_PLATE_RESULT (0x3050)	Refer to <i>ANPR Application_HCNetSDK User Manual</i> for details
Blocklist and Allowlist ANPR Alarm	vehicleMatchResult	COMM_VEHICLE_CONTROL_ALARM (0x3059)	
Vehicle Feature Recognition Alarm	vehicleRecognitionResult	COMM_VEHICLE_RECOG_RESULT (0x3062)	
Heat Map Alarm	heatmap	COMM_UPLOAD_HEATMAP_RESULT (0x4008)	Refer to <i>Heat Map Application_HCNetSDK User Manual</i> for details
Intersection Analysis Event	intersectionAnalysis	COMM_UPLOAD_HEATMAP_RESULT_INTERSECTION (0x4020)	
Face and ID Card	cardMatch	COMM_ID_INFO_ALARM (0x5200)	Refer to <i>Device Network SDK (Access Control on Card) _Developer Guide</i> for details

Alarm/Event Type	Value of <type>in Subscription Capability	ICommand (Command No.)	Alarm/Event Details
Recognition Alarm			
Hot Spare Exception Alarm	hotSpare	COMM_ALARM_HOT_SPARE (0x4006)	<u>NET_DVR_ALARM_HOT_SPARE</u>

7.2 Configure Exception Alarm

To monitor the device status, you can configure the exception alarm, such as supply voltage exception, PoE power exception, and so on. When the exception occurs, the configured linkage action will be triggered and the alarm information will be uploaded automatically.

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

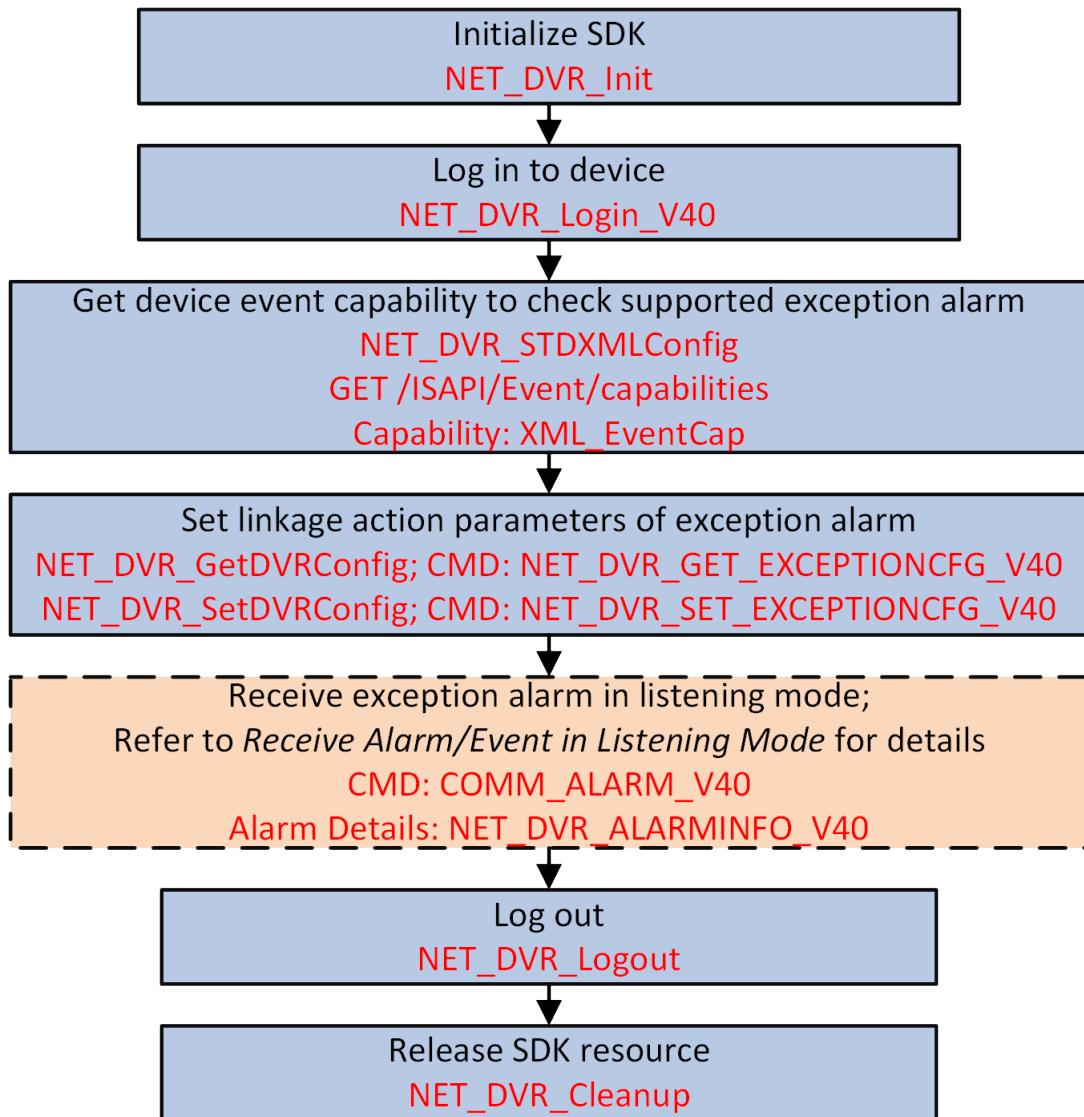


Figure 7-1 Programming Flow of Configuring Exception Alarm

1. Call **NET_DVR_STDXMLConfig** to pass through the request URL: **GET /ISAPI/Event/capabilities** for getting the event capability to check the supported exception alarm.
The event capability is returned in the message **XML_EventCap** by **IpOutBuffer**.
2. Call **NET_DVR_SetDVRConfig** with "NET_DVR_SET_EXCEPTIONCFG_V40" (command No.: 6178) and set **IpInBuffer** to **NET_DVR_EXCEPTION_V40** for setting the linkage action parameters of exception alarm.



Note

Before setting the linkage action parameters, you can call [NET_DVR_GetDVRConfig](#) with "NET_DVR_GET_EXCEPTIONCFG_V40" (command No.: 6177) for getting the default or configured linkage action parameters.

-
3. **Optional:** Configure arming mode (refer to [Receive Alarm/Event in Listening Mode](#)) and set **ICommand** in the registered alarm callback function ([MSGCallBack](#)) to "COMM_ALARM_V40" (command No.: 0x4007).

The exception alarm information is returned in the structure [NET_DVR_ALARMINFO_V40](#).

What to do next

Call [NET_DVR_Logout](#) and [NET_DVR_Cleanup](#) to log out and release the resources.

7.3 Configure Audio Exception Detection Alarm

When the sound in the environment is detected as the sudden increase or decrease of sound intensity, the audio exception detection alarm will be triggered. You can configure linkage actions, such as alarm output, uploading alarms to the center, etc., for the audio exception detection alarm.

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

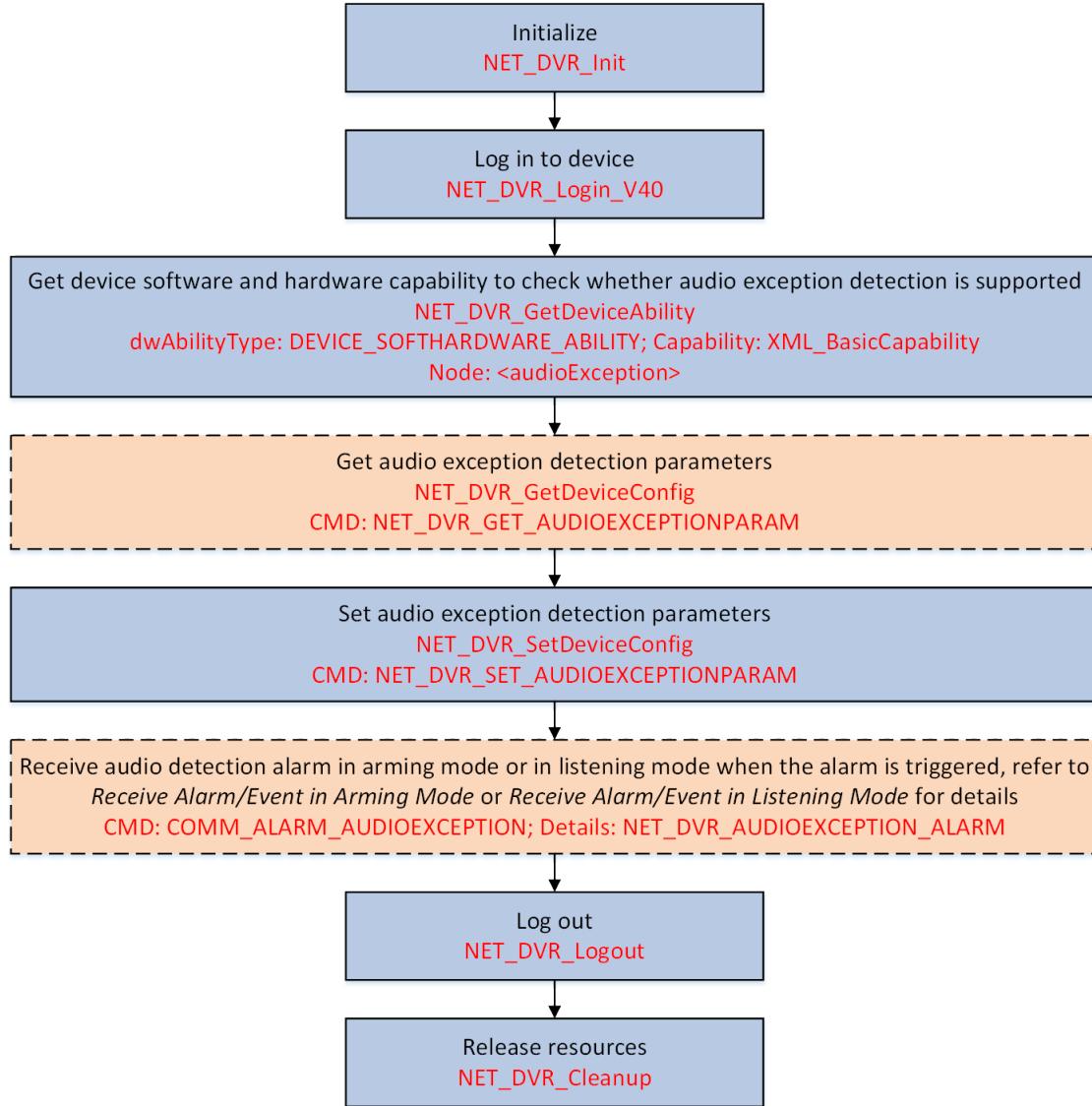


Figure 7-2 Programming Flow of Configuring Audio Exception Detection Alarm

1. Call **NET_DVR_GetDeviceAbility** and set the capability type (**dwAbilityType**) to "DEVICE_SOFTHARDWARE_ABILITY" (macro definition value: 0x001) for getting the device software and hardware capability to check if the audio exception detection is supported by the device.
The device software and hardware capability is returned in the message **XML_BasicCapability** by output buffer (**pOutBuf**).
If the node **<audioException>** is returned in the message and its value is true, it indicates that the audio exception detection is supported, and you can continue to perform this task.
Otherwise, the audio exception detection is not supported by the device, please end this task.

2. Optional: Call [**NET_DVR_GetDeviceConfig**](#) with "NET_DVR_GET_AUDIOEXCEPTIONPARAM" (command No.: 3366), and set the input parameter pointer (**IplInBuffer**) to the structure [**NET_DVR_CHANNEL_GROUP**](#) for getting the audio exception detection parameters, including detection rule, arming schedule, alarm linkage, and so on, for reference.

The audio exception detection parameters is returned in the structure [**NET_DVR_AUDIO_EXCEPTION**](#) by output parameter **IplOutBuffer**.

3. Call [**NET_DVR_SetDeviceConfig**](#) with "NET_DVR_SET_AUDIOEXCEPTIONPARAM" (command No.: 3367), set the input parameter pointer (**IplInBuffer**) to the structure [**NET_DVR_CHANNEL_GROUP**](#), and set the input parameter (**IplInParamBuffer**) to the structure [**NET_DVR_AUDIO_EXCEPTION**](#) for setting the audio exception detection parameters.



Note

- To receive the alarm in the platform, the linkage action must be set to "0x04" (upload to center).
- The above audio exception detection parameters can also be configured by logging in to the device via web browser.

4. Receive the audio exception alarm in arming mode (see [**Receive Alarm/Event in Arming Mode**](#)) or in listening mode (see [**Receive Alarm/Event in Listening Mode**](#)) when the alarm is triggered.



Note

- The command (**ICommand**) to receive audio exception alarms should be set to "COMM_ALARM_AUDIOEXCEPTION" (command No.: 0x1150) in the APIs [**NET_DVR_SetDVRMessageCallBack_V50**](#) and [**NET_DVR_StartListen_V30**](#).
 - For the alarm details, refer to the structure [**NET_DVR_AUDIOEXCEPTION_ALARM**](#).
-

What to do next

Call [**NET_DVR_Logout**](#) and [**NET_DVR_Cleanup**](#) to log out and release resources.

7.4 Configure Certificate Expiry Alarm

Generally, the device certificate is only valid in a specific period of time. You can configure the certificate expiry alarm to remind the user a few days in advance. When the certificate is expired, the alarm will be triggered and uploaded automatically.

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

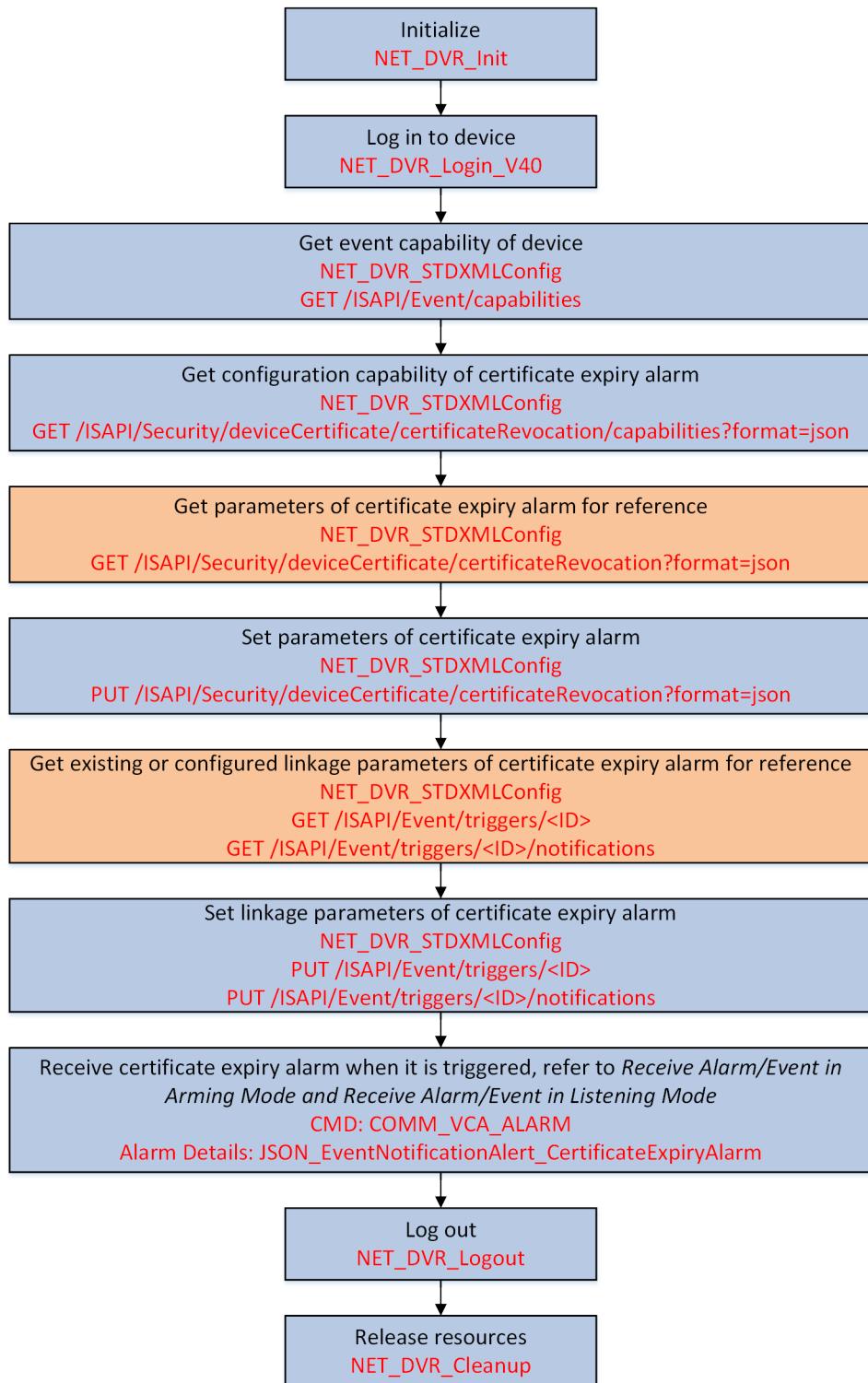


Figure 7-3 Programming Flow of Configuring Certificate Expiry Alarm

1. Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Event/capabilities](#) to get the event capability of the device and check whether the device supports certificate expiry alarm.

The event capability of the device is returned in the message [XML_EventCap](#) by the output parameter [IpOutputParam](#). If the certificate expiry alarm is supported, the node [<isSupportCertificateRevocation>](#) will be returned and its value is "true", then you can perform the following steps. Otherwise, please end this task.

2. Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Security/deviceCertificate/certificateRevocation/capabilities?format=json](#) to get the configuration capability of certificate expiry alarm and know the supported parameters that can be configured.

The configuration capability is returned in the message [JSON_CertificateRevocationCap](#) by the output parameter [IpOutputParam](#).

3. **Optional:** Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Security/deviceCertificate/certificateRevocation?format=json](#) to get the existing or configured parameters of certificate expiry alarm for reference.

The parameters are returned in the message [JSON_CertificateRevocation](#) by the output parameter [IpOutputParam](#).

4. Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/Security/deviceCertificate/certificateRevocation?format=json](#) and set [IpInBuffer](#) of [IpInputParam](#) to the message [JSON_CertificateRevocation](#) for setting parameters of certificate expiry alarm.

5. **Optional:** Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Event/triggers/<eventType>-<channelID>](#) or [/ISAPI/Event/triggers/<eventType>-<channelID>/notifications](#) to get the existing or configured linkage parameters of certificate expiry alarm for reference.



The <ID> in the request URL refers to the channel No., and it should be set to the format "certificateRevocation-<channelID>".

The parameters are returned in the message [XML_EventTrigger](#) or [XML_EventTriggerNotificationList](#) by the output parameter [IpOutputParam](#).

6. Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/Event/triggers/<eventType>-<channelID>](#) or [/ISAPI/Event/triggers/<eventType>-<channelID>/notifications](#) and set [IpInBuffer](#) of [IpInputParam](#) to the message [XML_EventTrigger](#) or [XML_EventTriggerNotificationList](#) for setting the linkage parameters of certificate expiry alarm.



The <ID> in the request URL refers to the channel No., and it should be set to the format "certificateRevocation-<channelID>".

7. **Optional:** Set [ICommand](#) of alarm/event callback function [MSGCallBack](#) to "COMM_VCA_ALARM" (command No.: 0x4993) for receiving certificate expiry alarm in arming mode (refer to [Receive Alarm/Event in Arming Mode](#) for details) or listening mode (refer to [Receive Alarm/Event in Listening Mode](#) for details).

The certificate expiry alarm information is called back in the message [JSON_EventNotificationAlert_CertificateExpiryAlarmMsg](#).

What to do next

Call [NET_DVR_Logout](#) and [NET_DVR_Cleanup](#) to log out of the device and release the resources.

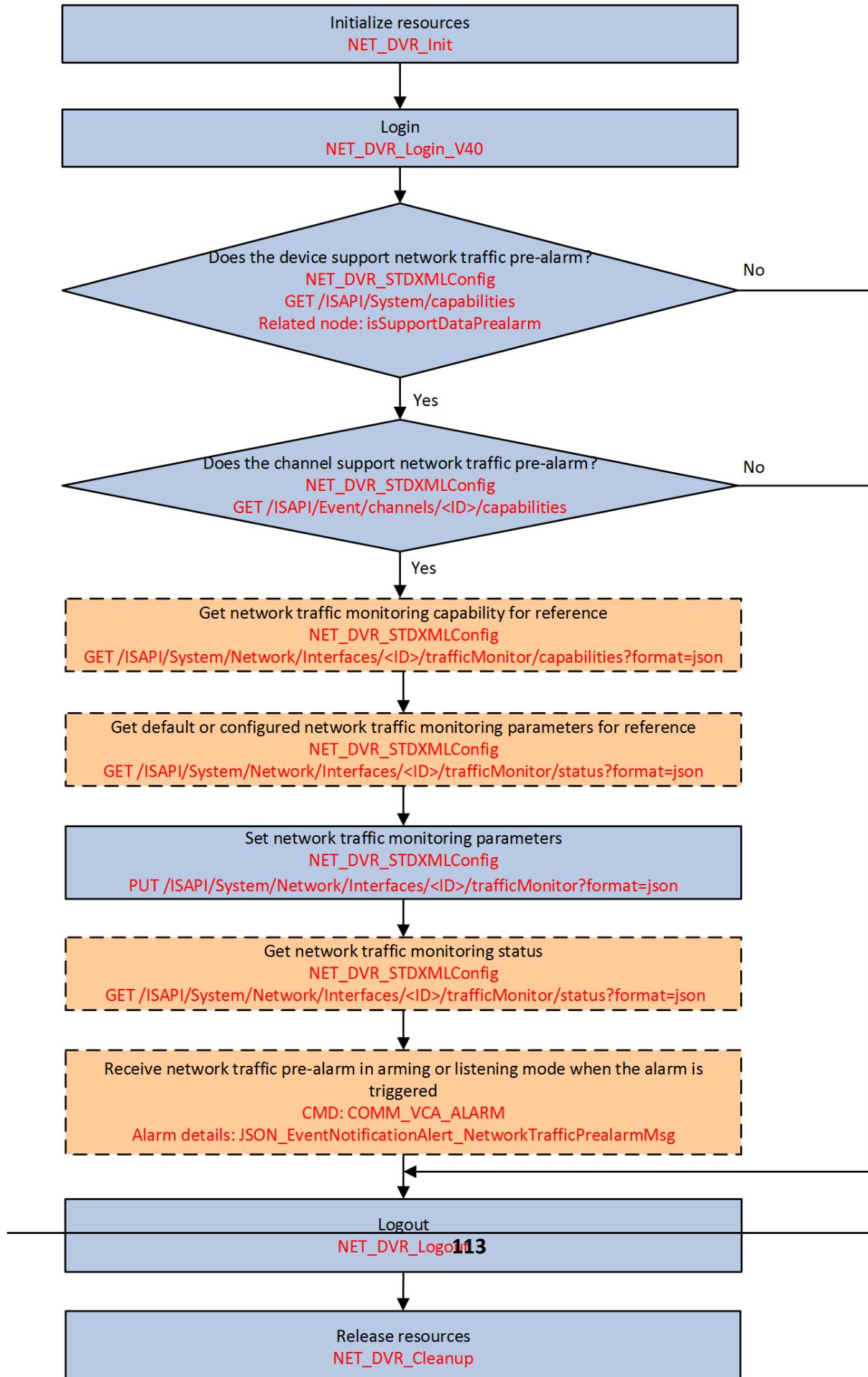
7.5 Configure Network Traffic Pre-alarm

You can set or search for data plans, monitor data usage, and set a traffic limit. When the limit is exceeded, the alarm will be triggered and uploaded automatically.

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 transmit the request URI: GET **/ISAPI/System/capabilities** to check whether the device supports network traffic pre-alarm.
2. Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Event/channels/<ID>/capabilities** to check whether the channel supports network traffic pre-alarm.
3. **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor/capabilities?format=json** to get network traffic monitoring capability for reference.
The capability is returned in the message **JSON_TrafficMonitorCap** by **IpOutBuffer**.
4. **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor?format=json** to get default or configured network traffic monitoring parameters for reference.
The parameters are returned in the message **JSON_TrafficMonitor** by **IpOutBuffer**.
5. Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor?format=json** to set network traffic monitoring parameters.
6. **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor/status?format=json** to get network traffic monitoring status.
The status is returned in the message **JSON_TrafficMonitorStatus** by **IpOutBuffer**.
7. **Optional:** Set **ICommand** of alarm/event callback function **MSGCallBack** to "COMM_VCA_ALARM" (command No.: 0x4993) and set for receiving network traffic alarm in arming mode (refer to **Receive Alarm/Event in Arming Mode** for details) or listening mode (refer to **Receive Alarm/Event in Listening Mode** for details).



Note

The traffic pre-alarm only supports the default linkage action (upload to center) and arming schedule (all-day schedule).

The network traffic alarm information is called back in the message

JSON_EventNotificationAlert_NetworkTrafficPrealarmMsg.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out of the device and release the resources.

7.6 Configure Vibration Detection Alarm

Vibration detection function can be applied to the scenes in which the real-time vibration status of devices should be detected in case people damage the devices on purpose. You can configure the arming schedule and linkage actions of the vibration detection alarm. For those triggered alarms, you can also search for the related pictures.

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

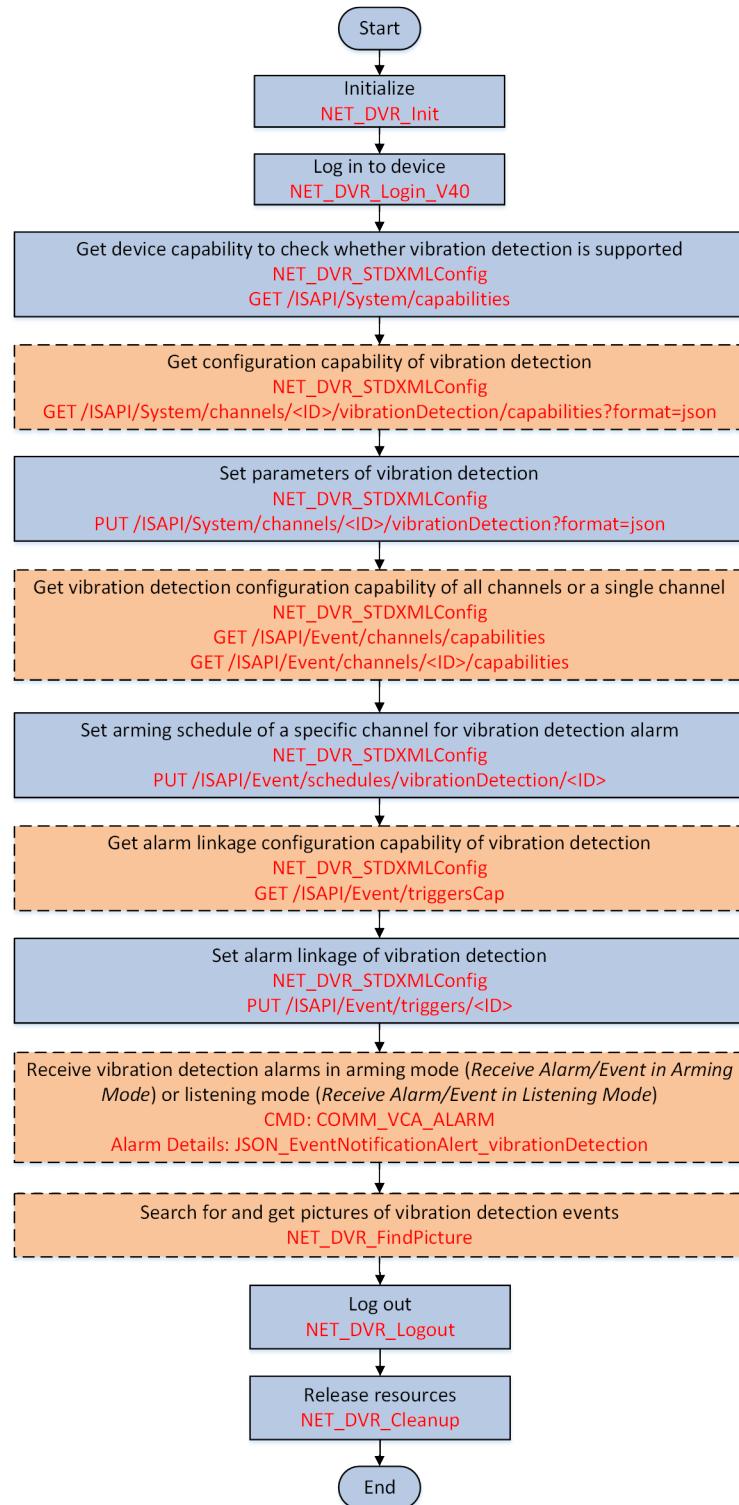


Figure 7-5 API Calling Flow of Configuring Vibration Detection Alarm

1. Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/System/capabilities** for getting the device capability to check whether vibration detection is supported by the device. The device capability is returned by **IpOutputParam** in the message **XML_DeviceCap**. If the node <isSupportVibrationDetection> is returned in the message and its value is "true", it indicates that the vibration detection is supported, and then you can continue to perform the following steps.
Otherwise, the vibration detection is not supported, please end this task.
2. **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/System/channels/<ID>/vibrationDetection/capabilities?format=json** for getting configuration capability of vibration detection.
The configuration capability of vibration detection is returned by **IpOutputParam** in the message **JSON_VibrationDetectionCap**.
3. Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/System/channels/<ID>/vibrationDetection?format=json** and set **IpInputParam** to the message **JSON_VibrationDetection** for setting parameters of vibration detection.
4. **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Event/channels/capabilities** or **/ISAPI/Event/channels/<ID>/capabilities** for getting event configuration capability of all channels or a single channel for knowing the configuration details or notices.
The event configuration capability is returned by **IpOutputParam** in the message **XML_ChannelEventCapList** or **XML_ChannelEventCap**.
5. Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Event/schedules/vibrationDetection/<ID>**, set the <ID> in the request URI in the format of "vibrationDetection-<channelID>", e.g., /ISAPI/Event/schedules/vibrationDetection/vibrationDetection-1, and set **IpInputParam** to the message **XML_Schedule** for setting arming schedule of a specific channel for vibration detection.
6. **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Event/triggersCap** for getting alarm linkage configuration capability of vibration detection.
The alarm linkage capability is returned by the output parameter pointer **IpOutputParam** in the message **XML_EventTriggersCap**.
7. Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Event/triggers/<eventType>-<channelID>** and set the <ID> in the request URI in the format of "vibrationDetection-<channelID>", e.g., /ISAPI/Event/triggers/vibrationDetection/vibrationDetection-1, and set **IpInputParam** to the message **XML_EventTrigger** for setting the alarm linkages of vibration detection alarms.
8. **Optional:** Set the uploading message type (**ICommand**) in the alarm callback function (**MSGCallBack**) to "COMM_VCA_ALARM" (macro definition value: 0x4993) for receiving vibration detection alarm in arming mode (refer to **Receive Alarm/Event in Arming Mode** for details) or in listening mode (refer to **Receive Alarm/Event in Listening Mode** for details) when alarm is triggered.



Note

The vibration detection alarm details is called back in the message ***JSON EventNotificationAlert vibrationDetection*** and the corresponding alarm type (**eventType**) is "vibrationDetection".

9. **Optional:** Call ***NET DVR FindPicture*** and set **byFileType** to "0x58" in the structure ***NET DVR FIND PICTURE PARAM*** to search for the pictures of the vibration detection events that trigger alarms.
- 1) Call ***NET DVR FindNextPicture V50*** to get the pictures.
 - 2) Call ***NET DVR CloseFindPicture*** to stop searching for pictures and release resources.

What to do next

Call ***NET DVR Logout*** and ***NET DVR Cleanup*** to log out of the device and release resources.

7.7 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

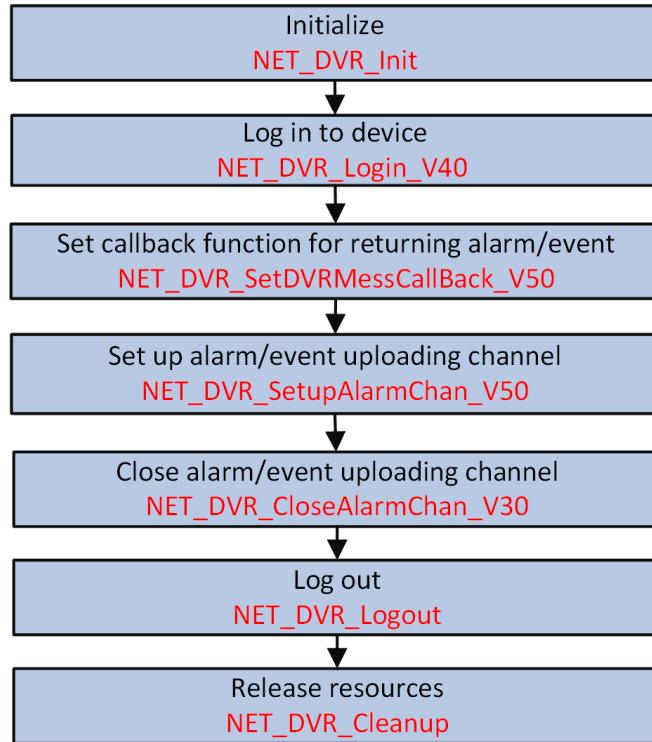


Figure 7-6 Programming Flow of Receiving Alarm/Event in Arming Mode

1. Call [NET_DVR_SetDVRMessageCallBack_V50](#) to set callback function for returning alarm/event information.



- 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_SETUPALARM_PARAM_V50 struSetupParamV50={0};
struSetupParamV50.dwSize=sizeof(NET_DVR_SETUPALARM_PARAM_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
IHandle = NET_DVR_SetupAlarmChan_V50(lUserID, &struSetupParamV50, NULL, strlen(szSubscribe));
}
else
{
//Subscribe
LIHandle = NET_DVR_SetupAlarmChan_V50(lUserID, &struSetupParamV50, szSubscribe, strlen(szSubscribe));
}

if (IHandle < 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(IHandle))
{
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.

7.8 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

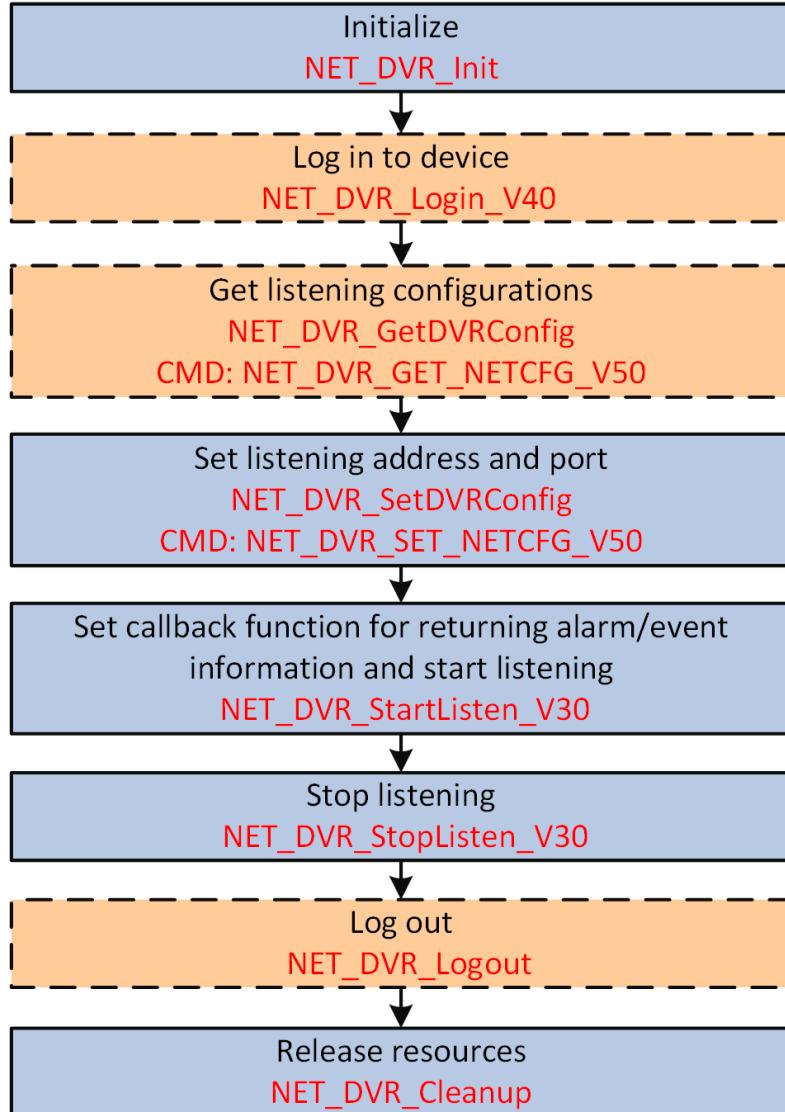


Figure 7-7 Programming Flow of Receiving Alarm/Event in Listening Mode

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.

7.9 Subscribe Alarm/Event in Arming Mode

For arming mode, the platform will connect to the devices automatically and send commands to the devices for uploading alarm/event information when the alarm is triggered or event occurred. To reduce the CPU and bandwidth usage of platform, and improve the device processing performance, the platform can subscribe alarm/event types to receive alarm/event information as required.

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

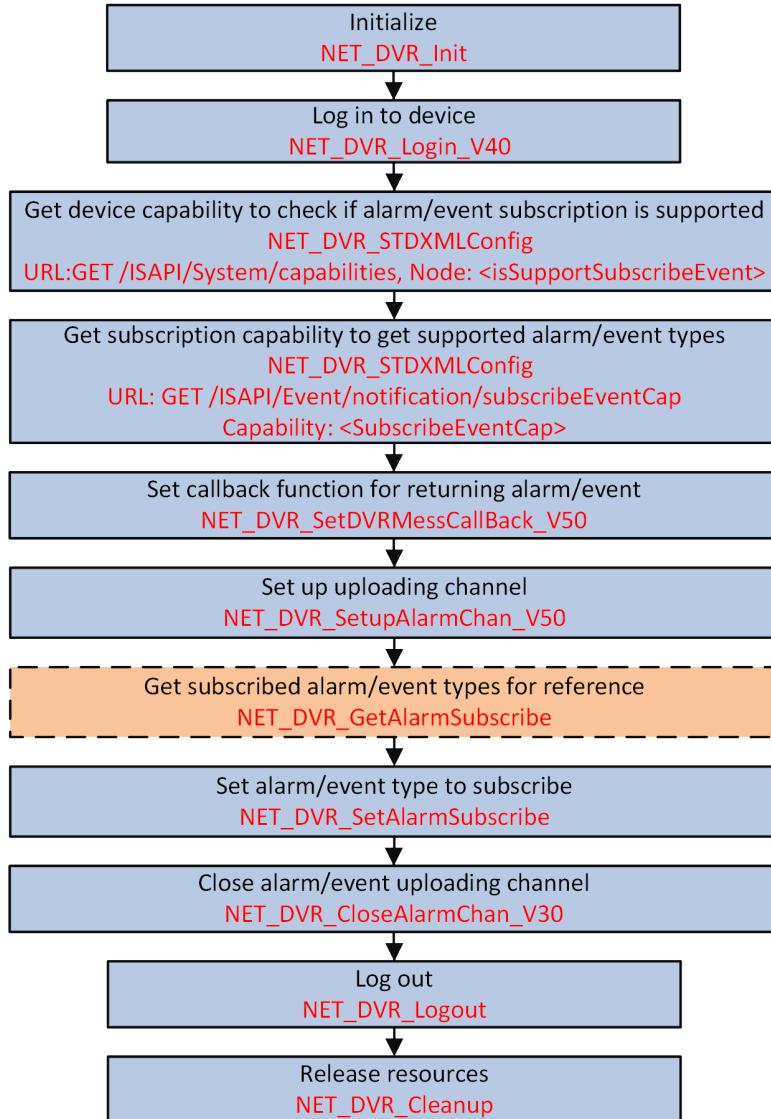


Figure 7-8 Programming Flow of Subscribing Alarm/Event in Arming Mode

1. Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/System/capabilities** for getting device capability to check if alarm/event subscription is supported.

The device capability is returned in the message **XML_DeviceCap** by the output parameter (**lpOutputParam**) pointer.

If the node **<isSupportSubscribeEvent>** is also returned in the message and its value is "true", it indicates that alarm/event subscription is supported by device, and you can continue to perform the following steps;

Otherwise, alarm/event subscription is not supported, please end the task.

2. Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Event/notification/subscribeEventCap** for getting subscription capability, which contains supported alarm/event types.

The alarm/event subscription capability is returned in the message **XML_SubscribeEventCap** by the output parameter (**IpOutputParam**) pointer.

3. Call **NET_DVR_SetDVRMessageCallBack_V50** to set callback function for returning alarm/event information or subscription failed 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.
- 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 *Supported Alarm/Event Types* for details).
- To receive the subscription result (subscription failed), the parameter **ICommand** (data type to be uploaded) in the configured callback function should be set to "COMM_ALARM_SUBSCRIBE_EVENT". And the result is returned in the message **XML_SubscribeEventResponse**

-
4. Call **NET_DVR_SetupAlarmChan_V50** to set up alarm/event uploading channel.

5. **Optional:** Call **NET_DVR_GetAlarmSubscribe** to get subscribed alarm/event types for reference.

6. Call **NET_DVR_SetAlarmSubscribe** to set alarm/event type to subscribe.

7. Call **NET_DVR_CloseAlarmChan_V30** to close alarm/event uploading channel and finishing receiving.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release resources.

7.10 Search for Alarms or Events

You can search for history alarms or events saved in the storage medium by specific conditions. You can also get the real-time alarm or event immediately when the alarm is triggered or the event occurs.

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 the capability type **dwAbilityType** to "DEVICE_ABILITY_INFO" (macro definition value: 0x011), and set the input parameter pointer **pInBuf** to the message **XML_Desc_EventAbility** for getting the event capability of the device to check whether searching for alarms or events is supported.

The capability is returned in the message **XML_EventAbility** by the output parameter pointer **pOutBuf**.

If the node <**AlarmSearch**> is returned in the message, it indicates that searching for alarms or events is supported, and you can continue to perform the following steps.

Otherwise, searching for alarms or events is not supported, please end this task.

2. Call **NET_DVR_StartRemoteConfig** with the command "NET_DVR_GET_ALARM_INFO" (command No.: 4193) and set **lpInBuffer** to the structure **NET_DVR_ALARM_SEARCH_COND** for setting up the persistent connection and set the callback function (**fRemoteConfigCallback**) for searching for alarms or events by specific conditions.
-



For getting the real-time alarm or event, **strStartTime** and **strStopTime** in the structure **NET_DVR_ALARM_SEARCH_COND** do not need to be configured.

The search result is returned in the structure **NET_DVR_ALARM_SEARCH_RESULT** by **lpBuffer** of the callback function.

3. When the needed information is obtained or searching failed, call **NET_DVR_StopRemoteConfig** to disconnect the persistent connection and stop searching.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out of the device and release the resources.

Chapter 8 File and Storage Management

8.1 Disk Quota

Get Disk Quota Information

- Call [NET_DVR_GetDVRConfig](#) with "NET_DVR_GET_DISK_QUOTA_CFG" (command No.: 1278). And the quota information is returned in the structure [NET_DVR_DISK_QUOTA_CFG](#) by the output pointer (**lpOutBuffer**).
- Call [NET_DVR_GetDVRConfig](#) with "NET_DVR_GET_DISK_QUOTA_CFG_V60" (command No.: 1292) and set **IChannel** to the channel No. The quota information is returned in the structure [NET_DVR_DISK_QUOTA_CFG_V60](#) by **lpOutBuffer**.

Set Disk Quota Parameter

- Call [NET_DVR_SetDVRConfig](#) with "NET_DVR_SET_DISK_QUOTA_CFG" (command No.: 1279) and set the input pointer (**lpInBuffer**) to [NET_DVR_DISK_QUOTA_CFG](#).
- Call [NET_DVR_SetDVRConfig](#) with "NET_DVR_SET_DISK_QUOTA_CFG_V60" (command No.: 1293), set **IChannel** to the channel No., and set **lpInBuffer** to [NET_DVR_DISK_QUOTA_CFG_V60](#).



Note

To check whether setting disk quota parameters is supported, you can call [NET_DVR_GetDeviceAbility](#) and set the capability type **dwAbilityType** to "DEVICE_SOFTWARE_HARDWARE_ABILITY" (macro definition value: 0x001) for getting the device software and hardware capability. The capability is returned in the message [XML_BasicCapability](#) by **pOutBuf**. If this function is supported, the node **QuotaRatio** will be returned and its value is 1.

8.2 Download Video Files

You can download the video files stored in the device to the local PC or back up the files for other processes.

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

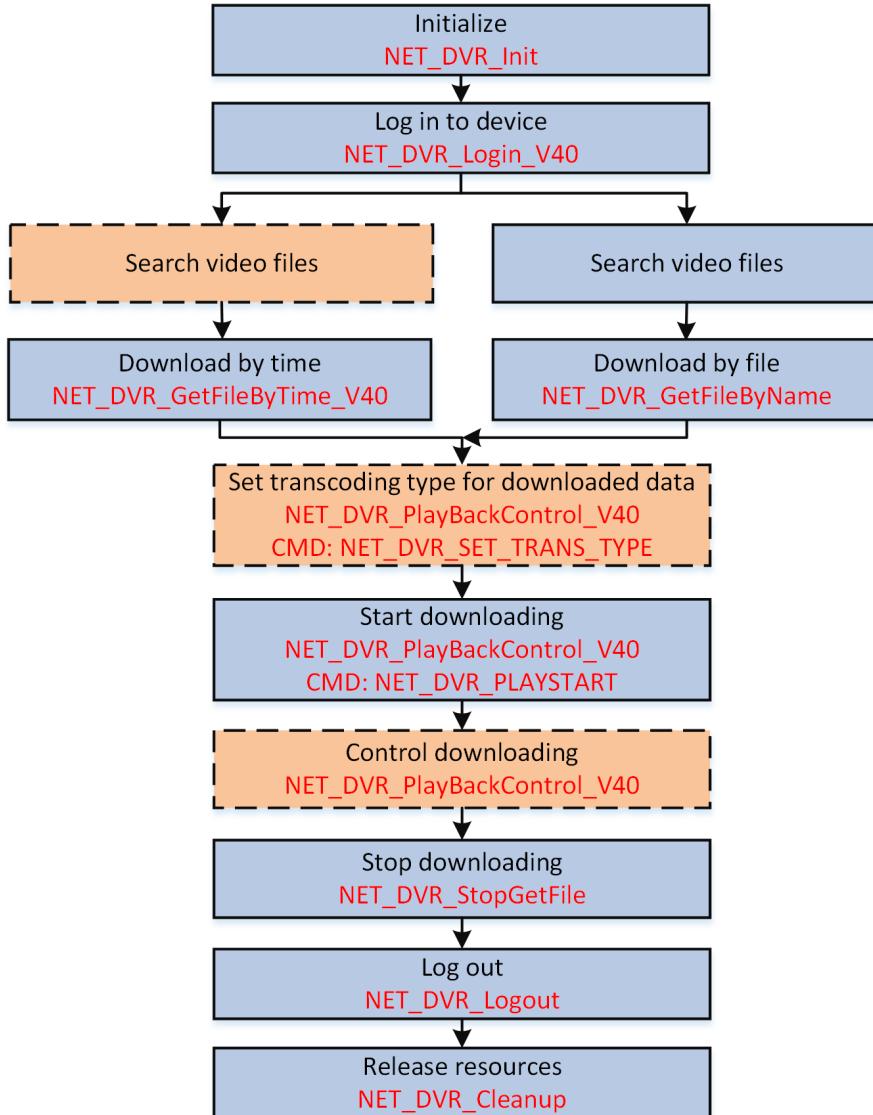


Figure 8-1 Programming Flow of Downloading Video Files

1. Perform one of the following operations to download video files by time or by file.
 - a. Search video files to get the time period that contains videos.
 - b. Call **NET_DVR_GetFileByTime_V40** to specify the start time and end time according to the found time period to download all video files.
- a. Search video files to get the file name.
- b. Call **NET_DVR_GetFileByName** to specify files for downloading according to the obtained file name.
- a. Call **NET_DVR_STDXMLConfig** to transmit **/ISAPI/ContentMgmt/download/capabilities** by GET method to get downloading capability to check whether the device supports downloading video files by file URL.

The capability is returned in the message **XML_DownloadAbility** by **IpOutputParam**. If it supports, the node <isSupportDownloadbyFileName> is returned and its value is "true", otherwise, downloading files by file URL is not supported.

- b. Call **NET_DVR_STDXMLConfig** to transmit **/ISAPI/ContentMgmt/download** by POST method and set **IpInputParam** to **XML_downloadRequest**.
-



If you choose this operation to download files, the following steps are not required.

2. **Optional:** Call **NET_DVR_PlayBackControl_V40** with the command of **NET_DVR_SET_TRANS_TYPE** (command No.: 32) to set the transcoding type for the data in the downloaded files.
-



The input parameter **IpInBuffer** refers to a 4-byte integer container format value, which contains 1-PS, 2-TS, 3-RTP, 5-MP4 (3GPP, it is supported only when downloading by file).

3. Call **NET_DVR_PlayBackControl_V40** with the command of **NET_DVR_PLAYSTART** (command No.: 1) to start downloading.
-



The input parameter **IpInBuffer** refers to a 4-byte integer offset, which is used to implement the ANR function.

4. **Optional:** Call **NET_DVR_PlayBackControl_V40** via different commands to control the downloading, such as pausing downloading, getting downloading progress, and so on.
-



When downloading by time, getting and setting downloading progress are not supported.

5. Call **NET_DVR_StopGetFile** to stop downloading.
-

Example

Sample Code of Downloading Video Files by Time

```
#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"); //IP address
struLoginInfo.wPort = 8000; //Service port
strcpy(struLoginInfo.sUserName, "admin"); //User name
strcpy(struLoginInfo.sPassword, "abcd1234"); //Password

//Device information, output parameters
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;
}

NET_DVR_PLAYCOND struDownloadCond={0};
struDownloadCond.dwChannel=1;

struDownloadCond.struStartTime.dwYear  = 2013;
struDownloadCond.struStartTime.dwMonth = 6;
struDownloadCond.struStartTime.dwDay   = 14;
struDownloadCond.struStartTime.dwHour  = 9;
struDownloadCond.struStartTime.dwMinute = 50;
struDownloadCond.struStartTime.dwSecond =0;
struDownloadCond.struStopTime.dwYear   = 2013;
struDownloadCond.struStopTime.dwMonth  = 6;
struDownloadCond.struStopTime.dwDay    = 14;
struDownloadCond.struStopTime.dwHour   = 10;
struDownloadCond.struStopTime.dwMinute = 7;
struDownloadCond.struStopTime.dwSecond = 0;

//-----
//Download by time
int hPlayback;
hPlayback = NET_DVR_GetFileByTime_V40(lUserID, "./test.mp4",&struDownloadCond);
if(hPlayback < 0)
{
    printf("NET_DVR_GetFileByTime_V40 fail,last error %d\n",NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//-----
//Start downloading
```

```
if(!NET_DVR_PlayBackControl_V40(hPlayback, NET_DVR_PLAYSTART, NULL, 0, NULL,NULL))
{
    printf("Play back control failed [%d]\n",NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

int nPos = 0;
for(nPos = 0; nPos < 100&&nPos>=0; nPos = NET_DVR_GetDownloadPos(hPlayback))
{
    printf("Be downloading... %d %%\n",nPos);
    Sleep(5000); //millisecond
}
if(!NET_DVR_StopGetFile(hPlayback))
{
    printf("failed to stop get file [%d]\n",NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}
if(nPos<0| |nPos>100)
{
    printf("download err [%d]\n",NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}
printf("Be downloading... %d %%\n",nPos);

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

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release resources.

8.3 HDD Management

The HDD (Hard Disk Drive) is used for storage. Before storing information in HDD, you must format it, and you can also reallocate its space for better storage management.

HDD Configuration

Get HDD management and configuration capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/Storage/hdd/capabilities</u> by GET method. The capability is returned in the message <u>XML_Cap_hddList</u> by IpOutputParam .
Get HDD parameters	Call <u>NET_DVR_GetDVRConfig</u> with "NET_DVR_GET_HDCFG_V50" (command No.: 4153) and set IChannel to the HDD group No. (starts from 0 and up to 33 HDDs are allowed in a group). The HDD parameters are returned in the structure <u>NET_DVR_HDCFG_V50</u> by the output buffer (IpOutBuffer)
Set HDD parameters	Call <u>NET_DVR_SetDVRConfig</u> with "NET_DVR_SET_HDCFG_V50" (command No.: 4154), set IChannel to the HDD group No. (starts from 0 and up to 33 HDDs are allowed in a group), and set IpInBuffer to <u>NET_DVR_HDCFG_V50</u> .
Get parameters of a HDD	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/Storage/hdd/<ID></u> by GET method. The parameters are returned in the message <u>XML_hdd</u> by IpOutputParam .
Set parameters of a HDD	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/Storage/hdd/<ID></u> by PUT method and set IpInputParam to <u>XML_hdd</u> .
Verify HDD encryption password	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/Storage/hdd/<ID>/encryptVerfy?format=json</u> by PUT method and set IpInputParam to <u>JSON_EncryptVerfy</u> .

HDD Formatting

Format a HDD	<ol style="list-style-type: none"> Call <i>NET_DVR_FormatDisk</i> to start formatting a HDD remotely. Call <i>NET_DVR_GetFormatProgress</i> to get the formatting progress of the HDD. Call <i>NET_DVR_CloseFormatHandle</i> to stop formatting the HDD.
Format HDDs in a batch	Call <i>NET_DVR_STDXMLConfig</i> to transmit <i>/ISAPI/ContentMgmt/Storage/hdd/specifyHddFormat?format=json</i> by PUT method and set <i>lpInputParam</i> to <i>JSON_HddFormatList</i> .
Format an encrypted HDD	Call <i>NET_DVR_STDXMLConfig</i> to transmit <i>/ISAPI/ContentMgmt/Storage/hdd/<ID>/encryptFormat?format=json</i> by PUT method and set <i>lpInputParam</i> to <i>JSON_EncryptFormat</i> .

HDD Clearing

Get HDD clearing capability	<p>Call <i>NET_DVR_STDXMLConfig</i> to transmit <i>/ISAPI/ContentMgmt/storage/hdd/clearingSpace/capabilities?format=json</i> by GET method.</p> <p>The capability is returned in the message <i>JSON_Cap_ClearingSpaceConfig</i> by <i>lpOutputParam</i>.</p>
Get HDD clearing parameters	<p>Call <i>NET_DVR_STDXMLConfig</i> to transmit <i>/ISAPI/ContentMgmt/storage/hdd/clearingSpace?format=json</i> by GET method.</p> <p>The parameters are returned in the message <i>JSON_ClearingSpaceConfig</i> by <i>lpOutputParam</i>.</p>
Set HDD clearing parameters	<p>Call <i>NET_DVR_STDXMLConfig</i> to transmit <i>/ISAPI/ContentMgmt/storage/hdd/clearingSpace?format=json</i> by PUT method and set <i>lpInputParam</i> to <i>JSON_ClearingSpaceConfig</i>.</p>

SHM (SkyHawk Health Management) Alarm Linkage Settings

Get alarm linkage parameters of HDD high temperature detection	Call <i>NET_DVR_STDXMLConfig</i> to transmit <i>/ISAPI/Event/triggers/highHDTemperature</i> by GET method.
--	--

	The parameters are returned in the message <u>XML_EventTrigger</u> by <u>IpOutputParam</u> .
Set alarm linkage parameters of HDD high temperature detection	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Event/triggers/highHDTemperature</u> by PUT method and set <u>IpInputParam</u> to <u>XML_EventTrigger</u> .
Get alarm linkage parameters of HDD low temperature detection	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Event/triggers/lowHDTemperature</u> by GET method. The parameters are returned in the message <u>XML_EventTrigger</u> by <u>IpOutputParam</u> .
Set alarm linkage parameters of HDD low temperature detection	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Event/triggers/lowHDTemperature</u> by PUT method and set <u>IpInputParam</u> to <u>XML_EventTrigger</u> .
Get alarm linkage parameters of HDD impact detection	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Event/triggers/hdImpact</u> by GET method. The parameters are returned in the message <u>XML_EventTrigger</u> by <u>IpOutputParam</u> .
Set alarm linkage parameters of HDD impact detection	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Event/triggers/hdImpact</u> by PUT method and set <u>IpInputParam</u> to <u>XML_EventTrigger</u> .
Get alarm linkage parameters of HDD bad sector detection	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Event/triggers/hdBadBlock</u> by GET method. The parameters are returned in the message <u>XML_EventTrigger</u> by <u>IpOutputParam</u> .
Set alarm linkage parameters of HDD bad sector detection	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Event/triggers/hdBadBlock</u> by PUT method and set <u>IpInputParam</u> to <u>XML_EventTrigger</u> .
Get alarm linkage parameters of HDD severe fault detection	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Event/triggers/severeHDFailure</u> by GET method. The parameters are returned in the message <u>XML_EventTrigger</u> by <u>IpOutputParam</u> .
Set alarm linkage parameters of HDD severe fault detection	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Event/triggers/severeHDFailure</u> by PUT method and set <u>IpInputParam</u> to <u>XML_EventTrigger</u> .

HDD Sync Status

Function	API
Get HDD data sync status	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/Storage/hdd/<ID>/syncStatus?format=json</u> by GET method.</p> <p>The progress is returned in the message <u>JSON_SyncStatus</u> by IpOutputParam.</p>

8.4 SSD Management

The SSD (solid state driver) is used for storage. Here introduces the URIs to implement the functions of SSD file system format, file system upgrade, SSD firmware upgrade, SSD S.M.A.R.T (self-monitoring, analysis, and reporting technology) detection, and so on.

SSD File System

Function	API
Format SSD file system	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/SSDFileSystem/format?format=json</u> by PUT method.</p>
Search for formatting progress of SSD file system	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/SSDFileSystem/format/status?format=json</u> by GET method.</p> <p>The progress information is returned in the message <u>JSON_SSDFormatStatus</u> by IpOutputParam.</p>
Get upgrade status of SSD file system	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/SSDFileSystem/upgrade?format=json</u> by GET method.</p> <p>The status is returned in the message <u>JSON_SSDUpgrade</u> by IpOutputParam.</p>
Upgrade SSD file system	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/SSDFileSystem/upgrade?format=json</u> by PUT method.</p>
Search for upgrade progress of SSD file system	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/SSDFileSystem/upgrade/status?format=json</u> by GET method.</p> <p>The progress information is returned in the message <u>JSON_SSDUpgradeStatus</u> by IpOutputParam.</p>
Search for SSD storage quotas	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/SSDFileSystem/capacity?format=json</u> by GET method.</p>

Function	API
	The storage quotas information is returned in the message JSON_SSDCapacity by IpOutputParam .

SSD Configuration

Function	API
Get SSD management capability	Call NET_DVR_STDXMLConfig to transmit /ISAPI/ContentMgmt/Storage/ssd/<ID>/capabilities?format=json by GET method. The capacity is returned in the message JSON_Cap_SSD by IpOutputParam .
Upgrade SSD firmware	Call NET_DVR_STDXMLConfig to transmit /ISAPI/System/SSDFileSystem/format/status?format=json by POST method.
Get SSD firmware upgrade progress	Call NET_DVR_STDXMLConfig to transmit /ISAPI/System/SSDFileSystem/upgrade?format=json by GET method.
Get a single SSD information	Call NET_DVR_STDXMLConfig to transmit /ISAPI/ContentMgmt/Storage/ssd/<ID> by GET method. The SSD information is returned in the message XML_ssdi by IpOutputParam .
Get all SSDs information	Call NET_DVR_STDXMLConfig to transmit /ISAPI/ContentMgmt/Storage/ssd by GET method. The SSD information is returned in the message XML_ssdl by IpOutputParam .

SSD S.M.A.R.T. Detection

Function	API
Run SSD S.M.A.R.T detection	Call NET_DVR_STDXMLConfig to transmit /ISAPI/ContentMgmt/Storage/ssd/<ID>/SMARTTest/start by PUT method and set IpInputParam to XML_SSDSMARTTest .
Get SSD S.M.A.R.T detection status	Call NET_DVR_STDXMLConfig to transmit /ISAPI/ContentMgmt/Storage/ssd/<ID>/SMARTTest/status by GET method. The status is returned in the message XML_SSD_SmartTestStatus by IpOutputParam .

8.5 Upload Files

You can upload files with different types, i.e., recording files, configuration files, pictures, and so on, of device to the cloud storage for some typical uses.

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

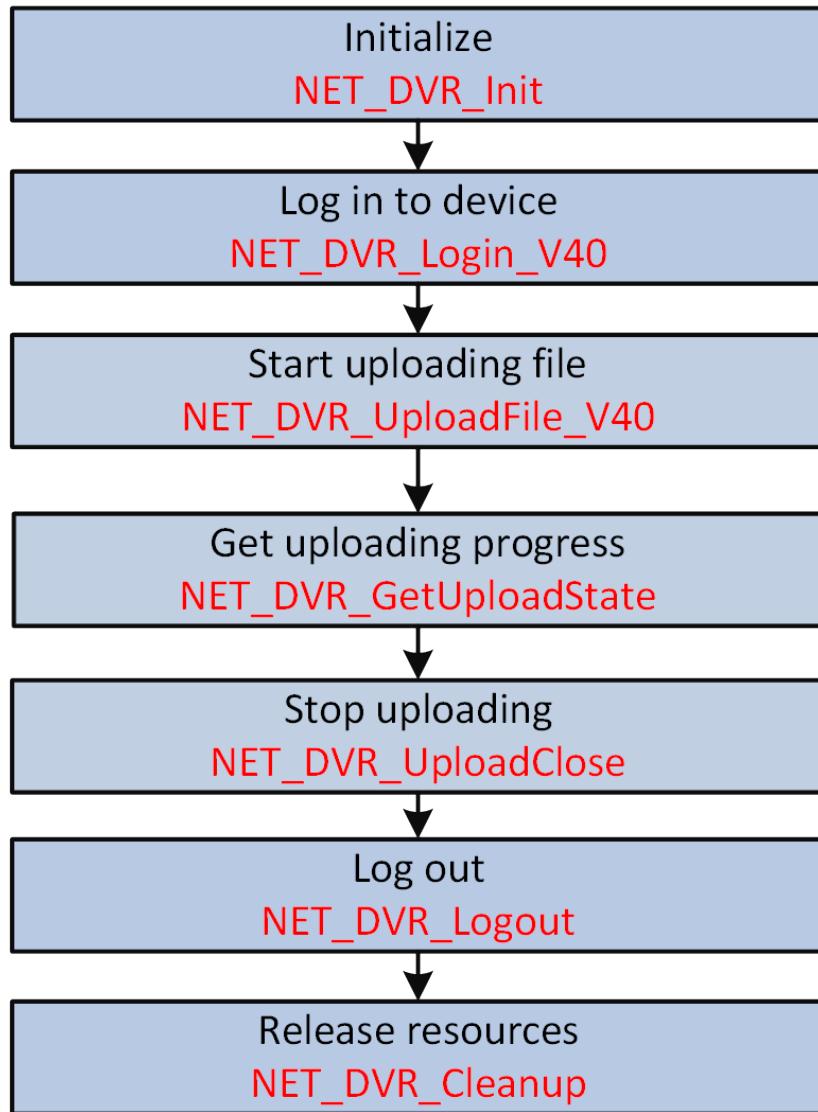


Figure 8-2 Programming Flow of Uploading Files

1. Call [NET_DVR_UploadFile_V40](#) to start uploading file.

2. Call [NET_DVR_GetUploadState](#) to get the uploading progress.



Note

If the progress is "4" (network disconnected), you should stop uploading first and perform step1 again when the network restored.

3. Call [NET_DVR_UploadClose](#) to stop uploading file.

What to do next

Call [NET_DVR_Logout](#) and [NET_DVR_Cleanup](#) to log out and release resources.

8.6 File Search

You can search video files stored in device by time, by event (or VCA event), by video tag, or perform VCA search to get the video files as required, and then remotely play back the videos. In addition, searching for pictures or pictures with smart information is also available.

You can also call [NET_DVR_STDXMLConfig](#) to transmit [/ISAPI/ContentMgmt/search](#) by POST method and set [IpInputParam](#) to [XML_VideoPic_CMSearchDescription](#).

The search results are returned in the message [XML_VideoPic_CMSearchResult](#) by [IpOutputParam](#).



Note

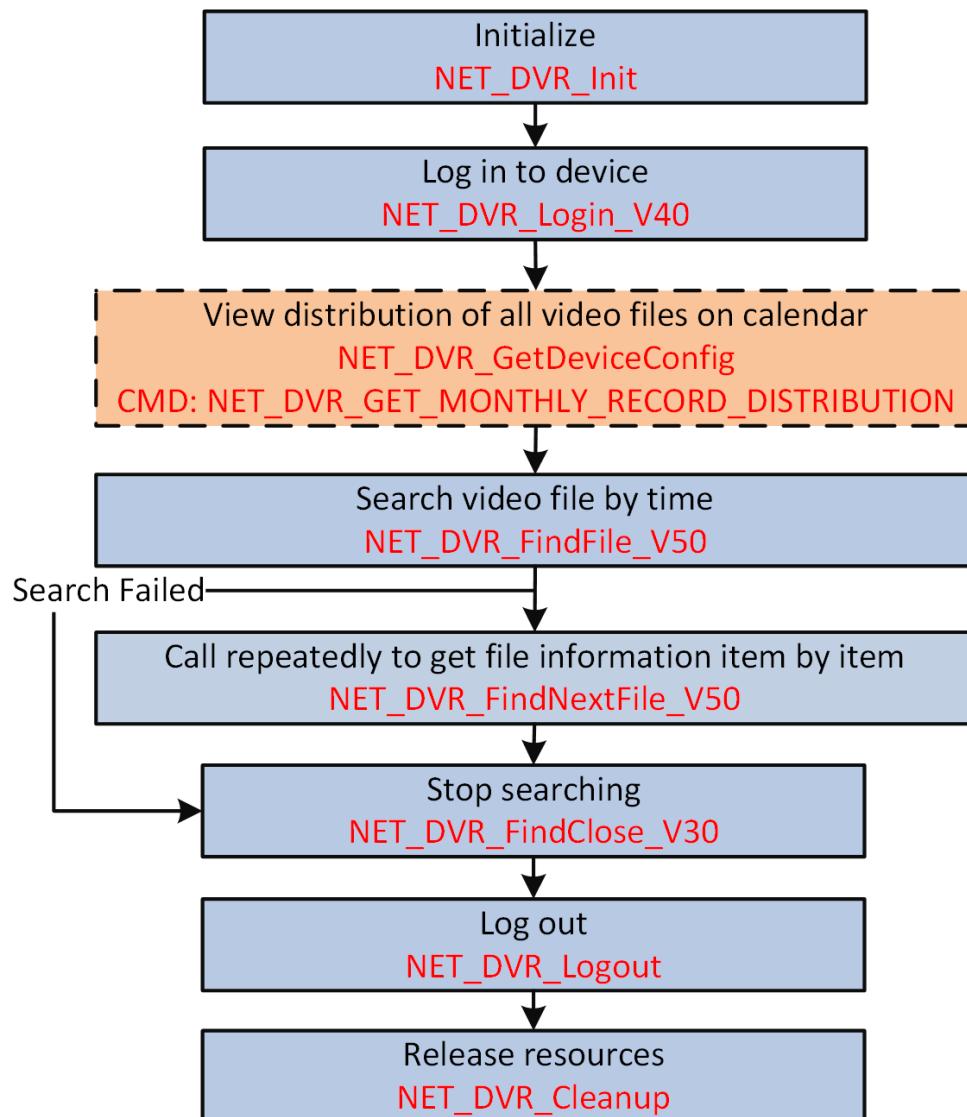
Before searching files via the above method, you can call [NET_DVR_STDXMLConfig](#) to transmit [/ISAPI/ContentMgmt/search/profile](#) by GET method to get the storage search capability ([XML_CMSearchProfile](#)) for reference.

8.6.1 Search Video Files by Time

You can set time and select the recording type configured in recording schedule to search the video files by time or by type. You can also call an API to view the distribution of all video files on the calendar to narrow the search range before searching by 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 8-3 Programming Flow of Searching Video Files by Time**

1. Optional: Call **NET_DVR_GetDeviceConfig** with
NET_DVR_GET_MONTHLY_RECORD DISTRIBUTION

(command No.: 6164) and set input parameter pointer (**lpInBuffer**) to the structure **NET_DVR_MRD_SEARCH_PARAM** for viewing the distribution of all video files on the calendar.



Viewing video distribution on calendar is not supported when login via ISAPI protocol.

The video distribution is returned in the structure **NET_DVR_MRD_SEARCH_RESULT** by output parameter pointer (**lpOutBuffer**).

2. Call **NET_DVR_FindFile_V50** to set time and recording type for search.



Note

When login via ISAPI protocol, only the continuously recorded videos can be searched.

The time periods with video are marked by blue blocks on the time bar.

3. Call **NET_DVR_FindNextFile_V50** to get the file information (such as file name, file size, start and end time, and so on) item by item.
4. Call **NET_DVR_FindClose_V30** to stop searching when searching ended or exception.

Example

Sample Code of Searching Video Files

```
#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"); //IP address
    struLoginInfo.wPort = 8000; //Service port
    strcpy(struLoginInfo.sUserName, "admin"); //User name
    strcpy(struLoginInfo.sPassword, "abcd1234"); //Password

    //Device information, output parameters
    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;
    }

    NET_DVR_FILECOND_V40 struFileCond={0};
    struFileCond.dwFileType = 0xFF;
```

```
struFileCond.lChannel = 1;
struFileCond.dwIsLocked = 0xFF;
struFileCond.dwUseCardNo = 0;
struFileCond.struStartTime.dwYear = 2011;
struFileCond.struStartTime.dwMonth = 3;
struFileCond.struStartTime.dwDay = 1;
struFileCond.struStartTime.dwHour = 10;
struFileCond.struStartTime.dwMinute = 6;
struFileCond.struStartTime.dwSecond = 50;
struFileCond.struStopTime.dwYear = 2011;
struFileCond.struStopTime.dwMonth = 3;
struFileCond.struStopTime.dwDay = 1;
struFileCond.struStopTime.dwHour = 11;
struFileCond.struStopTime.dwMinute = 7;
struFileCond.struStopTime.dwSecond = 0;

//-----
//Search video files
int lFindHandle = NET_DVR_FindFile_V40(lUserID, &struFileCond);
if(lFindHandle < 0)
{
    printf("find file fail,last error %d\n",NET_DVR_GetLastError());
    return;
}
NET_DVR_FINDDATA_V40 struFileData;
while(true)
{
    int result = NET_DVR_FindNextFile_V40(lFindHandle, &struFileData);
    if(result == NET_DVR_ISFINDING)
    {
        continue;
    }
    else if(result == NET_DVR_FILE_SUCCESS)
    {
        char strFileName[256] = {0};
        sprintf(strFileName, "./%s", struFileData.sFileName);
        saveRecordFile(lUserID, struFileData.sFileName, strFileName);
        break;
    }
    else if(result == NET_DVR_FILE_NOFOUND || result == NET_DVR_NOMOREFILE)
    {
        break;
    }
    else
    {
        printf("find file fail for illegal get file state");
        break;
    }
}

//Stop searching
if(lFindHandle >= 0)
```

```
{  
    NET_DVR_FindClose_V30(IFindHandle);  
}  
  
//Log out  
NET_DVR_Logout(IUserID);  
//Release SDK resources  
NET_DVR_Cleanup();  
return;  
}
```

What to do next

Call [NET_DVR_Logout](#) and [NET_DVR_Cleanup](#) to log out and release resources.

8.6.2 Search Video Files by Event

For device configured with event (i.e., motion detection) triggered recording schedule, you can search the video files recorded based on event to get the event details.

Before You Start

- Make sure you have configured the event triggered recording schedule for the device and set the linkage action of alarm/event to "recording".
- 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

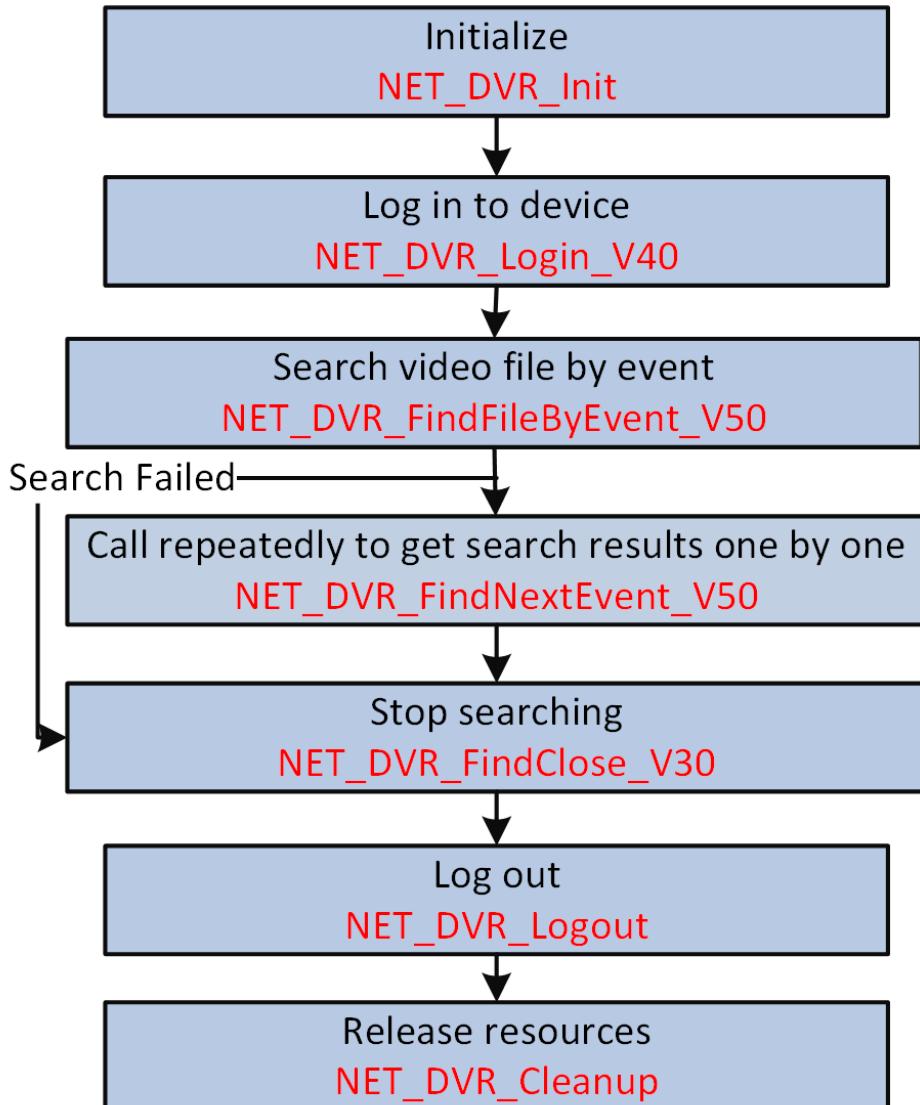


Figure 8-4 Programming Flow of Searching Video Files by Event

1. Call [NET_DVR_FindFileByEvent_V50](#) to start searching video files by event.
2. Call [NET_DVR_FindNextEvent_V50](#) repeatedly to get the information searched video files (such as event type, file size, start and end time, and so on) one by one.
3. Call [NET_DVR_FindClose_V30](#) to stop searching when searching ended or exception occurred.

What to do next

Call [NET_DVR_Logout](#) and [NET_DVR_Cleanup](#) to log out and release resources.

8.6.3 Search Video Files by VCA Event

You can set VCA rule to the video files and find the video that VCA event occurs, including motion, intrusion, and line crossing. This function helps to search out the video that you may be more concerned.

Before You Start

- Make sure you have set VCA rules and recording schedule, and set the linkage action of alarm or event to "recording".
- 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.

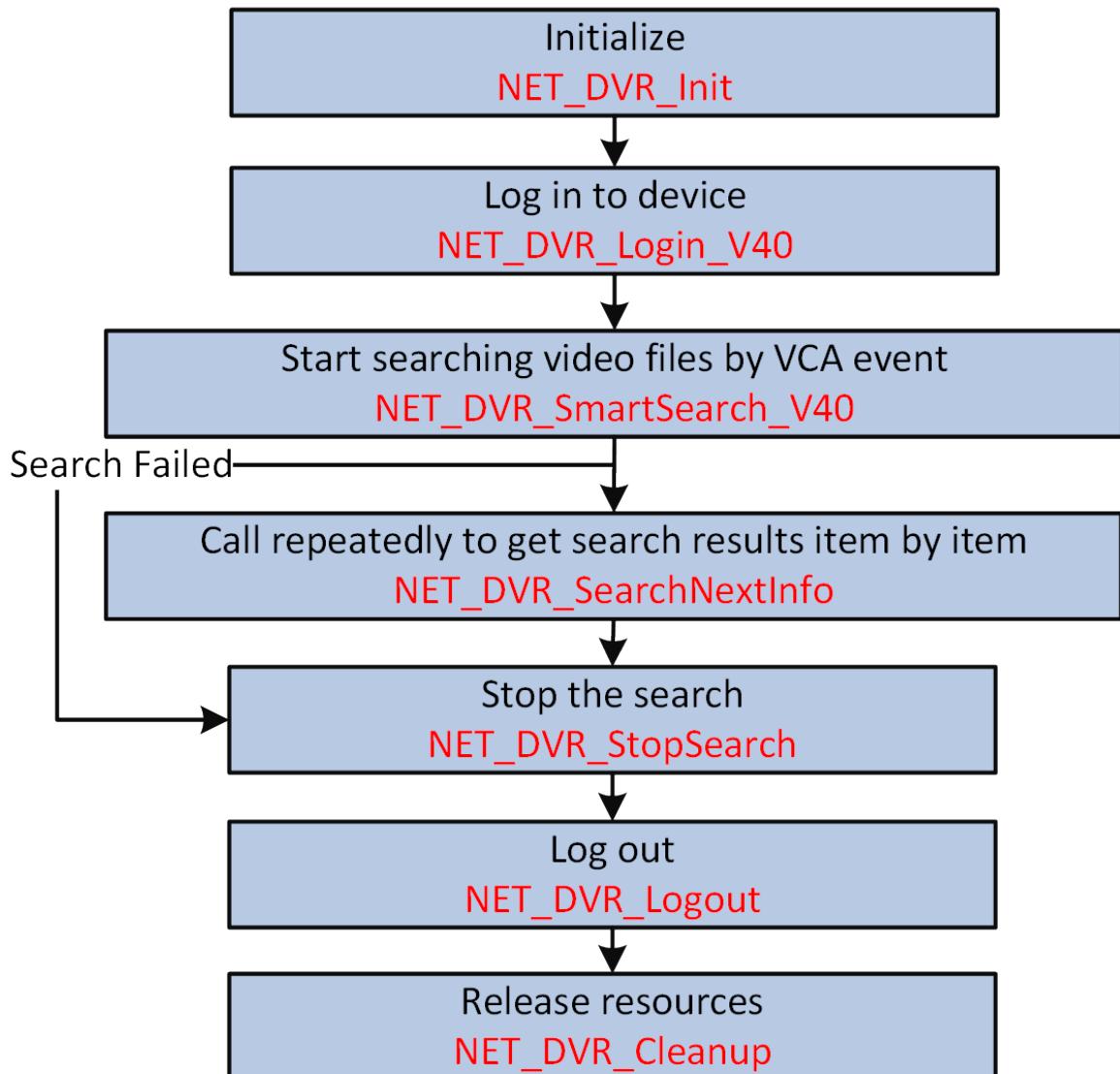
Steps

Figure 8-5 Programming Flow of Searching Video Files by VCA Event

1. Call `NET_DVR_SmartSearch_V40` to start searching video files by VCA event.
2. Call `NET_DVR_SearchNextInfo` repeatedly to get the video that VCA event occurs one by one.
3. Call `NET_DVR_StopSearch` to stop the search.

What to do next

Call `NET_DVR_Logout` and `NET_DVR_Cleanup` to log out and release resources.

8.6.4 Search Video Files by Tag

For the video files with important video point, you can add tag to mark the point for locating and fast review. So you can search the video files by the added tag name for playback.

Before You Start

- Make sure you have added tags to the video.
- 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_FindRecordLabel** to start searching video files by tag.
2. Call **NET_DVR_FindNextLabel** repeatedly to get the searched video tag for playback one by one.
3. Call **NET_DVR_StopFindLabel** to stop searching video tags.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release resources.

8.6.5 Search for Pictures

You can set different event or alarm types, specify search time period, or other conditions to search out the pictures that match the conditions.

Before You Start

- Make sure you have captured and saved the pictures in device.
- 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_FindPicture** to start searching for pictures.
2. Get searched pictures.
 - Call **NET_DVR_FindNextPicture_V50** repeatedly to get the searched pictures one by one.
 - Call **NET_DVR_GetPicture_V50** to get the searched pictures.
3. Call **NET_DVR_CloseFindPicture** to stop searching for pictures.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release resources.

8.6.6 Perform Picture Dual-VCA

You can secondary search for the pictures captured when the intelligent event occurs, such as vehicle detection, face detection, behavior analysis, and search by picture.

Before You Start

- Make sure you have set intelligent event rules and capture schedule, and set the linkage action of event to "capture".
- 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_SmartSearchPicture** to start picture dual-VCA.
2. Call **NET_DVR_FindNextSmartPicture_V50** to get the searched pictures one by one.
3. Call **NET_DVR_CloseSmartSearchPicture** to stop picture dual-VCA search.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release resources.

8.7 Working Mode Configuration

Function	API
Get configuration capability of current storage working mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/Storage/capabilities</u> by GET method. The capability is returned in the message <u>XML_Cap_storage</u> by <u>IpOutputParam</u> .
Get current working mode of storage.	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/storage</u> by PUT method. The information is returned in the message <u>XML_storage</u> by <u>IpOutputParam</u> .
Set current working mode of storage.	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/storage</u> by PUT method and set <u>IpInputParam</u> to <u>XML_storage</u> .
Start disk data reconstruction	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/Storage/dataReconstruction?format=json</u> by PUT method and set <u>IpInputParam</u> to <u>JSON_DataReconstruction</u> .
Get disk data reconstruction progress	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/Storage/dataReconstruction/progress?format=json</u> by GET method. The progress is returned in the message <u>JSON_Reconstruction_ProgressLists</u> by <u>IpOutputParam</u> .

8.8 Other File Operations

Lock or Unlock Video Files

Get video file locking and unlocking capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/record/control/locks/capabilities</u> by GET method. The capability is returned in the message <u>XML_LockCap</u> by <u>IpOutputParam</u> .
Lock or unlock video files by name	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/record/control/locks/name</u> by PUT method and set <u>IpInputParam</u> to <u>XML_LockByName</u> .

Add Remarks to Video File

Get capability of adding remarks to video files	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/record/control/remark/capabilities?format=json</u> by GET method. The capability is returned in the message <u>JSON_Cap_RemarkList</u> by <u>IpOutputParam</u> .
Add remarks to video files	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/ContentMgmt/record/control/remark?format=json</u> by PUT method and set <u>IpInputParam</u> to <u>JSON_RemarkList</u> .



Note

To check whether the device supports adding remarks to the video files, you can call [NET_DVR_STDXMLConfig](#) to transmit [/ISAPI/ContentMgmt/capabilities](#) by GET method to get the device storage capability ([XML_RacmCap](#)). If it supports, the node <isSupportRemark> will be returned and its value is "true".

Chapter 9 Parameter Configuration

This chapter provides the configurations of I/O, image, network, OSD (On Screen Display), channel, system, and so on.

9.1 Channel Settings

Basic Channel Parameters

Get all channels' attributes

Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AUXInfo/attributes/Channels**, the channels attributes are returned by output parameter pointer **lpOutBuffer** in **XML_ChannelInfoList**.

Get one channel's attributes

Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AUXInfo/attributes/Channels/<ID>**, the channel attributes are returned by output parameter pointer **lpOutBuffer** in **XML_ChannelInfo**.

Video Output Channel

Get video output channel parameters

Call **NET_DVR_GetDVRConfig** with
NET_DVR_GET_CCDPARAMCFG_EX
(command No.: 3368) and set **IChannel** to channel No. The parameters are returned in the structure **NET_DVR_CAMERAPARAMCFG_EX** by output buffer (**lpOutBuffer**).

Set video output channel parameters

Call **NET_DVR_SetDVRConfig** with
NET_DVR_SET_CCDPARAMCFG_EX
(command No.: 3369), set **IChannel** to channel No., and set input buffer (**lpInBuffer**) to the structure **NET_DVR_CAMERAPARAMCFG_EX**

Digital Channel

Get parameters of one digital channel

Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/ContentMgmt/InputProxy/channels/<ID>**. The parameters are returned by output parameter pointer **lpOutBuffer** in **XML_InputProxyChannel**.

Set parameters of one digital channel

Call **NET_DVR_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/ContentMgmt/InputProxy/channels/<ID>** and set input parameter pointer **IpInBuffer** to the message **XML_InputProxyChannel**.

Delete one digital channel

Call **NET_DVR_STDXMLConfig** to pass through the request URL: DELETE **/ISAPI/ContentMgmt/InputProxy/channels/<ID>**.

9.2 I/O Settings

The alarm input and alarm output of hybrid DVR, NVR, or CVR are allocated by device automatically after allocating the video and audio digital channel. You can configure the alarm input and output parameters, such as alarm input name, alarm output name, arming time, alarm device type, linkage actions, and so on.

Configure Alarm Input

Get Alarm Input Parameters

Call **NET_DVR_GetDVRConfig** with
NET_DVR_GET_IPALARMINCFG_V40

(command No.: 6183). And the alarm input parameters are returned by the **IpOutBuffer** in the structure .

Set Alarm Input Parameters

Call **NET_DVR_SetDVRConfig** with
NET_DVR_SET_ALARMINCFG_V40
(command No.: 6182) and set the input parameter pointer **IpInBuffer** to the structure .

Configure Alarm Output

Get Alarm Output Parameters

Call **NET_DVR_GetDVRConfig** with
NET_DVR_IPALARMOUTCFG_V40

(command No.: 6185). And the alarm output parameters are returned by the **IpOutBuffer** in the structure .

Set Alarm Output Parameters

Call **NET_DVR_SetDVRConfig** with
NET_DVR_SET_ALARMOUTCFG_V30
(command No.: 1027) and input parameter pointer **IpInBuffer** to the structure .

Configure Supplement Light Alarm Output

Get Configuration Capability of Supplement Light Alarm Output

Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Event/triggers/notifications/whiteLightAlarm/capabilities?format=json** . And the configuration capability is

returned in the message ***JSON_WhiteLightAlarmCap*** by the output parameter pointer (**IpOutputParam**).

Get Parameters of Supplement Light Alarm Output

Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: GET ***/ISAPI/Event/triggers/notifications/whiteLightAlarm?format=json***. And the configuration parameters is returned in the message ***JSON_WhiteLightAlarm*** by the output parameter pointer (**IpOutputParam**).

Set Parameters of Supplement Light Alarm Output

Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: PUT ***/ISAPI/Event/triggers/notifications/whiteLightAlarm?format=json***, and set the input parameter pointer ((**IpInputParam**) to the message ***JSON_WhiteLightAlarm***.

Configure Audible Alarm Output

Get Configuration Capability of Audible Alarm Output

Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: GET ***/ISAPI/Event/triggers/notifications/AudioAlarm/capabilities?format=json***. And the configuration capability is returned in the message ***JSON_AudioAlarmCap*** by the output parameter pointer (**IpOutputParam**).

Get Parameters of Audible Alarm Output

Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: GET ***/ISAPI/Event/triggers/notifications/AudioAlarm?format=json***. And the configuration parameters is returned in the message ***JSON_AudioAlarm*** by the output parameter pointer (**IpOutputParam**).

Set Parameters of Audible Alarm Output

Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: PUT ***/ISAPI/Event/triggers/notifications/AudioAlarm?format=json***, and set the input parameter pointer ((**IpInputParam**) to the message ***JSON_AudioAlarm***.

9.3 Video/Image Settings

Camera Video/Image Settings

Basic Parameters	Get camera image parameters	Call <i>NET_DVR_GetDVRConfig</i> with "NET_DVR_GET_CCDPARAMCFG" (command No.: 1067) and set IChannel to "0xFFFFFFFF". The camera image parameters are returned in the structure <i>NET_DVR_CAMERAPARAMCFG</i> by IpOutBuffer .
	Set camera image parameters	Call <i>NET_DVR_SetDVRConfig</i> with "NET_DVR_SET_CCDPARAMCFG" (command No: 1068), set IChannel to

		"0xFFFFFFFF", and set lplnBuffer to the structure <i>NET_DVR_CAMERAPARAMCFG</i> .
Correct defective pixel		Call <i>NET_DVR_RemoteControl</i> with "NET_DVR_DPC_CTRL" (command No.: 3410) and set lplnBuffer to the structure <i>NET_DVR_DPC_PARAM</i> .  Note This function is only available for certain camera models.
ISP Parameters	Get ISP (Image Signal Processing) parameters	Call <i>NET_DVR_GetDVRConfig</i> with "NET_DVR_GET_ISP_CAMERAPARAMCFG" (command No.: 3255). The ISP parameters are returned in the structure <i>NET_DVR_ISP_CAMERAPARAMCFG</i> by lpOutBuffer .
	Set ISP (Image Signal Processing) parameters	Call <i>NET_DVR_SetDVRConfig</i> with "NET_DVR_SET_ISP_CAMERAPARAMCFG" (command No.: 3256) and set lplnBuffer to the structure <i>NET_DVR_ISP_CAMERAPARAMCFG</i> .
Shielded Area Parameters	Get shielded area parameters of all channels	Call <i>NET_DVR_GetDVRConfig</i> with "NET_DVR_GET_VCA_MASK_REGION" (command No.: 167). The parameters are returned by lpOutBuffer in the structure <i>NET_VCA_MASK_REGION_LIST</i> .
	Get shielded area parameters of a channel by event	Call <i>NET_DVR_STDXMLConfig</i> to transmit /ISAPI/Intelligent/channels/<ID>/Shield/EventType by GET method. The parameters are returned by lpOutBuffer in the message <i>XML_Shield</i> .
	Set shielded area parameter of a channel by event	Call <i>NET_DVR_STDXMLConfig</i> to transmit /ISAPI/Intelligent/channels/<ID>/Shield/EventType by PUT method and set lplnBuffer to the message <i>XML_Shield</i> .

On Screen Display (OSD) Settings

Get OSD capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit /ISAPI/System/Video/capabilities by GET method.
--------------------	--

		The capability is returned in the message <u>XML_VideoCap</u> by IpOutputParam .
OSD Language	Get text language parameters of OSD	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/Video/inputs/OSDLanguage</u> by GET method. The OSD language parameters are returned in the message <u>XML_Language</u> by IpOutputParam .
	Set text language parameters of OSD	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/Video/inputs/OSDLanguage</u> by PUT method and set IpInputParam to the message <u>XML_Language</u> .
Intelligent Information Display	Display intelligent information on live video	Call <u>NET_DVR_RenderPrivateData</u> and set information type (iIntelType) as needed.
	Display intelligent thermal information on live video	Call <u>PlayM4_RenderPrivateDataEx</u> and set iIntelType to "RENDER_FIRE_DETCT" or "RENDER_TEM".
Smart Event Rule Display	Get capability of displaying rule information of smart events on video	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Streaming/channels/<ID>/smartOverlap/capabilities?format=json</u> by GET method. The capability is returned in the message <u>JSON_SmartOverlapCap</u> by IpOutputParam .  Note To check whether the function is supported by the device, you can call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/capabilities</u> by GET method to get device capability <u>XML_DeviceCap</u> . If it supports, the node <supportSmartOverlapChannles> will be returned and its value is "true".
	Get parameters of displaying rule information of smart events on video	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Streaming/channels/<ID>/smartOverlap?format=json</u> by GET method. The parameters are returned in the message <u>JSON_SmartOverlap</u> by the output parameter IpOutputParam .
	Set parameters of displaying rule information of smart events on video	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Streaming/channels/<ID>/smartOverlap?format=json</u> by PUT method and set IpInputParam to the message <u>JSON_SmartOverlap</u> .

Temperatur e Range Display	Get configuration capability of displayed temperature range	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/tempRange/capabilities</u> by GET method. The capability is returned in the message <u>XML_Cap_tempRange</u> by IpOutputParam .
	Get displayed temperature range	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/tempRange</u> by GET. The parameters are returned in the message <u>XML_tempRange</u> by IpOutputParam .
	Set displayed temperature range	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/tempRange</u> by PUT method and set IplInputParam to the message <u>XML_tempRange</u> .

Other Settings

Focus Settings	Get focus configuration capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/focusConfiguration/capabilities</u> by GET method. The capability is returned in the message <u>XML_Cap_FocusConfiguration</u> by IpOutputParam .
	Get focus parameters	Call <u>NET_DVR_GetDVRConfig</u> with "NET_DVR_GET_FOCUSMODECFG" (command No.: 3305). The focus parameters is returned by IpOutBuffer in the structure <u>NET_DVR_FOCUSMODE_CFG</u> .
	Set focus parameters	Call <u>NET_DVR_SetDVRConfig</u> with "NET_DVR_SET_FOCUSMODECFG" (command No.: 3306) and set IplInBuffer to the structure <u>NET_DVR_FOCUSMODE_CFG</u> .
	Get focus status	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/focusConfiguration/status?format=json</u> by GET method. The focus status is returned in the message <u>JSON_FocusStatus</u> by IpOutputParam .

Target Enhancement Settings	Get target enhancement capability	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/targetEnhancement/capabilities</u> by GET method.</p> <p>The capability is returned in the message <u>XML_TargetEnhancementCap</u> by IpOutputParam.</p>
	Get target enhancement parameters	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/targetEnhancement</u> by GET method.</p> <p>The target enhancement parameters are returned in the message <u>XML_TargetEnhancement</u> by IpOutputParam.</p>
	Set target enhancement parameters	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/targetEnhancement</u> by PUT method and set IpInputParam to the message <u>XML_TargetEnhancement</u>.</p>
Privacy Mask Region Settings	Get a privacy mask region's information	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/Video/inputs/channels/<ID>/privacyMask/regions/<ID></u> by GET method.</p> <p>The parameters of the privacy mask region are returned in the message <u>XML_PrivacyMaskRegion</u> by IpOutputParam.</p>
	Set a privacy mask region's information	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/System/Video/inputs/channels/<ID>/privacyMask/regions/<ID></u> by PUT method and set IpInputParam to the message <u>XML_PrivacyMaskRegion</u>.</p>
Distortion Correction Settings	Get distortion correction capability	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/lensDistortionCorrection/capabilities</u> by GET method.</p> <p>The capability is returned in the message <u>XML_Cap_LensDistortionCorrection</u> by IpOutputParam.</p>

	Get distortion correction parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/lensDistortionCorrection</u> by GET method. The parameters are returned in the message <u>XML_LensDistortionCorrection</u> by lpOutputParam .
	Set distortion correction parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/lensDistortionCorrection</u> by PUT method and set lpInputParam to the message <u>XML_LensDistortionCorrection</u> .
Image Adjustment Settings	Get image adjustment capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/color/capabilities</u> by GET method. The capability is returned in the message <u>XML_Cap_Color</u> by lpOutputParam .
	Get image adjustment parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/color</u> by GET method. The parameters are returned in the message <u>XML_Color</u> by lpOutputParam .
	Set image adjustment parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/color</u> by PUT method and set lpInputParam to the message <u>XML_Color</u> .
Mounting Scenario	Get configuration capability of mounting scenario mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/mountingScenario/capabilities</u> by GET method. The capability is returned in the message <u>XML_Cap_MountingScenario</u> by lpOutputParam .
	Get mounting scenario mode by channel	Call <u>NET_DVR_STDXMLConfig</u> to transmit <u>/ISAPI/Image/channels/<ID>/mountingScenario</u> by GET method.

	The parameters are returned in the message <i>XML_MountingScenario</i> by IpOutputParam .
Set mounting scenario mode by channel	Call <i>NET_DVR_STDXMLConfig</i> to transmit /ISAPI/Image/channels/<ID>/mountingScenario by PUT method and set IpInputParam to the message <i>XML_MountingScenario</i> .

9.4 Network Settings

Get Network Capability Set

Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: GET **/ISAPI/System/Network/capabilities**. And the capability is returned in the message ***XML_NetworkCap*** by the output parameter pointer (**IpOutputParam**).

Get Network Application Capability

Call ***NET_DVR_GetDeviceAbility***, set the capability type (**dwAbilityType**) to "DEVICE_NETAPP_ABILITY" (macro definition value: 0x00d), and set the input buffer pointer (**pInBuf**) to the message ***XML_Desc_NetAppAbility***. The network application capability is returned in the message ***XML_NetAppAbility*** by the output buffer pointer (**pOutBuf**).

Get Network Parameters

Call ***NET_DVR_SetDVRConfig*** with ***NET_DVR_GET_NETCFG_V50*** (command No.: 1015), the network parameters are returned in the structure ***NET_DVR_NETCFG_V50***.

Set Network Parameters

Call ***NET_DVR_SetDVRConfig*** with ***NET_DVR_SET_NETCFG_V50*** (command No.: 1016), and set the input parameter pointer **IpInBuffer** to the structure ***NET_DVR_NETCFG_V50***.

Get Parameters of Alarm Receiving Center

Call ***NET_DVR_SetDVRConfig*** with ***NET_DVR_GET_ALARMHOST_NETCFG_V50*** (command No.: 2224), and the parameters are returned in the structure ***NET_DVR_ALARMHOST_NETCFG_V50***.

Set Parameters of Alarm Receiving Center

Call ***NET_DVR_SetDVRConfig*** with

NET_DVR_SET_ALARMHOST_NETCFG_V50

(command No.: 2225) and set the input parameter pointer **IpInBuffer** to the structure **NET_DVR_ALARMHOST_NETCFG_V50**.

Import Network Certificate

Call **NET_DVR_UploadFile_V40**, set the file type (**dwUploadType**) to "UPLOAD_CERT_FILE", and set the input parameter (**IpInBuffer**) to the structure **NET_DVR_CERT_PARAM**.

Remotely Get Connection Socket IP

Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET [/ISAPI/System/Network/socketIP](#). And the socket IP information is returned in the message **XML_SocketIP** by the output parameter pointer (**IpOutputParam**).



To check if getting socket IP is supported by device, you can call **NET_DVR_GetSTDAbility** and set the **dwAbilityType** to "NET_DVR_GET_SYSTEM_CAPABILITIES" to get the device capability (**XML_DeviceCap**). If supports, the node <isSupportGetLinkSocketIP> will be returned.

9.4.1 Address Filter

IP Address Configuration

Get IP Address Filter Configuration Capability

Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET [/ISAPI/System/Network/capabilities](#). And the capability is returned in the message **XML_NetworkCap** by the output parameter pointer (**IpOutputParam**).

Get IP Address Filter Configuration Parameters

Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET [/ISAPI/System/Network/ipFilter](#). And the parameters are returned in the message **XML_IPFilter** by the output parameter pointer (**IpOutputParam**).

Set IP Address Filter

Call **NET_DVR_STDXMLConfig** to pass through the request URL: PUT [/ISAPI/System/Network/ipFilter](#), and set the input parameter (**IpInBuffer**) to the message **XML_IPFilter**.

Mac Address Configuration

Get MAC Address Filter Configuration Capability

Call **NET_DVR_GetSTDAbility**, set the **dwAbilityType** to NET_DVR_GET_MACFILTER_CAPABILITIES (value: 3643), and set condition parameter **IpCondBuffer** in the structure of to null. Then the capability is returned by the output parameter **IpOutBuffer** in the message **XML_Cap_MACFilter**.



Note

Check whether the device supports MAC address filter configuration via node `<isSupportMACFilter>` in the network capability set [XML_NetworkCap](#).

Get MAC Address Filter Configuration Parameters

Call [NET_DVR_GetSTDConfig](#) with the command of

`NET_DVR_GET_MACFILTER_CFG`

(command No.: 3644) and set the condition parameter **IpCondBuffer** in the structure of to null. Then the parameters are returned by the output parameter **IpOutBuffer** in the structure .

Set MAC Address Filter

Call [NET_DVR_SetSTDConfig](#) with the command of

`NET_DVR_SET_MACFILTER_CFG`

(command No.: 3645), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of to null and for setting MAC address filter.

9.4.2 Network Access

Access Protocol

Function	Description
Get Configuration Capability of Access Protocol	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/System/Network/Integrate/capabilities</u> . The configuration capability is returned in the message <u>XML_Cap_Integrate</u> by IpOutputParam .
Get Access Protocol Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/System/Network/Integrate</u> . The parameters are returned in the message <u>XML_Integrate</u> by IpOutputParam .
Set Access Protocol Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/System/Network/Integrate</u> and set IpInputParam to <u>XML_Integrate</u> .

Guarding Vision Access

Function	Description
Get the Guarding Vision access configuration capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/System/Network/EZVIZ/capabilities</u> .

Function	Description
	The capability is returned in the message <u>XML_Cap_EZVIZ</u> by IpOutputParam
Get Guarding Vision Access Parameters	Call <u>NET_DVR_GetDVRConfig</u> with "NET_DVR_GET_EZVIZ_ACCESS_CFG" (command No.: 3398). The Guarding Vision access parameters are returned in the structure <u>NET_DVR_EZVIZ_ACCESS_CFG</u> by IpOutBuffer .
Set Guarding Vision Access Parameters	Call <u>NET_DVR_SetDVRConfig</u> with "NET_DVR_SET_EZVIZ_ACCESS_CFG" (command No.: 3399), and set IpInBuffer to the structure <u>NET_DVR_EZVIZ_ACCESS_CFG</u> .
Edit Verification Code for Guarding Vision	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/System/Network/EZVIZ/secretKey?format=json</u> , and set IpInBuffer to the message <u>JSON_EZVIZSecretKey</u> for editing the verification code for Guarding Vision.  Note You can check whether device supports this function via the node <EZVIZSecretKey> in <u>XML_AccessProtocolAbility</u> .
Get Guarding Vision QR code	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/System/Network/EZVIZ/QRCode?format=json</u> or <u>/ISAPI/System/Network/EZVIZ/QRCode</u> . The information is returned in the message <u>JSON_EZVIZQRCode</u> by IpOutputParam or in binary format.



To check if the device supports accessing to Guarding Vision, you can get the network application capability ([XML_NetAppAbility](#)).

ISUP Server Access

Function	Description
Get ISUP Server Access Parameters	Call <u>NET_DVR_GetDVRConfig</u> with "NET_DVR_GET_CMS_CFG" (command No.: 2070).

Function	Description
	The ISUP server access parameters are returned in the structure <i>NET_DVR_CMS_PARAM</i> by IpOutBuffer .
Set ISUP Server Access Parameters	Call <i>NET_DVR_SetDVRConfig</i> with "NET_DVR_SET_CMS_CFG" (command No.: 2071) and set IpInBuffer to the structure <i>NET_DVR_CMS_PARAM</i> .



To check if the device supports accessing to ISUP server, you can get the network application capability (***XML_NetAppAbility***).

RTSP (Real-Time Streaming Protocol) Access

Function	Description
Get RTSP Parameters	Call <i>NET_DVR_GetRtspConfig</i> to get the RTSP parameters. The parameters are returned in the structure <i>NET_DVR_RTSPCFG</i> by the output buffer IpOutBuffer .
Set RTSP Parameters	Call <i>NET_DVR_SetRtspConfig</i> and set the input buffer IpInBuffer to the structure <i>NET_DVR_RTSPCFG</i> .

SNMP Access

Function	Description
Get batch configuration parameters of SNMP	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/System/Network/SNMP</i> . The parameter is returned in the message <i>XML_SNMP</i> by the output parameter IpOutputParam .
Get advanced configuration parameters of SNMP	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/System/Network/SNMP/advanced</i> . The parameter is returned in the message <i>XML_SNMPAdvanced</i> by the output parameter IpOutputParam .
Get all trap address parameters of SNMP in version 3	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/System/Network/SNMP/advanced/trapReceivers</i> .

Function	Description
	The parameters are returned in the message XML_SNMPTrapReceiverList by the output parameter lpOutputParam .
Set all trap address parameters of SNMP in version 3	Call NET_DVR_STDXMLConfig to transmit the request URI: PUT /ISAPI/System/Network/SNMP/advanced/trapReceivers , and set lpInBuffer to the message XML_SNMPTrapReceiverList .
Add a new trap address of SNMP in version 3	Call NET_DVR_STDXMLConfig to transmit the request URI: POST /ISAPI/System/Network/SNMP/advanced/trapReceivers , and set lpInBuffer to the message XML_SNMPTrapReceiver .
Delete all trap addresses of SNMP in version 3	Call NET_DVR_STDXMLConfig to transmit the request URI: DELETE /ISAPI/System/Network/SNMP/advanced/trapReceivers , and set lpInBuffer to NULL.
Get a single trap address's parameters of SNMP in version 3	Call NET_DVR_STDXMLConfig to transmit the request URI: GET /ISAPI/System/Network/SNMP/advanced/trapReceivers/<ID> . The parameter is returned in the message XML_SNMPTrapReceiver by the output parameter lpOutputParam .
Set parameters for a single trap address of SNMP in version 3	Call NET_DVR_STDXMLConfig to transmit the request URI: PUT /ISAPI/System/Network/SNMP/advanced/trapReceivers/<ID> , and set lpInBuffer to the message XML_SNMPTrapReceiver .
Delete a single address of SNMP in version 3	Call NET_DVR_STDXMLConfig to transmit the request URI: DELETE /ISAPI/System/Network/SNMP/advanced/trapReceivers/<ID> , and set lpInBuffer to NULL.

9.4.3 Network Interface

Table 9-1 PoE Port Configuration

Function	Description
Get PoE Parameters	Call NET_DVR_GetDVRConfig with NET_DVR_GET_POE_CFG

Function	Description
	(command No.: 6114). And the PoE parameters are returned by the IpOutBuffer in the structure <i>NET_DVR_POE_CFG</i> .
Set PoE Parameters	Call <i>NET_DVR_SetDVRConfig</i> with <i>NET_DVR_SET_POE_CFG</i> (command No.: 6115), and set the input parameter pointer IpInBuffer to the structure <i>NET_DVR_POE_CFG</i> .

Table 9-2 UPnP Port Configuration

Function	Description
Get UPnP Parameters	Call <i>NET_DVR_GetDVRConfig</i> with <i>NET_DVR_GET_NAT_CFG</i> (command No.: 6111). And the UPnP parameters are returned by the output buffer pointer (IpOutBuffer) in the structure .
Set UPnP Parameters	Call <i>NET_DVR_SetDVRConfig</i> with <i>NET_DVR_SET_NAT_CFG</i> (command No.: 6112) and set the input buffer pointer (IpInBuffer) to the structure .
Get UPnP Port Mapping Status	Call <i>NET_DVR_GetUpnpNatState</i> . And the mapping status is returned by the parameter IpState in the structure .

9.4.4 Network Service

Log Server

Function	Description
Get Log Server Configuration Capability	Call <i>NET_DVR_STDXMLConfig</i> to pass through the request URL: GET <i>/ISAPI/System/logServer/capabilities</i> . The capability is returned in the message <i>XML_LogServerCap</i> by the output parameter IpOutputParam .
Get Log Server Parameters	Call <i>NET_DVR_STDXMLConfig</i> to pass through the request URL: GET <i>/ISAPI/System/logServer</i> .

Function	Description
	The parameters are returned in the message <i>XML_LogServer</i> by the output parameter IpOutputParam .
Set Log Server Parameters	Call <i>NET_DVR_STDXMLConfig</i> to pass through the request URL: PUT /ISAPI/System/logServer and set the input parameter IpInputParam to the message <i>XML_LogServer</i> .

File Transfer Protocol (FTP) Server

Function	Description
Get FTP Server Configuration Capability	Call <i>NET_DVR_STDXMLConfig</i> to pass through the request URL: GET /ISAPI/System/Network/ftp/capabilities . The capability is returned in the message <i>XML_Cap_FTPNotificationList</i> by the output parameter IpOutputParam .
Get Parameters of A Specific FTP Server	Call <i>NET_DVR_STDXMLConfig</i> to pass through the request URL: GET /ISAPI/System/Network/ftp/<ID> . The parameters are returned in the message <i>XML_FTPNotification</i> by the output parameter IpOutputParam .
Set Parameters of A Specific FTP Server	Call <i>NET_DVR_STDXMLConfig</i> to pass through the request URL: PUT /ISAPI/System/Network/ftp/<ID> and set the input parameter IpInputParam to the message <i>XML_FTPNotification</i> .

SIP (Session Initiation Protocol) Service

Function	Description
Get SIP Parameters	Call <i>NET_DVR_GetDVRConfig</i> with "NET_DVR_GET_SIP_CFG_V50" (command No.: 16044) and set IChannel to 0xFFFFFFFF. The parameters are returned in the structure <i>NET_DVR_SIP_CFG_V50</i> by IpOutBuffer .
Set SIP Parameters	Call <i>NET_DVR_SetDVRConfig</i> with "NET_DVR_SET_SIP_CFG_V50" (command No.: 16045), set IChannel to 0xFFFFFFFF, and set IpInBuffer to the structure <i>NET_DVR_SIP_CFG_V50</i> .

9.4.5 Wireless Network

Wireless Hotspot

Get Wireless Hotspot Capability	Call <u>NET_DVR_GetSTDAbility</u> with "NET_DVR_GET_WIRELESSERVER_CAPABILITIES" (command No.: 3716) and set IpCondBuffer to a 4-byte wireless NIC No. The capability is returned in the message <u>XML_Cap_WirelessServer</u> by IpOutBuffer .
Get Wireless Hotspot Parameters	Call <u>NET_DVR_GetSTDConfig</u> with "NET_DVR_GET_WIRELESSERVER" (command No.: 3717) and set IpCondBuffer to a 4-byte wireless NIC No. The parameters are returned in the structure <u>NET_DVR_WIRELESSERVER</u> by IpOutBuffer .
Set Wireless Hotspot Parameters	Call <u>NET_DVR_SetSTDConfig</u> with "NET_DVR_SET_WIRELESSERVER" (command No.: 3718), set IpCondBuffer to a 4-byte wireless NIC No., and specify IpInBuffer as the structure <u>NET_DVR_WIRELESSERVER</u> .

Wireless Dial-up

Get Wireless Dial-up Configuration Capability	<ul style="list-style-type: none"> Option 1: Call <u>NET_DVR_GetSTDAbility</u> with "NET_DVR_GET_WIRELESSDIAL_CAPABILITIES" (command No.: 3580) and set IpCondBuffer to a 4-byte wireless NIC No. Option 2: Call <u>NET_DVR_STDXMLConfig</u> to transmit GET / <u>ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial/capabilities</u>. <p>The capability is returned in the message <u>XML_Cap_Dial</u> by IpOutBuffer or IpOutputParam.</p>
Get Wireless Dial-up Parameters	<ul style="list-style-type: none"> Option 1: Call <u>NET_DVR_GetSTDConfig</u> with "NET_DVR_GET_WIRELESS_DIAL" (command No.: 3578) and set IpCondBuffer to a 4-byte wireless NIC No. The parameters are returned in the structure <u>NET_DVR_WIRELESSDIAL_CFG</u> by IpOutBuffer. Option 2: Call <u>NET_DVR_STDXMLConfig</u> to transmit GET / <u>ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial</u>. The parameters are returned in the message <u>XML_Dial</u> by IpOutputParam.
Set Wireless Dial-up Parameters	• Option 1: Call <u>NET_DVR_SetSTDConfig</u> with "NET_DVR_SET_WIRELESS_DIAL" (command No.: 3579), set IpCondBuffer to a 4-byte wireless NIC No., and set the input parameter

	<p>IpInputParam to the structure <i>NET_DVR_WIRELESSDIAL_CFG</i>.</p> <ul style="list-style-type: none"> Option 2: Call <i>NET_DVR_STDXMLConfig</i> to transmit PUT / <i>ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial</i> and set IpInputParam to the message <i>XML_Dial</i>.
Get Wireless Dial-up Schedule Capability	<p>Call <i>NET_DVR_GetSTDAbility</i> with "NET_DVR_GET_WIRELESSDIAL_SCHEDULE_CAPABILITIES" (command No.: 3592) and set IpCondBuffer to a 4-byte wireless NIC No.</p> <p>The capability is returned in the message <i>XML_Cap_Schedule</i> by IpOutBuffer.</p>
Get Wireless Dial-up Schedule Parameters	<ul style="list-style-type: none"> Option 1: Call <i>NET_DVR_GetSTDConfig</i> with "NET_DVR_GET_WIRELESSDIAL_SCHEDULE" (command No.: 3581) and set IpCondBuffer to a 4-byte wireless NIC No. <p>The parameters are returned in the structure <i>NET_DVR_WIRELESSDIAL_SCHEDULE</i> by IpOutBuffer.</p> <ul style="list-style-type: none"> Option 2: Call <i>NET_DVR_STDXMLConfig</i> to transmit GET / <i>ISAPI/System/Network/WirelessDial/Interfaces/<ID>/schedule</i>. <p>The parameters are returned in the message <i>XML_Schedule</i> by IpOutputParam.</p>
Set Wireless Dial-up Schedule Parameters	<ul style="list-style-type: none"> Call <i>NET_DVR_SetSTDConfig</i> with "NET_DVR_SET_WIRELESSDIAL_SCHEDULE" (command No.: 3582), set IpCondBuffer to a 4-byte wireless NIC No., and set IpInputParam to the structure <i>NET_DVR_WIRELESSDIAL_SCHEDULE</i>. Option 2: Call <i>NET_DVR_STDXMLConfig</i> to transmit PUT / <i>ISAPI/System/Network/WirelessDial/Interfaces/<ID>/schedule</i> and set IpInputParam to the message <i>XML_Schedule</i>.
Get Wireless Dial-up Status Parameters	<p>Call <i>NET_DVR_GetSTDConfig</i> with "NET_DVR_GET_WIRELESSDIAL_STATUS" (command No.: 3583) and set IpCondBuffer to a 4-byte wireless NIC No.</p> <p>The parameters are returned in the structure <i>NET_DVR_WIRELESSDIAL_STATUS</i> by IpOutBuffer.</p>

Network Keepalive

Get Network Keepalive Capability	<p>Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URL: GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive/capabilities?format=json.</p> <p>The capability is returned in the message <i>JSON_KeepAliveCap</i> by the output parameter IpOutputParam.</p>
Get Network Keepalive Parameters	<p>Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URL: GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive?format=json.</p> <p>The parameters are returned in the message <i>JSON_KeepAlive</i> by the output parameter IpOutputParam.</p>
Set Network Keepalive Parameters	<p>Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URL: PUT /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive?format=json and set the input parameter IpInputParam to the message <i>JSON_KeepAlive</i>.</p>
Test Network Keepalive	<p>Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URL: POST /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive/test?format=json and set the input parameter IpInputParam to the message <i>JSON_KeepAlive</i>.</p>

Traffic Monitoring

Get Traffic Monitoring Capability	<p>Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URL: GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor/capabilities?format=json.</p> <p>The capability is returned in the message <i>JSON_TrafficMonitorCap</i> by the output parameter IpOutputParam.</p>
Get Network Keepalive Parameters	<p>Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URL: GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor?format=json.</p> <p>The parameters are returned in the message <i>JSON_TrafficMonitor</i> by the output parameter IpOutputParam.</p>
Set Network Keepalive Parameters	<p>Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URL: PUT /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor?format=json.</p>

	<code>trafficMonitor?format=json</code> and set the input parameter <code>IpInputParam</code> to the message <code>JSON_TrafficMonitor</code> .
Test Network Keepalive	<p>Call <code>NET_DVR_STDXMLConfig</code> to transmit the request URL: <code>GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor/status?format=json</code>.</p> <p>The parameters are returned in the message <code>JSON_TrafficMonitorStatus</code> by the output parameter <code>IpOutputParam</code>.</p>

9.4.6 Multicast

The multicast is to realize the group communication where information is addressed to a group of destination computers simultaneously.

Get Parameters of Multicast Based on NPQ

Call `NET_DVR_STDXMLConfig` to pass through the request URL: `GET /ISAPI/System/Network/extension`.

The parameters are returned in the message `XML_networkExtension` by the output parameter (`IpOutputParam`).

Set Parameters of Multicast Based on NPQ

Call `NET_DVR_STDXMLConfig` to pass through the request URL: `PUT /ISAPI/System/Network/extension` and set the input parameter (`IpInputParam`) to the message `XML_networkExtension`.

9.5 Peripherals Settings

Peripherals Configuration Capability

Peripherals Configuration Capability	<p>Call <code>NET_DVR_GetSTDAbility</code>, set <code>dwAbilityType</code> to "NET_DVR_GET_EXTERNALDEVICE_CAPABILITIES" (value: 3722) and set <code>IpCondBuffer</code> of structure <code>NET_DVR_STD_ABILITY</code> to a 4-byte channel No. for getting peripherals configuration capability.</p> <p>The capability is returned in the message <code>XML_Cap_ExternalDevice</code></p>
--------------------------------------	--

Built-in Supplement Light

Configuration Capability	<p>Call <u>NET_DVR_GetSTDAbility</u>, set dwAbilityType to "NET_DVR_GET_SUPPLEMENTLIGHT_CAPABILITIES" (value: 3727) and set IpCondBuffer of structure <u>NET_DVR_STD_ABILITY</u> to a 4-byte channel No. for getting configuration capability of built-in supplement light.</p> <p>The configuration capability is returned in the message <u>XML_Cap_SupplementLight</u> by IpOutBuffer of structure <u>NET_DVR_STD_ABILITY</u>.</p>
Configuration Parameters	<p>Call <u>NET_DVR_GetSTDConfig</u> with "NET_DVR_GET_SUPPLEMENTLIGHT" (command No.: 3728) and set IpCondBuffer of structure <u>NET_DVR_STD_CONFIG</u> to a 4-byte supplement No. for getting parameters of built-in supplement light.</p> <p>The configuration parameters are returned in the structure <u>NET_DVR_BUILTIN_SUPPLEMENTLIGHT</u> by IpOutBuffer of structure <u>NET_DVR_STD_CONFIG</u>.</p> <p>Call <u>NET_DVR_SetSTDConfig</u> with "NET_DVR_SET_SUPPLEMENTLIGHT" (command No.: 3729), set IpCondBuffer and IpInBuffer of structure <u>NET_DVR_STD_CONFIG</u> to a 4-byte supplement No. and <u>NET_DVR_BUILTIN_SUPPLEMENTLIGHT</u>, respectively, for setting parameters of built-in supplement light.</p>

POS (Point of Sale)

POS Capability	<p>Call <u>NET_DVR_GetDeviceAbility</u>, set dwAbilityType to "DEVICE_ABILITY_INFO" (value: 0x011), and set pInBuf to the message <u>XML_Desc_POSAbility</u> for getting POS capability.</p> <p>The POS capability is returned in the message <u>XML_POSAbility</u> by pOutBuf.</p>
Filter Rule of POS Information	<p>Call <u>NET_DVR_GetDVRCfg</u> with "NET_DVR_GET_POS_FILTER_CFG" (command No.: 6148) and set IChannel to the rule ID (which starts from 1) for getting filter rule of POS information.</p> <p>The filter rule parameters are returned in the structure <u>NET_DVR_POS_FILTER_CFG</u> by IpOutBuffer.</p> <p>Call <u>NET_DVR_SetDVRCfg</u> with "NET_DVR_SET_POS_FILTER_CFG" (command No.: 6149), set IChannel to the rule ID (which</p>

	<p>starts from 1), and set lplnBuffer to <u>NET_DVR_POS_FILTER_CFG</u> for setting filter rule of POS information.</p>
Connection Mode of DVR and POS	<p>Call <u>NET_DVR_GetDVRConfig</u> with "NET_DVR_GET_CONNECT_POS_CFG" (command No.: 6150) and set lChannel to the rule ID (which starts from 1) for getting connection mode of DVR and POS.</p> <p>The connection mode parameters are returned in the structure <u>NET_DVR_CONNECT_POS_CFG</u> by lpOutBuffer.</p> <p>Call <u>NET_DVR_SetDVRConfig</u> with "NET_DVR_SET_CONNECT_POS_CFG" (command No.: 6151), set lChannel to the rule ID (which starts from 1), and set lplnBuffer to <u>NET_DVR_CONNECT_POS_CFG</u> for setting connection mode of DVR and POS.</p>
Linkage between Filter Rule and Channel	<p>Call <u>NET_DVR_GetDeviceConfig</u> with "NET_DVR_GET_CHAN_FILTER_CFG" (command No.: 6152), set dwCount to 1, and set lplnBuffer to the structure <u>NET_DVR_CHANNEL_GROUP</u> for getting linkage information between filter rule and channel.</p> <p>The linkage information is returned in the structure <u>NET_DVR_CHAN_FILTER_CFG</u> by lpOutBuffer.</p> <p>Call <u>NET_DVR_SetDeviceConfig</u> with "NET_DVR_SET_CHAN_FILTER_CFG" (command No.: 6153), set dwCount to 1, set lplnBuffer to the structure <u>NET_DVR_CHANNEL_GROUP</u>, and set lplnParamBuffer to the structure <u>NET_DVR_CHAN_FILTER_CFG</u> for setting the linkage between filter rule and channel.</p>
Hidden Configuration of POS Information	<p>Call <u>NET_DVR_GetSTDAbility</u> and set dwAbilityType to "NET_DVR_GET_HIDDEN_INFORMATION_CAPABILITIES" (value: 3608) for getting hidden configuration capability of POS information.</p> <p>The configuration capability is returned in the message <u>XML_Cap_HiddenInformation</u> by lpOutBuffer of structure <u>NET_DVR_STD_ABILITY</u>.</p> <p>Call <u>NET_DVR_GetSTDConfig</u> with "NET_DVR_GET_HIDDEN_INFORMATION" (command No.: 6721) to get hidden parameters of POS information.</p>

The hidden parameters are returned in the structure **NET_DVR_HIDDEN_INFORMATION_CFG** by **lpOutBuffer** of structure **NET_DVR_STD_CONFIG**.

Call **NET_DVR_SetSTDConfig** with "NET_DVR_SET_HIDDEN_INFORMATION" (command No.: 6722), set **lpInBuffer** of structure **NET_DVR_STD_CONFIG** to **NET_DVR_HIDDEN_INFORMATION_CFG** for setting hidden parameters of POS information.

Chapter 10 System Configuration

USB Mode

Function	API
Get USB Mode Configuration Capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/System/usb/capabilities?format=json</i>
Get or Set USB Mode	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET or PUT <i>/ISAPI/System/usb?format=json</i>

Chapter 11 Advanced Functionality

This part shows some advanced functionality integrated based on HCNetSDK, such as data replenishment, protocol passthrough, and so on.

11.1 Data Replenishment

In the condition of disconnection between platform and NVR, the NVR saves the data of people counting, heat map, temperature, and vehicle. When the connection is resumed, the platform can get the specified data from NVR during disconnection period.

People Counting Data Replenishment

Get Capability of People Counting Data Replenishment

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/System/Video/inputs/channels/counting/collection/capabilities?format=json](#), the capability is returned by IpoOutBuffer in the message [JSON_PeopleCounting_CollectionDescriptionCap](#).



Before getting people counting data replenishment capability, you should check whether this function is supported by the device via the node <isSupportCountingCollection> in [XML_VideoCap](#), request URL: GET [/ISAPI/System/Video/capabilities](#).

Perform People Counting Data Replenishment

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: POST [/ISAPI/System/Video/inputs/channels/counting/collection?format=json](#), and the search condition parameters are inputted by IplnBuffer in [JSON_PeopleCounting_CollectionDescription](#).

Heat Map Data Replenishment

Get Capability of Heat Map Data Replenishment

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/System/Video/inputs/channels/heatMap/collection/capabilities?format=json](#)



Before getting people counting data replenishment capability, you should check whether this function is supported by the device via the node <isSupportHeatmapCollection> in [XML_VideoCap](#), URL: GET [/ISAPI/System/Video/capabilities](#).

Perform Heat Map Data Replenishment

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: POST [/ISAPI/System/Video/inputs/channels/heatMap/collection?format=json](#), and set the input parameters IplnBuffer to [JSON_HeatMap_CollectionDescription](#).

Temperature Data Replenishment

Get Temperature Data Replenishment Capability

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Thermal/temperature/collection/capabilities?format=json](#), the capability is returned in [JSON_Temperature_CollectionDescriptionCap](#).



Note

Before getting temperature data replenishment capability, you should check whether this function is supported by the device via the node <isSupportTemperatureCollection> in [XML_ThermalCap](#), URL: GET [/ISAPI/Thermal/capabilities](#).

Perform Temperature Data Replenishment

Call [NET_DVR_STDXMLConfig](#) to pass through the request URL: POST [/ISAPI/Thermal/temperature/collection?format=json](#), and set the input parameter **lpInBuffer** to [JSON_CollectionDescription](#).

Vehicle Detection Data Replenishment

Get Intelligent Search Capability

Call [NET_DVR_STDXMLConfig](#) to transmit [/ISAPI/SDT/Management/IntelligentSearch/capabilities?format=json](#) by GET method.

The capability is returned in [JSON_IntelligentSearchCap](#).

Perform Vehicle Detection Data Replenishment

Call [NET_DVR_STDXMLConfig](#) to transmit [/ISAPI/SDT/Management/IntelligentSearch?format=json](#) by POST method.



Note

Before searching for vehicle detection data, you should check whether the device supports this function. If it supports, the node <isSupportIntelligentSearch> will be returned in the intelligent management capability [JSON_Cap_IntelliManagement](#) (call [/ISAPI/SDT/Management/capabilities?format=json](#) by GET method to get) and its value is "true".

11.2 Integrate by Transmitting Text Protocol

The Device Network SDK support transmitting text protocol, including operation methods, request URLs, 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

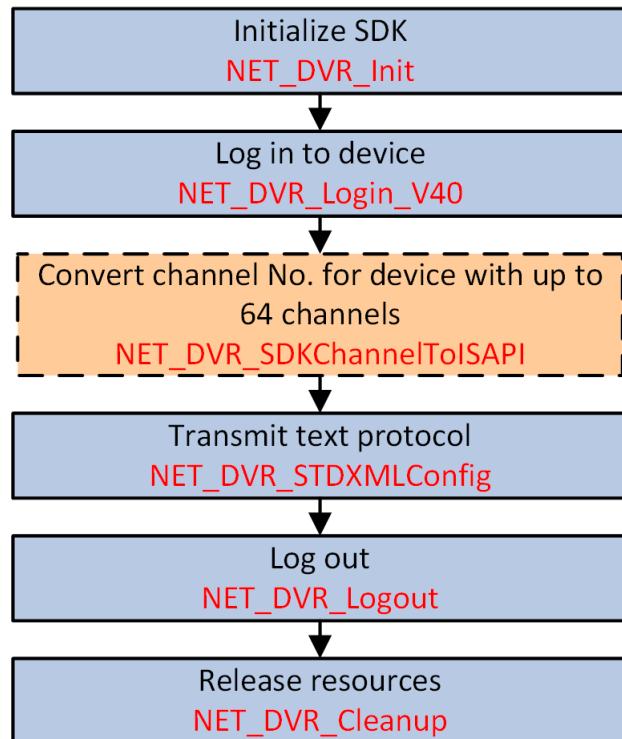


Figure 11-1 API Calling Flow of Integrating by Transmitting Text Protocol

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 12 API Reference

12.1 General APIs

12.1.1 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.

12.1.2 NET_DVR_CloseFormatHandle

Destroy the formatting handle and release the resources to stop formatting HDD.

API Definition

```
BOOL NET_DVR_CloseFormatHandle(  
    LONG  IFormatHandle  
);
```

Parameters

IFormatHandle

[IN] Handle returned by [**NET_DVR_FormatDisk**](#).

Return Value

Return *TRUE* for success, and return *FALSE* for failure.

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

12.1.3 NET_DVR_FormatDisk

Remotely format a HDD.

API Definition

```
LONG NET_DVR_FormatDisk(  
    LONG  lUserID,  
    LONG  lDiskNumber  
)
```

Parameters

lUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

lDiskNumber

[IN] HDD No., which starts from 0, 0xff-all HDDs (excepts the read-only HDDs).

Return Value

Return -1 for failure, and return other values as the parameters of [NET_DVR_CloseFormatHandle](#). If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

Remarks

If the network disconnected during formatting, the formatting will continue but you cannot get the formatting status.

See Also

[NET_DVR_GetFormatProgress](#)

12.1.4 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

IUserID

[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.

12.1.5 NET_DVR_GetDeviceConfig

Get device configuration information in batch (with sending data).

API Definition

```
BOOL NET_DVR_GetDeviceConfig(
    LONG   IUserID,
    DWORD  dwCommand,
    DWORD  dwCount,
    LPVOID lpInBuffer,
    DWORD  dwInBufferSize,
    LPVOID lpStatusList,
    LPVOID lpOutBuffer,
    DWORD  dwOutBufferSize
);
```

Parameters

IUserID

[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](#)

12.1.6 NET_DVR_GetDeviceStatus

Call this API to get device status.

API Definition

```
BOOL NET_DVR_GetDeviceStatus(
    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] Commands for getting the device status.

dwCount

[IN] Number of devices' statuses to be obtained.

lpInBuffer

[IN] Condition buffer of getting status, which is different according to different commands (**dwCommand**).

dwInBufferSize

[IN] Condition buffer size.

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 getting succeeded; if the value is larger than 0, it indicates getting failed.

lpOutBuffer

[OUT] Status details returned by device, which varies with the command (**dwCommand**), and one-to-one corresponds to the cameras that need to be searched.

If the **lpStatusList** value of a camera is larger than 0, the corresponding **lpOutBuffer** is invalid.

dwOutBufferSize

[IN] Output buffer size

Return Values

Returns *TRUE* for success, and returns *FALSE* for all failed. If returns *TRUE*, it does not mean that all settings are succeeded, you can check the value of **IpStatusList[n]** to judge which one is succeeded. If returning failed, you can call [***NET_DVR_GetLastError***](#) to get the error code.

Remarks

- If you want to get all devices' statuses, you should set the **dwCount** to 0xffffffff, set **lpInBuffer** to NULL, set **dwInBufferSize** to 0, set **lpStatusList** to NULL.
- For **lpOutBuffer**, the first 4-byte is the total number of structures returned by device, and the following bytes contain the structures details. If the configured output buffer is insufficient, only a part of structures will be returned.

12.1.7 [***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.

lChannel

[IN] Channel No. (NIC No.), which varies with different commands. 0xffffffff-invalid or all channels, 1-main NIC, 2-extended NIC.

lpOutBuffer

[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***](#)

12.1.8 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.

12.1.9 NET_DVR_GetFormatProgress

Get HDD formatting progress.

API Definition

```
BOOL NET_DVR_GetFormatProgress(  
    LONG lFormatHandle,
```

```
LONG *pCurrentFormatDisk,  
LONG *pCurrentDiskPos,  
LONG *pFormatStatic  
);
```

Parameters

IFormatHandle

[IN] Formatting handle, which is returned by [NET_DVR_FormatDisk](#).

pCurrentFormatDisk

[OUT] Pointer of HDD No. that is formatting, the No. starts from 0, and -1 indicates the initialization status.

pCurrentDiskPos

[OUT] Pointer of HDD formatting progress, its value is between 0 and 100.

pFormatStatic

[OUT] Pointer of HDD formatting status, values: 0-formatting, 1-formatted, 2-formatting failed, 3-starting formatting failed as the network exception or network HDD loss.

Return Value

Return *TRUE* for success, and return *FALSE* for failure.

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

12.1.10 NET_DVR_GetIPCProtoList_V41

Get the network camera protocol list supported by the device.

API Definition

```
BOOL NET_DVR_GetIPCProtoList_V41(  
    LONG          IUserID,  
    NET_DVR_IPC_PROTO_LIST_V41  IpProtoList  
)
```

Parameters

IUserID

[IN] Value returned by [NET_DVR_Login_V40](#)

IpProtoList

[IN] Network camera protocol list, refer to the structure [NET_DVR_IPC_PROTO_LIST_V41](#) for details.

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 is used to get the external network camera protocols supported by device. When calling this API, set **pBuffer** and **dwBufLen** in the structure [**NET_DVR_IPC_PROTO_LIST_V41**](#) to "null" and "0", and then get the valid number of network camera protocols. Finally, allocate the cache according to the number of protocols and call this API again to get protocol details.

12.1.11 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.

12.1.12 NET_DVR_GetLastErrorModelCode

Get the error code categorized by functional modules of the last operation.

API Definition

```
void NET_DVR_GetLastErrorModelCode(  
    DWORD    *pModelCode,  
    DWORD    *pMErrDevSelfEx  
)
```

Parameters

pModelCode

[OUT] Error code, which is categorized by functional modules.

pMErrDevSelfEx

[OUT] Extension of **pModelCode**. It is used to define the custom error code, which is categorized by functional modules.

Return Value

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

12.1.13 NET_DVR_GetLinkAddr

Get the IP address and port information of the stream sender and recipient.

API Definition

```
BOOL NET_DVR_GetLinkAddr(  
    LONG      ILinkHandle,  
    NET_DVR_LINK_KIND  enumLinkKind,  
    LPNET_DVR_LINK_ADDR  IpLinkAddr  
) ;
```

Parameters

ILinkHandle

Returned handle when live view, playback, two-way audio, or audio forward is enabled.

enumLinkKind

Stream types, see details in [NET_DVR_LINK_KIND](#).

IpLinkAddr

Pointer of the structure about IP address and port information of the stream sender and recipient, see details in [NET_DVR_LINK_ADDR](#).

Return Value

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

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

12.1.14 NET_DVR_GetNextRemoteConfig

Get the next search result.

API Definition

```
LONG NET_DVR_GetNextRemoteConfig(  
    LONG   IHandle,  
    void   *IpOutBuff,  
    DWORD  dwOutBuffSize  
) ;
```

Parameters

IHandle

[IN] Search handle, which is the value returned by [NET_DVR_StartRemoteConfig](#).

IpOutBuff

[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 NET_DVR_StopRemoteConfig can be called to end.
NET_SDK_GET_NEXT_STATUS_FAILED	1003	Getting data exception. The API NET_DVR_StopRemoteConfig 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.

12.1.15 NET_DVR_GetRemoteConfigState

Get the status of persistent connection.

API Definition

```
BOOL NET_DVR_GetRemoteConfigState(  
    LONG  lHandle,  
    void  *pState  
>;
```

Parameters

lHandle

[IN] Handle, which is returned by [***NET_DVR_StartRemoteConfig***](#).

pState

[OUT] A 4-byte status value, the returned statuses are different according to different remote configuration commands

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.

12.1.16 NET_DVR_GetRtspConfig

Get RTSP (Real-Time Streaming Protocol) parameters.

Structure Definition

```
BOOL NET_DVR_GetRtspConfig(  
    LONG      lUserID,  
    DWORD     dwCommand,  
    LPNET_DVR_RTSPCFG lpOutBuffer,  
    DWORD     dwOutBufferSize  
)
```

Parameters

lUserID

[IN] The value returned by [NET_DVR_Login_V40](#).

dwCommand

[IN] Reserved, set to 0.

lpOutBuffer

[OUT] Output buffer, see details in the structure [NET_DVR_RTSPCFG](#).

dwOutBufferSize

[IN] Size of the output buffer.

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.

12.1.17 NET_DVR_GetSDKLocalCfg

Get the HCNetSDK's local configuration parameters.

API Definition

```
BOOL NET_DVR_GetSDKLocalCfg(  
    NET_SDK_LOCAL_CFG_TYPE enumType,
```

```
void *IpOutBuff  
);
```

Parameters

enumType

[IN] Configuration options. Different values of configuration options correspond to different parameters, see details in [NET_SDK_LOCAL_CFG_TYPE](#).

IpOutBuff

[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](#)

12.1.18 NET_DVR_GetSTDAbility

Get the device capabilities.

API Definition

```
BOOL NET_DVR_GetSTDAbility(  
    LONG     IUserID,  
    DWORD    dwAbilityType,  
    NET_DVR_STD_ABILITY  lpAbilityParam  
);
```

Parameters

IUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

dwAbilityType

[IN] Capability types, which are different according to different functions.

lpAbilityParam

[IN/OUT] Capability details, including condition parameter, input parameter, output parameter, and so on (see details in the structure [NET_DVR_STD_ABILITY](#)), which are different according to different capability types.

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.

12.1.19 **NET_DVR_GetSTDConfig**

Get the device configuration information.

API Definition

```
BOOL NET_DVR_GetSTDConfig(  
    LONG          IUserID,  
    DWORD         dwCommand,  
    NET_DVR_STD_CONFIG  lpConfigParam  
)
```

Parameters

IUserID

[IN] Value returned by [**NET_DVR_Login_V40**](#).

dwCommand

[IN] Device configuration commands, which are different according to different configuration functions.

lpConfigParam

[IN][OUT] Set input and output parameters, which are different according to different configuration functions. For different configuration functions, the **lpCondBuffer** and **lpOutBuffer** in the **lpConfigParam** are also different. See the structure [**NET_DVR_STD_CONFIG**](#) for details.



Note

When getting configuration parameters, the **lpInBuffer** in the **lpConfigParam** is invalid, you can set it to 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.

See Also

[**NET_DVR_SetSTDConfig**](#)

12.1.20 NET_DVR_GetUpnpNatState

Get UPnP port mapping status.

API Definition

```
BOOL NET_DVR_GetUpnpNatState(  
    LONG          IUserID,  
    NET_DVR_UPNP_NAT_STATE  IpState  
);
```

Parameters

IUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

IpState

[OUT] UPnP port mapping status, see details in the structure .

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If returning failed, you can call [NET_DVR_GetLastError](#) to get the error code.

12.1.21 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](#)

12.1.22 NET_DVR_InquestGetEncryptState

Get encryption status of device stream.

API Definition

```
BOOL NET_DVR_InquestGetEncryptState(  
    LONG  lUserID,  
    LONG  lChannel,  
    BOOL  *bEncrypt  
);
```

Parameters

lUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

lChannel

[IN] Channel No., 0xffffffff-invalid or all channels.

bEncrypt

[OUT] Encryption status: TRUE-encrypted, FALSE-not encrypted.

Return Value

Return *TRUE* for success, and return *FALSE* for failure.

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

See Also

[NET_DVR_InquestStreamEncrypt](#)

12.1.23 NET_DVR_InquestSetSecretKey

Set encryption key for encrypting stream.

API Definition

```
BOOL NET_DVR_InquestSetSecretKey(  
    LONG          lUserID,  
    NET_DVR_INQUEST_SECRET_INFO  pSecretInfo  
);
```

Parameters

lUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

pSecretInfo

[IN] Encryption key information, see details in the structure .

Return Value

Return *TRUE* for success, and return *FALSE* for failure.

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

12.1.24 NET_DVR_InquestStreamEncrypt

Enable stream encryption.

API Definition

```
BOOL NET_DVR_InquestStreamEncrypt(  
    LONG lUserID,  
    LONG lChannel,  
    BOOL bEncrypt  
)
```

Parameters

lUserID

[IN] Value returned by [NET_DVR_Login_V40](#) .

lChannel

[IN] Channel No., 0xffffffff-invalid or all channels.

bEncrypt

[IN] Encryption flag: *TRUE*-encrypted, *FALSE*-not encrypted.

Return Value

Return *TRUE* for success, and return *FALSE* for failure.

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

See Also

[NET_DVR_InquestGetEncryptState](#)

12.1.25 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  IpDeviceInfo  
);
```

Parameters

pLoginInfo

[IN] Login parameters, including device address, user name, password, and so on. See details in the structure [NET_DVR_USER_LOGIN_INFO](#).

IpDeviceInfo

[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](#)

12.1.26 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***](#).

12.1.27 NET_DVR_RebootDVR

Reboot the device.

API Definition

```
BOOL NET_DVR_RebootDVR(  
    LONG  lUserID  
);
```

Parameters

lUserID

[IN] User ID, which is the returned value of [***NET_DVR_Login_V40***](#).

Return Values

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

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

12.1.28 NET_DVR_RemoteControl

Implement remote control.

API Definition

```
BOOL NET_DVR_RemoteControl(  
    LONG      lUserID,  
    DWORD     dwCommand,  
    LPVOID    lpInBuffer,  
    DWORD     dwInBufferSize  
);
```

Parameters

lUserID

[IN] Value returned by [***NET_DVR_Login_V40***](#).

dwCommand

[IN] Control commands. To realize different functions, the commands are different.

IplInBuffer

[IN] Input parameters, which vary with different control commands.

dwInBufferSize

[IN] Size of input parameters.

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.

12.1.29 NET_DVR_RenderPrivateData

Whether to display intelligent information on live video.

API Definition

```
BOOL NET_DVR_RenderPrivateData(  
    LONG  IRealHandle,  
    int   iIntelType,  
    BOOL  bTrue  
)
```

Parameters

IRealHandle

[IN] Live view handle, which is returned by [NET_DVR_RealPlay_V40](#).

iIntelType

[IN] Intelligent information types, which are enumerated in [PRIDATA_RENDER](#).

bTrue

[IN] Whether to enable displaying the intelligent information on live video: true, false.

Return Value

Return *TRUE* for success, and return *FALSE* for failure.

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

12.1.30 PlayM4_RenderPrivateDataEx

Whether to display intelligent thermal information on the live video.

API Definition

```
BOOL NET_DVR_RenderPrivateDataEx(  
    LONG lRealHandle,  
    int iIntelType,  
    int iSubType,  
    BOOL bTrue  
)
```

Parameters

lRealHandle

[IN] Live view handle, which is returned by [NET_DVR_RealPlay_V40](#).

iIntelType

[IN] Intelligent information types, which are enumerated in [PRIDATA_RENDER](#).

iSubType

[IN] Sub types of thermal information to display; when the **iIntelType** is set to "RENDER_FIRE_DETECT" (fire point detection information), the sub types are enumerated in [FIRE_ALARM](#); when the **iIntelType** is set to "RENDER_TEM" (rule information of temperature measurement), the sub types are enumerated in [TEM_FLAG](#).

bTrue

[IN] Whether to enable displaying the intelligent information on live video: true, false.

Return Value

Return *TRUE* for success, and return *FALSE* for failure.

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

12.1.31 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](#).

IInChannel

[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.

12.1.32 NET_DVR_SendRemoteConfig

Send data via the persistent connection.

API Definition

```
BOOL NET_DVR_SendRemoteConfig(  
    LONG   IHandle,  
    DWORD  dwDataType,  
    char   *pSendBuf,  
    DWORD  dwBufSize  
)
```

Parameters

IHandle

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.

12.1.33 NET_DVR_SendWithRecvRemoteConfig

Set condition to get result one by one.

API Definition

```
LONG NET_DVR_SendWithRecvRemoteConfig(
    LONG   IHandle,
    void   *IpInBuff,
    DWORD  dwInBuffSize,
    void   *IpOutBuff,
    DWORD  dwOutBuffSize,
    DWORD  *dwOutDataLen
);
```

Parameters

IHandle

[IN] Handle returned by [*NET_DVR_StartRemoteConfig*](#).

IpInBuff

[IN] Buffer to save input parameter structure, and this structure varies with the value of **dwCommand** in API [*NET_DVR_StartRemoteConfig*](#).

dwInBuffSize

[IN] Input buffer size, which equals to the size of input parameter structure (**IpInBuff**).

IpOutBuff

[OUT] Buffer to save output parameter structure, and this structure corresponds one-to-one with the input parameter structure (**IpInBuff**).

dwOutBuffSize

[IN] Output buffer size, which equals to the size of output parameter structure (**IpOutBuff**).

dwOutDataLen

[OUT] Pointer of actual received data size, and it cannot be set to "null".

Return Value

Return -1 for failure, and return other values as the status codes, refer to the table below.

Status Macro Definition	Status Code	Description
NET_SDK_CONFIG_STATUS_SUCCESS	1000	The data is obtained. The API <i>NET_DVR_SendWithRecvRemoteConfig</i>

Status Macro Definition	Status Code	Description
		should be called again to get the next item of data.
NET_SDK_CONFIG_STATUS_NEEDWAIT	1001	Waiting. The API NET_DVR_SendWithRecvRemoteConfig can be called again.
NET_SDK_CONFIG_STATUS_FINISH	1002	All data is obtained. The API <u>NET_DVR_StopRemoteConfig</u> can be called to end.
NET_SDK_CONFIG_STATUS_FAILED	1003	Getting data failed. The API NET_DVR_SendWithRecvRemoteConfig can be called to get the next item of data.
NET_SDK_CONFIG_STATUS_EXCEPTION	1004	Getting data exception. The API <u>NET_DVR_StopRemoteConfig</u> can be called to end.

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code for reference.

Remarks

To get all information, you should call this API repeatedly.

12.1.34 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.

12.1.35 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**.

IpStatusList

[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., **IpStatusList[2]** corresponds to **IpInBuffer[2]**.

If the parameter value is 0, it refers to setting succeeded, otherwise, this parameter value is the error code.

IpInParamBuffer

[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**](#)

12.1.36 NET_DVR_SetDeviceConfigEx

Set the device parameters in batch (sending data is supported).

API Definition

```
BOOL NET_DVR_SetDeviceConfigEx(
    LONG      IUserID,
    DWORD     dwCommand,
    DWORD     dwCount,
    NET_DVR_IN_PARAM  *IpInParam,
    NET_DVR_OUT_PARAM *IpOutParam
);
```

Parameters

IUserID

[IN] Value returned by [**NET_DVR_Login_V40**](#).

dwCommand

[IN] Device configuration commands, which are different according to different configuration functions.

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 64 cameras can be configured at a time.

IpInParam

[IN] Input parameter buffer, see details in the structure [NET_DVR_IN_PARAM](#) (the value of each member in the structure varies with the **dwCommand**).

IpOutParam

[IN] Output parameter buffer, see details in the structure [NET_DVR_OUT_PARAM](#) (the value of each member in the structure varies with the **dwCommand**).

Return Values

Returns *TRUE* for success, and returns *FALSE* for all failed. If returns *TRUE*, it does not mean that all settings are succeeded, you can check 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

Extended From: [NET_DVR_SetDeviceConfig](#)

12.1.37 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.

IChannel

[IN] Channel No. (NIC No.), which varies with different commands. 0xFFFFFFFF-invalid, 1-main NIC, 2-extended NIC.

IInBuffer

[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***](#)

12.1.38 NET_DVR_SetExceptionCallBack_V30

Sign up a window handle or callback function for receiving the message of exception, reconnection, and so on.

API Definition in Windows Operating System

```
BOOL NET_DVR_SetExceptionCallBack_V30(
    UINT          nMessage,
    HWND          hWnd,
    fExceptionCallBack cbExceptionCallBack,
    void         *pUser
);
```

API Definition in Linux Operating System

```
BOOL NET_DVR_SetExceptionCallBack_V30(
    UINT          nMessage,
    void*        hWnd,
    fExceptionCallBack cbExceptionCallBack,
    void         *pUser
);
```

Parameters

nMessage

[IN] Message, this parameter is reserved when programming in Linux operating system.

hWnd

[IN] Window handle for receiving exception message, this parameter is reserved when programming in operating system.

cbExceptionCallBack

[IN] Callback function for receiving exception message, see details in [*fExceptionCallBack*](#).

pUser

[IN] User data.

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 programming in Windows operating system, the parameter **hWnd** and **cbExceptionCallBack** can not be set to "NULL" at same time; for programming in Linux operating system, the parameter **cbExceptionCallBack** can not be set to "NULL", otherwise, no exception message will be received.

Example

Sample Code of Setting Callback Function to Receive Exception Messages

```
//Register callback function for receiving exception messages
NET_DVR_SetExceptionCallBack_V30(WM_NULL, NULL, g_ExceptionCallBack, NULL);

//Callback function receive exception message
void CALLBACK g_ExceptionCallBack(DWORD dwType, LONG lUserID, LONG lHandle, void *pUser)
{
    char tempbuf[256];
    ZeroMemory(tempbuf, 256);
    switch(dwType)
    {
        case EXCEPTION_AUDIOEXCHANGE: //Voice talk network exception
            sprintf(tempbuf,"Voice talk network exception!!!");
            TRACE("%s",tempbuf); //TODO: close voice talk
            break;
        case EXCEPTION_ALARM: //Alarm upload network exception
            sprintf(tempbuf,"Alarm upload network exception!!!");
            TRACE("%s",tempbuf); //TODO: close alarm upload
            break;
        case EXCEPTION_PREVIEW: //Network preview exception
            sprintf(tempbuf,"Network preview exception!!!");
            TRACE("%s",tempbuf); //TODO: close network preview
            break;
        case EXCEPTION_SERIAL: //Transparent channel transmission exception
            sprintf(tempbuf,"Transparent channel transmission exception!!!");
            TRACE("%s",tempbuf);
            //TODO: close transparent channel
            break;
        case EXCEPTION_RECONNECT: //Preview reconnect
            break;
        case EXCEPTION_DISCONNECT: //Preview disconnect
            break;
    }
}
```

```
    break;
default:
    break;
}
};
```

12.1.39 NET_DVR_SetRtspConfig

Set RTSP (Real-Time Streaming Protocol) parameters.

Structure Definition

```
BOOL NET_DVR_SetRtspConfig(
    LONG      IUserID,
    DWORD     dwCommand,
    LPNET_DVR_RTSPCFG  lpInBuffer,
    DWORD     dwInBufferSize
);
```

Parameters

IUserID

[IN] The value returned by [NET_DVR_Login_V40](#).

dwCommand

[IN] Reserved, set to 0.

lpInBuffer

[IN] Input buffer.

dwInBufferSize

[IN] The size of input buffer.

Return Values

Returns *TRUE* for success, and return *FALSE* for failure. If *FALSE* is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

12.1.40 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 12-1 NET_SDK_INIT_CFG_TYPE

enumType	Value	Description	IplnBuff
NET_SDK_INIT_CFG_ABILITY	1	Capability supported by SDK.	<u>NET_DVR_INIT_CFG_ABILITY</u>
NET_SDK_INIT_CFG_SDK_PATH	2	Set loading path for component libraries (supported by both Linux and Windows system).	<u>NET_DVR_LOCAL_SDK_PATH</u>
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., C:\libeay32.dll .
NET_SDK_INIT_CFG_SSLEAY_PATH	4	Set path (including library name) for ssleay32.dll (Windows), libssl.so (Linux), libssl.dylib (Mac) of OpenSSL in version 1.1.1 and 1.0.2.	Path in string format, e.g., C:\ssleay32.dll .

IplnBuff

[IN] Input parameter. Different parameter types correspond to different structures, see details in the table above.

Return Values

Returns *TURE* 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.

12.1.41 NET_DVR_SetSDKLocalCfg

Set the local parameters.

API Definition

```
BOOL NET_DVR_SetSDKLocalCfg(  
    NET_SDK_LOCAL_CFG_TYPE enumType,  
    void* const     lplnBuff  
)
```

Parameters

enumType

[IN] Configuration options. Different values of configuration options correspond to different SDK parameters, see details in [NET_SDK_LOCAL_CFG_TYPE](#).

lplnBuff

[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](#)

12.1.42 NET_DVR_SetSDKSecretKey

Set the key for the Device Network SDK users to decrypt the stream.

API Definition

```
BOOL NET_DVR_SetSDKSecretKey(  
    LONG  lUserID,  
    char  *sSecretKey  
)
```

Parameters

lUserID

[IN] User ID, which is the value returned by the login API such as [NET_DVR_Login_V40](#).

sSecretKey

[IN] 16-byte key.

Return Value

Return *TRUE* for success and *FALSE* for failure. If *FALSE* is returned, you can call [*NET_DVR_GetLastError*](#) to get the error code.

Remarks

If the stream is encrypted, you need to call this API to set the stream decryption key before live view.

12.1.43 NET_DVR_SetSTDConfig

Set the device parameters.

API Definition

```
BOOL NET_DVR_SetSTDConfig(  
    LONG          IUserID,  
    DWORD         dwCommand,  
    NET_DVR_STD_CONFIG  IpConfigParam  
)
```

Parameters

IUserID

[IN] Value returned by [*NET_DVR_Login_V40*](#).

dwCommand

[IN] Device configuration commands, which are different according to different configuration functions.

IpConfigParam

[IN][OUT] Set input and output parameters, which are different according to different configuration functions. For different configuration functions, the **IpCondBuffer** and **IpInBuffer** in the **IpConfigParam** are also different. See the structure [*NET_DVR_STD_CONFIG*](#) for details.



Note

When getting configuration parameters, the **IpOutBuffer** in the **IpConfigParam** is invalid, you can set it to "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.

See Also

[NET_DVR_GetSTDConfig](#)

12.1.44 NET_DVR_StartGetDevState

Enable device status checking.

API Definition

```
BOOL NET_DVR_StartGetDevState(  
    NET_DVR_CHECK_DEV_STATE *pParams  
)
```

Parameters

pParams

[IN] Parameters of checking device status. See details in the structure [NET_DVR_CHECK_DEV_STATE](#).

Return Value

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

After enabling, the SDK will regularly check the device status, and the obtained device status will be returned in the callback function of structure [NET_DVR_CHECK_DEV_STATE](#).

See Also

[NET_DVR_StopGetDevState](#)

12.1.45 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:

dwCommand Macro Definition	Value	Description	IpInBuffer Related Structure	IpBuffer Related Structure
NET_DVR_GET_ALL_RECORD_PASSBACK_TASK_MANUAL	6235	Get tasks of manually copying back videos	<u>NET_DVR_RECO_RD_PASSBACK_MANUAL_COND</u>	<u>NET_DVR_RECO_RD_PASSBACK_MANUAL_TASK_RET</u>

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.

12.1.46 NET_DVR_STDControl

Control remotely.

API Definition

```
BOOL NET_DVR_STDControl(
    LONG          lUserID,
    DWORD         dwCommand,
    LPNET_DVR_STD_CONTROL lpControlParam
);
```

Parameters

lUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

dwCommand

[IN] Control commands.

lpControlParam

[IN][OUT] Input parameters and output parameters of remote control. The parameters are different according to different control commands, see details in the structure of [NET_DVR_STD_CONTROL](#).

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.

12.1.47 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 **IpInputParam** and output parameter **IpOutputParam** are different when transmitting text protocol for implementing different functions, and each parameter corresponds to a component of text protocol, see the relations below:

Parameter of NET_DVR_STDXMLConfig		Component of Text Protocol
IpInputParam	IpRequestMethod (see in structure <u>NET_DVR_XML_CONFIG_INPUT</u>)	Method+URL E.g., GET /ISAPI/System/ capabilities
	IpInBuffer (see in structure <u>NET_DVR_XML_CONFIG_INPUT</u>)	Request Message
IpOutputParam	IpOutBuffer (see in structure <u>NET_DVR_XML_CONFIG_OUTPUT</u>)	Response Message
	IpStatusBuffer (see in structure <u>NET_DVR_XML_CONFIG_OUTPUT</u>)	Response Message

12.1.48 NET_DVR_StopGetDevState

Disable device status checking.

API Definition

```
BOOL NET_DVR_StopGetDevState()
{}
```

Parameters

None

Return Value

Return *TRUE* for success, and return *FALSE* for failure.

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

See Also

[NET_DVR_StartGetDevState](#)

12.1.49 NET_DVR_StopRemoteConfig

Disconnect the persistent connection to stop remote configuration, and release resources.

API Definition

```
BOOL NET_DVR_StopRemoteConfig(  
    LONG lHandle  
>);
```

Parameters

IHandle

[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.

12.2 Live View and PTZ APIs

12.2.1 NET_DVR_ChangeWndResolution

Send the window change notification to PlayCtrl LibrarySDK when the size of display window changed (only applicable for Linux system).

API Definition

```
BOOL NET_DVR_ChangeWndResolution(  
    LONG lRealHandle  
>);
```

Parameters

lRealHandle

[IN] Live view handle, which is returned by [NET_DVR_RealPlay_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.

12.2.2 NET_DVR_GetCurrentAudioCompress_V50

Get the effective audio encoding parameters of two-way audio.

API Definition

```
BOOL NET_DVR_GetCurrentAudioCompress_V50(
    LONG          lUserID,
    LPNET_DVR_AUDIO_CHANNEL *lpAudioChannel,
    LPNET_DVR_COMPRESSION_AUDIO lpCompressAudio
);
```

Parameters

lUserID

[IN] User information returned by [NET_DVR_Login_V40](#).

lpAudioChannel

[IN] Audio channel information, refer to the structure [NET_DVR_AUDIO_CHANNEL](#) for details.

lpCompressAudio

[OUT] Audio encoding parameters, refer to the structure [NET_DVR_COMPRESSION_AUDIO](#) for details.

Return Value

Return *TRUE* for success, and return *FALSE* for failure.

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

12.2.3 NET_DVR_PTZControl

Implement PTZ control.

API Definition

```
BOOL NET_DVR_PTZControl(
    LONG  lRealHandle,
    DWORD dwPTZCommand,
    DWORD dwStop
);
```

Parameters

IRealHandle

[IN] Value returned by [NET_DVR_RealPlay_V40](#).

dwPTZCommand

[IN] PTZ control commands, see details in the following table:

Command	Command No.	Description
LIGHT_PWRON	2	Turn on light.
WIPER_PWRON	3	Turn on wiper.
FAN_PWRON	4	Turn on fan.
HEATER_PWRON	5	Turn on heater.
AUX_PWRON1	6	Enable auxiliary device.
AUX_PWRON2	7	Enable auxiliary device.
ZOOM_IN	11	Zoom in.
ZOOM_OUT	12	Zoom out.
FOCUS_NEAR	13	Focus near.
FOCUS_FAR	14	Focus far.
IRIS_OPEN	15	Open iris.
IRIS_CLOSE	16	Close iris.
TILT_UP	21	Tilt up.
TILT_DOWN	22	Tilt down.
PAN_LEFT	23	Pan left.
PAN_RIGHT	24	Pan right.
UP_LEFT	25	Tilt up and pan left.
UP_RIGHT	26	Tilt up and pan right.
DOWN_LEFT	27	Tilt down and pan left.
DOWN_RIGHT	28	Tilt down and pan right.
PAN_AUTO	29	Auto-scan.
TILT_DOWN_ZOOM_IN	58	Tilt down and zoom in.

Command	Command No.	Description
TILT_DOWN_ZOOM_OUT	59	Tilt down and zoom out.
PAN_LEFT_ZOOM_IN	60	Pan left and zoom in.
PAN_LEFT_ZOOM_OUT	61	Pan left and zoom out.
PAN_RIGHT_ZOOM_IN	62	Pan right and zoom in.
PAN_RIGHT_ZOOM_OUT	63	Pan right and zoom out.
UP_LEFT_ZOOM_IN	64	Tilt up, pan left and zoom in.
UP_LEFT_ZOOM_OUT	65	Tilt up, pan left and zoom out.
UP_RIGHT_ZOOM_IN	66	Tilt up, pan right and zoom in.
UP_RIGHT_ZOOM_OUT	67	Tilt up, pan right and zoom out.
DOWN_LEFT_ZOOM_IN	68	Tilt down, pan left and zoom in.
DOWN_LEFT_ZOOM_OUT	69	Tilt down, pan left and zoom out.
DOWN_RIGHT_ZOOM_IN	70	Tilt down, pan right and zoom in.
DOWN_RIGHT_ZOOM_OUT	71	Tilt down, pan right and zoom out.
TILT_UP_ZOOM_IN	72	Tilt up and zoom in.
TILT_UP_ZOOM_OUT	73	Tilt up and zoom out.

dwStop

[IN] Whether to enable PTZ control: 0-no, 1-yes.

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 should start the live view.
- You should call this API for twice to perform each operation (including start and stop, which is determined by the parameter **dwStop**).

- The PTZ control commands correspond to the control codes between device and PTZ, the device will send the control codes to PTZ according to the configured decoder type and address. If the decoder of device and PTZ mismatches, you should configure the device decoder again. If the PTZ decoder is not supported by device, it is not available to call this API to control PTZ.
- By default, the PTZ movement is in maximum speed.

12.2.4 NET_DVR_PTZCruise

Set and start patrol during PTZ control.

API Definition

```
BOOL NET_DVR_PTZCruise(
    LONG  lRealHandle,
    DWORD dwPTZCruiseCmd,
    BYTE  byCruiseRoute,
    BYTE  byCruisePoint,
    WORD  wInput
);
```

Parameters

IRealHandle

[IN] Value returned by [**NET_DVR_RealPlay_V40**](#).

dwPTZCruiseCmd

[IN] Patrol commands, see details in the following table:

Command	Command No.	Description
FILL_PRE_SEQ	30	Add presets to the patrol.
SET_SEQ_DWELL	31	Set the dwell time at a preset.
SET_SEQ_SPEED	32	Set scanning speed between two presets.
CLE_PRE_SEQ	33	Remove the preset from the patrol.
RUN_SEQ	37	Start patrol.
STOP_SEQ	38	Stop patrol.

byCruiseRoute

[IN] Patrol, up to 32 patrols can be set (the No. starts from 1).

byCruisePoint

Preset, up to 32 presets can be added to one patrol (the No. starts from 1).

wInput

[IN] The values are different according to different patrol commands, maximum presets number-300, maximum dwell time-255, maximum speed-40.

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

- Before setting or calling patrol, at least two presets should be configured.
- The patrol commands correspond to the control codes between device and PTZ, the device will send the control codes to PTZ according to the configured decoder type and address. If the decoder of device and PTZ mismatches, you should configure the device decoder again. If the PTZ decoder is not supported by device, it is not available to call this API to set and start patrol.

See Also

[NET_DVR_PTZPreset](#)

12.2.5 NET_DVR_PTZPreset

Set and call preset during PTZ control.

API Definition

```
BOOL NET_DVR_PTZPreset(  
    LONG   lRealHandle,  
    DWORD  dwPTZPresetCmd,  
    DWORD  dwPresetIndex  
) ;
```

Parameters

lRealHandle

[IN] Value returned by [NET_DVR_RealPlay_V40](#).

dwPTZPresetCmd

Preset commands, see details in the following table:

Command	Command No.	Description
SET_PRESET	8	Set preset.
CLE_PRESET	9	Clear preset.
GOTO_PRESET	39	Call preset.

dwPresetIndex

[IN] Preset No., which starts from 1. And up to 300 presets can be configured.

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 preset commands correspond to the control codes between device and PTZ, the device will send the control codes to PTZ according to the configured decoder type and address. If the decoder of device and PTZ mismatches, you should configure the device decoder again. If the PTZ decoder is not supported by device, it is not available to call this API to set and call preset.

12.2.6 NET_DVR_RealPlay_V40

Start live view.

API Definition

```
LONG NET_DVR_RealPlay_V40(
    LONG          lUserID,
    NET_DVR_PREVIEWINFO  lpPreviewInfo,
    REALDATA CALLBACK     fRealDataCallBack_V30,
    void            *pUser
);
```

Parameters

lUserID

[IN] Value returned by [***NET_DVR_Login_V40***](#).

lpPreviewInfo

[IN] Live view parameters, see the definitions in the structure [***NET_DVR_PREVIEWINFO***](#).

fRealDataCallBack_V30

[IN] Stream data callback function, see details in [***REALDATA CALLBACK***](#).

pUser

[IN] User data.

Return Values

Return -1 for failure, and returns other values as the handle parameters of

[***NET_DVR_StopRealPlay***](#).

If -1 is returned, you can call [***NET_DVR_GetLastError***](#) to get the error code.

Remarks

- The callback function in this API can be set as null, so the callback function will not call back the stream data to user. But the user can still call [NET_DVR_SetRealDataCallBack](#) or [NET_DVR_SetStandardDataCallBack](#) to register a callback function to get the stream data.
- When the platform or system is offline, the device will keep the streaming connection for 10 seconds.
- For developing in Linux system, the live view API will import valid window handle and you should call [NET_DVR_ChangeWndResolution](#) to notify the PlayCtrl Library SDK to get the window size before changing the window size during live view. For developing in Window system, the window size is self-adaptive without calling any other API.

12.2.7 NET_DVR_RealPlaySpecial

Start live view (get stream from URLs) .

API Definition

```
LONG NET_DVR_RealPlaySpecial(
    LONG             lUserID,
    LPNET_DVR_PREVIEWINFO_SPECIAL lpPreviewInfo,
    REALDATACALLBACK      fRealDataCallBack_V30,
    void              *pUser
);
```

Parameters

lUserID

[IN] Value returned by [NET_DVR_Login_V40](#)

lpPreviewInfo

[IN] Live view parameters, see the definitions in the structure [NET_DVR_PREVIEWINFO_SPECIAL](#) .

fRealDataCallBack_V30

[IN] Stream data callback function, see details in [REALDATACALLBACK](#) .

pUser

[IN] User data.

Return Values

Return -1 for failure, and return other values as the handle parameters of [NET_DVR_StopRealPlay](#) . If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

Remarks

- The callback function in this API can be set to null, so the callback function will not call back the stream data to user. But the user can still call [**NET_DVR_SetRealDataCallBack**](#) or [**NET_DVR_SetStandardDataCallBack**](#) to register a callback function to get the stream data.
- When the client is offline, the device will keep the streaming connection for 10 seconds.
- For developing in Linux system, the live view API will import the valid window handle and you should call [**NET_DVR_ChangeWndResolutio**](#) to notify the player SDK to get the window size before changing the window size during live view. For developing in Windows system, the window size is self-adaptive without calling any other API.

See Also

[**NET_DVR_RealPlay_V40**](#)

12.2.8 NET_DVR_SetRealDataCallBack

Set a callback function for getting the real-time stream data.

API Definition

```
BOOL NET_DVR_SetRealDataCallBack(  
    LONG     IRealHandle,  
    fRealDataCallBack cbRealDataCallBack,  
    DWORD     dwUser  
>;
```

Parameters

IRealHandle

[IN] Value returned by [**NET_DVR_RealPlay_V40**](#).

cbRealDataCallBack

[IN] Stream data callback function, see details in [**fRealDataCallBack**](#).

dwUser

[IN] User data.

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

- This API provides the function of starting/stopping callback and processing data. When the callback function **fRealDataCallBack** is set to non-null, it will start callback and processing data. Otherwise, callback and data processing will be stopped.
- The first called back packet is a file header in 40-byte, which is for the following decoding, and then the encoded stream will be called back. The maximum size of callback data is 256 bytes.

12.2.9 NET_DVR_SetStandardDataCallBack

Register a callback function for getting the standard real-time stream data.

API Definition

```
BOOL NET_DVR_SetStandardDataCallBack(  
    LONG      IRealHandle,  
    fStdDataCallBack cbStdDataCallBack,  
    DWORD      dwUser  
)
```

Parameters

IRealHandle

[IN] Value returned by [NET_DVR_RealPlay_V40](#).

cbRealDataCallBack

[IN] Standard stream data callback function, see details in [fStdDataCallBack](#).

dwUser

[IN] User data.

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

- This API provides the function of starting/stopping callback and processing data. When the callback function **fStdDataCallBack** is set as non-null, it will start callback and processing data. Otherwise, callback and data processing will be stopped.
- The first called back packet is a file header in 40-byte, which is for the following decoding, and then the standard stream (with 12-byte RTP header) will be called back.
- The callback function of standard stream should be supported by device.

12.2.10 NET_DVR_StopRealPlay

Stop live view.

API Definition

```
BOOL NET_DVR_StopRealPlay(  
    LONG lRealHandle  
)
```

Parameters

lRealHandle

[IN] Live view handle, which is returned by [NET_DVR_RealPlay_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.

The following error codes may be returned: 0, 3, 12, 17, 41, 67, and 65. See details in [Device Network SDK Errors](#).

12.3 Search and Playback APIs

12.3.1 NET_DVR_CloseFindPicture

Stop picture search, and release the resources.

API Definition

```
BOOL NET_DVR_CloseFindPicture(  
    LONG lFindHandle  
)
```

Parameters

lFindHandle

[IN] Handle for searching by tag, which is returned by [NET_DVR_FindPicture](#).

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.

12.3.2 NET_DVR_CloseSmartSearchPicture

Stop picture dual-VCA, and release the resources.

API Definition

```
BOOL NET_DVR_CloseSmartSearchPicture(  
    LONG lFindHandle  
)
```

Parameters

lFindHandle

[IN] Handle for picture dual-VCA, which is returned by [NET_DVR_SmartSearchPicture](#).

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.

12.3.3 NET_DVR_FindClose_V30

Stop searching for files and release resources.

API Definition

```
BOOL NET_DVR_FindClose_V30(  
    LONG lFindHandle  
)
```

Parameters

lFindHandle

[IN] Handle for searching files, which is returned by [NET_DVR_FindFile_V50](#) or [NET_DVR_FindFileByEvent_V50](#).

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 are shown as the follows: 0, 3, 12, and 17.

See error details in [Device Network SDK Errors](#).

12.3.4 NET_DVR_FindFile_V50

Search for video files by file type or by time.

API Definition

```
LONG NET_DVR_FindFile_V50(  
    LONG          IUserID,  
    NET_DVR_FILECOND_V50  pFindCond  
)
```

Parameters

IUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

pFindCond

[IN] File information structure to be found, see details in the structure [NET_DVR_FILECOND_V50](#).

Return Values

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

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

The following error codes may be returned by this API: 0, 2, 3, 4, 7, 8, 9, 10, 12, 17, 19, 23, 41, 43, 44, 47, 72, and 73. For error details, refer to [Device Network SDK Errors](#).

Remarks

This API specifies the video files to be found. After calling this API, you can call [NET_DVR_FindNextFile_V50](#) to get the file details.

See Also

[NET_DVR_FindClose_V30](#)

12.3.5 NET_DVR_FindFileByEvent_V50

Search for video files by event.

API Definition

```
LONG NET_DVR_FindFileByEvent_V50(  
    LONG          IUserID,  
    NET_DVR_SEARCH_EVENT_PARAM_V50  lpSearchEventParam  
)
```

Parameters

IUserID

[IN] User ID, which is returned by [NET_DVR_Login_V40](#).

IpSearchEventParam

[IN] Structure of video file information to search, see details in the structure [NET_DVR_SEARCH_EVENT_PARAM_V50](#).

Return Values

Return -1 for failure, and return other value as the handle for playback.

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

Remarks

- After searching video files by event, you can call [NET_DVR_FindNextEvent_V50](#) to get the file information.
- The video files searched by event only support playback by time.

12.3.6 NET_DVR_FindNextEvent_V50

Get video files one by one after searching by event.

API Definition

```
LONG NET_DVR_FindNextEvent_V50(  
    LONG           IFindHandle,  
    NET_DVR_SEARCH_EVENT_RET_V50  IpSearchEventRet  
)
```

Parameters

IFindHandle

[IN] Handle for playback, which is returned by [NET_DVR_FindFileByEvent_V50](#).

IpSearchEventRet

[OUT] Search results, see details in the structure [NET_DVR_SEARCH_EVENT_RET_V50](#).

Return Values

Return -1 for failure, and return other values as the current getting status, see details in the following table.

Status	Value	Description
NET_DVR_FILE_SUCCESS	1000	Getting file information succeeded.
NET_DVR_FILE_NOFILE	1001	No file found.
NET_DVR_ISFINDING	1002	Searching. Please wait.
NET_DVR_NOMOREFILE	1003	No more file found. Search ended.
NET_DVR_FILE_EXCEPTION	1004	Search exception.
NET_DVR_FIND_TIMEOUT	1005	Search timed out.

If -1 is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

Remarks

The video files searched by event only support playback by time.

12.3.7 NET_DVR_FindNextFile_V50

Get video files one by one after searching by time.

API Definition

```
LONG NET_DVR_FindNextFile_V50(
    LONG          IFindHandle,
    NET_DVR_FINDDATA_V50  pFindData
);
```

Parameters

IFindHandle

[IN] Handle for playback, which is returned by [**NET_DVR_FindFile_V50**](#).

pFindData

[OUT] Search results, see details in the structure [**NET_DVR_FINDDATA_V50**](#).

Return Values

Return -1 for failure, and return other values as the current getting status, see details in the following table.

Status	Value	Description
NET_DVR_FILE_SUCCESS	1000	Getting file information succeeded.
NET_DVR_FILE_NOFILE	1001	No file found.

Status	Value	Description
NET_DVR_ISFINDING	1002	Searching. Please wait.
NET_DVR_NOMOREFILE	1003	No more file found. Search ended.
NET_DVR_FILE_EXCEPTION	1004	Search exception.
NET_DVR_FIND_TIMEOUT	1005	Search timed out.

If -1 is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

Remarks

- If you want to get all searched video files, you should repeatedly call this API.
- You can also get the linked card No. of current video file and the information of whether the file is locked via this API.
- Up to 4000 files can be searched for once.

12.3.8 NET_DVR_FindNextLabel

Get the searched video tags one by one.

API Definition

```
LONG NET_DVR_FindNextLabel(
    LONG          IFindHandle,
    NET_DVR_FINDLABEL_DATA *IpFindData
);
```

Parameters

IFindHandle

[IN] Handle for searching by tag, which is returned by [**NET_DVR_FindRecordLabel**](#).

IpFindData

[OUT] Pointer of searched video tags, refer to the structure for details.

Return Values

Return -1 for failure, and return other value as the getting status, see details in the table below.

Status	Value	Description
NET_DVR_FILE_SUCCESS	1000	Getting tag information succeeded.
NET_DVR_FILE_NOFIND	1001	No tag found.

Status	Value	Description
NET_DVR_ISFINDING	1002	Searching. Please wait.
NET_DVR_NOMOREFILE	1003	No more tag found. Search ended.
NET_DVR_FILE_EXCEPTION	1004	Search exception.
NET_DVR_FIND_TIMEOUT	1005	Search timed out.

If -1 is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

Remarks

You should repeatedly call this API if you want to get all searched video tags.

12.3.9 NET_DVR_FindNextPicture_V50

Get the searched pictures one by one.

API Definition

```
LONG NET_DVR_FindNextPicture_V50(
    LONG          IFindHandle,
    NET_DVR_FIND_PICTURE_V50  *IpFindData
);
```

Parameters

IFindHandle

[IN] Handle for searching for pictures, which is returned by [**NET_DVR_FindPicture**](#) .

IpFindData

[OUT] Pointer of searched pictures, refer to the structure for details.

Return Values

Return -1 for failure, and return other value as the getting status, see details in the table below.

Status	Value	Description
NET_DVR_FILE_SUCCESS	1000	Getting picture information succeeded.
NET_DVR_FILE_NOFIND	1001	No picture found.
NET_DVR_ISFINDING	1002	Searching. Please wait.
NET_DVR_NOMOREFILE	1003	No more picture found. Search ended.

Status	Value	Description
NET_DVR_FILE_EXCEPTION	1004	Search exception.
NET_DVR_FIND_TIMEOUT	1005	Search timed out.

If -1 is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

Remarks

You should repeatedly call this API if you want to get all searched pictures.

12.3.10 NET_DVR_FindNextSmartPicture_V50

Get the results of picture dual-VCA one by one.

API Definition

```
LONG NET_DVR_FindNextSmartPicture(
    LONG           IFindHandle,
    NET_DVR_SMART_SEARCH_PICTURE_RET_V50 *IpFindData
);
```

Parameters

IFindHandle

[IN] Handle for picture dual-VCA, which is returned by [**NET_DVR_SmartSearchPicture**](#).

IpFindData

[OUT] Pointer of picture dual-VCA results, refer to the structure for details.

Return Values

Return -1 for failure, and return other value as the getting status, see details in the table below.

Status	Value	Description
NET_DVR_FILE_SUCCESS	1000	Getting results succeeded.
NET_DVR_FILE_NOFIND	1001	No results.
NET_DVR_ISFINDING	1002	Executing. Please wait.
NET_DVR_NOMOREFILE	1003	No more result. Dual-VCA ended.
NET_DVR_FILE_EXCEPTION	1004	Dual-VCA exception.
NET_DVR_FIND_TIMEOUT	1005	Search timed out.

If -1 is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

Remarks

You should repeatedly call this API if you want to get all picture dual-VCA results.

12.3.11 NET_DVR_FindPicture

Search for pictures by type and time.

API Definition

```
LONG NET_DVR_FindPicture(  
    LONG          IUserID,  
    NET_DVR_FIND_PICTURE_PARAM *pFindParam  
)
```

Parameters

IUserID

[IN] User ID, which is returned by [NET_DVR_Login_V40](#).

pFindParam

[IN] Pointer of picture search conditions, refer to the structure for details.

Return Values

Return -1 for failure, and return other value as the parameter of [NET_DVR_FindNextPicture_V50](#) or [NET_DVR_CloseFindPicture](#).

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

Remarks

- This API is only used to search for the pictures in the local storage.
- After searching pictures, you can call [NET_DVR_FindNextPicture_V50](#) to get the search results.

12.3.12 NET_DVR_FindRecordLabel

Search for video tags.

API Definition

```
LONG NET_DVR_FindRecordLabel(  
    LONG          IUserID,  
    NET_DVR_FIND_LABEL *lpFindLabel  
)
```

Parameters

IUserID

[IN] User ID, which is returned by [NET_DVR_Login_V40](#).

IpFindLabel

[IN] Pointer of video tag information to search, refer to the structure for details.

Return Values

Return -1 for failure, and return other value as the parameter of [NET_DVR_FindNextLabel](#) or [NET_DVR_StopFindLabel](#).

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

Remarks

Up to 4000 video tags can be searched at one time.

12.3.13 NET_DVR_GetPicture_V50

Get the picture data.

API Definition

```
BOOL NET_DVR_GetPicture_V50(
    LONG      IUserID,
    NET_DVR_PIC_PARAM  IpPicParam
);
```

Parameters

IUserID

[IN] User ID, which is returned by [NET_DVR_Login_V40](#).

IpPicParam

[IN] Picture parameters, see details in the structure [NET_DVR_PIC_PARAM](#).

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.

12.3.14 NET_DVR_PlayBackByName

Playback by file.

API Definition

```
LONG NET_DVR_PlayBackByName(  
    LONG   lUserID,  
    char  *sPlayBackFileName,  
    HWND   hWnd  
)
```

Parameters

lUserID

Value returned by [NET_DVR_Login_V40](#).

sPlayBackFileName

[IN] Video file with specific name for playback, and up to 100 bytes are allowed by file name.

hWnd

[IN] Playback window handle. If it is set to null, the stream data is still received, but the data will not be decoded and displayed.

Return Values

Return -1 for failure, and return other values as the parameters of [NET_DVR_StopPlayBack](#).

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

The error codes may be returned by this API are shown as the follows: 0, 1, 3, 5, 7, 8, 9, 10, 14, 17, 33, 41, 43, 44, 47, 64, 65, 66, 72, and 73. See error descriptions in [Device Network SDK Errors](#).

Remarks

- This API only specifies the video files to play, so after calling this API, you must call [NET_DVR_PlayBackControl_V40](#) with the command of NET_DVR_PLAYSTART to start playback.
- After successfully calling this API, you can call [NET_DVR_SetPlayDataCallBack_V40](#) to register the callback function to get the stream data of video files and process the data by yourself as needed.

12.3.15 NET_DVR_PlayBackByTime_V50

Start playback by time.

API Definition

```
LONG NET_DVR_PlayBackByTime_V50(  
    LONG      lUserID,  
    NET_DVR_VOD_PARA_V50 *lpVodParam  
)
```

Parameters

IUserID

[IN] User ID, which is returned by [NET_DVR_Login_V40](#).

IpVodParam

[IN] Pointer of playback by time parameters, refer to the structure for details.

Return Values

Return -1 for failure, and return other value as the handle of [NET_DVR_PlayBackControl_V40](#) or [NET_DVR_StopPlayBack](#).

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

12.3.16 NET_DVR_PlayBackControl_V40

Control playback or video file downloading.

API Definition

```
BOOL NET_DVR_PlayBackControl_V40(
    LONG   IPlayHandle,
    DWORD  dwControlCode,
    LPVOID lpInBuffer,
    DWORD  dwInLen,
    LPVOID lpOutBuffer,
    DWORD  *lpOutLen
);
```

Parameters

IPlayHandle

[IN] Handle for playback or downloading, which is returned by [NET_DVR_PlayBackByTime_V50](#), [NET_DVR_PlayBackByName](#), [NET_DVR_GetFileByName](#), or [NET_DVR_GetFileByTime_V40](#).

dwControlCode

[IN] Control commands for playback or downloading, and different commands determine different input and output parameters, see details in the following table:

Command	Command No.	Description
NET_DVR_PLAYSTART	1	Start playback or downloading.
NET_DVR_PLAYPAUSE	3	Pause playback or downloading.

Command	Command No.	Description
NET_DVR_PLAYRESTART	4	Resume playback or downloading (playback speed will also resume)
NET_DVR_PLAYFAST	5	Fast forward (playback).
NET_DVR_PLAYSLOW	6	Slow forward (playback).
NET_DVR_PLAYNORMAL	7	Play.
NET_DVR_PLAYFRAME	8	Playback in single frame.
NET_DVR_PLAYSTARTAUDIO	9	Audio on (playback).
NET_DVR_PLAYSTOPAUDIO	10	Mute.
NET_DVR_PLAYAUDIOVOLUME	11	Adjust volume during playback, the value is ranging from 0 to 0xffff.
NET_DVR_PLAYSETPOS	12	Locate the playback/downloading position (not available for downloading by time).
NET_DVR_PLAYGETPOS	13	Get playback or downloading progress by file or by time.
NET_DVR_PLAYGETTIME	14	Get current played time (valid only when playback by file).
NET_DVR_PLAYGETFRAME	15	Get current played frames number (valid only when playback by file).
NET_DVR_GETTOTALFRAMES	16	Get the total frame number of the video (valid only when playback or downloading by file).
NET_DVR_GETTOTALTIME	17	Get the total time duration of the video (valid only when playback or downloading by file).
NET_DVR THROWBFRAME	20	Drop B frame.
NET_DVR_SETSPEED	24	Set playback or downloading speed, unit: kbps, minimum: 256 kbps, maximum: device bandwidth.
NET_DVR_KEEPALIVE	25	Keep the heartbeat with device (if the callback blocked, send once per two seconds).
NET_DVR_PLAYSETTIME	26	Locate according to absolute time.

Command	Command No.	Description
NET_DVR_PLAYGETTALLEN	27	Get the size of all video files in the time period for playback.
NET_DVR_PLAYSETTIME_V50	28	Locate time during playback.
NET_DVR_PLAY_FORWARD	29	Switch reverse playback to playback.
NET_DVR_PLAY_REVERSE	30	Switch playback to reverse playback.
NET_DVR_SET_TRANS_TYPE	32	Set transcoding type (call before NET_DVR_PLAYSTART).
NET_DVR_PLAY_CONVERT	33	Switch streams during playback or downloading (call before NET_DVR_PLAYSTART).
NET_DVR_START_DRAWFRAME	34	Start playback by extracting frame.
NET_DVR_STOP_DRAWFRAME	35	Stop playback by extracting frame.
NET_DVR_CHANGEWNDRESOLUTION	36	Send notification to player SDK when the window size changed (only available for developing in Linux system).
NET_DVR_VOD_DRAG_ING	37	Dragging
NET_DVR_VOD_DRAG_END	38	Dragging ended.

IplInBuffer

[IN] Pointer of input parameters, which are different according to different commands, see details in the table below:

Command	IplInBuf	IpOutBuf
NET_DVR_PLAYSTART	A 4-byte integer offset. If the offset is 0,it indicates the video playback or downloading starts from the initial position, otherwise, the playback or downloading starts	None.

Command	IpInBuf	IpOutBuf
	from the position that ANR occurred.	
NET_DVR_PLAYPAUSE	None.	None.
NET_DVR_PLAYRESTART	None.	None.
NET_DVR_PLAYFAST	None.	None.
NET_DVR_PLAYSLOW	None.	None.
NET_DVR_PLAYNORMAL	None.	None.
NET_DVR_PLAYFRAME	None.	None.
NET_DVR_PLAYSTARTAUDIO	None.	None.
NET_DVR_PLAYSTOPAUDIO	None.	None.
NET_DVR_PLAYAUDIOVOLUME	None.	None.
NET_DVR_PLAYSETPOS	A 4-byte integer progress value (from 0 to 100).	None.
NET_DVR_PLAYGETPOS	None.	A 4-byte integer progress value. If the value is ranging from 0 to 100, it refers to normal playback or downloading; if the value is larger than 100, it refers to playback or downloading exception. For playback or downloading by time, only the following progress value can be obtained: 0, 100 (ended), and 200 (exception).
NET_DVR_PLAYGETTIME	None.	A 4-byte integer time.
NET_DVR_PLAYGETFRAME	None.	A 4-byte integer frame number.
NET_DVR_GETTOTALFRAMES	None.	A 4-byte integer frame number.
NET_DVR_GETTOTALTIME	None.	A 4-byte integer time.
NET_DVR_THROWBFRAME	A 4-byte integer B frame number.	None.
NET_DVR_SETSPEED	A 4-byte integer speed.	None.

Command	IpInBuf	IpOutBuf
NET_DVR_KEEPALIVE	None.	None.
NET_DVR_PLAYSETTIME		None.
NET_DVR_PLAYGETTOTALEN	None.	An 8-byte integer length (unit: byte).
NET_DVR_PLAYSETTIME_V50		None.
NET_DVR_PLAY_FORWARD	Decode on application layer: ; Decode via SDK API: null	None.
NET_DVR_PLAY_REVERSE	Decode on application layer: ; Decode via SDK API: null	None.
NET_DVR_SET_TRANS_TYPE	A 4-byte integer packaging type: 1-PS, 2-TS, 3-RTP, 5-MP4 (3GPP, only available for playback by file).	None.
NET_DVR_PLAY_CONVERT		None.
NET_DVR_START_DRAWFRAME	or	None.
NET_DVR_STOP_DRAWFRAME		None.
NET_DVR_CHANGEWNDRESOLUTION	None.	None.
NET_DVR_VOD_DRAG_ING		None.
NET_DVR_VOD_DRAG_END		None.

dwInLen

[IN] Size of input parameter (when playback by extracting frame, you must set the correct length). This parameter is reserved during downloading control.

IpOutBuffer

[OUT] Pointer of output parameter.

IpOutLen

[OUT] Size of output parameters, which are different according to different commands, see details in the table above.

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 error codes may be returned by this API are shown as the follows: 0, 1, 3, 5, 7, 8, 9, 10, 14, 17, 23, 33, 41, 43, 44, 64, 65, 66, 72, and 73, see details in [***Device Network SDK Errors***](#).

Remarks

- For the integration based on OPEN ISAPI Protocol, only the following playback control commands are supported: NET_DVR_PLAYSTART (start playing), NET_DVR_PLAYPAUSE (pausing playing), NET_DVR_PLAYRESTART (resume playing), NET_DVR_PLAYFAST (fast forward), NET_DVR_PLAYSLOW (slow forward), NET_DVR_PLAYNORMAL (play in 1x speed).
- Before starting or stopping playback by extracting frame, you must send the current OSD time to device.
- If no window handle is imported by playback API, clearing the cache of PlayCtrl LibrarySDK to avoid delay is suggested when getting steam and decoding.
- For developing in Linux system, valid window handle will be imported by the playback API, so you should call this API with command NET_DVR_CHANGEWNDRESOLUTION to notify the PlayCtrl LibrarySDK to get the window size when it changed during playback.

12.3.17 NET_DVR_SearchNextInfo

Get the searched videos with VCA information one by one.

API Definition

```
LONG NET_DVR_SearchNextInfo(  
    LONG           ISearchHandle,  
    NET_DVR_SMART_SEARCH_RET  lpSmartSearchRet  
)
```

Parameters

ISearchHandle

[IN] Search handle, which is returned by [***NET_DVR_SmartSearch_V40***](#).

lpSmartSearchRet

[OUT] Pointer for saving the search results, refer to the structure for details.

Return Values

Return -1 for failure, and return other value as the getting status, see details in the table below.

Status	Value	Description
NET_DVR_FILE_SUCCESS	1000	Getting file information succeeded.
NET_DVR_FILE_NOFILE	1001	No file found.
NET_DVR_ISFINDING	1002	Searching. Please wait.
NET_DVR_NOMOREFILE	1003	No more file found. Search ended.
NET_DVR_FILE_EXCEPTION	1004	Search exception.

If -1 is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

Remarks

- Before calling this API, you must call [**NET_DVR_SmartSearch_V40**](#) to get the search handle.
- The searched videos can only be played back by time.

12.3.18 NET_DVR_SetPlayDataCallBack_V40

Sign up the callback function for getting the video data.

API Definition

```
BOOL NET_DVR_SetPlayDataCallBack_V40(
    LONG          IPlayHandle,
    fPlayDataCallBack_V40  cbPlayDataCallBack,
    void         *pUser
);
```

Parameters

IPlayHandle

[IN] Handle for playback, which is returned by [**NET_DVR_PlayBackByName**](#) or [**NET_DVR_PlayBackByTime_V50**](#).

fPlayDataCallBack

[IN] Video data callback function, see details in [**fPlayDataCallBack_V40**](#).

pUser

[IN] User data.

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 error codes may be returned by this API are shown as the follows: 0 and 3. See error descriptions in [**Device Network SDK Errors**](#).

Remarks

- Operations with long time consumption cannot be performed in the callback function.
- This API provides the function of starting/stopping callback and processing data. When the callback function is set as non-null, it will start callback and processing data. Otherwise, callback and data processing will be stopped.
- The first called back packet is a file header in 40-byte, which is for the following decoding, and then the encoded stream will be called back.

12.3.19 NET_DVR_SmartSearch_V40

Start VCA search.

API Definition

```
LONG NET_DVR_SmartSearch_V40(  
    LONG          IUserID,  
    NET_DVR_SMART_SEARCH_PARAM_V40  *IpSmartSearchParams  
)
```

Parameters

IUserID

[IN] User ID, which is returned by [NET_DVR_Login_V40](#).

IpSmartSearchParams

[IN] VCA search parameter structure, refer to for details.

Return Values

Return -1 for failure, and return other value as the parameter of [NET_DVR_SearchNextInfo](#) or [NET_DVR_StopSearch](#).

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

12.3.20 NET_DVR_SmartSearchPicture

Start picture dual-VCA.

API Definition

```
LONG NET_DVR_FindPicture(  
    LONG          IUserID,  
    NET_DVR_SMART_SEARCH_PICTURE_PARA  *pFindParam  
)
```

Parameters

IUserID

[IN] User ID, which is returned by [NET_DVR_Login_V40](#).

pFindParam

[IN] Pointer of picture dual-VCA conditions, refer to the structure for details.

Return Values

Return -1 for failure, and return other value as the parameter of [NET_DVR_FindNextSmartPicture_V50](#) or [NET_DVR_CloseSmartSearchPicture](#). If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

Remarks

After searching pictures, you can call [NET_DVR_FindNextSmartPicture_V50](#) to get the search results.

12.3.21 NET_DVR_StopFindLabel

Stop searching for video tags.

API Definition

```
BOOL NET_DVR_StopFindLabel(  
    LONG lFindHandle  
)
```

Parameters

lFindHandle

[IN] Handle for searching by tag, which is returned by [NET_DVR_FindRecordLabel](#).

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.

12.3.22 NET_DVR_StopPlayBack

Stop playback.

API Definition

```
BOOL NET_DVR_StopPlayBack(  
    LONG IPlayHandle  
)
```

Parameters

IPlayHandle

[IN] Handle for playback, which is returned by [NET_DVR_PlayBackByName](#) or [NET_DVR_PlayBackByTime_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 error codes may be returned by this API are shown as the follows: 0, 3, 12, 17, 41, 65, and 67.

See error descriptions in [Device Network SDK Errors](#).

12.3.23 NET_DVR_StopSearch

Stop VCA search.

API Definition

```
BOOL NET_DVR_StopSearch(  
    LONG ISearchHandle  
)
```

Parameters

ISearchHandle

[IN] Search handle, which is returned by [NET_DVR_SmartSearch_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.

The following error codes may be returned by this API: 0, 3, 12, and 17. For details, refer to [Device Network SDK Errors](#).

12.3.24 NET_DVR_UpdateRecordIndex

Refresh the video index manually.

API Definition

```
BOOL NET_DVR_UpdateRecordIndex(
    LONG   lUserID,
    DWORD  dwChannel
);
```

Parameters

lUserID

[IN] User ID, which is returned by [NET_DVR_Login_V40](#).

dwChannel

Device channel No.

Returned 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.

12.4 File Uploading and Downloading APIs

12.4.1 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.

12.4.2 NET_DVR_GetFileByName

Download video files by file name.

API Definition

```
LONG NET_DVR_GetFileByName(  
    LONG  lUserID,  
    char  *sDVRFileName,  
    char  *sSavedFileName  
)
```

Parameters

lUserID

Value returned by [**NET_DVR_Login_V40**](#).

sDVRFileName

[IN] Video file with specific name for downloading, and up to 100 bytes are allowed by file name.

sSavedFileName

[IN] Local saving path of downloaded files, it should be a absolute path (including file name).

Return Values

Returns -1 for failure, and returns other values as the parameters of [**NET_DVR_StopGetFile**](#).

If -1 is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

The error codes may be returned by this API are shown as the follows: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 17, 23, 33, 34, 41, 43, 44, 47, 72, 73, and 76. See error descriptions in [**Device Network SDK Errors**](#).

Remarks

- You can call the API [**NET_DVR_FindFile_V50**](#) to search video files before downloading the files.
- This API only specifies the video files to download, so after calling this API, you must call [**NET_DVR_PlayBackControl_V40**](#) with [**NET_DVR_PLAYSTART**](#) (command No.: 1) to start downloading.

12.4.3 NET_DVR_GetFileByTime_V40

Download video files by time.

API Definition

```
LONG NET_DVR_GetFileByTime_V40(
    LONG      lUserID,
    char     *sSavedFileName,
    LPNET_DVR_PLAYCOND pDownloadCond
);
```

Parameters

lUserID

Value returned by [NET_DVR_Login_V40](#).

sSavedFileName

[IN] Local saving path of downloaded files, it should be a absolute path (including file name).

pDownloadCond

[IN] Downloading condition, see details in the structure [NET_DVR_PLAYCOND](#).

Return Values

Return -1 for failure, and return other values as the parameters of [NET_DVR_StopGetFile](#).

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

The error codes may be returned by this API are shown as the follows: 0, 1, 3, 5, 7, 8, 9, 10, 14, 17, 34, 41, 43, 44, 47, 72, and 73. See error descriptions in [Device Network SDK Errors](#).

Remarks

- This API only specifies the video files to download, so after calling this API, you must call [NET_DVR_PlayBackControl_V40](#) with the command of NET_DVR_PLAYSTART to start downloading.
- If the size of saved video file is larger than the maximum limitation (by default, it is 1024 MB), the video will be automatically separated into multiple video segments and save as multiple files. The naming rule is to add a numeric identifier based on the original file name for each file segment (i.e., *_1.mp4, *_2.mp4). You can also call [NET_DVR_GetSDKLocalCfg](#) and [NET_DVR_SetSDKLocalCfg](#) (configuration type: NET_DVR_LOCAL_CFG_TYPE_GENERAL) to get and set the separation mode and maximum limitation.

12.4.4 NET_DVR_GetUploadResult

Get the uploaded data information.

API Definition

```
BOOL NET_DVR_GetUploadResult(
    LONG      lUploadHandle,
    LPVOID   lpOutBuffer,
```

```
    DWORD dwOutBufferSize  
);
```

Parameters

IUploadHandle

[IN] Handling for uploading files, which is returned by [NET_DVR_UploadFile_V40](#).

lpOutBuffer

[OUT] Buffer of uploaded data information, different uploading types (**dwUploadType**) correspond to different information.

dwOutBufferSize

[OUT] Buffer size

Return Value

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

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

12.4.5 NET_DVR_GetUploadState

Get the file uploading progress and status.

API Definition

```
LONG NET_DVR_GetUploadState(  
    LONG   IUploadHandle,  
    DWORD  *pProgress  
)
```

Parameters

IUploadHandle

[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 12-2 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.
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.

Return Value	Description
37	Person ID error (The ID is saved in customHumanID of FaceAppendData).
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.

12.4.6 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.

12.4.7 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.

12.4.8 NET_DVR_StopGetFile

Stop downloading video files.

API Definition

```
BOOL NET_DVR_StopGetFile(  
    LONG  lFileHandle  
)
```

Parameters

lFileHandle

[IN] Handle for downloading, which is returned by [NET_DVR_GetFileByName](#) or [NET_DVR_GetFileByTime_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.

The error codes may be returned by this API are shown as the follows: 0, 3, 12, 17, and 41. See error descriptions in [***Device Network SDK Errors***](#).

12.4.9 ***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.

12.4.10 ***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

lUserID

[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.

12.4.11 NET_DVR_UploadSend

Send data.

API Definition

```
LONG NET_DVR_UploadSend(
    LONG          IUploadHandle,
    LPNET_DVR_SEND_PARAM_IN pstruSendParamIN,
    void         *lpOutBuffer
);
```

Parameters

IUploadHandle

[IN] File uploading handle, returned by [NET_DVR_UploadFile_V40](#).

pstruSendParamIN

[IN] Uploaded data. See the structure [NET_DVR_SEND_PARAM_IN](#) for details.

lpOutBuffer

[OUT] Output parameter, reserved, set to NULL

Return Value

Return 0 for success, and return -1 for failure.

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

12.5 Alarm APIs

12.5.1 NET_DVR_CloseAlarmChan_V30

Close alarm uploading channel.

API Definition

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

Parameters

lAlarmHandle

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](#).

12.5.2 NET_DVR_GetAlarmSubscribe

Get the event/alarm subscription parameters.

API Definition

```
BOOL NET_DVR_GetAlarmSubscribe(  
    LONG     lAlarmHandle,  
    char     *pData,  
    DWORD    dwDataLen  
)
```

Parameters

IAlarmHandle

[IN] Value returned by [NET_DVR_SetupAlarmChan_V50](#)

pData

[OUT] Pointer to data buffer, see details in [XML_SubscribeEvent](#)

dwDataLen

[IN] Size of data buffer, unit: byte, it cannot be 0.

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.

12.5.3 NET_DVR_SetAlarmSubscribe

Set the event/alarm subscription parameters.

API Definition

```
BOOL NET_DVR_SetAlarmSubscribe(  
    LONG   IAlarmHandle,  
    char   *pData,  
    DWORD   dwDataLen  
)
```

Parameters

IAlarmHandle

[IN] Value returned by [NET_DVR_SetupAlarmChan_V50](#)

pData

[IN] Pointer to data buffer, see details in [XML_SubscribeEvent](#)

dwDataLen

[IN] Size of data buffer, unit: byte

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.

12.5.4 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 function via message mode (PostMessage).

```

switch(ICommand)
{
    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 ((struTSPlateResult.struPicInfo[i].dwDataLen != 0)&&(struTSPlateResult.struPicInfo[i].byType == 0))
    {
        sprintf(filename,"testPicPlate%d_%d.jpg",iNum,i);
        fSnapPicPlate=fopen(filename,"wb");
        fwrite(struTSPlateResult.struPicInfo[i].pBuffer, struTSPlateResult.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_SETUPALARM_PARAM struSetupParam={0};
struSetupParam.dwSize=sizeof(NET_DVR_SETUPALARM_PARAM);

//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](#)

12.5.5 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          lUserID,  
    NET_DVR_SETUPALARM_PARAM_V50  lpSetupParam,  
    char          *pData,  
    DWORD         dwDataLen,  
)
```

Parameters

lUserID

[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**.

12.5.6 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**).

12.5.7 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](#).

12.6 Two-Way Audio APIs

12.6.1 NET_DVR_AddDVR_V30

Add the audio channel of a device to the broadcast group for receiving the audio from PC.

API Definition

```
LONG NET_DVR_AddDVR_V30(  
    LONG  lUserID,  
    DWORD dwVoiceChan  
>;
```

Parameters

lUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

dwVoiceChan

[IN] Audio channel No., which starts from 1.

Return Values

Return -1 for failure, and return other values as the parameters of [NET_DVR_DelDVR_V30](#).

If -1 is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

The error codes may be returned by the API are as the follows: 0, 3, 7, 8, 9, 10, 12, 17, 41, and 47. See the error descriptions in [**Device Network SDK Errors**](#).

Remarks

- Before you can realize the audio broadcast function, you should call [**NET_DVR_ClientAudioStart_V30**](#) to collect the audio data from the local PC, and then call this API to add the device one by one and send the collected data to device.
- Up to 512 devices can be added.

12.6.2 NET_DVR_ClientAudioStart_V30

Enable collecting audio from PC for the audio broadcast.

API Definition

```
BOOL NET_DVR_ClientAudioStart_V30(
    fVoiceDataCallBack cbVoiceDataCallBack,
    void             *pUser
);
```

Parameters

cbVoiceDataCallBack

[IN] Audio data callback function, see details in [**fVoiceDataCallBack**](#).

pUser

[IN] Pointer of user data.

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 error codes may be returned by the API are as the follows: 0 and 12. See the error descriptions in [**Device Network SDK Errors**](#).

Remarks

- This API is not supported when programming in Linux operating system.
- When programming in Windows 7 operating system, if there is no external audio device, this API will return *FALSE*.
- Before you can realize the audio broadcast function, you should call this API to collect the audio data from the local PC, and then call [**NET_DVR_AddDVR_V30**](#) to add the device one by one and send the collected data to device.

See Also

[NET_DVR_ClientAudioStop](#)

12.6.3 NET_DVR_ClientAudioStop

Stop collecting audio from PC for audio broadcast.

API Definition

```
BOOL NET_DVR_ClientAudioStop(  
);
```

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 error codes may be returned by the API are as the follows: 0 and 12. See the error descriptions in [Device Network SDK Errors](#).

See Also

[NET_DVR_ClientAudioStart_V30](#)

12.6.4 NET_DelDVR_V30

Remove the audio channel of the device from the broadcast group.

API Definition

```
BOOL NET_DVR_DelDVR_V30(  
    LONG  lVoiceHandle  
);
```

Parameters

lVoiceHandle

[IN] Value returned by [NET_DVR_AddDVR_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.

12.6.5 NET_DVR_GetCurrentAudioCompress_V50

Get the effective audio encoding parameters of two-way audio.

API Definition

```
BOOL NET_DVR_GetCurrentAudioCompress_V50(
    LONG          lUserID,
    LPNET_DVR_AUDIO_CHANNEL *lpAudioChannel,
    LPNET_DVR_COMPRESSION_AUDIO lpCompressAudio
);
```

Parameters

lUserID

[IN] User information returned by [NET_DVR_Login_V40](#).

lpAudioChannel

[IN] Audio channel information, refer to the structure [NET_DVR_AUDIO_CHANNEL](#) for details.

lpCompressAudio

[OUT] Audio encoding parameters, refer to the structure [NET_DVR_COMPRESSION_AUDIO](#) for details.

Return Value

Return *TRUE* for success, and return *FALSE* for failure.

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

12.6.6 NET_DVR_EncodeMP3Frame

Encode MP3 audio data.

API Definition

```
BOOL NET_DVR_EncodeMP3Frame(
    unsigned int iType,
    unsigned char *pInBuffer,
    unsigned char *pOutBuffer
);
```

Parameters

iType

[IN] Audio encoding type.

pInBuffer

[IN] Input buffer.

pOutBuffer

[OUT] Output buffer.

Return Value

Return *TRUE* for success, and return *FALSE* for failure.

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

See Also

[***NET_DVR_DecodeMP3Frame***](#)

12.6.7 NET_DVR_DecodeMP3Frame

Decode MP3 audio data.

API Definition

```
BOOL NET_DVR_DecodeMP3Frame(  
    void      *pDecHandle,  
    unsigned char *pInBuffer,  
    unsigned char *pOutBuffer  
)
```

Parameters

pDecHandle

Audio decoding handle returned by [***NET_DVR_EncodeMP3Frame***](#).

pInBuffer

[IN] Input buffer.

pOutBuffer

[OUT] Output buffer.

Return Value

Return *TRUE* for success, and return *FALSE* for failure.

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

12.6.8 NET_DVR_InitG711Encoder

Initialize resources for G.711 audio encoding. This API is available for Linux operating system only.

API Definition

```
void* NET_DVR_InitG711Encoder(  
    NET_DVR_AUDIOENC_INFO *enc_info  
)
```

Parameters

enc_info

[OUT] Encoding information, see details in the structure [NET_DVR_AUDIOENC_INFO](#).

Return Values

Return -1 for failure, return other values as the handles for audio encoding, or return *NULL* for success.

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

See Also

[NET_DVR_ReleaseG711Encoder](#)
[NET_DVR_EncodeG711Frame](#)

12.6.9 NET_DVR_EncodeG711Frame

Encode audio in G.711 mode.

API Definition (For 32-Bit Windows Operating System)

```
BOOL NET_DVR_EncodeG711Frame(  
    unsigned int iType,  
    unsigned char *pInBuffer,  
    unsigned char *pOutBuffer  
)
```

Parameters (For 32-Bit Windows Operating System)

iType

[IN] Encoding algorithm: 0- μ -law algorithm, non-0-A-law algorithm.

pInBuffer

[IN] Input buffer, which is used to save the audio data in PCM format that sampled according to the sampling standard (sampling frequency is 16000, sample in 16-bit, single channel), and the data size is 320 bytes.

pOutBuffer

[OUT] Output buffer, the size of output data after encoding is 160 bytes.

API Definition (For Linux Operating System and 64-Bit Windows Operating System)

```
BOOL NET_DVR_EncodeG711Frame(  
    void *handle,  
    NET_DVR_AUDIOENC_PROCESS_PARAM *p_enc_proc_param  
)
```

Parameters (For Linux Operating System and 64-Bit Windows Operating System)

handle

[IN] Handle for audio encoding, which is returned by [NET_DVR_InitG711Encoder](#).

p_enc_proc_param

[IN][OUT] Encoding parameters, see details in the structure

[NET_DVR_AUDIOENC_PROCESS_PARAM](#).

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

- This API is mainly for two-way audio function. You can encode the original audio data by this API and send the encoded audio data to the platform or system. And then the [NET_DVR_DecodeG711Frame](#) can be called to decode the data after receiving the encoded audio data.
- Initialization is required before call this API, and after calling this API, you should also release the resource.

See Also

[NET_DVR_ReleaseG711Encoder](#)

Example

G.711 Encoding and Decoding (For Linux Operating System and 64-Bit Windows Operating System)

```
void Test_VoiceTalk_DecEncG711()  
{  
    printf("Test_VoiceTalk_DecEncG711\n");  
    const unsigned int AUDIO_CODEC_BUF_SIZE = 8192;  
    const unsigned int AUDIO_DEC_DATA_SIZE = 2048;  
    //*****Encode Audio in G.711 Mode*****/  
    LPVOID hEncoderInstance=0;  
    //Encoding information structure  
    NET_DVR_AUDIOENC_INFO enc_info;  
    //Initialize resources for encoding  
    hEncoderInstance = NET_DVR_InitG711Encoder(&enc_info);  
    printf("enc_info.in_frame_size = %d", enc_info.in_frame_size);  
  
    FILE *out_file;
```

```

FILE *in_file;
unsigned char *indata_buf;
unsigned char *outdata_buf;
int frame = 0;
char *inputfilename = "tst_8k_ch_male.wav";
//Input audio file with WAV format
char *outputfilename = "temp.g711";
//Output the encoded audio file
if ((in_file = fopen(inputfilename,"rb"))==0)
{
    printf("error opening output file %s\n",inputfilename);
    exit(99);
}
if ((out_file = fopen(outputfilename, "wb"))==0)
{
    printf("error opening output file %s\n",outputfilename);
    exit(99);
}

NET_DVR_AUDIOENC_PROCESS_PARAM enc_param;
memset(&enc_param, 0 ,sizeof(NET_DVR_AUDIOENC_PROCESS_PARAM));

indata_buf = (unsigned char *)malloc(AUDIO_CODEC_BUF_SIZE);
outdata_buf = (unsigned char *)malloc(AUDIO_CODEC_BUF_SIZE);
memset(indata_buf, 0, AUDIO_CODEC_BUF_SIZE * sizeof(unsigned char));
memset(outdata_buf, 0, AUDIO_CODEC_BUF_SIZE * sizeof(unsigned char));

enc_param.g711_type = 0;
enc_param.in_buf = indata_buf;
enc_param.out_buf = outdata_buf;

fseek(in_file, 44, SEEK_SET);
//Delete the header of WAV file

while(1)
{
    int num_read;
    //Read the file according to the frame size in encoding information structure. The input buffer size should be
    larger than this size.
    num_read = fread(indata_buf, sizeof(char), enc_info.in_frame_size ,in_file);
    if(num_read==0) {
        fclose(in_file);
        fclose(out_file);
        break;
    }

    BOOL ret = NET_DVR_EncodeG711Frame(hEncoderInstance, &enc_param);
    if(ret==FALSE)
    {
        printf("NET_DVR_EncodeG711Frame fail,err %d!\n", NET_DVR_GetLastError());
    }
    else {

```

```
int num_write;
//The enc_param.out_frame_size is the size of encoded output data, and the output buffer must be larger than
this size.
num_write = fwrite(outdata_buf,1,enc_param.out_frame_size,out_file);
printf("enc_param.out_frame_size = %d", enc_param.out_frame_size);
frame++;
printf("g711 Enc frame\n",frame);
}
}
//Release resources
BOOL ret = NET_DVR_ReleaseG711Encoder(hEncoderInstance);
if(ret==FALSE)
{
    printf("NET_DVR_ReleaseG711Encoder fail,err %d!\n", NET_DVR_GetLastError());
}

/******Decode Audio Data for G.711 Encoding*****/
LPVOID hDeclnstance = 0;
hDeclnstance = NET_DVR_InitG711Decoder();

char *inputfilename2 = "temp.g711";
char *outputfilename2 = "g711_ret.pcm";
if ((in_file = fopen(inputfilename2,"rb"))==0)
{
    printf("error opening output file %s\n",inputfilename2);
    exit(99);
}
/*Open the output file for the encoded MP2 data*/
if ((out_file = fopen(outputfilename2, "wb"))==0)
{
    printf("error opening output file %s\n",outputfilename2);
    exit(99);
}
NET_DVR_AUDIODEC_PROCESS_PARAM dec_proc_param;
memset(&dec_proc_param, 0 ,sizeof(NET_DVR_AUDIODEC_PROCESS_PARAM));
memset(indata_buf, 0, AUDIO_CODEC_BUF_SIZE * sizeof(char));
memset(outdata_buf, 0, AUDIO_CODEC_BUF_SIZE * sizeof(char));

dec_proc_param.g711_type = 0;
dec_proc_param.in_buf = indata_buf;
dec_proc_param.out_buf = outdata_buf;
dec_proc_param.in_data_size = 0;

int dwRead = fread(dec_proc_param.in_buf, sizeof(char), AUDIO_DEC_DATA_SIZE, in_file);
dec_proc_param.in_data_size += dwRead;

while(1)
{
    if(dwRead==0) {
        fclose(in_file);
        fclose(out_file);
        break;
    }
}
```

```

    }
    DWORD outsize = 0;
    BOOL err = NET_DVR_DecodeG711Frame(hDecInstance, &dec_proc_param);
    if (err == 1)
    {
        //The dec_proc_param.out_frame_size is decoded output data
        fwrite(dec_proc_param.out_buf, sizeof(char), dec_proc_param.out_frame_size, out_file);
        frame++;
        printf("g711 Dec frame(%d)\n",frame);
        printf("dec_proc_param.proc_data_size(%d)\n",dec_proc_param.proc_data_size);
        printf("dec_proc_param.out_frame_size(%d)\n",dec_proc_param.out_frame_size);
        //The dec_proc_param.proc_data_size is the size of data in the input buffer.
        dec_proc_param.in_data_size = dec_proc_param.in_data_size - dec_proc_param.proc_data_size;
        //Move the data in the inout buffer to overwrite the processed data.
        memmove(dec_proc_param.in_buf, dec_proc_param.in_buf + dec_proc_param.proc_data_size,
        dec_proc_param.in_data_size);

        if (dec_proc_param.in_data_size < AUDIO_DEC_DATA_SIZE )
        {
            dwRead = fread(dec_proc_param.in_buf + dec_proc_param.in_data_size, sizeof(char),
AUDIO_DEC_DATA_SIZE - dec_proc_param.in_data_size, in_file);
            dec_proc_param.in_data_size += dwRead;
        }
    }
    else
    {
        printf("decode err:0x%x", err);
    }
}

ret = NET_DVR_ReleaseG711Decoder(hDecInstance);
if (ret== FALSE)
{
    printf("NET_DVR_ReleaseG711Decoder fail,err %d!\n", NET_DVR_GetLastError());
}
}

```

12.6.10 NET_DVR_ReleaseG711Encoder

Release resources of G.711 audio encoding. This API is only available for Linux operating system.

API Definition

```

BOOL NET_DVR_ReleaseG711Encoder(
    void    *pEncodeHandle
);

```

Parameters

pEncodeHandle

[IN] Handle for audio encoding, which is returned by [NET_DVR_InitG711Encoder](#).

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.

See Also

[NET_DVR_EncodeG711Frame](#)

12.6.11 NET_DVR_InitG711Decoder

Initialize resources to decode audio data for G.711 encoding. This API is available for Linux operating system only.

API Definition

```
void* NET_DVR_InitG711Decoder()  
);
```

Return Values

Return *-1* for failure, and return other values as the handles for audio decoding, or return *NULL* for success.

If *-1* is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

See Also

[NET_DVR_ReleaseG711Decoder](#)
[NET_DVR_DecodeG711Frame](#)

12.6.12 NET_DVR_DecodeG711Frame

Decode the audio to G711 format.

API Definition (For 32-Bit Windows Operating System)

```
BOOL NET_DVR_DecodeG711Frame(  
    void      *pDecHandle,  
    unsigned char *pInBuffer,  
    unsigned char *pOutBuffer  
);
```

Parameters (For 32-Bit Windows Operating System)

pDecHandle

[IN] Handle for audio decoding, which is returned by the API [NET_DVR_InitG711Decoder](#).

pInBuffer

[IN] Input buffer, which is used to save the encoded data, and the data size is 160 bytes.

pOutBuffer

[OUT] Output buffer, which is used to save the PCM audio data sampled according to the sampling standard (sampling frequency is 16000, sample in 16-bit, single channel), and the data size is 320 bytes.

API Definition (For Linux Operating System and 64-Bit Windows Operating System)

```
BOOL NET_DVR_DecodeG711Frame(  
    void             *handle,  
    NET_DVR_AUDIODEC_PROCESS_PARAM  *p_dec_proc_param  
)
```

Parameters (For Linux Operating System and 64-Bit Windows Operating System)

handle

[IN] Handle for audio decoding, which is returned by [NET_DVR_InitG711Decoder](#).

p_dec_proc_param

[IN][OUT] Decoding parameters, see details in the structure [NET_DVR_AUDIODEC_PROCESS_PARAM](#).

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 error codes may be returned by this API are as the follows: 0, 3, and 17. See the error descriptions in [Device Network SDK Errors](#).

Remarks

- This API is mainly for two-way audio and audio forwarding function. You can encode the original audio data by [NET_DVR_EncodeG711Frame](#) and send the encoded audio data to the Client. While the Client can call this API to decode the data after receiving the encoded audio data.
- Initialization is required before call this API, and after calling this API, you should also release the resource.

See Also

[NET_DVR_ReleaseG711Decoder](#)

Example

G.711 Encoding and Decoding (For Linux Operating System and 64-Bit Windows Operating System)

```
void Test_VoiceTalk_DecEncG711()  
{  
    printf("Test_VoiceTalk_DecEncG711\n");  
    const unsigned int AUDIO_CODEC_BUF_SIZE = 8192;
```

```

const unsigned int AUDIO_DEC_DATA_SIZE = 2048;
/******************Encode Audio in G.711 Mode*****************/
LPVOID hEncoderInstance=0;
//Encoding information structure
NET_DVR_AUDIOENC_INFO enc_info;
//Initialize resources for encoding
hEncoderInstance = NET_DVR_InitG711Encoder(&enc_info);
printf("enc_info.in_frame_size = %d", enc_info.in_frame_size);

FILE *out_file;
FILE *in_file;
unsigned char *indata_buf;
unsigned char *outdata_buf;
int frame = 0;
char *inputfilename = "tst_8k_ch_male.wav";
//Input audio file with WAV format
char *outputfilename = "temp.g711";
//Output the encoded audio file
if ((in_file = fopen(inputfilename,"rb"))==0)
{
    printf("error opening output file %s\n",inputfilename);
    exit(99);
}
if ((out_file = fopen(outputfilename, "wb"))==0)
{
    printf("error opening output file %s\n",outputfilename);
    exit(99);
}

NET_DVR_AUDIOENC_PROCESS_PARAM enc_param;
memset(&enc_param, 0 ,sizeof(NET_DVR_AUDIOENC_PROCESS_PARAM));

indata_buf = (unsigned char *)malloc(AUDIO_CODEC_BUF_SIZE);
outdata_buf = (unsigned char *)malloc(AUDIO_CODEC_BUF_SIZE);
memset(indata_buf, 0, AUDIO_CODEC_BUF_SIZE * sizeof(unsigned char));
memset(outdata_buf, 0, AUDIO_CODEC_BUF_SIZE * sizeof(unsigned char));

enc_param.g711_type = 0;
enc_param.in_buf = indata_buf;
enc_param.out_buf = outdata_buf;

fseek(in_file, 44, SEEK_SET);
//Delete the header of WAV file

while(1)
{
    int num_read;
    //Read the file according to the frame size in encoding information structure. The input buffer size should be
    larger than this size.
    num_read = fread(indata_buf, sizeof(char), enc_info.in_frame_size ,in_file);
    if(num_read==0) {
        fclose(in_file);
}

```

```

fclose(out_file);
break;
}

BOOL ret = NET_DVR_EncodeG711Frame(hEncoderInstance, &enc_param);
if(ret==FALSE)
{
    printf("NET_DVR_EncodeG711Frame fail,err %d!\n", NET_DVR_GetLastError());
}
else {
    int num_write;
    //The enc_param.out_frame_size is the size of encoded output data, and the output buffer must be larger than
this size.
    num_write = fwrite(outdata_buf,1,enc_param.out_frame_size,out_file);
    printf("enc_param.out_frame_size = %d", enc_param.out_frame_size);
    frame++;
    printf("g711 Enc frame\n",frame);
}
}

//Release resources
BOOL ret = NET_DVR_ReleaseG711Encoder(hEncoderInstance);
if(ret==FALSE)
{
    printf("NET_DVR_ReleaseG711Encoder fail,err %d!\n", NET_DVR_GetLastError());
}

/******************Decode Audio Data for G.711 Encoding*****************/
LPVOID hDeclInstance = 0;
hDeclInstance = NET_DVR_InitG711Decoder();

char *inputfilename2 = "temp.g711";
char *outputfilename2 = "g711_ret.pcm";
if ((in_file = fopen(inputfilename2,"rb"))==0)
{
    printf("error opening output file %s\n",inputfilename2);
    exit(99);
}
/*Open the output file for the encoded MP2 data*/
if ((out_file = fopen(outputfilename2, "wb"))==0)
{
    printf("error opening output file %s\n",outputfilename2);
    exit(99);
}
NET_DVR_AUDIODEC_PROCESS_PARAM dec_proc_param;
memset(&dec_proc_param, 0 ,sizeof(NET_DVR_AUDIODEC_PROCESS_PARAM));
memset(indata_buf, 0, AUDIO_CODEC_BUF_SIZE * sizeof(char));
memset(outdata_buf, 0, AUDIO_CODEC_BUF_SIZE * sizeof(char));

dec_proc_param.g711_type = 0;
dec_proc_param.in_buf = indata_buf;
dec_proc_param.out_buf = outdata_buf;
dec_proc_param.in_data_size = 0;

```

```

int dwRead = fread(dec_proc_param.in_buf, sizeof(char), AUDIO_DEC_DATA_SIZE, in_file);
dec_proc_param.in_data_size += dwRead;

while(1)
{
    if(dwRead==0) {
        fclose(in_file);
        fclose(out_file);
        break;
    }
    DWORD outsize = 0;
    BOOL err = NET_DVR_DecodeG711Frame(hDeclInstance, &dec_proc_param);
    if (err == 1)
    {
        //The dec_proc_param.out_frame_size is decoded output data
        fwrite(dec_proc_param.out_buf, sizeof(char), dec_proc_param.out_frame_size, out_file);
        frame++;
        printf("g711 Dec frame(%d)\n",frame);
        printf("dec_proc_param.proc_data_size(%d)\n",dec_proc_param.proc_data_size);
        printf("dec_proc_param.out_frame_size(%d)\n",dec_proc_param.out_frame_size);
        //The dec_proc_param.proc_data_size is the size of data in the input buffer.
        dec_proc_param.in_data_size = dec_proc_param.in_data_size - dec_proc_param.proc_data_size;
        //Move the data in the inout buffer to overwrite the processed data.
        memmove(dec_proc_param.in_buf, dec_proc_param.in_buf + dec_proc_param.proc_data_size,
        dec_proc_param.in_data_size);

        if (dec_proc_param.in_data_size < AUDIO_DEC_DATA_SIZE )
        {
            dwRead = fread(dec_proc_param.in_buf + dec_proc_param.in_data_size, sizeof(char),
AUDIO_DEC_DATA_SIZE - dec_proc_param.in_data_size, in_file);
            dec_proc_param.in_data_size += dwRead;
        }
    }
    else
    {
        printf("decode err:0x%x", err);
    }
}

ret = NET_DVR_ReleaseG711Decoder(hDeclInstance);
if (ret== FALSE)
{
printf("NET_DVR_ReleaseG711Decoder fail,err %d!\n", NET_DVR_GetLastError());
}
}

```

12.6.13 NET_DVR_ReleaseG711Decoder

Release resources of audio decoding (for G.711 encoding). This API is only available for Linux operating system.

API Definition

```
BOOL NET_DVR_ReleaseG711Decoder(  
    LPVOID *pDecHandle  
)
```

Parameters

pDecHandle

[IN] Handle for audio decoding, which is returned by [NET_DVR_InitG711Decoder](#).

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.

See Also

[NET_DVR_InitG711Decoder](#)

[NET_DVR_DecodeG711Frame](#)

12.6.14 NET_DVR_InitG722Encoder

Initialize resources for G.722 audio encoding.

API Definition (For 32-Bit Windows Operating System)

```
void *NET_DVR_InitG722Encoder(  
)
```

API Definition (For Linux Operating System and 64-Bit Windows Operating System)

```
void *NET_DVR_InitG722Encoder(  
    NET_DVR_AUDIOENC_INFO *enc_info  
)
```

Parameters (For Linux Operating System and 64-Bit Windows Operating System)

enc_info

[IN] Encoding information, see details in the structure [NET_DVR_AUDIOENC_INFO](#).

Return Values

Return -1 for failure, and return other values as the handles for audio encoding, or return *NULL* for success.

If -1 is returned, you can call [***NET_DVR_GetLastError***](#) to get the error code.

See Also

[***NET_DVR_ReleaseG722Encoder***](#)
[***NET_DVR_EncodeG722Frame***](#)

12.6.15 ***NET_DVR_EncodeG722Frame***

Encode audio in G.722 mode.

API Definition (For 32-Bit Windows Operating System)

```
BOOL NET_DVR_EncodeG722Frame(  
    void      *pEncodeHandle,  
    unsigned char *pInBuffer,  
    unsigned char *pOutBuffer  
)
```

Parameters (For 32-Bit Windows Operating System)

pEncodeHandle

[IN] Handle for audio encoding, which is returned by the API [***NET_DVR_InitG722Encoder***](#).

pInBuffer

[IN] Input buffer, which is used to save the audio data in PCM format that sampled according to the sampling standard (sampling frequency is 16000, sample in 16-bit, single channel), and the data size is 1280 bytes.

pOutBuffer

[OUT] Output buffer, the size of output data after encoding is 80 bytes.

API Definition (For Linux Operating System and 64-Bit Windows Operating System)

```
BOOL NET_DVR_EncodeG722Frame(  
    void      *handle,  
    NET_DVR_AUDIOENC_PROCESS_PARAM *param  
)
```

Parameters (For Linux Operating System and 64-Bit Windows Operating System)

handle

[IN] Handle for audio encoding, which is returned by [***NET_DVR_InitG722Encoder***](#).

param

[IN][OUT] Encoding parameters, see details in the structure
[NET_DVR_AUDIOENC_PROCESS_PARAM](#).

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

- This API is mainly for two-way audio and audio forwarding function. You can encode the original audio data by this API and send the encoded audio data to the platform/system. And then the API [NET_DVR_DecodeG722Frame](#) can be called to decode the data after receiving the encoded audio data.
- Initialization is required before call this API, and after calling this API, you should also release the resource.

See Also

[NET_DVR_ReleaseG722Encoder](#)

Example

G.722 Encoding and Decoding (For Linux Operating System and 64-Bit Windows Operating System)

```
void Test_VoiceTalk_DecEncG722()
{
    printf("Test_VoiceTalk_DecEncG722\n");

    /*****Encode Audio in G.722 Mode*****/
    LPVOID hEnclnstance = 0;
    NET_DVR_AUDIOENC_INFO info_param;
    int blockcount= 0;

    hEnclnstance = NET_DVR_InitG722Encoder(&info_param);
    if ((long)hEnclnstance == -1)
    {
        printf("NET_DVR_InitG722Encoder fail,err %d!\n", NET_DVR_GetLastError());
        return;
    }

    NET_DVR_AUDIOENC_PROCESS_PARAM enc_proc_param;
    FILE *infile;
    FILE *outfile;
    char * infile_path = "tst_16k_ch_fmale.wav";
    char * outfile_path = "temp.g722";
    const unsigned int AUDIO_CODEC_BUFSIZE = 8192;
    const unsigned int HIK_CODEC722_IN_SAMPLES = 320;
    const unsigned int HIK_CODEC722_OUT_BITS = 160;
    const unsigned int AUDIO_DEC_DATA_SIZE = 2048;
    short encode_input[2048]; //20ms
```

```
unsigned char encoded_data[2048];

infile=fopen(infile_path,"rb");
outfile=fopen(outfile_path,"wb");
if(infile == NULL)
{
printf("infile open error.");
return ;
}
if(outfile == NULL)
{
printf("outfile open error.");
return ;
}

enc_proc_param.in_buf = (unsigned char *)encode_input;
enc_proc_param.out_buf = (unsigned char *)encoded_data;

while (1)
{
int encode_samples = fread((short *)enc_proc_param.in_buf,1,info_param.in_frame_size,infile);
if (encode_samples < 640)
{
    fclose(infile);
    fclose(outfile);
    break;
}
blockcount++;
    BOOL ret = NET_DVR_EncodeG722Frame(hEnclInstance, &enc_proc_param);
if (ret != 1)
{
    exit(-1);
}

fwrite(
    enc_proc_param.out_buf,
    sizeof(unsigned char),
    enc_proc_param.out_frame_size,
    outfile
);
printf("g722 Enc frame %d \n", blockcount);
}

NET_DVR_ReleaseG722Encoder(hEnclInstance);

/******************Decode Audio Data for G.722 Encoding*****************/
LPVOID hDeclnstance = 0;
hDeclnstance = NET_DVR_InitG722Decoder();
if (hDeclnstance == 0)
{
printf("NET_DVR_InitG722Decoder fail,err %d!\n", NET_DVR_GetLastError());
return;
```

```

}

NET_DVR_AUDIODEC_PROCESS_PARAM dec_proc_param;
char * infile_path2 = "temp.g722";
char * outfile_path2 = "g722_ret.pcm";
unsigned char decode_input[2048];
short decode_data[2048];
infile=fopen(infile_path2,"rb");
outfile=fopen(outfile_path2,"wb");
if(infile == NULL)
{
    printf("infile open error.");
    return ;
}
if(outfile == NULL)
{
    printf("outfile open error.");
    return ;
}
dec_proc_param.in_buf = (unsigned char *)decode_input;
dec_proc_param.out_buf = (unsigned char *)decode_data;
dec_proc_param.reserved[0] = 0;
dec_proc_param.in_data_size = 0;
int decode_samples = fread(dec_proc_param.in_buf, sizeof(char), AUDIO_DEC_DATA_SIZE,infile);
dec_proc_param.in_data_size += decode_samples;
blockcount= 0;
while (1)
{
    if (decode_samples == 0){
        fclose(infile);
        fclose(outfile);
        break;
    }
    BOOL ret = NET_DVR_DecodeG722Frame(hDeclInstance ,&dec_proc_param);

    if (ret == 1)
    {
        //dec_proc_param->out_buf=output;
        fwrite(
            dec_proc_param.out_buf,
            sizeof(unsigned char),
            dec_proc_param.out_frame_size,
            outfile
        );
        blockcount++;
        printf("g722 Dec frame %d \n", blockcount);

        dec_proc_param.in_data_size = dec_proc_param.in_data_size - dec_proc_param.proc_data_size;
        memmove(dec_proc_param.in_buf, dec_proc_param.in_buf + dec_proc_param.proc_data_size,
        dec_proc_param.in_data_size);//

        if (dec_proc_param.in_data_size < AUDIO_DEC_DATA_SIZE )
    }
}

```

```
{  
    decode_samples = fread(dec_proc_param.in_buf + dec_proc_param.in_data_size, sizeof(char),  
AUDIO_DEC_DATA_SIZE - dec_proc_param.in_data_size,infile);  
    dec_proc_param.in_data_size += decode_samples;  
  
}  
}  
else  
{  
  
}  
}  
NET_DVR_ReleaseG722Decoder(hDeclInstance);  
}
```

12.6.16 NET_DVR_ReleaseG722Encoder

Release resources of G.722 audio encoding. This API is available for both the Windows and Linux operating system.

API Definition

```
void NET_DVR_ReleaseG722Encoder(  
    void *pEncodeHandle  
);
```

Parameters

pEncodeHandle

[IN] Handle for audio encoding, which is returned by [NET_DVR_InitG722Encoder](#).

Return Values

No return values. But if calling failed, you can call [NET_DVR_GetLastError](#) to get the error code.

See Also

[NET_DVR_EncodeG722Frame](#)

12.6.17 NET_DVR_InitG722Decoder

Initialize resources to decode audio data for G.722 encoding.

API Definition (For 32-Bit Windows Operating System)

```
void *NET_DVR_InitG722Decoder(  
    int nBitrate  
)
```

Parameters (For 32-Bit Windows Operating System)

nBitrate

[IN] Sampling frequency during encoding, which should be defined as 16000.

API Definition (For Linux Operating System and 64-Bit Windows Operating System)

```
void* NET_DVR_InitG722Decoder(  
)
```

Return Values

Return -1 for failure, and return other values as the handles for audio decoding, or return *NULL* for success.

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

See Also

[NET_DVR_ReleaseG722Decoder](#)
[NET_DVR_DecodeG722Frame](#)

12.6.18 NET_DVR_DecodeG722Frame

Decode the audio to G722 format.

API Definition (For 32-Bit Windows Operating System)

```
BOOL NET_DVR_DecodeG722Frame(  
    void      *pDecHandle,  
    unsigned char *pInBuffer,  
    unsigned char *pOutBuffer  
)
```

Parameters (For 32-Bit Windows Operating System)

pDecHandle

[IN] Handle for audio decoding, which is returned by the API [NET_DVR_InitG722Decoder](#).

pInBuffer

[IN] Input buffer, which is used to save the encoded data, and the data size is 80 bytes.

pOutBuffer

[OUT] Output buffer, which is used to save the PCM audio data sampled according to the sampling standard (sampling frequency is 16000, sample in 16-bit, single channel), and the data size is 1280 bytes.

API Definition (For Linux Operating System and 64-Bit Windows Operating System)

```
BOOL NET_DVR_DecodeG722Frame(  
    void           *handle,  
    NET_DVR_AUDIODEC_PROCESS_PARAM *param  
)
```

Parameters (For Linux Operating System and 64-Bit Windows Operating System)

handle

[IN] Handle for audio decoding, which is returned by [NET_DVR_InitG722Decoder](#).

param

[IN][OUT] Decoding parameters, see details in the structure

[NET_DVR_AUDIODEC_PROCESS_PARAM](#).

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 error codes may be returned by this API are as the follows: 0, 3, and 17. See the error descriptions in [Device Network SDK Errors](#).

Remarks

- This API is mainly for two-way audio and audio forwarding function. You can encode the original audio data by [NET_DVR_EncodeG722Frame](#) and send the encoded audio data to the Client. While the Client can call this API to decode the data after receiving the encoded audio data.
- Initialization is required before call this API, and after calling this API, you should also release the resource.

See Also

[NET_DVR_ReleaseG722Decoder](#)

Example

G.722 Encoding and Decoding (For Linux Operating System and 64-Bit Windows Operating System)

```
void Test_VoiceTalk_DecEncG722()  
{  
    printf("Test_VoiceTalk_DecEncG722\\n");  
  
    /*****Encode Audio in G.722 Mode*****/  
    LPVOID hEnclnstance = 0;  
    NET_DVR_AUDIOENC_INFO info_param;  
    int blockcount= 0;
```

```
hEnInstance = NET_DVR_InitG722Encoder(&info_param);
if ((long)hEnInstance == -1)
{
printf("NET_DVR_InitG722Encoder fail,err %d!\n", NET_DVR_GetLastError());
    return;
}

NET_DVR_AUDIOENC_PROCESS_PARAM enc_proc_param;
FILE      *infile;
FILE      *outfile;
char * infile_path = "tst_16k_ch_fmale.wav";
char * outfile_path = "temp.g722";
const unsigned int AUDIO_CODEC_BUF_SIZE = 8192;
const unsigned int HIK_CODEC722_IN_SAMPLES = 320;
const unsigned int HIK_CODEC722_OUT_BITS = 160;
const unsigned int AUDIO_DEC_DATA_SIZE = 2048;
short encode_input[2048]; //20ms
unsigned char encoded_data[2048];

infile=fopen(infile_path,"rb");
outfile=fopen(outfile_path,"wb");
if(infile == NULL)
{
printf("infile open error.");
return ;
}
if(outfile == NULL)
{
printf("outfile open error.");
return ;
}

enc_proc_param.in_buf  = (unsigned char *)encode_input;
enc_proc_param.out_buf = (unsigned char *)encoded_data;

while (1)
{
int encode_samples = fread((short *)enc_proc_param.in_buf,1,info_param.in_frame_size,infile);
if (encode_samples < 640)
{
    fclose(infile);
    fclose(outfile);
    break;
}
blockcount++;
    BOOL ret = NET_DVR_EncodeG722Frame(hEnInstance, &enc_proc_param);
if (ret != 1)
{
    exit(-1);
}
```

```

fwrite(
    enc_proc_param.out_buf,
    sizeof(unsigned char),
    enc_proc_param.out_frame_size,
    outfile
);
printf("g722 Enc frame %d \n", blockcount);
}

NET_DVR_ReleaseG722Encoder(hEnclInstance);

/******************Decode Audio Data for G.722 Encoding*****************/
LPVOID hDeclInstance = 0;
hDeclInstance = NET_DVR_InitG722Decoder();
if (hDeclInstance == 0)
{
    printf("NET_DVR_InitG722Decoder fail,err %d!\n", NET_DVR_GetLastError());
    return;
}

NET_DVR_AUDIODEC_PROCESS_PARAM dec_proc_param;
char * infile_path2 = "temp.g722";
char * outfile_path2 = "g722_ret.pcm";
unsigned char decode_input[2048];
short decode_data[2048];
infile=fopen(infile_path2,"rb");
outfile=fopen(outfile_path2,"wb");
if(infile == NULL)
{
    printf("infile open error.");
    return ;
}
if(outfile == NULL)
{
    printf("outfile open error.");
    return ;
}
dec_proc_param.in_buf = (unsigned char *)decode_input;
dec_proc_param.out_buf = (unsigned char *)decode_data;
dec_proc_param.reserved[0] = 0;
dec_proc_param.in_data_size = 0;
int decode_samples = fread(dec_proc_param.in_buf, sizeof(char), AUDIO_DEC_DATA_SIZE,infile);
dec_proc_param.in_data_size += decode_samples;
blockcount= 0;
while (1)
{
    if (decode_samples == 0){
        fclose(infile);
        fclose(outfile);
        break;
    }
    BOOL ret = NET_DVR_DecodeG722Frame(hDeclInstance ,&dec_proc_param);
}

```

```

if (ret == 1)
{
    //dec_proc_param->out_buf=output;
    fwrite(
        dec_proc_param.out_buf,
        sizeof(unsigned char),
        dec_proc_param.out_frame_size,
        outfile
    );
    blockcount++;
    printf("g722 Dec frame %d \n", blockcount);

    dec_proc_param.in_data_size = dec_proc_param.in_data_size - dec_proc_param.proc_data_size;
    memmove(dec_proc_param.in_buf, dec_proc_param.in_buf + dec_proc_param.proc_data_size,
    dec_proc_param.in_data_size);//

    if (dec_proc_param.in_data_size < AUDIO_DEC_DATA_SIZE )
    {
        decode_samples = fread(dec_proc_param.in_buf + dec_proc_param.in_data_size, sizeof(char),
AUDIO_DEC_DATA_SIZE - dec_proc_param.in_data_size,infile);
        dec_proc_param.in_data_size += decode_samples;

    }
    else
    {

    }
    NET_DVR_ReleaseG722Decoder(hDeclInstance);
}

```

12.6.19 NET_DVR_ReleaseG722Decoder

Release resources of audio decoding (for G.722 encoding). This API is available for both the Windows and Linux operating system.

API Definition

```

void NET_DVR_ReleaseG722Decoder(
    void *pDecHandle
);

```

Parameters

pDecHandle

[IN] Handle for decoding audio, which is returned by [**NET_DVR_InitG722Decoder**](#).

Return Values

No return values. But if calling failed, you can call [***NET_DVR_GetLastError***](#) to get the error code.

See Also

[***NET_DVR_DecodeG722Frame***](#)

12.6.20 NET_DVR_InitG726Encoder

Initialize resources for G.726 audio encoding. This API is available for both the Windows and Linux operating system.

API Definition

```
void* NET_DVR_InitG726Encoder(  
    void *pEncModule  
)
```

Parameters

pEncModule

[OUT] Handle of encoding module, which can be set as input parameter when encoding.

Return Values

Return -1 for failure, and returns other values as the handles for audio encoding.

If -1 is returned, you can call [***NET_DVR_GetLastError***](#) to get the error code.

See Also

[***NET_DVR_ReleaseG726Encoder***](#)

[***NET_DVR_EncodeG726Frame***](#)

12.6.21 NET_DVR_EncodeG726Frame

Encode audio in G.726 mode. This API is available for both the Windows and Linux operating system.

API Definition

```
BOOL NET_DVR_EncodeG726Frame(  
    void *pEncHandle,  
    unsigned char *pInBuffer,  
    unsigned char *pOutBuffer,  
    BYTE byReset  
)
```

Parameters

pEncHandle

[IN] Handle for encoding audio, which is returned by [NET_DVR_InitG726Encoder](#).

pInBuffer

[IN] Input buffer pointer, which is used to save the audio data with PCM format that sampled according to the sampling standard (sampling frequency is 16000, sample in 16-bit, single channel). The predefined input data size is 640 bytes.

pOutBuffer

[OUT] Output buffer pointer, the encoded output data size is 80 bytes.

byReset

[IN] Whether to reset. The first frame should be reset.

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

- This API is mainly for two-way audio and audio forwarding function. You can encode the original audio data by this API and send the encoded audio data to the platform/system. And then the API [NET_DVR_DecodeG726Frame](#) can be called to decode the data after receiving the encoded audio data.
- Initialization is required before call this API, and after calling this API, you should also release the resource.

See Also

[NET_DVR_ReleaseG726Encoder](#)

Example

G.726 Encoding and Decoding (For Linux Operating System)

```
void Test_VoiceTalk_DecEncG726()
{
    const unsigned int AUDIO_CODEC_BUF_SIZE = 1024;
    /*****Encode Audio in G.726 Mode*****/
    LPVOID hEncoder=0, hEncoderInstance=0;
    hEncoderInstance = NET_DVR_InitG726Encoder(&hEncoder);
    if(hEncoder==0 || hEncoderInstance==0)
    {
        printf("NET_DVR_InitG726Encoder fail,err %d!\n", NET_DVR_GetLastError());
    }

    FILE *Fi;
    FILE *Fo;
    unsigned char *enc_inbuf;
```

```

unsigned char *enc_outbuf;
char *inputfilename = "tst_8k_ch_male.pcm";
char *outputfilename = "temp.g726";
if ((enc_inbuf = (unsigned char *) calloc(AUDIO_CODEC_BUF_SIZE, sizeof(char))) == NULL)
{
    printf("Error in memory allocation!\n");
    exit(0);
}
if ((enc_outbuf = (unsigned char *) calloc(AUDIO_CODEC_BUF_SIZE, sizeof(char))) == NULL)
{
    printf("Error in memory allocation!\n");
    exit(0);
}
if ((Fi = fopen(inputfilename, "rb")) == NULL)
{
    printf("file open error!\n");
    exit(0);
}
if ((Fo = fopen(outputfilename, "wb")) == NULL)
{
    printf("file open error!\n");
    exit(0);
}
printf("G726_EBCIN_DECOUT_SIZE = %d\n", G726_EBCIN_DECOUT_SIZE); // 640
printf("G726_ENC_OUT_SIZE = %d\n", G726_ENC_OUT_SIZE); // 80
printf("G726_EBCIN_DECOUT_SIZE = %d\n", G726_DEC_IN_SIZE); // 80
int reset = 1;
for (int cur_blk = 0; ; cur_blk++)
{
    reset = (reset == 1 && cur_blk == 0) ? 1 : 0;
    if (fread(enc_inbuf, sizeof(char), G726_EBCIN_DECOUT_SIZE, Fi) < G726_EBCIN_DECOUT_SIZE)
    {
        fclose(Fi);
        fclose(Fo);
        break;
    }
}

BOOL ret = NET_DVR_EncodeG726Frame(hEncoder, enc_inbuf, enc_outbuf, reset);

if (ret == 1)
{
    fwrite(enc_outbuf, sizeof(char), G726_ENC_OUT_SIZE, Fo);
}
else
{
    printf("encode error!\n");
    break;
}

printf("g726 Enc frame: %d\n", cur_blk+1);
}

```

```
NET_DVR_ReleaseG726Encoder(hEncoderInstance);

/*****Decode Audio Data for G.726 Encoding*****/
LPVOID hDec = 0, hDeclnstance = 0;
hDeclnstance = NET_DVR_InitG726Decoder(&hDec);
if (hDec == 0 || hDeclnstance == 0)
{
printf("NET_DVR_InitG726Decoder fail,err %d!\n", NET_DVR_GetLastError());
}

char *inputfilename2 = "temp.g726";
char *outputfilenam2= "g726_ret.pcm";
if ((Fi = fopen(inputfilename2, "rb")) == NULL)
{
    printf("file open error!\n");
    exit(0);
}
if ((Fo = fopen(outputfilenam2, "wb")) == NULL)
{
    printf("file open error!\n");
    exit(0);
}
memset(enc_inbuf, 0, AUDIO_CODEC_BUF_SIZE);
memset(enc_outbuf, 0, AUDIO_CODEC_BUF_SIZE);
reset = 1; //The first frame should be reset
for (int cur_blk = 0; ; cur_blk++)
{
    reset = (reset == 1 && cur_blk == 0) ? 1 : 0;
    if (fread(enc_inbuf, sizeof(char), G726_DEC_IN_SIZE, Fi) < G726_DEC_IN_SIZE)
    {
        fclose(Fi);
        fclose(Fo);
        break;
    }
    BOOL ret = NET_DVR_DecodeG726Frame(hDec, enc_inbuf, enc_outbuf, reset);

    if (ret == 1)
    {
        fwrite(enc_outbuf, sizeof(char), G726_EBCIN_DECOUT_SIZE, Fo);
    }
    else
    {
        printf("decode err:0x%x", ret);
        break;
    }
    printf("g726 Dnc frame: %d\n", cur_blk);
}
NET_DVR_ReleaseG726Decoder(hDeclnstance);
}
```

12.6.22 NET_DVR_ReleaseG726Encoder

Release resources of G.726 audio encoding. This API is available for both the Windows and Linux operating system.

API Definition

```
void NET_DVR_ReleaseG726Encoder(  
    void *pEncHandle  
)
```

Parameters

pEncHandle

[IN] Handle for audio encoding, which is returned by [NET_DVR_InitG726Encoder](#).

Return Values

No return values. But if calling failed, you can call [NET_DVR_GetLastError](#) to get the error code.

12.6.23 NET_DVR_InitG726Decoder

Initialize resources to decode audio data for G.726 encoding. This API is available for both the Windows and Linux operating system.

API Definition

```
void* NET_DVR_InitG726Decoder(  
    void **pDecMoudle  
)
```

Parameters

pDecMoudle

[OUT] Handle of decoding module, which can be set as the input parameter when decoding.

Return Values

Return -1 for failure, and returns other values as the handles for decoding.
If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

See Also

[NET_DVR_ReleaseG726Decoder](#)
[NET_DVR_DecodeG726Frame](#)

12.6.24 NET_DVR_DecodeG726Frame

Decode the audio to G726 format. This API is available for both the Windows and Linux operating system.

API Definition

```
BOOL NET_DVR_DecodeG726Frame(  
    void      *pDecHandle,  
    unsigned char *pInBuffer,  
    unsigned char *pOutBuffer,  
    BYTE      byReset  
)
```

Parameters

pDecHandle

[IN] Handle for decoding audio, which is returned by [NET_DVR_InitG726Decoder](#).

pInBuffer

[IN] Input buffer pointer. The encoded data size is 80 bytes.

pOutBuffer

[OUT] Output buffer pointer, get the PCM audio data according to the sampling standard (sampling frequency is 8000, sampling in 16-bit and single channel). The predefined output data size is 640 bytes.

byReset

[IN] Whether to reset. The first frame should be reset.

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

- This API is mainly for two-way audio and audio forwarding function. You can encode the original audio data by [NET_DVR_EncodeG726Frame](#) and send the encoded audio data to the platform or system. And this API can be called to decode the data after receiving the encoded audio data.
- Initialization is required before call this API, and after calling this API, you should also release the resource.

See Also

[NET_DVR_ReleaseG726Decoder](#)

Example

G.726 Encoding and Decoding (For Linux Operating System)

```

void Test_VoiceTalk_DecEncG726()
{
    const unsigned int AUDIO_CODEC_BUF_SIZE = 1024;
    /*****Encode Audio in G.726 Mode*****/
    LPVOID hEncoder=0, hEncoderInstance=0;
    hEncoderInstance = NET_DVR_InitG726Encoder(&hEncoder);
    if(hEncoder==0 || hEncoderInstance==0)
    {
        printf("NET_DVR_InitG726Encoder fail,err %d!\n", NET_DVR_GetLastError());
    }

    FILE *Fi;
    FILE *Fo;
    unsigned char *enc_inbuf;
    unsigned char *enc_outbuf;
    char *inputfilename = "tst_8k_ch_male.pcm";
    char *outputfilename = "temp.g726";
    if ((enc_inbuf = (unsigned char *) calloc(AUDIO_CODEC_BUF_SIZE, sizeof(char))) == NULL)
    {
        printf("Error in memory allocation!\n");
        exit(0);
    }
    if ((enc_outbuf = (unsigned char *) calloc(AUDIO_CODEC_BUF_SIZE, sizeof(char))) == NULL)
    {
        printf("Error in memory allocation!\n");
        exit(0);
    }
    if ((Fi = fopen(inputfilename, "rb")) == NULL)
    {
        printf("file open error!\n");
        exit(0);
    }
    if ((Fo = fopen(outputfilename, "wb")) == NULL)
    {
        printf("file open error!\n");
        exit(0);
    }
    printf("G726_EBCIN_DECOUT_SIZE = %d\n", G726_EBCIN_DECOUT_SIZE); // 640
    printf("G726_ENC_OUT_SIZE = %d\n", G726_ENC_OUT_SIZE); // 80
    printf("G726_EBCIN_DECOUT_SIZE = %d\n", G726_DEC_IN_SIZE); // 80
    int reset = 1;
    for (int cur_blk = 0; ; cur_blk++)
    {
        reset = (reset == 1 && cur_blk == 0) ? 1 : 0;
        if (fread(enc_inbuf, sizeof(char), G726_EBCIN_DECOUT_SIZE, Fi) < G726_EBCIN_DECOUT_SIZE)
        {
            fclose(Fi);
            fclose(Fo);
            break;
        }
    }

    BOOL ret = NET_DVR_EncodeG726Frame(hEncoder, enc_inbuf, enc_outbuf, reset);
}

```

```

if (ret == 1)
{
    fwrite(enc_outbuf, sizeof(char), G726_ENC_OUT_SIZE, Fo);
}
else
{
    printf("encode error!\n");
    break;
}

printf("g726 Enc frame: %d\n", cur_blk+1);

}

NET_DVR_ReleaseG726Encoder(hEncoderInstance);

/*****************Decode Audio Data for G.726 Encoding*****************/
LPVOID hDec = 0, hDeclInstance = 0;
hDeclInstance = NET_DVR_InitG726Decoder(&hDec);
if (hDec == 0 || hDeclInstance == 0)
{
    printf("NET_DVR_InitG726Decoder fail,err %d!\n", NET_DVR_GetLastError());
}

char *inputfilename2 = "temp.g726";
char *outputfilenam2= "g726_ret.pcm";
if ((Fi = fopen(inputfilename2, "rb")) == NULL)
{
    printf("file open error!\n");
    exit(0);
}
if ((Fo = fopen(outputfilenam2, "wb")) == NULL)
{
    printf("file open error!\n");
    exit(0);
}
memset(enc_inbuf, 0, AUDIO_CODEC_BUF_SIZE);
memset(enc_outbuf, 0, AUDIO_CODEC_BUF_SIZE);
reset = 1; //The first frame should be reset
for (int cur_blk = 0; ; cur_blk++)
{
    reset = (reset == 1 && cur_blk == 0) ? 1 : 0;
    if (fread(enc_inbuf, sizeof(char), G726_DEC_IN_SIZE, Fi) < G726_DEC_IN_SIZE)
    {
        fclose(Fi);
        fclose(Fo);
        break;
    }
    BOOL ret = NET_DVR_DecodeG726Frame(hDec, enc_inbuf, enc_outbuf, reset);

    if (ret == 1)

```

```
{  
    fwrite(enc_outbuf, sizeof(char), G726_EBCIN_DECOUT_SIZE, Fo);  
}  
else  
{  
    printf("decode err:0x%x", ret);  
    break;  
}  
printf("g726 Dnc frame: %d\n", cur_blk);  
}  
  
NET_DVR_ReleaseG726Decoder(hDeclInstance);  
}
```

12.6.25 NET_DVR_ReleaseG726Decoder

Release resources of audio decoding (for G.726 encoding). This API is available for both the Windows and Linux operating system.

API Definition

```
void NET_DVR_ReleaseG726Decoder(  
    void *pDecHandle  
);
```

Parameters

pDecHandle

[IN] Handle for decoding audio, which is returned by [NET_DVR_InitG726Decoder](#).

Return Values

No return values. But if calling failed, you can call [NET_DVR_GetLastError](#) to get the error code.

See Also

[NET_DVR_DecodeG726Frame](#)

12.6.26 NET_DVR_SetVoiceDataCallBack

Set a callback function for collecting the audio data.

API Definition

```
BOOL NET_DVR_SetVoiceDataCallBack(  
    LONG          lVoiceComHandle,  
    BOOL          bNeedCBNoEncData,  
    fVoiceDataCallBack cbVoiceDataCallBack,
```

```
void *pUser  
);
```

Parameters

IVoiceComHandle

[IN] Value returned by [NET_DVR_StartVoiceCom_V30](#) or [NET_DVR_StartVoiceCom_MR_V30](#).

bNeedCBNoEncData

[IN] Audio data types need to be called back: 0-Encoded audio data. 1-Original PCM data. For audio forwarding, only the encoded audio data can be called back. So the parameter is invalid for the audio forwarding.

cbVoiceDataCallBack

[IN] Audio data callback function, see details in [fVoiceDataCallBack](#).

pUser

[IN] Pointer of user data.

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.

See Also

[NET_DVR_StartVoiceCom_MR_V30](#)

[NET_DVR_StartVoiceCom_V30](#)

12.6.27 NET_DVR_StartVoiceCom_MR_V30

Enable audio forwarding function and get the encoded audio data.

API Definition

```
LONG NET_DVR_StartVoiceCom_MR_V30(  
    LONG     lUserID,  
    DWORD    dwVoiceChan,  
    fVoiceDataCallBack cbVoiceDataCallBack,  
    void     *pUser  
)
```

Parameters

lUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

dwVoiceChan

[IN] Audio channel No. For the two-way audio channels of device, it starts from 1. For the network channels of device, it starts from the value of (Initial two-way audio channel No. (**byStartDTalkChan**)+network channel No.-1).

For example, when the Client starts two-way audio with the network camera linked by network channel No.02 via the NVR, the **dwVoiceChan** equals to **byStartDTalkChan+1**.

cbVoiceDataCallBack

[IN] Audio data callback function, see details in [**fVoiceDataCallBack**](#). The obtained audio data is encoded, so you should call the audio decoding API to decode the audio data to PCM data.

pUser

[IN] Pointer of user data.

Return Values

Returns -1 for failure, and returns other values as the handle parameters of [**NET_DVR_VoiceComSendData**](#) or [**NET_DVR_StopVoiceCom**](#).

If -1 is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

Remarks

- Before enabling the audio forwarding function, you can call [**NET_DVR_SetDVRConfig**](#) with the command of NET_DVR_SET_COMPRESSCFG_AUD (command No.: 1058) to set the audio encoding type for device in the structure [**NET_DVR_COMPRESSION_AUDIO**](#).
- When the audio is encoded to G722 format, the sampling frequency is 16000, and audio is sampled in 16-bit and single channel. The definition of audio format is:

```
const int SAMPLES_PER_SECOND = 16000;
const int CHANNEL = 1;
const int BITS_PER_SAMPLE = 16;
WAVEFORMATEX m_wavFormatEx;
m_wavFormatEx.cbSize = sizeof(m_wavFormatEx);
m_wavFormatEx.nBlockAlign = CHANNEL * BITS_PER_SAMPLE / 8;
m_wavFormatEx.nChannels = CHANNEL;
m_wavFormatEx.nSamplesPerSec = SAMPLES_PER_SECOND;
m_wavFormatEx.wBitsPerSample = BITS_PER_SAMPLE;
m_wavFormatEx.nAvgBytesPerSec = SAMPLES_PER_SECOND*m_wavFormatEx.nBlockAlign
```

- When the audio is encoded to G711 format, the sampling frequency is 8000, and audio is sampled in 16-bit and single channel. The definition of audio format is:

```
const int SAMPLES_PER_SECOND_G711_MU = 8000;
const int CHANNEL = 1;
const int BITS_PER_SAMPLE = 16;
WAVEFORMATEX m_wavFormatEx;
m_wavFormatEx.cbSize = sizeof(m_wavFormatEx);
m_wavFormatEx.nBlockAlign = CHANNEL * BITS_PER_SAMPLE / 8;
m_wavFormatEx.nChannels = CHANNEL;
m_wavFormatEx.nSamplesPerSec = SAMPLES_PER_SECOND_G711_MU;
m_wavFormatEx.wBitsPerSample = BITS_PER_SAMPLE;
m_wavFormatEx.nAvgBytesPerSec = SAMPLES_PER_SECOND_G711_MU* m_wavFormatEx.nBlockAlign;
```

See Also

[NET_DVR_StopVoiceCom](#)

12.6.28 NET_DVR_StartVoiceCom_V30

Enable two-way audio function.

API Definition

```
LONG NET_DVR_StartVoiceCom_V30(  
    LONG      lUserID,  
    DWORD     dwVoiceChan,  
    BOOL      bNeedCBNoEncData,  
    fVoiceDataCallBack cbVoiceDataCallBack,  
    void      *pUser  
)
```

Parameters

lUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

dwVoiceChan

[IN] Audio channel No. For the two-way audio channels of device, it starts from 1. For the network channels of device, it starts from the value of (Initial two-way audio channel No. (**byStartDTalkChan**) + network channel No.-1).

For example, when the Client starts two-way audio with the network camera linked by network channel 02 via the NVR, the **dwVoiceChan** = **byStartDTalkChan** + 1.

bNeedCBNoEncData

[IN] Audio data types need to be called back: 0-Encoded audio data. 1-Original PCM data.

cbVoiceDataCallBack

[IN] Audio data callback function, see details in [fVoiceDataCallBack](#).

pUser

[IN] Pointer of user data.

Return Values

Return -1 for failure, and return other values as the handle parameters of

[NET_DVR_StopVoiceCom](#).

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

Remarks

- This API is supported for programming in 64-bit Windows/Linux operating system and 32-bit Windows operating system.
- Before enabling the audio forwarding function, you can call [**NET_DVR_SetDVRConfig**](#) with the command of NET_DVR_SET_COMPRESSCFG_AUD (command No.: 1058) to set the audio encoding type for device in the structure [**NET_DVR_COMPRESSION_AUDIO**](#).
- When the audio is encoded to G722 format, the sampling frequency is 16000, and audio is sampled in 16-bit and single channel. The definition of audio format is:

```
const int SAMPLES_PER_SECOND = 16000;
const int CHANNEL = 1;
const int BITS_PER_SAMPLE = 16;
WAVEFORMATEX m_wavFormatEx;
m_wavFormatEx.cbSize = sizeof(m_wavFormatEx);
m_wavFormatEx.nBlockAlign = CHANNEL * BITS_PER_SAMPLE / 8;
m_wavFormatEx.nChannels = CHANNEL;
m_wavFormatEx.nSamplesPerSec = SAMPLES_PER_SECOND;
m_wavFormatEx.wBitsPerSample = BITS_PER_SAMPLE;
m_wavFormatEx.nAvgBytesPerSec = SAMPLES_PER_SECOND*m_wavFormatEx.nBlockAlign
```

- When the audio is encoded to G711 or G726 format, the sampling frequency is 8000, and audio is sampled in 16-bit and single channel. The definition of audio format is:

```
const int SAMPLES_PER_SECOND_G711_MU = 8000;
const int CHANNEL = 1;
const int BITS_PER_SAMPLE = 16;
WAVEFORMATEX m_wavFormatEx;
m_wavFormatEx.cbSize = sizeof(m_wavFormatEx);
m_wavFormatEx.nBlockAlign = CHANNEL * BITS_PER_SAMPLE / 8;
m_wavFormatEx.nChannels = CHANNEL;
m_wavFormatEx.nSamplesPerSec = SAMPLES_PER_SECOND_G711_MU;
m_wavFormatEx.wBitsPerSample = BITS_PER_SAMPLE;
m_wavFormatEx.nAvgBytesPerSec = SAMPLES_PER_SECOND_G711_MU* m_wavFormatEx.nBlockAlign;
```

- This API cannot be called by multiple threads at same time.

See Also

[**NET_DVR_StopVoiceCom**](#)

12.6.29 NET_DVR_StopVoiceCom

Stop two-way audio or forwarding audio.

API Definition

```
BOOL NET_DVR_StopVoiceCom(
    LONG lVoiceComHandle
);
```

Parameters

IVoiceComHandle

Value returned by [NET_DVR_StartVoiceCom_V30](#) or [NET_DVR_StartVoiceCom_MR_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.

See Also

[NET_DVR_StartVoiceCom_MR_V30](#)

[NET_DVR_StartVoiceCom_V30](#)

12.6.30 NET_DVR_VoiceComSendData

Forward the audio data.

API Definition

```
BOOL NET_DVR_VoiceComSendData(  
    LONG   IVoiceComHandle,  
    char   *pSendBuf,  
    DWORD  dwBufSize  
)
```

Parameters

IVoiceComHandle

[IN] Value returned by [NET_DVR_StartVoiceCom_MR_V30](#).

pSendBuf

[IN] Buffer for saving audio data.

dwBufSize

[IN] Audio data size. When the audio is encoded to G722 format, the size of data sent for each time is 80 bytes. When the audio is encoded to G726 format, the size of data sent for each time is 160 bytes.

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

This API is used to send the encoded audio data to device, so for different audio encoding types, the API used to decode the audio data are also different.

12.7 Remote Maintenance APIs

12.7.1 NET_DVR_CloseUpgradeHandle

Close the remote upgrading handle and release resources.

API Definition

```
BOOL NET_DVR_CloseUpgradeHandle(  
    LONG lUpgradeHandle  
)
```

Parameters

IUpgradeHandle

[in] The value returned by [NET_DVR_Upgrade_V50](#).

Return Values

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

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

The possible error codes returned by the API are listed as below:

Error Type	Value	Error Message
NET_DVR_NOERROR	0	No error.
NET_DVR_NOINIT	3	SDK is not initialized.
NET_DVR_ORDER_ERROR	12	API calling order error.
NET_DVR_PARAMETER_ERROR	17	Parameter error. Input or output parameter in the SDK API is NULL.

12.7.2 NET_DVR_FindDVRLog_V50

Search for log files.

API Definition

```
LONG NET_DVR_FindDVRLog(  
    LONG          lUserID,  
    NET_DVR_FIND_LOG_COND *pFindCond  
)
```

Parameters

IUserID

[IN] User ID, which is returned by [NET_DVR_Login_V40](#).

pFindCond

[IN] Pointer of log search conditions, refer to the structure for details.

Return Values

Return -1 for failure, and return other value as the parameter of [NET_DVR_FindNextLog_V50](#) or [NET_DVR_FindLogClose_V30](#).

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

12.7.3 NET_DVR_FindLogClose_V30

Stop searching for logs and release log search resources.

API Definition

```
BOOL NET_DVR_FindLogClose_V30(  
    LONG ILogHandle  
)
```

Parameters

ILogHandle

[in] Handle of log search, which is returned by [NET_DVR_FindDVRLog_V50](#).

Return Values

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

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

12.7.4 NET_DVR_FindNextLog_V50

Get the searched log files one by one.

API Definition

```
LONG NET_DVR_FindNextLog_50(  
    LONG      ILogHandle,  
    NET_DVR_LOG_V50  *lpLogData  
)
```

Parameters

ILogHandle

[IN] Handle for log search, which is returned by [NET_DVR_FindDVRLog_V50](#).

IpLogData

[OUT] Pointer for saving the log files, refer to the structure for details.

Return Values

Return -1 for failure, and return other value as the getting status, see details in the table below.

Status	Value	Description
NET_DVR_FILE_SUCCESS	1000	Getting log file succeeded.
NET_DVR_FILE_NOFIND	1001	No log found.
NET_DVR_ISFINDING	1002	Searching. Please wait.
NET_DVR_NOMOREFILE	1003	No more log found. Search ended.
NET_DVR_FILE_EXCEPTION	1004	Search exception.

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

Remarks

You should repeatedly call this API if you want to get all searched log files.

12.7.5 NET_DVR_GetUpgradeProgress

Get the progress of remote upgrade.

API Definition

```
int NET_DVR_GetUpgradeProgress(  
    LONG lUpgradeHandle  
)
```

Parameters

IUpgradeHandle

[IN] Upgrade handle, which is returned by [NET_DVR_Upgrade_V50](#).

Return Values

Return -1 for failure, and return other values as the upgrade progress percentage (from 0 to 100).

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

12.7.6 NET_DVR_GetUpgradeState

Get the status of remote upgrade.

API Definition

```
int NET_DVR_GetUpgradeState(  
    LONG lUpgradeHandle  
>;
```

Parameters

IUpgradeHandle

[IN] Upgrade handle, which is returned by [NET_DVR_Upgrade_V50](#).

Return Values

Return -1 for failure, and return other values as the upgrade status, see details in the table below:

Table 12-3 Upgrade Status

Status Value	Description
1	Upgraded.
2	Upgrading.
3	Upgrade failed.
4	Network disconnected. Unknown status.
5	Upgrade packet language mismatched.
6	Writing to flash failed.
7	Upgrade packet type mismatched.
8	Upgrade packet version mismatched.
9	The file is locked.
10	Backup exception.
11	Memory full.

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code. And if the status code 9, 10, or 11 is returned, please contact our supports for help.

12.7.7 NET_DVR_SetLogFile

Write log files.

API Definition

```
BOOL NET_DVR_SetLogFile(  
    DWORD  nLogLevel,  
    char   *strLogDir,  
    BOOL   bAutoDel  
)
```

Parameters

nLogLevel

[IN] Log level: 0-disable log (default), 1-output error log only, 2-output error and debug log, 3-output all logs (i.e., error, debug, and information).

strLogDir

[IN] Saving path of log file. The default path of Windows operating system is **C:\|SdkLog\|**, and the default path of Linux operating system is **/home/sdklog/**.

bAutoDel

[IN] Whether to enable automatically deleting log files when the maximum number reached: "TRUE" (default)-yes and delete the exceeded number of log files; "FALSE"-no and compress the log files when the number reaches 50.

Return Values

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

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

Remarks

- The log file path must be absolute path, and should be finished with "\\\" (Windows) or "/" (Linux). It is suggested to manually create file firstly. If no file path is allocated, it will use the default path. If the path is changed, it will use the new path when writing the next file.
- You can call the API for multiple times to create new log files and up to 10 files can be created at the same time.

12.7.8 NET_DVR_Upgrade_V50

Remotely upgrade device firmware.

API Definition

```
LONG NET_DVR_Upgrade_V50(  
    DWORD          IUserID,  
    NET_DVR_UPGRADE_PARAM  lpUpgradeParam  
)
```

Parameters

IUserID

[in] Value returned by [NET_DVR_Login_V40](#).

IpUpgradeParam

[in] Upgrade parameters, see details in the structure [NET_DVR_UPGRADE_PARAM](#).

Return Value

Return -1 for failure, and return other values as the parameters of

[NET_DVR_GetUpgradeProgress](#).

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

12.8 Callback Function

12.8.1 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.

12.8.2 DEV_WORK_STATE_CB

Callback function of device status information

Callback Function Definition

```
typedef BOOL (CALLBACK *DEV_WORK_STATE_CB)(  
    void          *pUserData,  
    LONG          lUserID,  
    NET_DVR_WORKSTATE_V40  *lpWorkState  
)
```

Parameters

pUserData

[OUT] Custom data.

lUserID

[OUT] User ID, which is returned by [NET_DVR_Login_V40](#).

lpWorkState

[OUT] Device status information. See details in the structure [NET_DVR_WORKSTATE_V40](#). If getting device status failed, this parameter's value is null. You can directly call [NET_DVR_GetLastError](#) to get the error code.

12.8.3 fExceptionCallBack

Exception message callback function.

Callback Function Definition

```
typedef void(CALLBACK *fExceptionCallBack)(  
    DWORD  dwType,  
    LONG   lUserID,  
    LONG   lHandle,
```

```
void *pUser
);
```

Callback Function Parameters

dwType

[OUT] Message types of exception or reconnection, see details in the table below:

dwType	Value	Description
EXCEPTION_EXCHANGE	0x8000	User interaction exception
EXCEPTION_AUDIOEXCHANGE	0x8001	Two-way audio exception
EXCEPTION_ALARM	0x8002	Alarm exception
EXCEPTION_PREVIEW	0x8003	Live view exception
EXCEPTION_SERIAL	0x8004	Transparent channel exception
EXCEPTION_RECONNECT	0x8005	Reconnecteing during live view
EXCEPTION_ALARMRECONNECT	0x8006	Reconnecting during alarm
EXCEPTION_SERIALRECONNECT	0x8007	Transparent channel reconnecting
SERIAL_RECONNECTSUCCESS	0x8008	Transparent channel reconnected
EXCEPTION_PLAYBACK	0x8010	Playback exception
EXCEPTION_DISKFMT	0x8011	HDD formatted
EXCEPTION_PASSIVEDECODE	0x8012	Passive decoding exception
EXCEPTION_EMAILTEST	0x8013	Email test exception
EXCEPTION_BACKUP	0x8014	Backup exception
PREVIEW_RECONNECTSUCCESS	0x8015	Live view reconnected
ALARM_RECONNECTSUCCESS	0x8016	Alarm reconnected
RESUME_EXCHANGE	0x8017	User interaction resumed
NETWORK_FLOWTEST_EXCEPTION	0x8018	Network traffic flow detection exception
EXCEPTION_PICPREVIEWRECONNECT	0x8019	Reconnecting during picture preview
PICPREVIEW_RECONNECTSUCCESS	0x8020	Reconnected during picture preview
EXCEPTION_PICPREVIEW	0x8021	Picture preview exception
EXCEPTION_MAX_ALARM_INFO	0x8022	No more alarm information can be stored

dwType	Value	Description
EXCEPTION_LOST_ALARM	0x8023	Alarm loss
EXCEPTION_PASSIVETRANSRECONNECT	0x8024	Reconnecting during passive transcoding
PASSIVETRANS_RECONNECTSUCCESS	0x8025	Reconnected during passive transcoding
EXCEPTION_PASSIVETRANS	0x8026	Passive transcoding exception
EXCEPTION_RELOGIN	0x8040	Logging in again
RELOGIN_SUCCESS	0x8041	Logged in
EXCEPTION_PASSIVEDECODE_RECONNNECT	0x8042	Reconnected during passive decoding
EXCEPTION_CLUSTER_CS_ARMFAILED	0x8043	Cluster exception alarm
EXCEPTION_RELOGIN_FAILED	0x8044	Login again failed, stopped logging in
EXCEPTION_PREVIEW_RECONNECT_CLOSED	0x8045	Disabled live view reconnection function
EXCEPTION_ALARM_RECONNECT_CLOSED	0x8046	Disabled alarm reconnection function
EXCEPTION_SERIAL_RECONNECT_CLOSED	0x8047	Disabled transparent channel reconnection function
EXCEPTION_PIC_RECONNECT_CLOSED	0x8048	Disabled display reconnection function
EXCEPTION_PASSIVE_DECODE_RECONNECT_CLOSED	0x8049	Disabled passive decoding reconnection function
EXCEPTION_PASSIVE_TRANS_RECONNECT_CLOSED	0x804a	Disable passive transcoding reconnection
EXCEPTION_VIDEO_DOWNLOAD	0x804b	Video downloading exception.

IUserID

[OUT] Login ID.

IHandle

[OUT] Handle of exception types.

pUser

[OUT] User data.

12.8.4 fLoginResultCallBack

Login Status Callback Function

Member	Data Type	Description
IUserID	LONG	User ID, which is returned by <u>NET_DVR_Login_V40</u> .
dwResult	DWORD	Login status: 0-asynchronously logging in failed, 1-asynchronously logged in.
lpDeviceInfo	<u>NET_DVR_DEVICEINFO_V40</u>	Device information, such as serial No., channel, capability, and so on.
pUser	void*	User data.

12.8.5 fPlayDataCallBack_V40

Video data callback function.

Callback Function Definition

```
typedef void (CALLBACK *fPlayDataCallBack_V40)(
    LONG   IPlayHandle,
    DWORD  dwDataType,
    BYTE   *pBuffer,
    DWORD  dwBufSize,
    void   *pUser
);
```

Parameters

IPlayHandle

[IN] Current playback handle.

dwDataType

[OUT] Data type, see details in the following table:

dwDataType	Value	Description
NET_DVR_SYSHEAD	1	System header data.
NET_DVR_STREAMDATA	2	Stream data (including video and audio stream, and video stream)

dwDataType	Value	Description
NET_DVR_CHANGE_FORWARD	10	Switch stream to normal playback.
NET_DVR_CHANGE_REVERSE	11	Switch stream to reverse playback.

pBuffer

[OUT] Pointer of buffer to save data.

dwBufSize

[OUT] Buffer size.

pUser

[OUT] User data.

See Also

[NET_DVR_SetPlayDataCallBack_V40](#)

12.8.6 fRealDataCallBack

Stream data callback function.

Callback Function Definition

```
typedef void(CALLBACK *fRealDataCallBack)(  
    LONG   lRealHandle,  
    DWORD  dwDataType,  
    BYTE   *pBuffer,  
    DWORD  dwBufSize,  
    DWORD  *dwser  
)
```

Parameters

lRealHandle

[OUT] Current live view handle.

dwDataType

[OUT] Data type, see details in the following table.

Type Macro Definition	Value	Description
NET_DVR_SYSHEAD	1	System header data.
NET_DVR_STREAMDATA	2	Stream data (including video & audio stream, or video stream).
NET_DVR_AUDIOSTREAMDATA	3	Audio stream data.
NET_DVR_METADATA_DATA	107	Metadata data packet transmitted via ISAPI.
NET_DVR_PRIVATE_DATA	112	Private data, including VCA information.

pBuffer

[OUT] Pointer of buffer to save data.

dwBufSize

[OUT] Buffer size.

dwUser

[OUT] User data.

Remarks

This callback function is not applicable for the operations or API that may take a long time.

12.8.7 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.

dwType	lpBuffer
NET_SDK_CALLBACK_TYPE_STATUS	If dwBufLen is 4, lpBuffer is 4-byte connection status; if dwBufLen is 8, lpBuffer consists of 4-byte connection status and 4-byte error code. The connection status is enumerated in <u>NET_SDK_CALLBACK_STATUS_NORMAL</u> .
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 (dwCommand) in <u>NET_DVR_StartRemoteConfig</u> .

dwBufLen

[OUT] Buffer size.

pUserData

[OUT] User data.

12.8.8 fStdDataCallBack

Standard stream data callback function.

Callback Function Definition

```
typedef void(CALLBACK *fStdDataCallBack)(
    LONG   lRealHandle,
    DWORD  dwDataType,
    BYTE   *pBuffer,
    DWORD  dwBufSize,
```

```

    DWORD *dwser
);
```

Parameters

IRealHandle

[OUT] Current live view handle.

dwDataType

[OUT] Data type, see details in the following table.

Type Macro Definition	Value	Description
NET_DVR_SYSHEAD	1	System header data.
NET_DVR_STREAMDATA	2	Stream data (including video & audio stream, or video stream).
NET_DVR_STD_VIDEODATA	4	Standard video stream data (standard 264, MPEG4).
NET_DVR_STD_AUDIODATA	5	Standard audio stream data (G722 audio data).
NET_DVR_PRIVATE_DATA	112	Private data, including VCA information.
Others	0x00-G711U Audio, 0x04-G723 Audio, 0x08-G711A Audio, 0x0b-PCM16 Audio, 0x0e-Mpeg(MP2L2) Audio, 0x12-G729 Audio, 0x1A-MJPEG Audio, 0x68-AAC Audio, and so on.	payload type in standard RTP, no transformation. The data of all types should be imported to player SDK when decoding and displaying stream.

pBuffer

[OUT] Pointer of buffer to save data.

dwBufSize

[OUT] Buffer size.

dwUser

[OUT] User data.

Remarks

This callback function is not applicable for the operations or API that may take a long time.

12.8.9 fVoiceDataCallBack

Audi data callback function.

Callback Function Definition

```
typedef void(CALLBACK *fVoiceDataCallBack)(  
    char    *pRecvDataBuffer,  
    DWORD   dwBufSize,  
    void    *pUser  
)
```

Parameters

pRecvDataBuffer

[OUT] Pointer of buffer to save the audio data (PCM) collected from the local PC.

dwBufSize

[OUT] Audio data size.

pUser

[OUT] Pointer of user data.

12.8.10 MSGCallBack

Alarm/event information callback function.

Callback Function Definition

```
typedef void(CALLBACK *MSGCallBack)(  
    LONG      ICommand,  
    NET_DVR_ALARMER *pAlarmer,  
    char      *pAlarmInfo,  
    DWORD     dwBufLen,  
    void      *pUser  
)
```

Parameters

ICommand

[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.

12.8.11 REALDATACALLBACK

Stream data callback function.

Callback Function Definition

```
typedef void(CALLBACK *REALDATACALLBACK)(  
    LONG   IRealHandle,  
    DWORD  dwDataType,  
    BYTE   *pBuffer,  
    DWORD  dwBufSize,  
    void   *pUser  
>;
```

Parameters

IRealHandle

[OUT] Current live view handle, which is returned by [NET_DVR_RealPlay_V40](#) .

dwDataType

[OUT] Data type, see details in the following table.

Type Macro Definition	Value	Description
NET_DVR_SYSHEAD	1	System header data.
NET_DVR_STREAMDATA	2	Stream data (including video & audio stream, or video stream).
NET_DVR_AUDIOSTREAMDATA	3	Audio stream data.
NET_DVR_PRIVATE_DATA	112	Private data, including VCA information.

pBuffer

[OUT] Pointer of buffer to save data.

dwBufSize

[OUT] Buffer size.

pUser

[OUT] User data.

Remarks

This callback function is not applicable for the operations or API that may take a long time.

Appendix A. Data Structure

A.1 DATE_TIME

Date and Time Structure

Member	Data Type	Description
year	short	Year.
month	short	Month.
dayOfWeek	short	Days of the week: 0-Sunday, 1-Monday, 2-Tuesday, 3-Wednesday, 4-Thursday, 5-Friday, 6-Saturday.
day	short	Day.
hour	short	Hour.
minute	short	Minute.
second	short	Second.
millisecond	short	Millisecond.

A.2 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>NET_ALARM_RECORD</u> <u>FILE LOSS</u>	Video loss alarm information, the value of dwAlarmType in <u>NET_DVR_ALARMINFO_DEV_V40</u> is 8.
struStreamException	<u>NET_ALARM_STREAM_EXCEPTION</u>	Streaming exception alarm information, the value of dwAlarmType in <u>NET_DVR_ALARMINFO_DEV_V40</u> is 9.

Member	Data Type	Description
struResourceUsage	<u>NET_ALARM_RESOURCE_USAGE</u>	Resource usage alarm information, the value of dwAlarmType in <u>NET_DVR_ALARMINFO_DEV_V40</u> is 10.
struRecordException	<u>NET_ALARM_RECORD_EXCEPTION</u>	Recording exception alarm information, the value of dwAlarmType in <u>NET_DVR_ALARMINFO_DEV_V40</u> is 12.

A.3 NET_ALARM_RECORD_EXCEPTION

Structure about Recording Exception Alarm Information

Member	Data Type	Description
byReason	BYTE	Exception reason: 0-video volume full, 1-video volume exception, 2-no available video volume.
byRes1	BYTE[]	Reserved, set to 0. The maximum array length is 3 bytes.
sVolumeName	BYTE[]	Video volume name, the maximum array length is "MAX_VOLUMENAME_LEN" (32 bytes).
dwVolumeID	DWORD	Video volume ID, or HDD No.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 452 bytes.

A.4 NET_ALARM_RECORDFILE_LOSS

Structure about Video Loss Alarm Information

Member	Data Type	Description
struInspectStart	<u>NET_DVR_TIME_EX</u>	Start time of video loss check.
struInspectEnd	<u>NET_DVR_TIME_EX</u>	End time of video loss check.
struIP	<u>NET_DVR_IPADDR_UNION</u>	IP address of video loss channel.
dwChanNo	DWORD	Channel No.

Member	Data Type	Description
dwIDIndex	DWORD	Encoder ID.
sName	BYTE[]	Encoder name, the maximum array length is "STREAM_ID_LEN" (32 bytes).
struLossStartTime	<u>NET_DVR_TIME_EX</u>	Start time of video loss.
struLossEndTime	<u>NET_DVR_TIME_EX</u>	End time of video loss.
dwLostNum	DWORD	Number of lost video files, 0xffffffff-all video files are lost.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 240 bytes.

A.5 NET_ALARM_RESOURCE_USAGE

Structure about Resource Usage Alarm Information

Member	Data Type	Description
byLevel	BYTE	Usage alarm level: 0-normal, 1-alarm level 1, 2-alarm level 2, 3-alarm level 3.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 491 bytes.

A.6 NET_ALARM_STREAM_EXCEPTION

Structure about Video Exception Alarm Information

Member	Data Type	Description
struIP	<u>NET_DVR_IPADDR_UNION</u>	IP address of video exception channel.
dwChanNo	DWORD	Channel No.
dwIDIndex	DWORD	Encoder ID.
sName	BYTE[]	Encoder name, the maximum array length is "STREAM_ID_LEN" (32 bytes).

Member	Data Type	Description
byExceptionCase	BYTE	Exception reason: 0-data writing exception, 1-network exception.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 307 bytes.

A.7 NET_DVR_ADDRESS

IP address and port parameter structure.

```
struct{
    NET_DVR_IPADDR struIP;
    WORD      wPort;
    BYTE      byRes[2];
}NET_DVR_ADDRESS,*LPNET_DVR_ADDRESS;
```

Members

struIP

IP address, see details in the structure .

wPort

Port number.

byRes

Reserved.

A.8 NET_DVR_ADVANCE_COND_UNION

Condition union for searching by group.

Structure Definition

```
union{
    BYTE      byLen[36];
    NET_VCA_HUMAN_FEATURE struHumanFeature;
    NET_DVR_OBJECT_FEATURE struObjectFeature;
}NET_DVR_ADVANCE_COND_UNION,*LPNET_DVR_ADVANCE_COND_UNION;
```

Members

byLen

Union size, which is 36bytes.

struHumanFeature

Human body features, refer to the structure NET_VCA_HUMAN_FEATURE in the HCNetSDK user manual of face applications for details.

struObjectFeature

Object features, refer to the structure for details.

See Also

A.9 NET_DVR_AES_KEY_INFO

Encryption key information structure of AES algorithm

Structure Definition

```
struct{
    BYTE      sAESKey[16];
    BYTE      byRes[64];
}NET_DVR_AES_KEY_INFO, *LPNET_DVR_AES_KEY_INFO;
```

Members

sAESKey

Stream encryption key.

byRes

Reserved.

A.10 NET_DVR_AGC_PARAM

AGC parameter structure

Structure Definition

```
struct{
    BYTE  bySceneType;
    BYTE  byLightLevel;
    BYTE  byGainLevel;
    BYTE  byRes[5];
}NET_DVR_AGC_PARAM,*LPNET_DVR_AGC_PARAM;
```

Members

bySceneType

AGC mode: 1-normal scene, 2-highlight scene, 3-manual

byLightLevel

Brightness level, value range: [1,100]; it is valid when **bySceneType** is set to "3".

byGainLevel

Gain level, value range: [1,100]; it is valid when **bySceneType** is set to "3".

byRes

Reserved, set to 0.

A.11 NET_DVR_ALRAM_FIXED_HEADER

Structure About Constant Alarm Information

Member	Data Type	Description
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 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	<u>NET_DVR_TIME_EX</u>	Alarm time
uStruAlarm	Union (<u>Table 13-1</u>)	Alarm information union
pRes	DWORD*	Reserved.
byTimeDiffFlag	BYTE	Whether the time difference parameter is valid: 0-invalid, 1-valid.

Member	Data Type	Description
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 byISO8601 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 byISO8601 is "1".
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 5 bytes.

Table A-1 Union about Alarm Information Structures (uStruAlarm)

Member	Data Type	Description
byUnionLen	Array of BYTE	Union size, which is 116 bytes.
struIOAlarm	Struct (Table 13-2)	Structure about alarm input parameters
struAlarmChannel	Struct (Table 13-3)	Structure about alarm channel parameters
struAlarmHardDisk	Struct (Table 13-4)	Structure about HDD alarm parameters
struRecordingHost	Struct (Table 13-5)	Structure about alarm parameters of education sharing system
struVoltageInstable	Struct (Table 13-6)	Structure about alarm parameters of supply voltage exception
struPTLocking	Struct (Table 13-7)	Structure about parameters of PTZ locked alarm

Table A-2 Structure about Alarm Input Parameters (struIOAlarm)

Member	Data Type	Description
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 pAlarmData in NET_DVR_ALARMINFO_V40 , 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 pAlarmData of

Member	Data Type	Description
		<u>NET_DVR_ALARMINFO_V40</u> , each channel is represented by 4 bytes.

Table A-3 Structure about Alarm Channel Parameters (struAlarmChannel)

Member	Data Type	Description
dwAlarmChanNum	DWORD	The number of alarm channels. It is used for calculating the number of all alarm channels by pAlarmData of <u>NET_DVR_ALARMINFO_V40</u> , 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 Window operating system and 64-bit Linux operating system.

Table A-4 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 pAlarmData of <u>NET_DVR_ALARMINFO_V40</u> , each alarm HDD is represented by 4 bytes.

Table A-5 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	<u>NET_DVR_TIME_EX</u>	Recording end time.

Table A-6 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-7 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.12 NET_DVR_ALARM_HOT_SPARE

Hot spare (N+1 mode) exception alarm information structure

Structure Definition

```
struct{
    DWORD      dwSize;
    DWORD      dwExceptionCase;
    NET_DVR_IPADDR struDeviceIP;
    BYTE       byRes[256];
}NET_DVR_ALARM_HOT_SPARE, *LPNET_DVR_ALARM_HOT_SPARE;
```

Members

dwSize

Structure size

dwExceptionCase

Alarm reason: 0- network exception

struDeviceIP

IP address of exception device, refer to the structure [**NET_DVR_IPADDR_UNION**](#)

byRes

Reserved, set it to 0.

A.13 NET_DVR_ALARM_ISAPI_INFO

Structure about Alarm Information Transmitted Based on Text Protocol

Member	Data Type	Description
pAlarmData	char*	Alarm information based on text protocol (XML or JSON message without binary data).
dwAlarmDataLen	DWORD	Alarm data length.
byDataType	BYTE	Alarm data type: 0-invalid, 1-XML, 2-JSON.
byPicturesNumber	BYTE	The number of pictures (number of pPicPackData returned). When this member is 1, only one structure of <u>NET_DVR_ALARM_ISAPI_PICD</u> ATA will be returned by pPicPackData . When this member is larger than 1, multiple structures of <u>NET_DVR_ALARM_ISAPI_PICD</u> ATA will be returned by pPicPackData .
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 2 bytes.

Member	Data Type	Description
pPicPackData	void*	Alarm picture structure, see <u>NET_DVR_ALARM_ISAPI_PICD ATA</u> for details.
byRes	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.14 NET_DVR_ALARM_ISAPI_PICDATA

Structure about Alarm Picture Data Transmitted Based on Text Protocol

Member	Data Type	Description
dwPicLen	DWORD	Alarm picture data length.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 4 bytes.
szFilename	Array of char	Picture file saving path, including file name. The maximum size is 256 bytes.
pPicData	BYTE*	Pointer that pointing to the uploaded image data.

A.15 NET_DVR_ALARM_RS485CFG

Structure about the RS-485 parameters of the security control panel

Structure Definition

```
struct{
    DWORD dwSize;
    BYTE sDeviceName[NAME_LEN];
    WORD wDeviceType;
    WORD wDeviceProtocol;
    DWORD dwBaudRate;
    BYTE byDataBit;
```

```
BYTE  byStopBit;
BYTE  byParity;
BYTE  byFlowcontrol;
BYTE  byDuplex;
BYTE  byWorkMode;
BYTE  byChannel;
BYTE  bySerialType;
BYTE  byMode;
BYTE  byOutputDataType;
BYTE  byAddress;
BYTE  byStairsOutputDataType;
BYTE  byRes[32];
}NET_DVR_ALARM_RS485CFG, *LPNET_DVR_ALARM_RS485CFG;
```

Members

dwSize

Structure size

sDeviceName

Front-end device name

wDeviceType

Front-end device type

wDeviceProtocol

Front-end device protocol, got by NET_DVR_GetDeviceProtoList

dwBaudRate

Baud rate(bps), 0- 50, 1- 75, 2- 110, 3- 150, 4- 300, 5- 600, 6- 1200, 7- 2400, 8- 4800, 9- 9600,
10- 19200, 11- 38400, 12- 57600, 13- 76800, 14- 115.2k

byDataBit

byDataBit

byStopBit

Stop bit: 0- 1 bit, 1- 2 bits

byParity

Parity or not: 0- none, 1- odd parity, 2- even parity

byFlowcontrol

Flow control or not: 0-none, 1-software flow control, 2- hardware flow control

byDuplex

0- half duplex, 1- full duplex

byWorkMode

Work mode: 0- console, 1- transparent channel, 2- Elevator control, 3- Card reader, 4- Access control module

byChannel

RS485 channel No

bySerialType

Serial port type: 0- 485, 1- 232

byMode

Connection mode: 0- Connects to the card reader, 1- Connects to the client, 2- Connects to the extension module, 3- Connects to the access controller, 4-Connects to the elevator controller, 0xffff- invalid.

byOutputDataType

Output data type of the card reader: 1 (invalid), 2 (output card No.), 3 (output employee No.).

This member is valid when the value of the member **byMode** is 0.

byAddress

Address of the serial port.

byStairsOutputDataType

Output data type of the elevator controller: 0 (invalid), 1 (output floor No.), 2 (output card No.).

This member is valid when the value of the member **byMode** is 4.

byRes

Reserved. The member size is 32 bytes.

A.16 NET_DVR_ALARM_SEARCH_COND

Structure about alarm search condition parameters.

Structure Definition

```
struct{
    DWORD      dwSize;
    NET_DVR_TIME_SEARCH_COND strStartTime;
    NET_DVR_TIME_SEARCH_COND strStopTime;
    DWORD      dwAlarmComm;
    char       sAlarmUID[64];
    WORD       wEventType;
    WORD       wSubEventType;
    BYTE       bySupport;
    BYTE       byNoBoundary;
    BYTE       byRes[122];
}NET_DVR_ALARM_SEARCH_COND, *LPNET_DVR_ALARM_SEARCH_COND;
```

Members

dwSize

Structure size.

strStartTime

Start time, see details in the structure [**NET_DVR_TIME_SEARCH_COND**](#). If this member is NULL, it indicates that the start time is not a filtering condition.

strStopTime

End time, see details in the structure [**NET_DVR_TIME_SEARCH_COND**](#). If this member is NULL, it indicates that the end time is not a filtering condition.

dwAlarmComm

Alarm command, currently the following commands are supported: "COMM_VCA_ALARM" (0x4993), "COMM_UPLOAD_FACESNAP_RESULT" (0x1112), and "COMM_SNAP_MATCH_ALAR" (0x2902). If this member is NULL, it indicates that the alarm command is not a filtering condition.

sAlarmUID

UID which is returned by the device when uploading alarms, and its length is 64 bytes. The UID can be made up of time (it should be accurate to millisecond) and random number.

wEventType

Event type: 0-all events, 1-multi-target-type detection (mixedTargetDetection), 2-smart radar detection and video analysis (radarVideoDetection). This member is valid when **dwAlarmComm** is "COMM_VCA_ALARM" (0x4993).

wSubEventType

Sub event type: 0-all sub events, other values depend on the member **wEventType**.

bySupport

Whether to enable uploading for function demonstration: 0-no, 1-yes. Bit0 refers to the multi-target-type detection alarm (mixedTargetDetection). Some smart radar detection and video analysis cameras do not support uploading multi-target-type detection alarm by default, and this member should be set to 1 for enabling uploading and function demonstration. For other devices, this member does not need to be parsed. If the device supports uploading multi-target-type detection alarms, the alarms can be uploaded directly.

byNoBoundary

Whether form data is contained in the returned alarm information: 0-form data is contained, 1-form data is not contained. This member is valid when **dwAlarmComm** is "COMM_VCA_ALARM" (0x4993).

byRes

Reserved.

A.17 NET_DVR_ALARM_SEARCH_RESULT

Structure about alarm search result parameters.

Structure Definition

```
struct{
    DWORD      dwSize;
    DWORD      dwAlarmComm;
    NET_DVR_ALARMER struAlarmer;
    DWORD      dwAlarmLen;
    char       *pAlarmInfo;
    BYTE       byRes[128];
}NET_DVR_ALARM_SEARCH_RESULT, *LPNET_DVR_ALARM_SEARCH_RESULT;
```

Members

dwSize

Structure size.

dwAlarmComm

Alarm command, currently the following commands are supported: "COMM_VCA_ALARM" (0x4993), "COMM_UPLOAD_FACESNAP_RESULT" (0x1112), and "COMM_SNAP_MATCH_ALAR" (0x2902). If this member is NULL, it indicates that the alarm command is not a filtering condition.

struAlarmer

Alarm device information, see details in the structure [NET_DVR_ALARMER](#).

dwAlarmLen

Data length of alarm or event information.

pAlarmInfo

Alarm or event information.

byRes

Reserved.

A.18 NET_DVR_ALARMHOST_NETCFG_V50

Structure about parameters of alarm receiving center.

Structure Definition

```
struct{
    DWORD      dwSize;
    NET_DVR_ALARMHOST_NETPARAM_V50   struNetCenter[MAX_CENTERNUM/*4*/];
    BYTE       byRes1[128];
}NET_DVR_ALARMHOST_NETCFG_V50, *LPNET_DVR_ALARMHOST_NETCFG_V50;
```

Members

dwSize

Structure size.

struNetCenter

Parameters of alarm receiving center, up to 4 alarm receiving centers can be set for security control panel and only 1 alarm receiving center can be set for access controller, refer to the structure [**NET_DVR_ALARMHOST_NETPARAM_V50**](#) for details.

byRes1

Reserved.

A.19 NET_DVR_ALARMHOST_NETPARAM_V50

Network parameter structure of alarm receiving center.

Structure Definition

```
struct{
    DWORD      dwSize;
    NET_DVR_IPADDR struIP;
    WORD       wPort;
    BYTE       byAddressType;
    BYTE       byEnable;
    BYTE       byDomainName[MAX_DOMAIN_NAME/*64*/];
    BYTE       byReportProtocol;
    BYTE       byDevID[ACCOUNTNUM_LEN_32/*32*/];
    BYTE       byProtocolVersion;
    BYTE       byRes1[3];
    BYTE       byEHomeKey[NET_SDK_EHOME_KEY_LEN/*32*/];
    BYTE       byRes2[28];
}NET_DVR_ALARMHOST_NETPARAM_V50, *LPNET_DVR_ALARMHOST_NETPARAM_V50;
```

Members

dwSize

Structure size.

struIP

IP address of upload center, valid when **byAddressType** is 0 or 1.

wPort

Port No. of upload center.

byAddressType

Address type: 0-no meaning, 1-ipv4/ipv6 address, 2-domain name.

byEnable

Enable the upload center: 0-no, 1-yes.

byDomainName

Domain name, valid when **byAddressType** is 2.

byReportProtocol

Report protocol: 1-private, 2-NAL2300, 3-eHome (access controller only supports EHome).

byDevID

Device ID, valid for security control panel when protocol is NAL2300, and valid length is 9; for access controller, the valid length is 32.

Valid characters: 0 to 9, a to f, A to F.

byProtocolVersion

EHome protocol version: 0-invalid, 1-V2.0, 2-V4.0, 3-V5.0.

byRes1

Reserved.

byEHomeKey

EHome key, which is used for EHome protocol version 5.0, can only be configured and cannot be obtained.

byRes2

Reserved.

A.20 NET_DVR_ALARMINCFG_V40

Alarm input parameters (expended) structure.

Structure Definition

```
struct{
    WORD      dwSize;
    BYTE      sAlarmInName[NAME_LEN/*32*/];
    BYTE      byAlarmType;
    BYTE      byAlarmInHandle;
    BYTE      byChannel;
    BYTE      byInputType;
    WORD      dwHandleType;
    WORD      dwMaxRelAlarmOutChanNum;
    WORD      dwRelAlarmOutChanNum;
    WORD      dwRelAlarmOut[MAX_ALARMOUT_V40/*4128*/];
    NET_DVR_SCHEDTIME struAlarmTime[MAX_DAYS/*7*/][MAX_TIMESEGMENT_V30/*8*/];
    WORD      dwMaxRecordChanNum;
    WORD      dwCurRecordChanNum;
    WORD      dwRelRecordChan[MAX_CHANNUM_V40/*512*/];
    WORD      dwMaxEnablePtzCtrlNun;
```

```

DWORD      dwEnablePresetChanNum;
NET_DVR_PRESETCHAN_INFO struPresetChanInfo[MAX_CHANNUM_V40/*512*/];
BYTE       byPresetDurationTime[MAX_CHANNUM_V40/*512*/];
BYTE       byRes2[4];
DWORD      dwEnableCruiseChanNum;
NET_DVR_CRUISECHAN_INFO struCruiseChanInfo[MAX_CHANNUM_V40/*512*/];
DWORD      dwEnablePtzTrackChanNum;
NET_DVR_PTZTRACKCHAN_INFO struPtzTrackInfo[MAX_CHANNUM_V40/*512*/];
WORD       wEventType[NET_SDK_MAX_EVENT_NUM/*64*/];
BYTE       byRes[128];
}NET_DVR_ALARMINCFG_V40,*LPNET_DVR_ALARMINCFG_V40;

```

Members

dwSize

Structure size

sAlarmInName

Alarm input name

byAlarmType

Alarm device type: 0-NO, 2- NC

byAlarmInHandle

Whether to handle the alarm? 0- No, 1- Yes

byChannel

Alarm input triggered smart recognition channel

byInputType

Input type: 0-relay (corresponding to sensor alarm type), 1-pulse mode (corresponding to pulse alarm type)

dwHandleType

Exception handling

0x00: no response

0x01: monitor warning

0x02: audible warning

0x04: notify the surveillance center

0x08: trigger alarm output

0x10: send Jpeg snapshot by E-mail

0x20: interact with wireless sound and light alarm

0x40: e-map linkage (supported by PCNVR only)

0x200: capture image and upload to FTP

E.g. dwHandleType==0x01|0x04, it indicates that when alarm triggered, there is monitor warning, and alarm is uploaded to center

dwMaxRelAlarmOutChanNum

The maximum channel output of devices triggered by alarm (read-only)

dwRelAlarmOutChanNum

The number of output channels triggered by alarm

dwRelAlarmOut

The output channel No. triggered by alarm, for example, when dwRelAlarmOut[8]=0xffffffff, it indicated that configure dwRelAlarmOut[0] to dwRelAlarmOut[7] alarm output channel.

struAlarmTime

Arming time, 7days/week, 8 time segments/day, see details in structure .

dwMaxRecordChanNum

Max. number of device linked recording channel (read-only)

dwRelRecordChan

Actually triggered recording channels

dwMaxEnablePtzCtrlNun

The maximum number of available presets

dwEnablePresetChanNum

The number of used presets

struPresetChanInfo

Channel that enable preset function, when **dwEnablePresetChanNum** is 0, it indicates that no preset enabled, and when **dwEnablePresetChanNum** is not 0, it indicates that struPresetChanInfo[0]~struPresetChanInfo [dwEnablePresetChanNum-1] are valid. See details in structure .

byPresetDurationTime

Preset dwell time, ranges from 0 to 20, unit: second, the default time is 10s

byRes2

Reserved, set to 0

dwEnableCruiseChanNum

Number of enabled patrols

struCruiseChanInfo

Channel that enabled patrol function:when **dwEnableCruiseChanNum** is not 0, struCruiseChanInfo[0]~struCruiseChanInfo [dwEnableCruiseChanNum-1] is valid, when **dwEnableCruiseChanNum** is 0, the patrol function of channel is not enabled. See details in structure .

dwEnablePtzTrackChanNum

Number of enabled patterns

struPtzTrackInfo

Channel that enabled pattern function: when **dwEnablePtzTrackChanNum** is not 0, struPtzTrackInfo[0]~struPtzTrackInfo[dwEnablePresetChanNum-1] is valid, when **dwEnablePtzTrackChanNum** is 0, pattern function of channel is not enabled. See details in structure

wEventType

Composite event type, each bit corresponds to one event type.

byRes

Reserved, set to 0.

A.21 NET_DVR_ALARMINFO_DEV

Device Alarm Information Structure

Memeber	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.
struTime		Alarm time
byRes	Array of BYTE	Reserved, set to 0.
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.22 NET_DVR_ALARMINFO_DEV_V40

Structure about CVR Alarm Information

Member	Data Type	Description
dwAlarmType	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, 11-CVR exception recovered, 12-recording exception.
struTime	<u>NET_DVR_TIME</u>	Alarm time
uSubAlarmInfo	<u>NET_ALARM_CVR_SU_BINFO_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.23 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

Member	Data Type	Description
		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 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.24 NET_DVR_ALARMINFO_V40

Structure About Uploaded Alarm Information

Member	Data Type	Description
struAlarmFixedHeader	<u>NET_DVR_ALRAM_FIXED_HEADER</u>	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-8 Relations Between pAlarmData and dwAlarmType

dwAlarmType	Description	pAlarmData
0, 23	Alarm input alarm, pulse alarm	dwTrigerAlarmOutNum*(DWORD) Alarm output No., +dwTrigerRecordChanNum*(DWORD) 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*(DWORD) 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.25 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 <u>NET_DVR_Login_V40</u> , 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.26 NET_DVR_ALARMOUTCFG_V30

Alarm output parameters structure.

Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       sAlarmOutName[NAME_LEN/*32*/];
    DWORD      dwAlarmOutDelay;
    NET_DVR_SCHEDTIME struAlarmOutTime[MAX_DAYS/*7*/][MAX_TIMESEGMENT_V30/*8*/];
    BYTE       byAlarmOutHandle;
    BYTE       byNormalStatus;
    BYTE       byRes[14];
}NET_DVR_ALARMOUTCFG_V30,*LPNET_DVR_ALARMOUTCFG_V30;
```

Members

dwSize

Structure size

sAlarmOutName

Name

dwAlarmOutDelay

Alarm output delay: 0-"5s", 1-"10s", 2-"30s", 3-"1 minute", 4-"2 minutes", 5-"5 minutes", 6-"10 minutes", 7-manual, the default value is 5 minutes.

struAlarmOutTime

Time period of activating alarm output, see details in .

byAlarmOutHandle

Whether to handle alarm: 0-yes, 1-no

byNormalStatus

Alarm output type: 0-remain open, 1-remain closed

byRes

Reserved, set as 0.

A.27 NET_DVR_AREA_SMARTSEARCH_COND_UNION

VCA search condition union.

Structure Definition

```
union{
    BYTE      byLen[6144];
    BYTE      byMotionScope[64][96];
    NET_DVR_TRAVERSE_PLANE_SEARCHCOND struTraversPlaneCond;
    NET_DVR_INTRUSION_SEARCHCOND     struIntrusionCond;
```

```
NET_DVR_FACEDETECTION_SEARCHCOND struFaceSnapCond;  
}NET_DVR_AREA_SMARTSEARCH_COND_UNION,*LPNET_DVR_AREA_SMARTSEARCH_COND_UNION;
```

Members

byLen

Union size, which is 6144 bytes.

byMotionScope

Motion detection region, the array size is 96*64, for P video standard, only 22*18 array is valid, for N video standard, only 22*15 array is valid. Value: 1-set as motion detection region, 0-not set as motion detection region.

struTraversPlaneCond

Conditions for searching videos recorded based on line crossing detection, refer to the structure for details.

struIntrusionCond

Conditions for searching videos recorded based on intrusion detection, refer to the structure for details.

struFaceSnapCond

Conditions for searching videos recorded based on face detection, refer to the structure for details.

Remarks

- For P video standard, the image with the size 704*576 is divided into 22*18 blocks; for N video standard, the image with the size 704*480 is divided in to 22*15 blocks. And then set each block as motion detection region or not.
- For P video standard, if the image size is 1280*720, when setting motion detection region, the image should be zoomed out to 704*576.

See Also

A.28 NET_DVR_ATMFINDINFO

Condition structure for searching files with ATM information.

Structure Definition

```
struct{  
    BYTE byTransactionType;  
    BYTE byRes[3];  
    DWORD dwTransationAmount;  
}NET_DVR_ATMFINDINFO, *LPNET_DVR_ATMFINDINFO;
```

Members

byTransactionType

Transaction type: 0-all, 1-search, 2-withdraw, 3-deposit, 4-change password, 5-transfer, 6-search without card, 7-deposit without card, 8-retract cash, 9-retract card, 10-custom.

byRes

Reserved, set to 0.

dwTransactionAmount

Transaction amount, -1: search without displaying amount.

See Also

[NET_DVR_SPECIAL_FINDINFO_UNION](#)

A.29 NET_DVR_AUDIOENC_INFO

Audio encoder information structure

Structure Definition

```
struct{
    DWORD  in_frame_size;
    int    reserved[16];
}NET_DVR_AUDIOENC_INFO, LPNET_DVR_AUDIOENC_INFO;
```

Members

in_frame_size

Data size of an input frame, unit: byte.

reserved

Reserved.

A.30 NET_DVR_AUDIODEC_INFO

Audio decoding information structure

Structure Definition

```
struct{
    int  nchans;
    int  sample_rate;
    int  aacdec_profile;
```

```
int reserved[16];
}NET_DVR_AUDIODEC_INFO, LPNET_DVR_AUDIODEC_INFO;
```

Members

nchans

Number of audio channels.

sample_rate

Sampling rate.

aacdec_profile

Reserved.

reserved

Reserved.

A.31 NET_DVR_AUDIOENC_PROCESS_PARAM

Audio encoding parameter structure

Structure Definition

```
struct{
    unsigned char      *in_buf;
    unsigned char      *out_buf;
    DWORD             out_frame_size;
    int                g726enc_reset;
    int                g711_type;
    ENUM_AUDIO_ENC_MODE enc_mode;
    int                reserved[16];
}NET_DVR_AUDIOENC_PROCESS_PARAM, LPNET_DVR_AUDIOENC_PROCESS_PARAM;
```

Members

in_buf

Input data buffer, and its size should be larger than or equal to **in_frame_size** in the structure of .

out_buf

Output data buffer, its size should be larger than or equal to input data buffer (**in_buf**).

out_frame_size

Size of an encoded frame, unit: byte.

g726enc_reset

Resetting switch.

g711_type

G711 encoding type: 0-U law, 1-A law. This parameter is for G711 audio encoding only.

enc_mode

Audio encoding mode.

reserved

Reserved.

A.32 NET_DVR_AUDIODEC_PROCESS_PARAM

Audio decoding parameter structure

Structure Definition

```
struct{
    unsigned char      *in_buf;
    unsigned char      *out_buf;
    DWORD             in_data_size;
    DWORD             proc_data_size;
    DWORD             out_frame_size;
    NET_DVR_AUDIODEC_INFO dec_info;
    int               g726dec_reset;
    int               g711_type;
    int               reserved[16];
}NET_DVR_AUDIODEC_PROCESS_PARAM, LPNET_DVR_AUDIODEC_PROCESS_PARAM;
```

Members

in_buf

Input data buffer, the recommended size for input data buffer is 8192 bytes.

out_buf

Output data buffer, the recommended size for output data buffer is 8192 bytes.

in_data_size

Size of input data buffer, unit: byte.

proc_data_size

Size of processed data in decoding library, unit: byte. If the value of this parameter is smaller than that of **in_data_size**, it indicates that the data processing is not completed.

out_frame_size

Size of a decoded frame, unit: byte.

dec_info

Decoding information, see details in the structure .

g726dec_reset

Resetting switch.

g711_type

G711 encoding type: 0-U law, 1-A law. This parameter is for G711 decoding only.

reserved

Reserved.

A.33 NET_DVR_AUDIOEXCEPTION_ALARM

Audio exception alarm information structure

Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byAlarmType;
    BYTE       byRes1;
    WORD       wAudioDecibel;
    NET_VCA_DEV_INFO struDevInfo;
    WORD       wDevInfolvmsChannelEx;
    BYTE       byRes[62];
}NET_DVR_AUDIOEXCEPTION_ALARM, *LPNET_DVR_AUDIOEXCEPTION_ALARM;
```

Members

dwSize

Structure size

byAlarmType

Alarm type: 1-audio loss detection, 2-sudden increase of sound intensity, 3-sudden decrease of sound intensity

byRes1

Reserved, set to 0.

wAudioDecibel

Sound intensity (valid when **byAlarmType** is 2 or 3), which is ranging from 0 to 100.

struDevInfo

Alarm device information, refer to the structure [NET_VCA_DEV_INFO](#)

wDevInfolvmsChannelEx

Extended channel No., and the maximum channel No. is 255.

byRes

Reserved, set to 0.

A.34 NET_VCA_AUDIO_ABNORMAL

Table A-9 Structure about Detection Parameters for Sudden Increase of Sound Intensity

Member	Data Type	Description
wDecibel	WORD	Sound intensity.
bySensitivity	BYTE	Sensitivity, value range: [1,100].
byAudioMode	BYTE	Sound detection mode: 0-enable sensitivity detection, 1-enable decibel threshold detection, 2-enable both.
byEnable	BYTE	Whether to enable sudden increase of sound intensity detection: 0-no, 1-yes.
byThreshold	BYTE	Sound intensity threshold, value range: [0,100].
byRes	Array [BYTE]	Reserved, set to 0. The maximum size is 54 bytes.

A.35 NET_DVR_AUDIO_CHANNEL

Audio channel information structure

Structure Definition

```
struct{
    DWORD dwChannelNum;
    BYTE byres[32];
}NET_DVR_AUDIO_CHANNEL, *LPNET_DVR_AUDIO_CHANNEL;
```

Members

dwChannelNum

Channel No.

byres

Reserved, set to 0.

A.36 NET_DVR_AUDIO_EXCEPTION

Table A-10 Structure about Audio Exception Detection Parameters

Member	Data Type	Description
dwSize	DWORD	Structure size.
byEnableAudioInException	BYTE	Whether to enable audio exception detection alarm: 0-no, 1-yes.
byRes1	Array [BYTE]	Reserved, set to 0. The maximum size is 3 bytes.
struAudioAbnormal	<u>NET_VCA_AUDIO_ABNORMAL</u>	Parameters of sudden increase of sound intensity detection.
struAlarmSched	<u>NET_DVR_SCHEDTIME</u>	Arming schedule. Up to 7 (the value of the macro definition "MAX_DAYS") days can be configured with arming schedules, and up to 8 (the value of the macro definition "MAX_TIMESEGMENT_V30") time periods can be configured for each day.
struHandleException	<u>NET_DVR_HANDLEEXCEPTION_V41</u>	Exception handling method.
dwMaxRelRecordChanNum	DWORD	The maximum number of recording channels (read-only) that can be triggered by the alarm.
dwRelRecordChanNum	DWORD	The actually supported number of recording channels that can be triggered by the alarm.
byRelRecordChan	Array [DWORD]	Recording channels triggered by the alarm. The supported maximum number of channels is 64. The first dwRelRecordChanNum elements in the array indicates the numbers of recording

Member	Data Type	Description
		channels. For example, when dwRelRecordChanNum is 5, you can configure dwRelRecordChan[0] to dwRelRecordChan[4] as the recording channels.
struAudioSteepDrop	<i>NET_DVR_AUDIO_STEEP_DRO</i> <i>P</i>	Parameters of sudden decrease of sound intensity detection.
byRes2	Array [BYTE]	Reserved, set to 0. The maximum size is 24 bytes.

A.37 NET_DVR_AUDIO_STEEP_DROP

Table A-11 Structure about Parameters for Sudden Decrease of Sound Intensity Detection

Member	Data Type	Description
bySensitivity	BYTE	Sensitivity, value range: [1,100].
byEnable	BYTE	Whether to enable sudden decrease of sound intensity detection: 0-no, 1-yes.
byRes	Array [BYTE]	Reserved, set to 0. The maximum size is 6 bytes.

See Also

[NET_DVR_AUDIO_EXCEPTION](#)

A.38 NET_DVR_AUXILIARY_DEV_UPGRADE_PARAM

Auxiliary device upgrade structure.

Structure Definition

```
struct{
    DWORD dwSize;
    DWORD dwDevNo;
    BYTE byDevType;
    BYTE byRes[131];
}NET_DVR_AUXILIARY_DEV_UPGRADE_PARAM,*LPNET_DVR_AUXILIARY_DEV_UPGRADE_PARAM;
```

Members

dwSize

Structure size

dwDevNo

Device No.

byDevType

Upgraded device type: 0-keyboard, 1-mechanical movement, 2-network module, 3-router, 4-zone, 5-RS-485 wireless extension module

byRes

Reserved, set to 0.

A.39 NET_DVR_BACKLIGHT

Backlight compensation (BLC) parameter structure

Structure Definition

```
struct{
    BYTE   byBacklightMode;
    BYTE   byBacklightLevel;
    BYTE   byRes1[2];
    DWORD  dwPositionX1;
    DWORD  dwPositionY1;
    DWORD  dwPositionX2;
    DWORD  dwPositionY2;
    BYTE   byRes2[4];
}NET_DVR_BACKLIGHT, *LPNET_DVR_BACKLIGHT;
```

Members

byBacklightMode

BLC mode: 0-disabled, 1-top, 2-bottom, 3-left, 4-right, 5-middle, 6-custom, 10-enable, 11-auto, 12-BLC in multiple areas.

byBacklightLevel

BLC level, which is between 0x0 and 0xF.

byRes1

Reserved, set to 0.

dwPositionX1

X-coordinate of position 1.

dwPositionY1

Y-coordinate of position 1.

dwPositionX2

X-coordinate of position 2.

dwPositionY2

Y-coordinate of position 2.

byRes2

Reserved, set to 0.

A.40 NET_DVR_BEHAVIOR_INFO

Structure about abnormal event detection picture information.

Structure Definition

```
struct{
    NET_VCA_RECT struVcaRect;
    BYTE     byRes2[240];
}NET_DVR_BEHAVIOR_INFO, *LPNET_DVR_BEHAVIOR_INFO;
```

Members

struVcaRect

Thumbnail coordinates (normalized), refer to the structure for details.

byRes2

Reserved, set to 0.

See Also

A.41 NET_DVR_BEHAVIOUR_COND

Behavior analysis condition structure.

Structure Definition

```
struct{
    DWORD dwSearchType;
    BYTE  byHumanMisinfo;
    BYTE  byRes[251];
}NET_DVR_BEHAVIOUR_COND, *LPNET_DVR_BEHAVIOUR_COND;
```

Members

dwSearchType

Search type, represented by bit, value: 0-no, 1-yes.

- dwSearchType&0x1: Line Crossing Detection
- dwSearchType&0x2: Intrusion Detection
- dwSearchType&0x4: Unattended Baggage
- dwSearchType&0x8: Object Removal
- dwSearchType&0x10: Region Entrance
- dwSearchType&0x20 : Region Exiting
- dwSearchType&0x40: Illegal Parking
- dwSearchType&0x80: Loitering
- dwSearchType&0x100: People Gathering
- dwSearchType&0x200: Fast Moving
- dwSearchType&0x400: Absence Detection
- dwSearchType&0x800: People Number Changed
- dwSearchType&0x1000: Violent Motion Detection
- dwSearchType&0x2000: People Running
- dwSearchType&0x4000: People Falling Down
- dwSearchType&0x8000: Overstay Detection
- dwSearchType&0x10000: Situation Analysis

byHumanMisinfo

Whether to get false alarms only: 0-no, 1-yes. For intrusion detection, line crossing detection, region entrance detection, and region exiting detection, if filtering false alarms is enabled, the device will store false alarms locally without uploading.

byRes

Reserved.

Remarks

For functions like intrusion detection, line crossing detection, and region exiting detection, if false alarm prevention is enabled, the false alarm will be filtered and stored locally instead of being uploaded.

A.42 NET_DVR_BINOC_RECTIFY_PARAM

Structure about correction parameters of the dual-lens image.

Structure Definition

```
struct{
    float fCamInternalMatrix[3][3];
    float fDistCoeffs[8];
    float fRotateMatrix[3][3];
    float fProjectMatrix[3][4];
    BYTE byRes[64];
}NET_DVR_BINOC_RECTIFY_PARAM, *LPNET_DVR_BINOC_RECTIFY_PARAM;
```

Members

fCamInternalMatrix

Matrix of camera internal parameters (including focal length and principle point coordinates).

fDistCoeffs

Lens distortion coefficient.

fRotateMatrix

Rotation matrix after correction.

fProjectMatrix

Projection matrix after correction.

byRes

Reserved.

A.43 NET_DVR_BUF_INFO

Structure about buffer information.

Structure Definition

```
struct{
    void* pBuf;
    DWORD nLen;
}NET_DVR_BUF_INFO,*LPNET_DVR_BUF_INFO;
```

Members

pBuf

Buffer pointer

nLen

Buffer size

See Also

[NET_DVR_IN_PARAM](#)

NET_DVR_OUT_PARAM**A.44 NET_DVR_BUILTIN_SUPPLEMENTLIGHT**

Configuration parameter structure of supplement light

Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byMode;
    BYTE       byBrightnessLimit;
    BYTE       bySupplementLightMode;
    BYTE       byMixedLightRegulatMode;
    BYTE       byLrLightBrightness;
    BYTE       byHighLrLightBrightness;
    BYTE       byHighBrightnessLimit;
    BYTE       byLowLrLightBrightness;
    NET_DVR_SCHEDULE_DAYTIME struSchedTime;
    BYTE       byLowBrightnessLimit;
    BYTE       byWhiteLightBrightness;
    BYTE       byIrLightbrightnessLimit;
    BYTE       byWhiteLightbrightnessLimit
    BYTE       byRes1[252];
}NET_DVR_BUILTIN_SUPPLEMENTLIGHT, *LPNET_DVR_BUILTIN_SUPPLEMENTLIGHT;
```

Members**dwSize**

Structure size.

byMode

Supplement light mode: 0-scheduled, 1-enabled, 2-disabled, 3-auto (non-photosensitive)

byBrightnessLimit

Brightness range of supplement light: [0,100]; it is available only when **byMode** is set to 0 or 1, or **byMixedLightRegulatMode** is set to 1.

bySupplementLightMode

Light type: 0-white light, 1-hybrid light

byMixedLightRegulatMode

Brightness control mode of hybrid supplement light: 0-auto, 1-manual; it is available when **bySupplementLightMode** is set to 1.

byLrLightBrightness

Brightness range of IR supplement light: [0,100]; it is available when **byMixedLightRegulatMode** is set to 1.

byHighLrLightBrightness

Brightness range of far IR light; it is available when **byMixedLightRegulatMode** is set to 1.

byHighBrightnessLimit

Brightness range of far white light; it is available when **byMixedLightRegulatMode** is set to 1.

byLowLrLightBrightness

Brightness range of far near light; it is available when **byMixedLightRegulatMode** is set to 1.

struCchedTime

Scheduled time period (refer to the structure NET_DVR_SCHEDULE_DAYTIME for **NET_DVR_SCHEDULE_DAYTIME** details), during which, the supplement light is enabled automatically; it is available when **byMode** is set to 0.

byLowBrightnessLimit

Brightness range of near white light; it is available when **byMixedLightRegulatMode** is set to 1.

byWhiteLightBrightness

White light brightness

byIrLightbrightnessLimit

IR light brightness limit

byWhiteLightbrightnessLimit

White light brightness limit

byRes1

Reserved, set to 0.

A.45 NET_DVR_BV_CORRECT_PARAM

Structure about the correction parameters of the left camera and the right camera.

Structure Definition

```
struct{
    DWORD      dwSize;
    float       fReprojectMatrix[4][4];
    NET_DVR_BINOC_RECTIFY_PARAM struLCamParam;
    NET_DVR_BINOC_RECTIFY_PARAM struRCamParam;
    BYTE       byLensType;
    BYTE       byRes1[3];
    float      fRotateMatrix[3][3];
    float      fTransMatrix[3];
    DWORD      dwOrilImgWidth;
    DWORD      dwOrilImgHeight;
    BYTE       byRes[196];
}NET_DVR_BV_CORRECT_PARAM,*LPNET_DVR_BV_CORRECT_PARAM;
```

Members

dwSize

Structure size.

fReprojectMatrix

Reprojection matrix.

struLCamParam

Correction parameters of the left camera, see details in the structure

[NET_DVR_BINOC_RECTIFY_PARAM](#).

struRCamParam

Correction parameters of the right camera, see details in the structure

[NET_DVR_BINOC_RECTIFY_PARAM](#).

byLensType

Focal length of the lens: 0—"unknown", 1—"8mm", 2—"12mm", 3—"16mm", 4—"25mm", 5—"35mm", 6—"50mm", 7—"4mm", 8—"6mm", 9—"2.0mm", 10—"2.8mm", 11—"4.3mm".

byRes1

Reserved.

fRotateMatrix

Rotation matrix of the left camera relative to the right camera after dual-lens calibration.

fTransMatrix

Translation matrix. **RotateMatrix** and **TransMatrix** are used to describe the position of the left camera relative to the right camera

dwOrIImgWidth

Width of the original calibrated image. For example, if the resolution of the image is 1920*1080, the value of this member is 1920.

dwOrIImgHeight

Height of the original calibrated image. For example, if the resolution of the image is 1920/1080, the value of this member is 1080.

byRes

Reserved.

A.46 NET_DVR_CAMERAPARAMCFG

Camera parameter structure

Structure Size

```
struct{
    DWORD      dwSize;
    NET_DVR_VIDEOEFFECT struVideoEffect;
    NET_DVR_GAIN struGain;
    NET_DVR_WHITEBALANCE struWhiteBalance;
    NET_DVR_EXPOSURE struExposure;
    NET_DVR_GAMMACORRECT struGammaCorrect;
    NET_DVR_WDR struWdr;
    NET_DVR_DAYNIGHT struDayNight;
    NET_DVR_BACKLIGHT struBackLight;
    NET_DVR_NOISEREMOVE struNoiseRemove;
    BYTE      byPowerLineFrequencyMode;
    BYTE      byIrisMode;
    BYTE      byMirror;
    BYTE      byDigitalZoom;
    BYTE      byDeadPixelDetect;
    BYTE      byBlackPwl;
    BYTE      byEptzGate;
    BYTE      byLocalOutputGate;
    BYTE      byCoderOutputMode;
    BYTE      byLineCoding;
    BYTE      byDimmerMode;
    BYTE      byPaletteMode;
    BYTE      byEnhancedMode;
    BYTE      byDynamicContrastEN;
    BYTE      byDynamicContrast;
    BYTE      byJPEGQuality;
    NET_DVR_CMOSMODECFG struCmosModeCfg;
    BYTE      byFilterSwitch;
    BYTE      byFocusSpeed;
    BYTE      byAutoCompensationInterval;
    BYTE      bySceneMode;
}NET_DVR_CAMERAPARAMCFG, *LPNET_DVR_CAMERAPARAMCFG;
```

Members

dwSize

Structure size.

struVideoEffect

Video effect parameters, refer to the structure [NET_DVR_VIDEOEFFECT](#) for details.

struGain

Gain parameters, refer to the structure [NET_DVR_GAIN](#) for details.

struWhiteBalance

White balance parameters, refer to the structure [NET_DVR_WHITEBALANCE](#) for details.

struExposure

Exposure parameters, refer to the structure [NET_DVR_EXPOSURE](#) for details.

struGammaCorrect

Gamma correction parameters, refer to the structure [NET_DVR_GAMMACORRECT](#) for details.

struWdr

Wide dynamic range (WDR) parameters, refer to the structure [NET_DVR_WDR](#) for details.

struDayNight

Day/night auto switch parameters, refer to the structure [NET_DVR_DAYNIGHT](#) for details.

struBackLight

Backlight compensation (BLC) parameters, refer to the structure [NET_DVR_BACKLIGHT](#) for details.

struNoiseRemove

Digital noise reduction parameters, refer to the structure [NET_DVR_NOISEREMOVE](#) for details.

byPowerLineFrequencyMode

Power supply frequency: 0-50 Hz; 1-60 Hz.

byIrisMode

Iris mode: 0: auto; 1: manual, 2: P-Iris1, 3: Union 3-9mm F1.6-2.7 (T5280-PQ1) [IPC5.1.7], 4: Union 2.8-12mm F1.6-2.7 (T5289-PQ1) [IPC5.1.7], 5: private 3.8-16mm F1.5(HV3816P-8MPIR), 6: private 11-40mm F1.7 (HV1140P-8MPIR), 7: private 2.7-12mm F1.2(TV2712P-MPIR), 8: MZ5721D-12MPIR, 9: MZ1555D-12MPIR, 10: MZ5721D-12MPIR(RS485), 11: MZ1555D-12MPIR(RS485).

byMirror

Mirror: 0-off, 1-left and right, 2-top and bottom, 3-center.

byDigitalZoom

Digital zoom: 0: enable, 1: ×2, 2: ×4.

byDeadPixelDetect

Whether to enable the dead pixel detection: 0-no, 1-yes.

byBlackPwl

Black level compensation parameter, which is between 0 and 255.

byEptzGate

Whether to enable E-PTZ: 0-disable, 1-enable.

byLocalOutputGate

Local output status:

0: disable local output

1: enable local BNC output

2: disable HDMI® output

20: enable HDMI®(720P50) output
21: enable HDMI®(720P60) output
22: enable HDMI®(1080I60) output
23: enable HDMI®(1080I50) output
24: enable HDMI®(1080P24) output
25: enable HDMI®(1080P25) output
26: enable HDMI®(1080P30) output
27: enable HDMI®(1080P50) output
28: enable HDMI®(1080P60) output
40: SDI_720P50
41: SDI_720P60
42: SDI_1080I50
43: SDI_1080I60
44: SDI_1080P24
45: SDI_1080P25
46: SDI_1080P30
47: SDI_1080P50
48: SDI_1080P60
49: SDI_720P25
50: SDI_720P30
51: SDI_1080I25
52: SDI_1080I30
60: YPBPR_720P50
61: YPBPR_720P60
62: YPBPR_1080I50
63: YPBPR_1080I60
64: YPBPR_1080P24
65: YPBPR_1080P25
66: YPBPR_1080P30
67: YPBPR_1080P50
68: YPBPR_1080P60
69: YPBPR_720P25
70: YPBPR_720P30
71: HDMI_1080P
72: HDMI_720P
73: HDMI_2160P

byCoderOutputMode

Encoder FPGA output mode: 0-directly output, 3-moving pixel.

byLineCoding

Whether to enable line coding: 0-no, 1-yes.

byDimmerMode

Dimming mode (available for thermal device): 0-semi-automatic, 1-automatic.

byPaletteMode

Palette mode (available for thermal device): 0-white heat, 1-black heat, 2-palette 2, ..., 8-palette 8.

byEnhancedMode

Enhancement mode (available for thermal device): 0-disable, 1, 2, 3, 4.

byDynamicContrastEN

Whether to enable dynamic contrast enhancement: 0-no, 1-yes.

byDynamicContrast

Dynamic contrast, its value is between 0 and 100.

byJPEGQuality

JPEG picture quality, its value is between 0 and 100. This member is not available for intelligent traffic cameras with version 3.5 or above.

struCmosModeCfg

Camera lens configuration parameters in CMOS mode, refer to the structure

[NET_DVR_CMOSMODECFG](#) for details.

byFilterSwitch

Whether to enable filtering: 0-no, 1-yes.

byFocusSpeed

Focus adjusting speed, its value is between 0 and 10.

byAutoCompensationInterval

Time interval of auto shutter compensation, its value is between 1 and 120, unit: minute.

bySceneMode

Scene mode: 0-outdoor, 1-indoor

A.47 NET_DVR_CAMERAPARAMCFG_EX

Camera configuration parameter structure

Structure Definition

```

struct{
    DWORD      dwSize;
    NET_DVR_VIDEOEFFECT struVideoEffect;
    NET_DVR_GAIN      struGain;
    NET_DVR_WHITEBALANCE struWhiteBalance;
    NET_DVR_EXPOSURE   struExposure;
    NET_DVR_GAMMACORRECT struGammaCorrect;
    NET_DVR_WDR       struWdr;
    NET_DVR_DAYNIGHT  struDayNight;
    NET_DVR_BACKLIGHT struBackLight;
    NET_DVR_NOISEREMOVE struNoiseRemove;
    BYTE        byPowerLineFrequencyMode;
    BYTE        byIrisMode;
    BYTE        byMirror;
    BYTE        byDigitalZoom;
    BYTE        byDeadPixelDetect;
    BYTE        byBlackPwl;
    BYTE        byEptzGate;
    BYTE        byLocalOutputGate;
    BYTE        byCoderOutputMode;
    BYTE        byLineCoding;
    BYTE        byDimmerMode;
    BYTE        byPaletteMode;
    BYTE        byEnhancedMode;
    BYTE        byDynamicContrastEN;
    BYTE        byDynamicContrast;
    BYTE        byJPEGQuality;
    NET_DVR_CMOSMODECFG struCmosModeCfg;
    BYTE        byFilterSwitch;
    BYTE        byFocusSpeed;
    BYTE        byAutoCompensationInterval;
    BYTE        bySceneMode;
    NET_DVR_DEFOGCFG      struDefogCfg;
    NET_DVR_ELECTRONICSTABILIZATION struElectronicStabilization;
    NET_DVR_CORRIDOR_MODE_CCD struCorridorMode;
    BYTE        byExposureSegmentEnable;
    BYTE        byBrightCompensate;
    BYTE        byCaptureModeN;
    BYTE        byCaptureModeP;
    NET_DVR_SMARTIR_PARAM    struSmartIRParam;
    NET_DVR_PIRIS_PARAM      struPIrisParam;
    NET_DVR_LASER_PARAM_CFG struLaserParam;
    NET_DVR_FFC_PARAM        struFFCParam;
    NET_DVR_DDE_PARAM        struDDEParam;
    NET_DVR_AGC_PARAM        struAGCParam;
    BYTE        byLensDistortionCorrection;
    BYTE        byDistortionCorrectionLevel;
    BYTE        byCalibrationAccurateLevel;
    BYTE        byZoomedInDistantViewLevel;
}

```

```
NET_DVR_SNAP_CAMERAPARAMCFG    struSnapCCD;
NET_DVR_OPTICAL_DEHAZE        struOpticalDehaze;
NET_DVR_THERMOMETRY_AGC       struThermAGC;
BYTE                          byFusionMode;
BYTE                          byHorizontalFOV;
BYTE                          byVerticalFOV;
BYTE                          byBrightnessSuddenChangeSuppression
BYTE                          byGPSEnabled
BYTE                          byRes2[155];
}NET_DVR_CAMERAPARAMCFG_EX, *LPNET_DVR_CAMERAPARAMCFG_EX;
```

Members

dwSize

Structure size.

struVideoEffect

Video effect parameter, refer to the structure [NET_DVR_VIDEOEFFECT](#) for details.

struGain

Gain parameter, refer to the structure [NET_DVR_GAIN](#) for details.

struWhiteBalance

White balance parameter, refer to the structure [NET_DVR_WHITEBALANCE](#) for details.

struExposure

Exposure parameter, refer to the structure [NET_DVR_EXPOSURE](#) for details.

struGammaCorrect

Gamma correction parameters, refer to the structure [NET_DVR_GAMMACORRECT](#) for details.

struWdr

Wide dynamic range (WDR) parameters, refer to the structure [NET_DVR_WDR](#) for details.

struDayNight

Day/night auto switch parameters, refer to the structure [NET_DVR_DAYNIGHT](#) for details.

struBackLight

Backlight compensation (BLC) parameters, refer to the structure [NET_DVR_BACKLIGHT](#) for details.

struNoiseRemove

Digital noise reduction parameters, refer to the structure [NET_DVR_NOISEREMOVE](#) for details.

byPowerLineFrequencyMode

Power supply frequency: 0-50 Hz; 1-60 Hz.

byIrisMode

Iris mode: 0: auto; 1: manual, 2: P-Iris1, 3: Union 3-9mm F1.6-2.7 (T5280-PQ1) [IPC5.1.7], 4: Union 2.8-12mm F1.6-2.7 (T5289-PQ1) [IPC5.1.7], 5: private 3.8-16mm F1.5(HV3816P-8MPIR),

6: private 11-40mm F1.7 (HV1140P-8MPIR), 7: private 2.7-12mm F1.2(TV2712P-MPIR), 8: MZ5721D-12MPIR, 9: MZ1555D-12MPIR, 10: MZ5721D-12MPIR(RS485), 11: MZ1555D-12MPIR(RS485).

byMirror

Mirror: 0-off, 1-left and right, 2-top and bottom, 3-center.

byDigitalZoom

Digital zoom: 0: enable, 1: ×2, 2: ×4.

byDeadPixelDetect

Whether to enable the dead pixel detection: 0-no, 1-yes.

byBlackPwl

Black level compensation parameter, which is between 0 and 255.

byEptzGate

Whether to enable E-PTZ: 0-disable, 1-enable.

byLocalOutputGate

Local output status:

0: disable local output

1: enable local BNC output

2: disable HDMI® output

20: enable HDMI®(720P50) output

21: enable HDMI®(720P60) output

22: enable HDMI®(1080I60) output

23: enable HDMI®(1080I50) output

24: enable HDMI®(1080P24) output

25: enable HDMI®(1080P25) output

26: enable HDMI®(1080P30) output

27: enable HDMI®(1080P50) output

28: enable HDMI®(1080P60) output

40: SDI_720P50

41: SDI_720P60

42: SDI_1080I50

43: SDI_1080I60

44: SDI_1080P24

45: SDI_1080P25

46: SDI_1080P30

47: SDI_1080P50

48: SDI_1080P60

byCoderOutputMode

Encoder FPGA output mode: 0-directly output, 3-moving pixel.

byLineCoding

Whether to enable line coding: 0-no, 1-yes.

byDimmerMode

Dimming mode (available for thermal device): 0-semi-automatic, 1-automatic.

byPaletteMode

Palette mode (available for thermal device): 0-white heat, 1-black heat, 2-palette 2, ..., 8-palette 8, 9-fusion 1,10-rainbow, 11-fusion 2, 12-rust 1, 13-rust red 2, 14-sepia, 15-color 1, 16-color 2, 17-ice and fire,18-rain, 19-red hot, 20-green hot, 21-dark blue, 22-color 3

byEnhancedMode

Enhancement mode (available for thermal device): 0-disable, 1, 2, 3, 4.

byDynamicContrastEN

Whether to enable dynamic contrast enhancement: 0-no, 1-yes.

byDynamicContrast

Dynamic contrast, its value is between 0 and 100.

byJPEGQuality

JPEG picture quality, its value is between 0 and 100. This member is not available for intelligent traffic cameras with version 3.5 or above.

struCmosModeCfg

Camera lens configuration parameters in CMOS mode, refer to the structure

[**NET_DVR_CMOSMODECFG**](#) for details.

byFilterSwitch

Whether to enable filtering: 0-no, 1-yes.

byFocusSpeed

Focus adjusting speed, its value is between 0 and 10.

byAutoCompensationInterval

Time interval of auto shutter compensation, its value is between 1 and 120, unit: minute.

bySceneMode

Scene mode: 0-outdoor, 1-indoor

struDefogCfg

Defog parameters, refer to the structure [**NET_DVR_DEFOGCFG**](#) for details.

struElectronicStabilization

Electronic image stabilization parameters, refer to the structure

[**NET_DVR_ELECTRONICSTABILIZATION**](#) for details.

struCorridorMode

Rotate mode parameter, refer to the structure [NET_DVR_CORRIDOR_MODE_CCD](#) for details.

byExposureSegmentEnable

Whether to adjust exposure time and gain by step: 0-no, 1-yes. For example, to enhance the exposure, first lengthen the exposure time to the medium value, and then enlarge the gain to the medium value, finally enhance the exposure and gain to the maximum level.

byBrightCompensate

Brightness, value range: [0,100].

byCaptureModeN

Video input format (NTSC standard): 0-close, 1- 640×480 @25fps, 2- 640×480 @30ps, 3- 704×576 @25fps, 4- 704×480 @30fps, 5- 1280×720 @25fps, 6- 1280×720 @30fps, 7- 1280×720 @50fps, 8- 1280×720 @60fps, 9- 1280×960 @15fps, 10- 1280×960 @25fps, 11- 1280×960 @30fps, 12- 1280×1024 @25fps, 13- 1280×1024 @30fps, 14- 1600×900 @15fps, 15- 1600×1200 @15fps, 16- 1920×1080 @15fps, 17- 1920×1080 @25fps, 18- 1920×1080 @30fps, 19- 1920×1080 @50fps, 20- 1920×1080 @60fps, 21- 2048×1536 @15fps, 22- 2048×1536 @20fps, 23- 2048×1536 @24fps, 24- 2048×1536 @25fps, 25- 2048×1536 @30fps, 26- 2560×2048 @25fps, 27- 2560×2048 @30fps, 28- 2560×1920 @7.5fps, 29- 3072×2048 @25fps, 30- 3072×2048 @30fps, 31- 2048×1536 @12.5, 32- 2560×1920 @6.25, 33- 1600×1200 @25, 34- 1600×1200 @30, 35- 1600×1200 @12.5, 36- 1600×900 @12.5, 37- 1600×900 @15, 38- 800×600 @25, 39- 800×600 @30, 40- 4000×3000 @12.5fps, 41- 4000×3000 @15fps, 42- 4096×2160 @20fps, 43- 3840×2160 @20fps, 44- 960×576 @25fps, 45- 960×480 @30fps, 46- 752×582 @25fps, 47- 768×494 @30fps, 48- 2560×1440 @25fps, 49- 2560×1440 @30fps, 50- $720P$ @100fps, 51- $720P$ @120fps, 52- 2048×1536 @50fps, 53- 2048×1536 @60fps, 54- 3840×2160 @25fps, 55- 3840×2160 @30fps, 56- 4096×2160 @25fps, 57- 4096×2160 @30fps, 58- 1280×1024 @50fps, 59- 1280×1024 @60fps, 60- 3072×2048 @50fps, 61- 3072×2048 @60fps, 62- 3072×1728 @25fps, 63- 3072×1728 @30fps, 64- 3072×1728 @50fps, 65- 3072×1728 @60fps, 66- 336×256 @50fps, 67- 336×256 @60fps, 68- 384×288 @50fps, 69- 384×288 @60fps, 70- 640×512 @50fps, 71- 640×512 @60fps, 72- 2592×1944 @25fps, 73- 2592×1944 @30fps, 74- 2688×1536 @25fps, 75- 2688×1536 @30fps, 76- 2592×1944 @20fps, 77- 2592×1944 @15fps, 78- 2688×1520 @20fps, 79- 2688×1520 @15fps, 80- 2688×1520 @25fps, 81- 2688×1520 @30fps, 82- 2720×2048 @25fps, 83- 2720×2048 @30fps, 84- 336×256 @25fps, 85- 384×288 @25fps, 86- 640×512 @25fps, 87- 1280×960 @50fps, 88- 1280×960 @60fps, 89- 1280×960 @100fps, 90- 1280×960 @120fps, 91- 4000×3000 @20fps, 92- 1920×1200 @25fps, 93- 1920×1200 @30fps, 94- 2560×1920 @25fps, 95- 2560×1920 @20fps, 96- 2560×1920 @30fps, 97- 1280×1920 @25fps, 98- 1280×1920 @30fps, 99- 4000×3000 @24fps, 100- 4000×3000 @25fps, 101- 4000×3000 @10fps, 102- 384×288 @30fps, 103- 2560×1920 @15fps, 104- 2400×3840 @25fps, 105- 1200×1920 @25fps, 106- 4096×1800 @30fps, 107- 3840×1680 @30fps, 108- 2560×1120 @30fps, 109- 704×320 @30fps, 110- 1280×560 @30fps, 111- 4096×1800 @25fps, 112- 3840×1680 @25fps, 113- 2560×1120 @25fps, 114- 704×320 @25fps, 115- 1280×560 @25fps, 116- 2400×3840 @24fps, 117- 3840×2400 @24fps, 118- 3840×2400 @25fps, 119- 2560×1920 @12.5fps, 120- 2560×2048 @12fps, 121- 2560×2048 @15fps,

122-2560 × 1536@25fps, 123-2560 × 1536@30fps, 124-2256 × 2048@25fps, 125-2256 × 2048@30fps, 126-2592 × 2592@12.5fps, 127-2592 × 2592@15fps, 128 - 640 × 512@30fps, 129-2048 × 1520@30fps, 130-2048 × 1520@25fps, 131-3840 × 2160@24fps, 132-2592 × 1520@25fps, 133-2592 × 1520@30fps, 134-2592 × 1536@25fps, 135-2592 × 1536@30fps, 136-640 × 960@25fps, 137-640 × 960@24fps, 141-2688 × 1520@12.5fps, 142-2992 × 2192@25fps, 143-2992 × 2192@30fps, 144-3008 × 2160@25fps, 145-3008 × 2160@30fps, 146-3072 × 1728@20fps, 147-2560 × 1440@20fps, 148-2160 × 3840@25fps, 149-2160 × 3840@30fps, 150-7008 × 1080@25fps, 151-7008 × 1080@30fps, 152-3072 × 2048@20fps, 153-1536 × 864@25fps, 154-2560 × 1920@24fps, 155-2400 × 3840@30fps, 156-3840 × 2400@30fps, 157-3840 × 2160@15fps, 158-384 × 288@8.3fps, 159-640 × 512@8.3fps, 160-160 × 120@8.3fps, 161-1024 × 768@8.3fps, 162-640 × 480@8.3fps

byCaptureModeP

Video input format (PAL standard), the available values are same as that of **byCaptureModeN**.

struSmartIRParam

Smart IR parameters, refer to the structure [**NET_DVR_SMARTIR_PARAM**](#) for details.

struPIrisParam

IR iris parameters, refer to the structure [**NET_DVR_PIRIS_PARAM**](#) for details.

struLaserParam

Laser parameters, refer to the structure [**NET_DVR_LASER_PARAM_CFG**](#) for details.

struFFCParam

FFC parameters, refer to the structure [**NET_DVR_FFC_PARAM**](#) for details.

struDDEParam

DDE parameters, refer to the structure [**NET_DVR_DDE_PARAM**](#) for details.

struAGCParam

AGC parameters, refer to the structure [**NET_DVR_AGC_PARAM**](#) for details.

byLensDistortionCorrection

Whether to enable lens distortion correction: 0-no, 1-yes.

byDistortionCorrectionLevel

Distortion correction level: 0-reserved, 1-level 1, 2-level 2, 3-level 3, 255-custom

byCalibrationAccurateLevel

Distortion correction strength, value range: [0,100].

byZoomedInDistantViewLevel

Remote zooming level, value range: [0,100].

struSnapCCD

CCD parameters of capture camera, refer to the structure [**NET_DVR_SNAP_CAMERAPARAMCFG**](#) for details.

struOpticalDehaze

Optical defog parameters, refer to the structure [NET_DVR_OPTICAL_DEHAZE](#) for details.

struThermAGC

AGC parameters of temperature measurement, refer to the structure [NET_DVR_THERMOMETRY_AGC](#) for details.

byFusionMode

Visual and thermal image fusion mode: 0-thermal image mode, 1-fusion mode, 2- PIP mode, 3-visible light mode, 4-black and white fusion mode, 5-jungle (fusion color mode), 6-desert (fusion color mode), 7-snow (fusion color mode), 8-sea (fusion color mode), 9-city (fusion color mode).

byHorizontalFOV

Horizontal FOV, value range: [0,100].

byVerticalFOV

Vertical FOV, value range: [0,100].

byBrightnessSuddenChangeSuppression

Whether to enable suppression of brightness sudden change: 0-no, 1-yes.

byGPSEnabled

Whether to enable GPS: 0-no, 1-yes

byRes2

Reserved, set to 0.

A.48 NET_DVR_CERT_ADDITION_PARAM

Additional certification information structure.

Structure Definition

```
struct{
    DWORD dwSize;
    char csCustomID[64];
    BYTE byRes1[2] ;
    BYTE byCertificateMode ;
    BYTE byPrivateKeyMode;
    BYTE byPassword[64];
    BYTE byRes[128];
}NET_DVR_CERT_ADDITION_PARAM, *LPNET_DVR_CERT_ADDITION_PARAM;
```

Members

dwSize

Structure size.

csCustomID

Custom ID.

byRes1

Reserved.

byCertificateMode

Certificate mode: 0-self-signed certificate, 1-certificate and private key. This member is valid when **wCertType** in the structure **NET_DVR_CERT_PARAM** is "1" (certificate).

byPrivateKeyMode

Private key mode: 0-independent key, 1-PKCS#12. This member is valid when **byCertificateMode** is "1" (certificate and private key method).

byPassword

Password. If **byPrivateKeyMode** is 0, this member refers to the private key password; if **byPrivateKeyMode** is 1, this member refers to the PKCS#12 password.

byRes

Reserved.

A.49 NET_DVR_CERT_PARAM

Certificate parameter structure.

Structure Definition

```
struct{
    DWORD      dwSize;
    WORD       wCertFunc;
    WORD       wCertType;
    BYTE       byFileType;
    BYTE       byRes1[2];
    BYTE       byAddition;
    NET_DVR_CERT_ADDITION_PARAM *pStruParam;
    BYTE       byRes[28];
}NET_DVR_CERT_PARAM, *LPNET_DVR_CERT_PARAM;
```

Members

dwSize

Structure size.

wCertFunc

Certificate function: 0-802.1x (for Wi-Fi access), 1-HTTPS, 3-ieee802.1x (for wired LAN access), 4-SSH

wCertType

Certificate type: 0-CA, 1-certificate, 2-private key file

byFileType

File type: 0-PEM, 1-PFX

byRes1

Reserved

byAddition

Whether to use additional structure: 0-no, 1-yes

pStruAdditionParam

Pointer of the additional information structure, see details in the structure

NET_DVR_CERT_ADDITION_PARAM

byRes

Reserved, set to 0

Remarks

- When importing custom CA certificate, the member **wCertType** should be set to 0 (CA), **byAddition** should be set to 1, and the member **csCustomID** in the structure **NET_DVR_CERT_ADDITION_PARAM** should be configured.
- When importing client/server self-signed certificate, the member **wCertType** should be set to 1 (certificate), **byAddition** should be set to 1; in the structure **NET_DVR_CERT_ADDITION_PARAM**, the member **byCertificateMode** should be set to 0 (self-signed certificate), and **csCustomID** should be configured.
- When importing client/server certificate along with the PKCS#12 private key, the member **wCertType** should be set to 1 (certificate), **byAddition** should be set to 1; in the structure **NET_DVR_CERT_ADDITION_PARAM**, the member **byCertificateMode** should be set to 1 (certificate and private key), **byPrivateKeyMode** should be set to 1 (PKCS#12), **byPassword** should be set to the PKCS#12 password, and **csCustomID** should be configured.
- When importing client/server certificate along with the independent private key, you should perform the following two steps.
 1. Import certificate file: the member **wCertType** should be set to 1 (certificate), **byAddition** should be set to 1; in the structure **NET_DVR_CERT_ADDITION_PARAM**, the member **byCertificateMode** should be set to 1 (certificate and private key), **byPrivateKeyMode** should be set to 0 (independent key), and **csCustomID** should be configured.
 2. Import the private key file of the certificate: the member **wCertType** should be set to 2 (private key file), **byAddition** should be set to 1; in the structure **NET_DVR_CERT_ADDITION_PARAM**, the member **byCertificateMode** should be set to 1 (certificate and private key), **byPrivateKeyMode** should be set to 0 (independent key), **byPassword** should be set to the password of the private key file, and **csCustomID** should be configured.

- When downloading CA certificate, the member **wCertType** should be set to 0 (CA), and the member **csCustomID** in the structure ***NET_DVR_CERT_ADDITION_PARAM*** should be configured.
- When downloading client/server certificate, the member **wCertType** should be set to 1 (certificate), and the member **csCustomID** in the structure ***NET_DVR_CERT_ADDITION_PARAM*** should be configured.

A.50 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.51 NET_DVR_CHANNELSTATE_V30

Structure of channel state information

Table A-12 Channel Status Structure

Member	Data Type	Description
byRecordStatic	BYTE	Whether the channel is recording: 0-no, 1-yes.
bySignalStatic	BYTE	Video signal status: 0-normal, 1-lost.
byHardwareStatic	BYTE	Channel hardware status: 0-normal, 1-exception (e.g. DSP exception).
byRes1	BYTE	A reserved byte. Set to 0.
dwBitRate	DWORD	Actual bit rate.
dwLinkNum	DWORD	The total number of clients connected with the channel.
struClientIP	<u>NET_DVR_IPADDR_UNION []</u>	Client IP address. The maximum length is 6 bytes.
dwIPLinkNum	DWORD	Total number of connections of the digital channel.
byExceedMaxLink	BYTE	Whether the maximum connections reached (Up to 6 connections are allowed): 0-no, 1-yes.
byRes	BYTE	Reserved, set to 0. The maximum length is 3 bytes.
dwAllBitRate	DWORD	The sum of actual bit rate of this channel.
dwChannelNo	DWORD	The current channel No., 0xffffffff-invalid.

A.52 NET_DVR_CHAN_FILTER_CFG

Structure about linking POS rules and channels

Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byFilterID;
    BYTE       byFontSize;
    BYTE       byShowPosInfo;
    BYTE       byOverlayMode;
    DWORD      dwDelayTime;
    NET_DVR_POS OSD_REGION struOsdPosInfo;
    NET_DVR_RGB_COLOR struOsdColor;
    NET_DVR_SCHEDTIME struAlarmSched[MAX_DAYS][MAX_TIMESEGMENT_V30/*8*/];
    NET_DVR_HANDLEEXCEPTION_V40 struHandleException;
```

```
DWORD      dwMaxRelRecordChanNum;
DWORD      dwRelRecordChanNum;
DWORD      dwRelRecordChan[MAX_CHANNUM_V30/*64*/];
DWORD      dwTimeOut;
BYTE       byRes[60];
}NET_DVR_CHAN_FILTER_CFG,*LPNET_DVR_CHAN_FILTER_CFG;
```

Members

dwSize

Structure size.

byFilterID

POS filtering rule ID.

byFontSize

Font size: 0- 8*16, 1- 16*32, 2- 32*64.

byShowPosInfo

Whether to overlay POS information in stream: 0-no, 1-yes.

byOverlayMode

Overlaying mode: 0-scrolling, 1-page turning.

dwDelayTime

Delay time, value ranges from 5 to 3600, unit:s.

struOsdPosInfo

POS information overlapping area.

struOsdColor

Color of overlapped POS information.

struAlarmSched

Arming schedule, up to 8 periods can be scheduled for each day.

struHandleException

Exception handling type.

dwMaxRelRecordChanNum

The maximum number of video channels that can be triggered by alarm (read-only).

dwRelRecordChanNum

Actual number of video channels that can be triggered by alarm.

dwRelRecordChan

Video channels that can be triggered by alarm, the first dwRelRecordChanNum channels can be configured.

dwTimeOut

Timeout time, value ranges from 5 to 3600, unit: s. Interval during which no POS data can be received.

byRes

Reserved.

Remarks

- When the value of **dwMaxRelRecordChanNum** is larger than 64, the channels will be divided into groups and each group will be numbered (**dwGroup**) in the structure **NET_DVR_CHANNEL_GROUP**, and up to 64 channels can be contained in a group.
- The video channel No. links with the group No., e.g., when the group No. is 0, the No. of channels in the group is between 1 and 64; when the group No. is 1, the No. of channels in the group is between 65 and 128, and so on. If the group No. is 0xffffffff, it indicates that the No. of channels in current group and the following channels is invalid.

A.53 NET_DVR_CHANNEL_GROUP

Channel No. and group No. information structure.

Structure Definition

```
struct{  
    DWORD dwSize;  
    DWORD dwChannel;  
    DWORD dwGroup;  
    BYTE byID;  
    BYTE byRes1[3];  
    DWORD dwPositionNo;  
    BYTE byRes[56];  
}NET_DVR_CHANNEL_GROUP,*LPNET_DVR_CHANNEL_GROUP;
```

Members

dwSize

Structure size

dwChannel

Channel No.

dwGroup

Group No., which starts from 0, 0-group No.1, 1-group No.2, ...

byID

Device region configuration ID

byRes1

Reserved, set to 0.

dwPositionNo

Scene No., for network camera, it is 0; for speed dome, it starts from 1.

byRes

Reserved, set to 0.

A.54 NET_DVR_CHECK_DEV_STATE

Table A-13 Structure about Checking Parameters of Device Status

Member	Data Type	Description
dwTimeout	DWORD	Time interval to check device status for once, unit: millisecond. The default value is 0, which represents 30,000 ms. The minimum time interval is 1,000 ms.
fnStateCB	<u>DEV_WORK_STATE_CB</u>	Callback function of device status information.
pUserData	void*	Custom data.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 60 bytes.

A.55 NET_DVR_CMOSMODECFG

Camera lens parameter structure of CMOS mode

Structure Definition

```
struct{
    BYTE byCaptureMod;
    BYTE byBrightnessGate;
    BYTE byCaptureGain1;
    BYTE byCaptureGain2;
    DWORD dwCaptureShutterSpeed1;
    DWORD dwCaptureShutterSpeed2;
    BYTE byRes[4];
}NET_DVR_CMOSMODECFG, *LPNET_DVR_CMOSMODECFG;
```

Members**byCaptureMod**

Capture mode: 0-mode 1, 1-mode 2.

byBrightnessGate

Brightness threshold.

byCaptureGain1

Capture gain 1, its value is between 0 and 100.

byCaptureGain2

Capture gain 2, its value is between 0 and 100.

dwCaptureShutterSpeed1

Shutter speed 1 for capture.

dwCaptureShutterSpeed2

Shutter speed 2 for capture.

byRes

Reserved, set to 0.

A.56 NET_DVR_CMS_PARAM

Platform access parameter structure

Structure Definition

```
struct{
    DWORD      dwSize;
    NET_DVR_IPADDR struAddr;
    WORD       wServerPort;
    BYTE       bySeverProtocolType;
    BYTE       byStatus;
    BYTE       sDeviceId[NAME_LEN/*32*/];
    char      sPassWord[PASSWD_LEN/*16*/];
    BYTE       sPlatformEhomeVersion[NAME_LEN/*32*/];
    BYTE       byNetWork;
    BYTE       byAddressType;
    BYTE       byProtocolVersion;
    BYTE       byRes1;
    BYTE       sDomainName[MAX_DOMAIN_NAME/*64*/];
    BYTE       byEnable;
    BYTE       byRes[172];
}NET_DVR_CMS_PARAM,*LPNET_DVR_CMS_PARAM;
```

Members

dwSize

Structure size

struAddr

Platform IP address, see details in the structure , it is valid when **byAddressType** is 0 or 1.

wServerPort

Platform listening port number.

byServerProtocolType

Platform protocol type: 1-private, 2-Ehome.

byStatus

Device registration status in the platform: 1-unregistered, 2- registered

sDeviceId

Device ID, which is provided by the platform.

sPassword

Password, which is provided by platform.

sPlatformEhomeVersion

EHome protocol version information of this platform.

byNetwork

Network mode: 0-reserved, 1-auto, 2-wired network in priority, 3-wired network, 4- mobile network (3G network)

byAddressType

Address type: 0-reserved, 1-ipv4/ipv6 address, 2-domain name

byProtocolVersion

Protocol version: 0-reserved, 1-Version 2.0, 2-Version 4.0

byRes1

Reserved, set to 0

sDomainName

Platform server domain name, it is valid when **byAddressType** is 2.

byEnable

Whether to enable: 0-no, 1-yes

byRes

Reserved, set to 0

A.57 NET_DVR_COLOR

Structure about image parameters.

Structure Definition

```
struct{
    BYTE  byBrightness;
    BYTE  byContrast;
    BYTE  bySaturation;
    BYTE  byHue;
}NET_DVR_COLOR, *LPNET_DVR_COLOR;
```

Members

byBrightness

Brightness, ranges from 0 to 255

byContrast

Contrast, ranges from 0 to 255

bySaturation

Saturation, ranges from 0 to 255

byHue

Hue, ranges from 0 to 255

A.58 NET_DVR_COMPRESSION_AUDIO

Audio parameter structure of two-way audio.

Structure Definition

```
struct{
    BYTE byAudioEncType;
    BYTE byAudioSamplingRate;
    BYTE byAudioBitRate;
    BYTE byres[4];
    BYTE bySupport;
}NET_DVR_COMPRESSION_AUDIO, *LPNET_DVR_COMPRESSION_AUDIO;
```

Members

byAudioEncType

Audio encoding type: 0-G722, 1-G711_U, 2-G711_A, 5-MP2L2, 6-G726, 7-AAC, 8-PCM, 9-G722, 10-G723, 11-G729, 12-AAC_LC, 13-AAC_LD, 14-Opus, 15-MP3, 16-ADPCM.

byAudioSamplingRate

Audio sampling rate: 0-default, 1-16kHz, 2-32kHz, 3-48kHz, 4-44.1kHz, 5-8kHz.

byAudioBitRate

Audio encoding bit rate, refer to [BITRATE_ENCODE_INDEX](#) for details.

byres

Reserved, set to 0.

bySupport

Audio data size.

Remarks

After setting the audio parameter for two-way audio, you must reboot the device to take effect.

A.59 NET_DVR_COMPRESSION_INFO_V30

Structure about stream encoding parameters.

Structure Definition

```
struct{
    BYTE      byStreamType;
    BYTE      byResolution;
    BYTE      byBitrateType;
    BYTE      byPicQuality;
    DWORD     dwVideoBitrate;
    DWORD     dwVideoFrameRate;
    WORD      wIntervalFrame;
    BYTE      byIntervalBPF;
    BYTE      byres1;
    BYTE      byVideoEncType;
    BYTE      byAudioEncType;
    BYTE      byVideoEncComplexity;
    BYTE      byEnableSvc;
    BYTE      byFormatType;
    BYTE      byAudioBitRate;
    BYTE      bySteamSmooth;
    BYTE      byAudioSamplingRate;
    BYTE      bySmartCodec;
    BYTE      byDepthMapEnable;
    WORD      wAverageVideoBitrate;
}NET_DVR_COMPRESSION_INFO_V30, *LPNET_DVR_COMPRESSION_INFO_V30;
```

Members

byStreamType

Stream type: 0- video stream, 1-video and audio stream, 0xfe-auto (same as the source).

For event information encoding parameter (**struEventRecordPara**), the highest-bit of **byStreamType** indicates whether to enable event information encoding. That is, when setting **struEventRecordPara**:

byStreamType&0x80 == 0: disable event information encoding

(byStreamType&0x80 == 1)&&(byStreamType&0x7f == 0) : enable event information encoding and set stream type to video stream;

(byStreamType&0x80 == 1)&&(byStreamType&0x7f == 1) : enable event information encoding and set stream type to video and audio stream;

byStreamType==0fe: enable event information encoding and set stream type as same as the source.

byResolution

Resolution: CIF(528 × 384/528 × 320), 1-CIF(352 × 288/352 × 240), 2-QCIF(176 × 144/176 × 120), 3-4CIF(704 × 576/704 × 480) or D1(720 × 576/720 × 486), 4-2CIF(704 × 288/704 × 240), 6-QVGA(320 × 240), 7-QQVGA(160 × 120), 12-384 × 288, 13-576 × 576, 16-VGA(640 × 480), 17-UXGA(1600 × 1200), 18-SVGA(800 × 600), 19-HD720P(1280 × 720), 20-XVGA(1280 × 960), 21-HD900P(1600 × 900), 22-1360 × 1024, 23-1536 × 1536, 24-1920 × 1920, 27-1920 × 1080p, 28-2560 × 1920, 29-1600 × 304, 30-2048 × 1536, 31-2448 × 2048, 32-2448 × 1200, 33-2448 × 800, 34-XGA(1024 × 768), 35-SXGA(1280 × 1024), 36-WD1(960 × 576/960 × 480), 37-1080i(1920 × 1080), 38-WXGA(1440 × 900), 39-HD_F(1920 × 1080/1280 × 720), 40-HD_H(1920 × 540/1280 × 360), 41-HD_Q(960 × 540/630 × 360), 42-2336 × 1744, 43-1920 × 1456, 44-2592 × 2048, 45-3296 × 2472, 46-1376 × 768, 47-1366 × 768, 48-1360 × 768, 49-WSXGA+, 50-720 × 720, 51-1280 × 1280, 52-2048 × 768, 53-2048 × 2048, 54-2560 × 2048, 55-3072 × 2048, 56-2304 × 1296, 57-WXGA(1280 × 800), 58-1600 × 600, 59-1600 × 900, 60-2752 × 2208, 61-384 × 288, 62-4000 × 3000, 63-4096 × 2160, 64-3840 × 2160, 65-4000 × 2250, 66-3072 × 1728, 67-2592 × 1944, 68-2464 × 1520, 69-1280 × 1920, 70-2560 × 1440, 71-1024 × 1024, 72-160 × 128, 73-324 × 240, 74-324 × 256, 75-336 × 256, 76-640 × 512, 77-2720 × 2048, 78-384 × 256, 79-384 × 216, 80-320 × 256, 81-320 × 180, 82-320 × 192, 83-512 × 384, 84-325 × 256, 85-256 × 192, 86-640 × 360, 87-1776x1340, 88-1936x1092, 89-2080x784, 90-2144x604, 91-1920 × 1200, 92-4064 × 3040, 93-3040 × 3040, 94-3072 × 2304, 95-3072 × 1152, 96-2560 × 2560, 97-2688 × 1536, 98-2688 × 1520, 99-3072 × 3072, 100-3392 × 2008, 101-4000 × 3080, 102-960 × 720, 103-1024 × 1536, 104-704 × 1056, 105-352 × 528, 106-2048 × 1530, 107-2560 × 1600, 108-2800 × 2100, 109-4088 × 4088, 110-4000 × 3072, 111-960 × 1080(1080p Lite), 112-640 × 720(half 720p), 113-640 × 960, 114-320 × 480, 115-3840 × 2400, 116-3840 × 1680, 117-2560 × 1120, 118-704 × 320, 119-1200 × 1920, 120-480 × 768, 121-768 × 480, 122-320 × 512, 123-512 × 320, 124-4096 × 1800, 125-1280 × 560, 126-2400 × 3840, 127-480 × 272, 128-512 × 272, 129-2592 × 2592, 130-1792 × 2880, 131-1600 × 2560, 132-2720 × 1192, 133-3MP(1920 × 1536/2048 × 1536), 134-5MP(2560 × 1944), 137-4096 × 1200, 138-3840 × 1080, 139-2720 × 800, 140-512 × 232, 141-704 × 200, 142-512 × 152, 143-2048 × 896, 144-2048 × 600, 145-1280 × 376, 150-8208 × 3072, 151-4096 × 1536, 152-6912 × 2800, 153-3456 × 1400, 154-480 × 720, 155-800 × 450, 156-480 × 270, 157-2560 × 1536, 160-3264 × 2448, 161-288 × 320, 162-144 × 176, 163-480 × 640, 164-240 × 320, 165-120 × 160, 166-576 × 720, 167-720 × 1280, 168-576 × 960, 169-2944 × 1656, 170-432 × 240, 171-2160 × 3840, 172-1080 × 1920, 173-7008 × 1080, 174-3504 × 540, 175-1752 × 270, 176-876 × 135, 177-4096 × 1440, 178-4096 × 1080, 179-1536 × 864, 180-180 × 240, 181-360 × 480, 182-540 × 720, 183-720 × 960, 184-960 × 1280, 185-1080 × 1440, 186-3200 × 1800, 187-1752 × 272, 188-872 × 136, 189-1280 × 1440, 215-1080 × 720, 216-360 × 640, 0xffff-auto (current stream resolution).

Note

- When the index number of the current resolution exceeds 254, the returned/configured value of this member is 254 and the returned/configured value of **dwResolution** in the structure **NET_DVR_MULTI_STREAM_COMPRESSIONCFG** is the index number of the current resolution.
 - When the index number of the current resolution is smaller than 254, the returned/configured value of this member is the index number of the current resolution.
-

byBitrateType

Bit rate type: 0- variable bit rate, 1-constant bit rate, 0xfe: auto

byPicQuality

Image quality: 0-best, 1-better, 2-good, 3-normal, 4-worse, 5-bad, 0xfe-auto.

dwVideoBitrate

Video bitrate: 0-Reserved, 1-16K (Reserved), 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K, 23-2048K, 24-3072K, 25-4096K, 26-8192K, 27-16384K, 0xfffffffffe-auto.

dwVideoFrameRate

Video frame rate: 0-all, 1-1/16, 2-1/8, 3-1/4, 4-1/2, 5-1, 6-2, 7-4, 8-6, 9-8, 10-10, 11-12, 12-16, 13-20, 14-15, 15-18, 16—22, 17-25, 18-30, 19-35, 20-40, 21-45, 22-50, 23-55, 24-60, 25-3, 26-5, 27-7, 28-9, 29-100, 30-120, 31-24, 32-48, 33-8.3, 0xfffffffffe-auto.

wIntervalFrameI

Interval of I frame, 0xffffe-auto (same with source), 0xffff-invalid.

byIntervalBPF

Frame format: 0-BBP, 1-BP, 2-single P frame, 0xff-invalid.

byres1

Reserved, set to 0.

byVideoEncType

Video encoding format: 0- private H.264, 1- standard H.264, 2- standard MPEG4, 7- M-JPEG, 8- MPEG2, 9- SVAC, 10- standard h265, 0xfe- auto (same with source), 0xff- invalid.

byAudioEncType

Audio encoding format: 0-G.722, 1-G.711_U, 2- G.711_A, 5- MP2L2, 6-G.726, 7-AAC, 8-PCM, 12- AAC_LC, 13-AAC_LD, 14-Opus, 15-MP3, 0xfe-auto, 0xff-invalid.

byVideoEncComplexity

Video encoding complexity: 0-low, 1-medium, 2-high, 0xfe-auto (same as the source).

byEnableSvc

Whether to enable SVC (Scalable Video Coding): 0-no, 1-yes, 2-auto

byFormatType

Container format: 1-raw stream, 2-RTP, 3-PS, 4-TS, 5-private, 6-FLV, 7-ASF, 8-3GP, 0xff-invalid.

byAudioBitRate

Audio encoding bit rate, refer to [BITRATE_ENCODE_INDEX](#) for details.

bySteamSmooth

Stream smooth level, the value is between 1 and 100, 1-clear, 100-smooth.

byAudioSamplingRate

Audio sampling rate: 0-default, 1-16kHz, 2-32kHz, 3-48kHz, 4-44.1kHz, 5-8kHz.

bySmartCodec

Whether to enable high performance encoding (enable high performance means Smart 264 when **byVideoEncType==1**: enable Smart264; **byVideoEncType==10**: Smart265): 0-disable, 1-enable.

When **bySmartCodec** is 1 and **byVideoEncType** is 1, you can configure upper-limit bitrate (**dwVideoBitrate**) and average bit rate (**wAverageVideoBitrate**) for variable bitrate, and configure upper-limit bit rate (**dwVideoBitrate**) for constant bit rate.

byDepthMapEnable

Whether to enable depth map: 0-no, 1-yes; If it is enabled, the video parameters of sub-stream of channel No.2 cannot be configured, and by default, the resolution of output depth map is 960 × 540.

wAverageVideoBitrate

Average video bit rate (it is valid when **bySmartCodec** is enabled): 0-0K, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K, 23-2048K, 24-2560K, 25-3072K, 26-4096K, 27-5120K, 28-6144K, 29-7168K, 30-8192K.

Highest-bit (bit15)==1: custom stream; bit0 to bit14: stream value.

Remarks

- The supports of video resolution, frame rate, audio and video encoding type, container format, and so on, depend on the device encoding capability.
- After setting the video encoding type, you should reboot the device to take effect.
- The average bit rate is used to control video size. When setting this parameter, you'd better refer to the recommended value, which varies under different devices. You can get the average bit rate from the dynamic capability of front-end device.
- If Smart264 or Smart265 is enabled, the following functions are not available, such as ROI, SVC, main/sub-stream smooth, high frame rate, resolution 2048 × 1536, vehicle detection, multi-target-type detection, traffic enforcement, heat map, and so on.

A.60 NET_DVR_COMPRESSIONCFG_V30

Channel encoding parameter structure

Structure Definition

```
struct{
    WORD           dwSize;
    NET_DVR_COMPRESSION_INFO_V30 struNormHighRecordPara;
    NET_DVR_COMPRESSION_INFO_V30 struNormLowRecordPara;
    NET_DVR_COMPRESSION_INFO_V30 struEventRecordPara;
```

```
NET_DVR_COMPRESSION_INFO_V30 struNetPara;  
}NET_DVR_COMPRESSIONCFG_V30, *LPNET_DVR_COMPRESSIONCFG_V30;
```

Members

dwSize

Structure size

struNormHighRecordPara

Encoding parameters of video stream, that is, the encoding parameters of sub-stream, refer to the structure [NET_DVR_COMPRESSION_INFO_V30](#) for details.

struNormLowRecordPara

Encoding parameters of video stream, that is, the encoding parameters of sub-stream, refer to the structure [NET_DVR_COMPRESSION_INFO_V30](#) for details.

struEventRecordPara

Encoding parameters of event or alarm stream. When the alarm is triggered or event occurred, the main stream will automatically switch to the encoding settings of event or alarm stream.

Refer to the structure [NET_DVR_COMPRESSION_INFO_V30](#) for details.

struNetPara

Stream encoding parameters for network transmission, that is, the encoding parameters of sub-stream, refer to the structure [NET_DVR_COMPRESSION_INFO_V30](#) for details.

A.61 NET_DVR_CONNECT_POS_CFG

Structure about POS connection configuration.

Structure Definition

```
struct{  
    DWORD      dwSize;  
    BYTE       byConnectMode;  
    BYTE       byRes1[3];  
    NET_DVR_POS_CONNECTMODE_UNION struPosConnMode;  
    BYTE       byRes[64];  
}NET_DVR_CONNECT_POS_CFG,*LPNET_DVR_CONNECT_POS_CFG;
```

Members

dwSize

Structure size

byConnectMode

Connection mode: 0-network listening (TCP), 1-network receiving, 2-port receiving, 3-network listening (UDP), 4-network detection (sniff) , 5-multicasting, 6- USB_232 port mode

byRes1

Reserved, set it to 0.

struPosConnMode

Connection mode configuration.

byRes

Reserved, set it to 0.

A.62 NET_DVR_CORRIDOR_MODE_CCD

Rotate mode parameter structure

Structure Definition

```
struct{
    BYTE   byEnableCorridorMode;
    BYTE   byRes[11];
}NET_DVR_CORRIDOR_MODE_CCD, *LPNET_DVR_CORRIDOR_MODE_CCD;
```

Members**byEnableCorridorMode**

Whether to enable rotate mode: 0-no, 1-yes.

byRes

Reserved, set to 0.

Remarks

When the rotate mode is enable, the image will rotate 90 degree in anticlockwise, for example, 1280 × 720 will rotate to 720 × 1280. This mode can enlarge vertical surveillance range, and it is applicable to corridors, roads, and so on.

A.63 NET_DVR_CRUISECHAN_INFO**Structure about The Information of Channel That Called Patrol**

Member	Data Type	Description
dwEnableCruiseChan	DWORD	Channel that called patrol.
dwCruiseNo	DWORD	Patrol No., 0xffffffff-invalid.

A.64 NET_DVR_DAYNIGHT

Day/night auto switch parameter structure

Structure Definition

```
struct{
    BYTE   byDayNightFilterType;
    BYTE   bySwitchScheduleEnabled;
    BYTE   byBeginTime;
    BYTE   byEndTime;
    BYTE   byDayToNightFilterLevel;
    BYTE   byNightToDayFilterLevel;
    BYTE   byDayNightFilterTime;
    BYTE   byBeginTimeMin;
    BYTE   byBeginTimeSec;
    BYTE   byEndTimeMin;
    BYTE   byEndTimeSec;
    BYTE   byAlarmTrigState;
}NET_DVR_DAYNIGHT, *LPNET_DVR_DAYNIGHT;
```

Members

byDayNightFilterType

Day/night auto switch mode: 0-day, 1-night, 2-auto (photoresistance mode), 3-scheduled, 4-alarm input triggered, 5-auto mode 2 (no photosensitivity), 6-black light, 7-auto (black light), 8-scheduled (black light).

bySwitchScheduleEnabled

Scheduled mode: 0-enable, 1-disable.

byBeginTime

Start time of scheduled mode, unit: hour, its value is between 0 and 23.

byEndTime

End time of scheduled mode, unit: hour, its value is between 0 and 23.

byDayToNightFilterLevel

Level for switching day to night, for network camera, the level is between 0 and 7; for speed dome, the level is between 1 and 3.

byNightToDayFilterLevel

Level for switching night to day, for network camera, the level is between 0 and 7; for speed dome, the level is between 1 and 3.

byDayNightFilterTime

Switching time interval: 60 s

byBeginTimeMin

Start time of scheduled mode, unit: minute, its value is between 0 and 59.

byBeginTimeSec

Start time of scheduled mode, unit: second, its value is between 0 and 59.

byEndTimeMin

End time of scheduled mode, unit: minute, its value is between 0 and 59.

byEndTimeSec

End time of scheduled mode, unit: second, its value is between 0 and 59.

byAlarmTrigState

Alarm input triggering status: 0-day, 1-night.

A.65 NET_DVR_DAYTIME

Time parameter structure.

Structure Definition

```
struct{
    BYTE  byHour;
    BYTE  byMinute;
    BYTE  bySecond;
    BYTE  byRes;
    WORD   wMilliSecond;
    BYTE  byRes1[2];
}NET_DVR_DAYTIME, *LPNET_DVR_DAYTIME;
```

Members

byHour

Hour, ranges from 0 to 24

byMinute

Minute, ranges from 0 to 60

bySecond

Second, ranges from 0 to 60

byRes

Reserved, set to 0

wMilliSecond

Millisecond, ranges from 0 to 1000

byRes1

Reserved, set to 0

A.66 NET_DVR_DEC_DDNS_DEV

Configuration parameter structure about getting stream by DDNS.

Structure Definition

```
struct{
    NET_DVR_DEV_DDNS_INFO      struDdnsInfo;
    NET_DVR_STREAM_MEDIA_SERVER struMediaServer;
}NET_DVR_DEC_DDNS_DEV,*LPNET_DVR_DEC_DDNS_DEV;
```

Members

struDdnsInfo

DDNS configuration parameters, see the structure [NET_DVR_DEV_DDNS_INFO](#) for details.

struMediaServer

Stream media server configuration parameters, see the structure [NET_DVR_STREAM_MEDIA_SERVER](#) for details.

See Also

[NET_DVR_DEC_STREAM_MODE](#)

A.67 NET_DVR_DEFOGCFG

Defog parameter structure

Structure Definition

```
struct{
    BYTE  byMode;
    BYTE  byLevel;
    BYTE  byRes[6];
}NET_DVR_DEFOGCFG, *LPNET_DVR_DEFOGCFG;
```

Members

byMode

Defog mode: 0-disable, 1-auto, 2-remain enabled

byLevel

Defog level, value range: [0,100]

byRes

Reserved, set to 0.

A.68 NET_DVR_DEVICECFG_V50

Structure about Device Basic Parameters

Member	Data Type	Description
dwSize	DWORD	Structure size.
sDVRName	BYTE	DVR name. The maximum length is "NET_DEV_NAME_LEN" (64 bytes).
dwDVRID	DWORD	DVR ID, which is for remote control. The ID ranges from 0 to 99 (v1.4) or 0 to 255 (v1.5).
dwRecycleRecord	DWORD	Whether to enable continuous recording: 0-no, 1-yes.
sSerialNumber	BYTE	Serial No. The maximum length is "SERIALNO_LEN" (48 bytes).
dwSoftwareVersion	DWORD	Software version No., the high 16-bit represents the major version and the low 16-bit represents minor version.
dwSoftwareBuildDate	DWORD	Software build date, whose format is 0xYYYYMMDD.
dwDSPSoftwareVersion	DWORD	DSP software version No., the high 16-bit represents the major version and the low 16-bit represents minor version.
dwDSPSoftwareBuildDate	DWORD	DSP software build date, whose format is 0xYYYYMMDD.
dwPanelVersion	DWORD	Front panel version No., the high 16-bit represents the major version and the low 16-bit represents minor version.
dwHardwareVersion	DWORD	Hardware version No., the high 16-bit represents the major version and the low 16-bit represents minor version.
byAlarmInPortNum	BYTE	Number of DVR alarm inputs.
byAlarmOutPortNum	BYTE	Number of DVR alarm outputs.
byRS232Num	BYTE	Number of DVR RS-232 serial ports.

Member	Data Type	Description
byRS485Num	BYTE	Number of DVR RS-485 serial ports.
byNetworkPortNum	BYTE	Number of network interfaces.
byDiskCtrlNum	BYTE	Number of DVR HDD controllers.
byDiskNum	BYTE	Number of DVR HDDs.
byDVRTypE	BYTE	DVR types: 1-DVR, 2-ATM DVR, 3-DVS.
byChanNum	BYTE	Number of DVR channels.
byStartChan	BYTE	Start channel No.
byDecordChans	BYTE	Number of DVR decoding channels.
byVGANum	BYTE	Number of VGA ports.
byUSBNum	BYTE	Number of USB ports.
byAuxoutNum	BYTE	Number of auxiliary ports.
byAudioNum	BYTE	Number of audio ports.
byIPChanNum	BYTE	Maximum number of digital channels.
byZeroChanNum	BYTE	Number of zero channels.
bySupport	BYTE	<p>Capability supporting information. The bitwise AND result: 0-not support, 1-support.</p> <ul style="list-style-type: none"> • bySupport&0x1: whether it supports VCA search • bySupport&0x2: whether it supports backup • bySupport&0x4: whether it supports getting encoding configuration capability • bySupport&0x8: whether it supports multi-NIC • bySupport&0x10: whether it supports remote SADP • bySupport&0x20: whether it supports RAID card. • bySupport&0x40: whether it supports IPSAN search • bySupport&0x80: whether it supports RTP over RTSP

Member	Data Type	Description
byEsataUseage	BYTE	Esata's usage: 0-for backup, 1-for recording.
byIPCPlug	BYTE	Whether to enable plug and play: 0-no, 1-yes.
byStorageMode	BYTE	Storage mode: 0-disk array mode, 1-disk quota, 2-frame extracting.
bySupport1	BYTE	<p>Capability supporting information. The bitwise AND result: 0-not support, 1-support.</p> <ul style="list-style-type: none"> • bySupport1&0x1: whether it supports SNMP v30 • bySupport1&0x2: whether it supports distinguishing the playback and downloading • bySupport1&0x4: whether it supports setting arming priority • bySupport1&0x8: whether it supports extending the arming schedule of intelligent device • bySupport1&0x10: whether it supports multiple disks (more than 33) • bySupport1&0x20: whether it supports RTSP over HTTP
wDevType	WORD	Extended device type.
byDevTypeName	BYTE	Device type name. The maximum length is "NET_DEV_TYPE_NAME_LEN" (64 bytes).
bySupport2	BYTE	<p>Capability supporting information. The bitwise AND result: 0-not support, 1-support.</p> <p>bySupport2&0x1: whether it supports text overlay</p>
byAnalogAlarmInPortNum	BYTE	Number of analog alarm inputs.
byStartAlarmInNo	BYTE	Start No. of analog alarm inputs.
byStartAlarmOutNo	BYTE	Start No. of analog alarm outputs.
byStartIPAlarmInNo	BYTE	Start No. of digital alarm inputs.

Member	Data Type	Description
byStartIPAlarmOutNo	BYTE	Start No. of digital alarm outputs.
byHighIPChanNum	BYTE	Number of digital channels.
byEnableRemotePowerOn	BYTE	Whether to enable remote power on when the device is in sleep mode: 0-no, 1-yes.
byRes	BYTE	Reserved. The maximum length is 256 bytes.

A.69 NET_DVR_DEV_DDNS_INFO

Configuration parameter structure about DDNS.

Structure Definition

```
struct{
    BYTE   byDevAddress[MAX_DOMAIN_NAME/*64*/];
    BYTE   byTransProtocol;
    BYTE   byTransMode;
    BYTE   byDdnsType;
    BYTE   byRes1;
    BYTE   byDdnsAddress[MAX_DOMAIN_NAME/*64*/];
    WORD   wDdnsPort;
    BYTE   byChanType;
    BYTE   byFactoryType;
    DWORD  dwChannel;
    BYTE   byStreamId[STREAM_ID_LEN/*32*/];
    BYTE   sUserName[NAME_LEN/*32*/];
    BYTE   sPassword[PASSWD_LEN/*16*/];
    WORD   wDevPort;
    BYTE   byRes2[2];
}NET_DVR_DEV_DDNS_INFO,*LPNET_DVR_DEV_DDNS_INFO;
```

Members

byDevAddress

Device domain name

byTransProtocol

Transfer protocol type: 0-TCP, 1-UDP, 2-multicast

byTransMode

Stream mode: 0-main stream, 1-sub-stream

byDdnsType

DDNS type: 0-IPServer, 1-Dyndns, 2- PeanutHull, 3- NO-IP, 4-hiDDNS

byRes1

Reserved

byDdnsAddress

DDNS address

wDdnsPort

DDNS port number

byChanType

Channel type: 0-normal channel, 1-channel-zero, 2-stream ID

byFactoryType

Front-end device manufacturer

dwChannel

Device channel No.

byStreamId

Stream ID, it is valid only when the **byChanType** is 2.

sUserName

User name for log in to device

sPassword

Device password.

wDevPort

Device port number

byRes2

Reserved

See Also

[NET_DVR_DEC_DDNS_DEV](#)

A.70 NET_DVR_DEC_STREAM_DEV_EX

Configuration parameter structure about getting stream from device.

Structure Definition

```
struct{
    NET_DVR_STREAM_MEDIA_SERVER  struStreamMediaSrvCfg;
```

```
NET_DVR_DEV_CHAN_INFO_EX    struDevChanInfo;
}NET_DVR_DEC_STREAM_DEV_EX,*LPNET_DVR_DEC_STREAM_DEV_EX;
```

Members

struStreamMediaSrvCfg

Stream media server configuration parameters, see the structure [NET_DVR_STREAM_MEDIA_SERVER](#) for details.

struDevChanInfo

Device channel configuration parameters, see the structure [NET_DVR_DEV_CHAN_INFO_EX](#) for details.

See Also

[NET_DVR_DEC_STREAM_MODE](#)

A.71 NET_DVR_DEC_STREAM_MODE

Configuration parameter union about streaming mode.

Structure Definition

```
union{
    NET_DVR_DEC_STREAM_DEV_EX  struDecStreamDev;
    NET_DVR_PU_STREAM_URL     struUrlInfo;
    NET_DVR_DEC_DDNS_DEV       struDdnsDeclInfo;
    BYTE                      byRes[300];
}NET_DVR_DEC_STREAM_MODE,*LPNET_DVR_DEC_STREAM_MODE;
```

Members

struDecStreamDev

Get stream from device or stream media server by IP address or domain name, see the structure [NET_DVR_DEC_STREAM_DEV_EX](#) for the configuration details.

struUrlInfo

Get stream from device or stream media server by URL, see the structure [NET_DVR_PU_STREAM_URL](#) for the configuration details.

struDdnsDeclInfo

Get stream from device by DDNS, see the structure [NET_DVR_DEC_DDNS_DEV](#) for the configuration details.

byRes

Reserved, set to 0.

A.72 NET_DVR_DEFOCUS_ALARM

Defocus alarm information structure

Structure Definition

```
struct{
    DWORD      dwSize;
    NET_VCA_DEV_INFO struDevInfo;
    WORD       wDevInfolvmsChannelEx;
    BYTE       byRes1[2];
    DWORD      dwRelativeTime;
    DWORD      dwAbsTime;
    BYTE       byTimeDiffFlag;
    char       cTimeDifferenceH;
    char       cTimeDifferenceM;
    BYTE       byRes[49];
}NET_DVR_DEFOCUS_ALARM,*LPNET_DVR_DEFOCUS_ALARM;
```

Members

dwSize

Structure size.

struDevInfo

Front-end device information, refer to the structure [NET_VCA_DEV_INFO](#) for details.

wDevInfolvmsChannelEx

Extended the parameter **bylvmsChannel** in [NET_VCA_DEV_INFO](#), its value range is extended.

byRes1

Reserved, set to 0.

dwRelativeTime

Time of UTC ± 00:00, which is valid only when the value of **byTimeDiffFlag** is "1".

dwAbsTime

Local time.

byTimeDiffFlag

Whether the time difference parameter is valid: 0-invalid, 1-valid.

cTimeDifferenceH

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 **byTimeDiffFlag** is "1".

cTimeDifferenceM

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 **byTimeDiffFlag** is "1".

byRes

Reserved, set to 0.

A.73 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
bySubProto	BYTE	Transmission protocol type of sub-stream: 0-private protocol (default), 1-RTSP, 2-private protocol+RTSP
bySupport	BYTE	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. <ul style="list-style-type: none"> • bySupport&0x1: whether supports VCA search. • bySupport&0x2: whether supports backup.

Member	Data Type	Description
		<ul style="list-style-type: none"> • bySupport&0x4: whether supports getting encoding parameters. • bySupport&0x8: whether supports dual-NIC. • bySupport&0x10: whether supports remote SADP. • bySupport&0x20: whether supports RAID card. • bySupport&0x40: whether supports searching in IPSAN directory. • bySupport&0x80: whether supports RTP over RTSP.
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"> • bySupport1&0x1: whether supports SNMP with version 30. • bySupport1&0x2: whether supports playback and downloading video files. • bySupport1&0x4: whether supports setting the arming priority. • bySupport1&0x8: whether supports extending the arming time period. • bySupport1&0x10: whether supports multiple HDDs (more than 33). • bySupport1&0x20: whether supports RTP over RTSP. • bySupport1&0x80: whether supports license plate recognition alarm.
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"> • bySupport2&0x1: whether supports getting stream via URL. • bySupport2&0x2: whether supports FTP with version 40. • bySupport2&0x4: whether supports ANR.

Member	Data Type	Description
		<ul style="list-style-type: none"> • bySupport2&0x20: whether supports getting device status. • bySupport2&0x40: whether supports encrypting stream.
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"> • bySupport3&0x1: whether supports multi-stream. • bySupport3&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). • bySupport3&0x20: whether supports getting stream via DDNS.
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"> • byMultiStreamProto&0x1: whether supports third-stream. • byMultiStreamProto&0x2: whether supports fourth-stream. • byMultiStreamProto&0x40: whether supports main stream. • byMultiStreamProto&0x80: whether supports sub-stream.
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	Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not

Member	Data Type	Description
		<p>supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> • bySupport4&0x01: whether all stream types support RTSP and private protocol. • bySupport4&0x02: whether the device supports transmitting form format data via API (NET_DVR_STDXMLConfig). • bySupport4&0x10: whether supports loading network disk by domain name.
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"> • byLanguageType ==0: this field is not supported by device. • byLanguageType&0x1: whether supports Chinese. • byLanguageType&0x2: whether supports English.
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.74 NET_DVR_DEVSERVER_CFG

Structure of module service configuration

Structure Definition

```
struct{
    DWORD dwSize;
    BYTE byIrLampServer;
    BYTE bytelnetServer;
    BYTE byABFServer;
    BYTE byEnableLEDStatus;
    BYTE byEnableAutoDefog;
    BYTE byEnableSupplementLight;
    BYTE byEnableDeicing;
    BYTE byEnableVisibleMovementPower;
    BYTE byEnableThermalMovementPower;
    BYTE byEnablePtzPower;
    BYTE byPowerSavingControl;
    BYTE byCaptureWithSupplimentLightEnabled;
    BYTE byRes[244];
}NET_DVR_DEVSERVER_CFG,*LPNET_DVR_DEVSERVER_CFG;
```

Members

dwSize

Structure size

byIrLampServer

IR lamp settings: 0- disable, 1- enable

bytelnetServer

Telnet settings: 0- disable, 1- enable

byABFServer

ABF settings: 0- enable, 1- disable

byEnableLEDStatus

LED light setting: 0-disable, 1-enable

byEnableAutoDefog

Auto defog control: 0- enable, 1- disable

byEnableSupplementLight

Supplement light control: 0- enable, 1- disable

byEnableDeicing

Enable deicing function: 0- No, 1- Yes

byEnableVisibleMovementPower

Enable visible light function: 0- No, 1- Yes

byEnableThermalMovementPower

Enable thermal imaging function: 0- No, 1- Yes

byEnablePtzPower

Enable PTZ: 0- No, 1- Yes

byPowerSavingControl

Low-power consumption strategy: 0-reserved, 1-sleepy mode, 2-low-power mode. When in low-power mode, the visible light, thermal imaging, and PTZ function will all take effect.

byCaptureWithSupplimentLightEnabled

Enable snapshot supplement light or not: 0-disable, 1-enable

byRes

reserved

Remarks

- Only admin user can set the telnet.
- ABF: Auto Back Focus, which realizes the focus adjustment of the sensor. While, normal focus mode only realizes the auto or manual focus adjustment of the lens.

A.75 NET_DVR_DEV_CHAN_INFO_EX

Structure about front-end device information.

Structure Definition

```
struct{
    BYTE   byChanType;
    BYTE   byStreamId[STREAM_ID_LEN/*32*/];
    BYTE   byRes1[3];
    DWORD  dwChannel;
    BYTE   byRes2[24];
    BYTE   byAddress[MAX_DOMAIN_NAME/*64*/];
    WORD   wDVRPort;
    BYTE   byChannel;
    BYTE   byTransProtocol;
    BYTE   byTransMode;
    BYTE   byFactoryType;
    BYTE   byDeviceType;
    BYTE   byDispChan;
    BYTE   bySubDispChan;
    BYTE   byResolution;
    BYTE   byRes[2];
    BYTE   sUserName[NAME_LEN/*32*/];
```

```
BYTE sPassword[PASSWD_LEN/*16*/];
}NET_DVR_DEV_CHAN_INFO_EX,*LPNET_DVR_DEV_CHAN_INFO_EX;
```

Members

byChanType

Channel type: 0-normal channel, 1-channel-zero, 2-stream ID, 3-local input source

byStreamId

Stream ID, it is valid only when the **byChanType** is 2.

byRes1

Reserved

dwChannel

Channel No., it is valid only when the byChanType is 0, 1, 3. If the channel type is local input source, this member is the local source index No.

byRes2

Reserved

byAddress

Device IP address or domain name

wDVRPort

Device port number

byChannel

Invalid

byTransProtocol

Transfer protocol type: 0-TCP, 1-UDP

byTransMode

Stream mode: 0-main stream, 1-sub-stream

byFactoryType

Front-end device manufacturer

byDeviceType

Device type (for MVC only): 1-network camera, 2-encoding device

byDispChan

Display channel No.

bySubDispChan

Sub display channel No.

byResolution

Resolution: 1- CIF, 2- 4CIF, 3- 720P, 4- 1080P, 5- 500W

byRes

Reserved, set to 0.

sUserName

User name for log in to device.

sPassword

Device password.

See Also

[NET_DVR_DEC_STREAM_DEV_EX](#)

A.76 NET_DVR_DIRECTION

Direction structure.

Structure Definition

```
struct {
    NET_VCA_POINT struStartPoint;
    NET_VCA_POINT struEndPoint;
}NET_DVR_DIRECTION, *LPNET_DVR_DIRECTION;
```

Members

struStartPoint

Start point of the direction, see details in the structure .

struEndPoint

End point of the direction, see details in the structure .

A.77 NET_DVR_DISKSTATE

Table A-14 HDD Status Structure

Member	Data Type	Description
dwVolume	DWORD	HDD capacity, unit: MB.
dwFreeSpace	DWORD	Free space of HDD, unit: MB.
dwHardDiskStatic	DWORD	HDD status: 0-active, 1-sleepy, 2-exception, 3-sleepy HDD error, 4-unformatted, 5-network disconnected, 6-formatting.

A.78 NET_DVR_DISK_QUOTA

Disk quota information structure

Structure Definition

```
struct{
    BYTE   byQuotaType;
    BYTE   byRes1[5];
    WORD   wStoragePeriod;
    DWORD  dwHCapacity;
    DWORD  dwLCapacity;
    DWORD  dwHUsedSpace;
    DWORD  dwLUsedSpace;
    BYTE   byQuotaRatio;
    BYTE   byRes2[21];
}NET_DVR_DISK_QUOTA,*LPNET_DVR_DISK_QUOTA;
```

Members

byQuotaType

Disk quota type: 1-by capacity, 2-by proportion, 3-by time.

byRes1

Reserved.

wStoragePeriod

Video saving period, unit: day, it is valid when **byQuotaType** is 3.

dwHCapacity

High 32-bit of allocated disk capacity, unit: MB

dwLCapacity

Low 32-bit of allocated disk capacity, unit: MB

dwHUsedSpace

High 32-bit of used disk capacity, unit: MB

dwLUsedSpace

Low 32-bit of used disk capacity, unit: MB

byQuotaRatio

Allocated disk proportion, unit: %

byRes2

Reserved.

See Also

A.79 NET_DVR_DISK_QUOTA_CFG

Disk quota settings structure

Structure Definition

```
struct{
    DWORD      dwSize;
    NET_DVR_DISK_QUOTA struPicQuota;
    NET_DVR_DISK_QUOTA struRecordQuota;
    BYTE       res2[60];
}NET_DVR_DISK_QUOTA_CFG,*LPNET_DVR_DISK_QUOTA_CFG;
```

Members

dwSize

Structure size.

struPicQuota

Disk capacity for saving pictures, refer to the structure for details.

struRecordQuota

Disk capacity for saving videos, refer to the structure for details.

res2

Reserved, set to 0.

A.80 NET_DVR_DISK_QUOTA_CFG_V60

Structure about disk quota parameters.

Structure Definition

```
struct{
    DWORD      dwSize;
    NET_DVR_DISK_QUOTA_V60 struPicQuota;
    NET_DVR_DISK_QUOTA_V60 struRecordQuota;
    NET_DVR_DISK_QUOTA_V60 struAddInfoQuota;
    NET_DVR_DISK_QUOTA_V60 struPubInfoFile;
    BYTE       byRes[256];
}NET_DVR_DISK_QUOTA_CFG_V60,*LPNET_DVR_DISK_QUOTA_CFG_V60;
```

Members

dwSize

Structure size.

struPicQuota

Picture quota, see details in the structure [NET_DVR_DISK_QUOTA_V60](#).

struRecordQuota

Video quota, see details in the structure [NET_DVR_DISK_QUOTA_V60](#).

struAddInfoQuota

Additional information quota, see details in the structure [NET_DVR_DISK_QUOTA_V60](#). This member is only used for the cloud storage server, and currently only additional information of heat map and people counting is supported.

struPubInfoFile

Information release file quota, see details in the structure [NET_DVR_DISK_QUOTA_V60](#).

byRes

Reserved, set to 0.

A.81 NET_DVR_DISK_QUOTA_V60

Structure about the disk quota information.

Structure Definition

```
struct{
    BYTE byQuotaType;
    BYTE byRes1[5];
    WORD wStoragePeriod;
    DWORD dwHCapacity;
    DWORD dwLCapacity;
    DWORD dwHUsedSpace;
    DWORD dwLUsedSpace;
    BYTE byQuotaRatio;
    BYTE byRes2[23];
}NET_DVR_DISK_QUOTA_V60, *LPNET_DVR_DISK_QUOTA_V60;
```

Members

byQuotaType

Disk quota type: 1-by capacity, 2-by proportion, 3-by time.

byRes1

Reserved.

wStoragePeriod

Video saving period, unit: day, it is valid when **byQuotaType** is 3.

dwHCapacity

High 32-bit of allocated disk capacity, unit: MB.

dwLCapacity

Low 32-bit of allocated disk capacity, unit: MB.

dwHUsedSpace

High 32-bit of used disk capacity, unit: MB.

dwLUsedSpace

Low 32-bit of used disk capacity, unit: MB.

byQuotaRatio

Allocated disk proportion, unit: %.

byRes2

Reserved.

A.82 NET_DVR_DDE_PARAM

DDE parameter structure

Structure Definition

```
struct{
    BYTE byMode;
    BYTE byNormalLevel;
    BYTE byExpertLevel;
    BYTE byRes[5];
}NET_DVR_DDE_PARAM,*LPNET_DVR_DDE_PARAM;
```

Members

byMode

DDE mode: 1-disabled, 2-normal mode, 3-expert mode.

byNormalLevel

Normal mode level, value range: [1,100], it is valid when **byMode** is set to 2.

byExpertLevel

Expert mode level, value range: [1,100], it is valid when **byMode** is set to 3.

byRes

Reserved, set to 0.

A.83 NET_DVR_DDNS_STREAM_CFG

Structure about IP Server and stream server configuration.

Structure Definition

```

struct{
    BYTE      byEnable;
    BYTE      byRes1[3];
    NET_DVR_IPADDR  struStreamServer;
    WORD      wStreamServerPort;
    BYTE      byStreamServerTransmitType;
    BYTE      byRes2;
    NET_DVR_IPADDR  struIPServer;
    WORD      wIPServerPort;
    BYTE      byRes3[2];
    BYTE      sDVRName[NAME_LEN/*32*/];
    WORD      wDVRNameLen;
    WORD      wDVRSerialLen;
    BYTE      sDVRSerialNumber[SERIALNO_LEN/*48*/];
    BYTE      sUserName[NAME_LEN/*32*/];
    BYTE      sPassWord[PASSWD_LEN/*16*/];
    WORD      wDVRPort;
    BYTE      byRes4[2];
    BYTE      byChannel;
    BYTE      byTransProtocol;
    BYTE      byTransMode;
    BYTE      byFactoryType;
}NET_DVR_DDNS_STREAM_CFG, *LPNET_DVR_DDNS_STREAM_CFG;

```

Members

byEnable

Whether to enable: 0-no, 1-yes

byRes1

Reserved, set to 0

struStreamServer

IP address of stream media server, refer to the structure [NET_DVR_IPADDR_UNION](#) for details.

wStreamServerPort

Stream server port number

byStreamServerTransmitType

Transmission protocol type of stream media server: 0-TCP, 1-UDP

byRes2

Reserved, set to 0

struIPServer

IP address of IP Server, refer to the structure [NET_DVR_IPADDR_UNION](#) for details.

wIPServerPort

IP Server port No.

byRes3

Reserved, set to 0

sDVRName

Device name

wDVRNameLen

Device name size

wDVRSerialLen

Device serial No. size

sDVRSerialNumber

Device serial No.

sUserName

Device user name

sPassWord

Device password

wDVRRPort

Device port No.

byRes4

Reserved, set to 0

byChannel

Device channel No.

byTransProtocol

Transmission protocol type: 0-TCP, 1-UDP

byTransMode

Steam type: 0-main stream, 1-sub-stream

byFactoryType

Network camera protocol list.

A.84 NET_DVR_DEVICEINFO_V40

Device Parameter Structure (V40)

Member	Data Type	Description
struDeviceV30	<u>NET_DVR_DEVICEINFO_V30</u>	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 bySupportLock 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 bySupportLock 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-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 bySupport&0x1 indicates that this member is reserved; the result of bySupport&0x2 indicates that whether it supports uploading changes: 0-not support, 1-support. This member is the capability set extension.

Member	Data Type	Description
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 iResidualValidity 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.
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 bySupportStreamEncrypt&0x1 indicates

Member	Data Type	Description
		whether to support RTP/TLS streaming, the result of bySupportStreamEncrypt&0x2 indicates whether to support SRTP/UDP streaming, and the result of bySupportStreamEncrypt&0x4 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.85 NET_DVR_DNMODE

Structure about day and night mode parameters of motion detection .

Structure Definition

```
struct{
    BYTE byObjectSize;
    BYTE byMotionSensitive;
    BYTE byRes[6];
}NET_DVR_DNMODE, *LPNET_DVR_DNMODE;
```

Members

byObjectSize

Object size, the ratio between motion object and configured motion region, ranges from 0 to 100 (only detect the object with size larger than **byObjectSize**)

byMotionSensitive

Motion detection sensitivity, ranges from 1 to 100, the larger the more sensitivity, 0xff-disabled

byRes

Reserved, set to 0

See Also

[NET_DVR_MOTION_MULTI_AREAPARAM](#)

A.86 NET_DVR_DPC_PARAM

Structure of Defective Point Correction Parameters

Member	Data Type	Description
dwSize	DWORD	Structure size
dwChannel	DWORD	Channel number
wCtrlType	WORD	Control type, details as follows:
byDPCMode	BYTE	DPC Mode: 0-Manual, 1-Auto Manual Correction: default mode, correct manually by using wCtrlType operation type Auto Correction: correct automatically, in this mode, the parameters wCtrlType and struPoint are invalid.
byRes	BYTE	Reserved, set to 0.
struPoint	<u>NET_VCA_POINT</u>	The input plane coordinates of the image, normalize to 0-1. It is valid when wCtrlType value "DPC_POINT".
byRes1	Array of BYTE	Reserved, set to 0.

Remarks

Control type, details as follows:

wCtrlType Macro Definition	Value	Description
DPC_CORRECT	1	Defective pixel correction
DPC_CORRECT_CANCEL	2	Cancel correction
DPC_CROSS_DISPALY_OPEN	3	Enable the crossed display of defective pixel detection.
DPC_CROSS_DISPALY_CLOSE	4	Disable the crossed display of defective pixel detection.
DPC_POINT	5	Defective pixel correction coordinates
DPC_UP	6	The defective pixel coordinate is upward offset.
DPC_DOWN	7	The defective pixel coordinate is downward offset.
DPC_RIGHT	8	The defective pixel coordinate is offset towards right.
DPC_LEFT	9	The defective pixel coordinate is offset towards left.
DPC_ALL_CORRECT	10	All the defective pixels correction.
DPC_SAVE	11	Save the defective pixels.

A.87 NET_DVR_ELECTRONICSTABILIZATION

Parameter structure of electronic image stabilization

Structure Definition

```
struct{
    BYTE byEnable;
    BYTE byLevel;
    BYTE byRes[6];
}NET_DVR_ELECTRONICSTABILIZATION,*LPNET_DVR_ELECTRONICSTABILIZATION;
```

Members

byMode

Whether to enable electronic image stabilization: 0-no, 1-yes.

byLevel

Level, value range: [0, 100]

byRes

Reserved, set to 0.

A.88 NET_DVR_ETHERNET_V30**Ethernet Configuration Structure**

Member	Data Type	Description
struDVRIP	<u>NET_DVR_IPADDR_UN</u> <u>ION</u>	Device IP address
struDVRIPMask	<u>NET_DVR_IPADDR_UN</u> <u>ION</u>	Mask of device IP address
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.89 NET_DVR_EVENT_CAPTURE_V40

Structure about parameters for capturing event pictures.

Structure Definition

```
struct{
    NET_DVR_JPEGPARA      struJpegPara;
    DWORD                  dwPicInterval;
    NET_DVR_REL_CAPTURE_CHAN_V40 struRelCaptureChan[MAX_PIC_EVENT_NUM/*32*/];
    NET_DVR_REL_CAPTURE_CHAN_V40 struAlarmInCapture[MAX_ALARMIN_CAPTURE/*16*/];
    DWORD                  dwMaxGroupNum;
    BYTE                   byCapTimes;
```

```
    BYTE          byRes[59];
}NET_DVR_EVENT_CAPTURE_V40, *LPNET_DVR_EVENT_CAPTURE_V40;
```

Members

struJpegPara

Quality of the captured event picture, see details in the structure [NET_DVR_JPEGPARA](#).

dwPicInterval

Time interval of capturing event pictures which is between 500 and 65535, unit: ms.

struRelCaptureChan

The array subscript are shown below: 0-motion detection, 1-tampering alarm, 2-video loss, 3-PIR alarm, 4-wireless alarm, 5-distress alarm, 6-intelligent capture, 7-face detection, 8-line crossing detection, 9-intrusion detection, 10-scene change detection, 11-region entering detection, 12-region exiting detection, 13-loitering detection, 14-people gathering detection, 15-rapid motion detection, 16-parking detection, 17-unattended baggage detection, 18-object removal detection. For example, **struRelCaptureChan[0]** indicates that the motion detection will trigger capture, see details in the structure [NET_DVR_REL_CAPTURE_CHAN_V40](#).

struAlarmInCapture

Alarm input that triggered capture. For example, **struAlarmInCapture[0]** indicates alarm input 1, and so forth, see details in the structure [NET_DVR_REL_CAPTURE_CHAN_V40](#).

dwMaxGroupNum

Maximum number of alarm input groups supported by the device. Each group contains 16 alarm inputs.

byCapTimes

Number of captured pictures.

byRes

Reserved.

A.90 NET_DVR_EXCEPTION_V40

Structure of exception parameters(expanded).

Sturcture Definition

```
struct{
    DWORD          dwSize;
    DWORD          dwMaxGroupNum;
    NET_DVR_HANDLEEXCEPTION_V41  struExceptionHandle[MAX_EXCEPTIONNUM_V30];
    BYTE           byRes[128];
}NET_DVR_EXCEPTION_V40,*LPNET_DVR_EXCEPTION_V40;
```

Members

dwSize

Structure size

dwMaxGroupNum

Max. number of exception type groups supported by the device (read only)

struExceptionHandle

Exception handle mode, see the structure [**NET_DVR_HANDLEEXCEPTION_V41**](#) for details. The exception types are shown below:

array[0]-HDD full

array[1]- HDD fault

array[2]- Network broken

array[3]- IP address conflict

array[4]- Illegal access

array[5]- Video input/output standard mismatch

array[6]- Video signal exception

array[7]- Recording exception

array[8]- RAID exception

array[9]- Recording resolution does not match with which of front-end camera

array[10]- Over driving

array[11]- Hot standby exception (for N+1)

array[12]- Temperature exception

array[13]- Sub-system exception

array[14]- Fan exception

array[15]- POE power supply exception

array[16]- POC exception

array[17]- Supply voltage exception

byRes

Reserved, set to 0.

A.91 NET_DVR_EXPOSURE

CCD exposure control parameter structure.

Structure Definition

```
struct{
    BYTE  byExposureMode;
```

```
BYTE byAutoApertureLevel;
BYTE byRes[2];
DWORD dwVideoExposureSet;
DWORD dwRes;
}NET_DVR_BACKGROUND_PIC_CFG,*LPNET_DVR_BACKGROUND_PIC_CFG;
```

Members

byExposureMode

Exposure mode: 0-manual, 1-auto.

byAutoApertureLevel

Auto iris sensitivity, its value is between 0 and 10.

byRes

Reserved, set to 0.

dwVideoExposureSet

Custom video exposure time (unit: us), it is the minimum value when in auto exposure mode.

dwExposureUserSet

Custom exposure time (unit: us), it is snapshot shutter speed when snapshot in CCD mode.

dwRes

Reserved, set to 0.

A.92 NET_DVR_EZVIZ_ACCESS_CFG

Guarding Vision access parameter structure

Structure Definition

```
struct{
    DWORD dwSize;
    BYTE byEnable;
    BYTE byDeviceStatus;
    BYTE byAllowRedirect;
    BYTE byDomainName[MAX_DOMAIN_NAME/*64*/];
    BYTE byRes1;
    BYTE byVerificationCode[NET_SDK_MAX_VERIFICATION_CODE_LEN/*32*/];
    BYTE byNetMode;
    BYTE byOfflineStatus;
    BYTE byRes2[2];
    BYTE byOperateCode[NET_SDK_MAX_OPERATE_CODE_LEN/*64*/];
    BYTE byRes[344];
}NET_DVR_EZVIZ_ACCESS_CFG,*LPNET_DVR_EZVIZ_ACCESS_CFG;
```

Members

dwSize

Structure size.

byEnable

Whether to access to Guarding Vision: 0-no, 1-yes.

byDeviceStatus

Device status: 0-reserved, 1-online, 2-offline

byAllowRedirect

Whether supports redirection: 0-reserved, 1-yes, 2-no

byDomainName

Domain name

byRes1

Reserved.

byVerificationCode

Verification code for Guarding Vision.

byNetMode

Network mode: 0-reserved; 1-auto, 2-wired network in priority, 3-wired network, 4-wireless network.

byOfflineStatus

Device offline details: 0-reserved, 1-invalid verification code. This member is read-only, and it is valid only when **DeviceStatus** is 2.

byRes2

Reserved

byOperateCode

Operation code, which is used for binding device.

byRes

Reserved.

A.93 NET_DVR_FACEDETECTION_SEARCHCOND

Condition structure for searching videos recorded based on face detection.

Structure Definition

```
struct{
    NET_VCA_POLYGON struFacePolygon;
    DWORD           dwPreTime;
```

```
DWORD      dwDelayTime;
BYTE       byRes[5972];
}NET_DVR_FACEDETECTION_SEARCHCOND,*LPNET_DVR_FACEDETECTION_SEARCHCOND;
```

Members

struFacePolygon

Face detection region parameters, refer to the structure for details.

dwPreTime

Pre-alarm time, unit: second.

dwDelayTime

Post-alarm time, unit: second.

byRes

Reserved, set to 0.

See Also

A.94 NET_DVR_FACE_PIC_DATA_INFO

Condition structure for face thumbnail dual-VCA.

Structure Definition

```
struct{
    DWORD      dwImageLen;
    NET_VCA_RECT struVcaRect;
    DWORD      dwFaceScore;
    BYTE       byVcaRectOnly;
    BYTE       byRes1[3];
    DWORD      dwPID;
    DWORD      dwFaceSearchNum;
    NET_VCA_RECT struMultiVcaRect[NET_DVR_MAX_FACE_SEARCH_NUM/*5*/];
    BYTE       byRes[136];
    DWORD      *pImage;
}NET_DVR_FACE_PIC_DATA_INFO, *LPNET_DVR_FACE_PIC_DATA_INFO;
```

Members

dwImageLen

Picture size, it is valid when **byVcaRectOnly** is "0".

struVcaRect

Position coordinates of face thumbnail in the picture (normalized coordinates), it is valid when **byVcaRectOnly** is "0". Refer to the structure for details.

dwFaceScore

Similarity, value range: from 0 to 100.

byVcaRectOnly

Whether to upload picture ID and coordinates: 0-no, 1-yes.

byRes1

Reserved.

dwPID

Picture ID, it is valid when **byVcaRectOnly** is "1".

dwFaceSearchNum

Number of regions to search face, it is valid when **byVcaRectOnly** is "1".

struMultiVcaRect

Position coordinates of face thumbnails in the picture (normalized coordinates), it is valid when **byVcaRectOnly** is "1". Each array indicates a face thumbnail. Refer to the structure for details.

byRes

Reserved, set to 0.

pImage

Buffer for saving picture data, the picture format is JPG, it is valid when **byVcaRectOnly** is "0".

See Also

A.95 NET_DVR_FACE_SUB_PIC_INFO

Structure about human body recognition results.

Structure Definition

```
struct{
    DWORD      dwSimilarity;
    NET_VCA_RECT struVcaRect;
    BYTE       byRes2[236];
}NET_DVR_FACE_SUB_PIC_INFO, *LPNET_DVR_FACE_SUB_PIC_INFO;
```

Members

dwSimilarity

Similarity, value range: from 0 to 100.

struVcaRect

Face region coordinates (normalized coordinates). refer to the structure for details.

byRes2

Reserved, set to 0.

See Also

A.96 NET_DVR_FFC_PARAM

FFC parameter structure

Structure Definition

```
struct{
    BYTE  byMode;
    BYTE  byRes1;
    WORD  wCompensateTime;
    BYTE  byRes2[4];
}NET_DVR_FFC_PARAM,*LPNET_DVR_FFC_PARAM;
```

Members

byMode

FFC mode: 1-scheduled mode, 2-temperature mode, 3-disabled

byRes1

Reserved, set to 0.

wCompensateTime

Time (it is valid when the **byMode** is set to "1"), unit: minute, value: 10, 20, 30, 40, 50, 60, 120, 180, 240.

byRes2

Reserved, set to 0.

A.97 NET_DVR_FILECOND_V50

File search condition structure.

Structure Definition

```
struct{
    NET_DVR_STREAM_INFO      struStreamID;
    NET_DVR_TIME_SEARCH_COND struStartTime;
    NET_DVR_TIME_SEARCH_COND struStopTime;
    BYTE                    byFindType;
    BYTE                    byDrawFrame;
```

```
BYTE           byQuickSearch;
BYTE           byStreamType;
DWORD          dwFileType;
DWORD          dwVolumeNum;
BYTE           byIsLocked;
BYTE           byNeedCard;
BYTE           byOnlyAudioFile;
BYTE           bySpecialFindInfoType;
char            szCardNum[32];
char            szWorkingDeviceGUID[16];
NET_DVR_SPECIAL_FINDINFO_UNION uSpecialFindInfo;
DWORD          dwTimeout;
BYTE           byRes[252];
}NET_DVR_FILECOND_V50, *LPNET_DVR_FILECOND_V50;
```

Members

struStreamID

Stream ID or channel No., see details in the structure of [NET_DVR_STREAM_INFO](#).

struStartTime

Start time, see details in the structure of [NET_DVR_TIME_SEARCH_COND](#).

struStopTime

End time, see details in the structure of [NET_DVR_TIME_SEARCH_COND](#).

byFindType

File storage type for search: 0-search in normal volume, 1-search in storage volume, 2-search in N+1 hot spare.

byDrawFrame

Whether to extract the frame: 0-no, 1-yes.

byQuickSearch

Whether to enable searching by calendar: 0-no, 1-yes.

byStreamType

Stream types: 0-main stream, 1-sub-stream, 2-third stream, 0xff-all.

dwFileType

File types.

dwVolumeNum

Storage volume No., it is valid only when **byFindType** is 1.

byIsLocked

Whether to lock the file: 0-no, 1-yes, 0xff-lock or unlock all files.

byNeedCard

Whether the card search is required: 0-no, 1-yes.

byOnlyAudioFile

Audio or video file: 0-video file, 1-audio file.

bySpecialFindInfoType

Search condition type: 0-invalid, 1-search files with ATM information

szCardNum

Card No., it is valid only when **byNeedCard** is 1.

szWorkingDeviceGUID

Working station GUID, which is obtained from N+1 hot spare, it is valid only when **byFindType** is 2.

uSpecialFindInfo

Specific search condition union, see details in [NET_DVR_SPECIAL_FINDINFO_UNION](#).

dwTimeout

Timeout time of searching for files, value range: [5000,15000], 0-no changes of timeout time.

Unit: millisecond.

byRes

Reserved.

Remarks

For login based on ISAPI protocol, only the continuously recorded video can be searched.

Related API

[NET_DVR_FindFile_V50](#)

A.98 NET_DVR_FIND_LOG_COND

Structure about log search conditions.

Structure Definition

```
struct{
    DWORD      dwSelectMode;
    DWORD      dwMainType;
    DWORD      dwSubType;
    NET_DVR_TIME_V50 struStartTime;
    NET_DVR_TIME_V50 struEndTime;
    BOOL       bOnlySmart;
    BYTE       byRes[128];
}NET_DVR_FIND_LOG_COND, *LPNET_DVR_FIND_LOG_COND;
```

Members

dwSelectMode

Search mode: 0-search all, 1-search by type, 2-search by time, 3-search by time and type.

dwMainType

Major log type.

dwSubType

Minor log type.

struStartTime

Start time of search, refer to the structure for details.

struStopTime

End time of search, refer to the structure for details.

bOnlySmart

Whether to search for log files with SMART information only: 0-no, 1-yes.

byRes

Reserved, set to 0.

Related API

[NET_DVR_FindDVRLog_V50](#)

A.99 NET_DVR_FINDDATA_V50

Structure about file search information.

Structure Definition

```
struct{
    char          sFileName[100];
    NET_DVR_TIME_SEARCH struStartTime;
    NET_DVR_TIME_SEARCH struStopTime;
    NET_DVR_ADDRESS   struAddr;
    DWORD          dwFileSize;
    BYTE           byLocked;
    BYTE           byFileType;
    BYTE           byQuickSearch;
    BYTE           byStreamType;
    DWORD          dwFileIndex;
    char          sCardNum[32];
    BYTE           byRes[256];
}NET_DVR_FINDDATA_V50,*LPNET_DVR_FINDDATA_V50;
```

Members

sFileName

File name, it is invalid when searching by time.

struStartTime

File start time, see details in the structure .

struStopTime

File stop time, see details in the structure .

struAddr

Video segment address, see details in the structure , which is used for cluster playback.

dwFileSize

File size.

byLocked

Whether the file is locked: 0-no, 1-yes.

byFileType

Video file type, it is valid when searching by time. 0-continuous recording, 1- motion detection, 2- alarm triggered, 3- motion detection | alarm, 4-motion detection & alarm, 5-command triggered, 6- manual recording, 7-VCA recording, 10-PIR alarm, 11-wireless alarm, 12-panic alarm, 13-all, 14-intelligent traffic events, 15-line crossing, 16-intrusion, 17-sound exception, 18-scene change, 19-line crossing|intrusion|face detection|sound exception|scene change, 20-face detection, 21-sensor, 22-callback, 23-copy back recording, 24-video tampering, 25-POS recording, 26-region entrance, 27-region exiting, 28-loitering detection, 29-people gathering, 30-fast moving, 31-parking detection, 32-unattended baggage, 33-object removal, 34-fire source detection, 35-tampering detection, 36-ship detection, 37-temperature pre-alarm, 38-temperature alarm, 39-fight detection, 40-getting up detection, 41-sleepy detection, 42-temperature difference alarm, 43-offline temperature measurement alarm, 44-zone alarm, 45-panic alarm, 46-inquiry service, 47-getting up detection, 48-climbing detection, 49-in-toilet overtime, 50-running detection, 51-playground overstay detection, 75-dredger detection alarm. It is valid when searching by time.



Note

When logging in to device by ISAPI method, only continuous recording is supported.

byQuickSearch

Search result type, 0-result of normal search, 2-result of searching by time.

byStreamType

Stream type: 0-main stream, 1-sub-stream, 2-third stream.

dwFileIndex

File No.

sCardNum

Card No.

byRes

Reserved.

Remarks

When logging in to device by ISAPI method, the following parameters **byLocked**, **byQuickSearch**, **byStreamType**, **dwFileIndex**, and **sCard** are not supported.

Related API

[NET_DVR_FindNextFile_V50](#)

A.100 NET_DVR_FIND_LABEL

Condition structure for searching video tags.

Structure Definition

```
struct{
    DWORD      dwSize;
    LONG       lChannel;
    NET_DVR_TIME struStartTime;
    NET_DVR_TIME struStopTime;
    BYTE       sLabelName[LABEL_NAME_LEN/*40*/];
    BYTE       byDrawFrame;
    BYTE       byISO8601;
    signed char cStartTimeDifferenceH;
    signed char cStartTimeDifferenceM;
    signed char cStopTimeDifferenceH;
    signed char cStopTimeDifferenceM;
    BYTE       byRes[34];
}NET_DVR_FIND_LABEL,*LPNET_DVR_FIND_LABEL;
```

Members

dwSize

Structure size.

lChannel

No. of channel to be searched.

struStartTime

Start time of search, refer to the structure for details.

struStopTime

End time of search, refer to the structure for details.

sLabelName

Video tag name, if this parameter is set to "null", it indicates that all tags between the start time and end time will be searched.

byDrawFrame

Whether to extract frame: 0-no, 1-yes.

byISO8601

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.

cStartTimeDifferenceH

Time difference between start time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStartTimeDifferenceM

Time difference between start time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStopTimeDifferenceH

Time difference between stop time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStopTimeDifferenceM

Time difference between stop time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

byRes

Reserved, set to 0.

Related API

[NET_DVR_FindRecordLabel](#)

A.101 NET_DVR_FINDLABEL_DATA

Video tag information structure.

Structure Definition

```
struct{
    BYTE      sLabelName[LABEL_NAME_LEN/*40*/];
    NET_DVR_TIME      struTimeLabel;
    NET_DVR_LABEL_IDENTIFY struLabelIdentify;
    BYTE      byISO8601;
    signed char      cTimeDifferenceH;
    signed char      cTimeDifferenceM;
```

```
    BYTE      byRes1[32];
}NET_DVR_FINDLABEL_DATA,*LPNET_DVR_FINDLABEL_DATA;
```

Members

sTagName

Video tag name.

struTimeLabel

Video tag time, refer to the structure for details.

struLabelIdentify

Video tag ID, refer to the structure for details.

byISO8601

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.

cTimeDifferenceH

Time difference between tag time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cTimeDifferenceM

Time difference between tag time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

byRes

Reserved, set to 0.

Related API

[NET_DVR_FindNextLabel](#)

A.102 NET_DVR_FIND_PICTURE_PARAM

Picture search condition structure.

Structure Definition

```
struct{
    DWORD      dwSize;
    LONG       lChannel;
    BYTE       byFileType;
    BYTE       byNeedCard;
    BYTE       byProvince;
    BYTE       byEventType;
    BYTE       byRes1;
    BYTE       sCardNum[CARDNUM_LEN_V30/*40*/];
    NET_DVR_TIME struStartTime;
```

```

NET_DVR_TIME struStopTime;
DWORD      dwTrafficType;
DWORD      dwVehicleType;
DWORD      dwIllegalType;
BYTE       byLaneNo;
BYTE       bySubHvtType;
BYTE       byRes2[2];
char       sLicense[MAX_LICENSE_LEN/*16*/];
BYTE       byRegion;
BYTE       byCountry;
BYTE       byArea;
BYTE       byISO8601;
signed char cStartTimeDifferenceH;
signed char cStartTimeDifferenceM;
signed char cStopTimeDifferenceH;
signed char cStopTimeDifferenceM;
}NET_DVR_FIND_PICTURE_PARAM,*LPNET_DVR_FIND_PICTURE_PARAM;

```

Members

dwSize

Structure size.

lChannel

Channel No.

byFileType

Picture capture types: 0-continuous capture, 1-capture based on motion detection, 2- capture based on alarm, 3-capture based on motion detection or alarm, 4-capture based on motion detection and alarm, 6-manual capture, 9-intelligent picture, 10-capture based on PIR alarm, 11-capture based on wireless alarm, 12-capture based on panic alarm, 0xa-capture during live view, 0xd-capture based on face detection, 0xe-captured based on line crossing detection, 0xf-captured based on intrusion detection, 0x10-capture based on scene change detection, 0x11-capture during playback, 0x12-capture based on VCA detection, 0x13-capture based on region entrance detection, 0x14-capture based on region exiting detection, 0x15-capture based on loitering detection, 0x16-capture based on people gathering detection, 0x17-capture based on fast moving detection, 0x18-capture based on parking detection, 0x19-captured based on unattended baggage detection, 0x1a-captured based on object removal detection, 0x1b-capture based on license plate detection, 0x1c-capture based on multi-target-type detection, 0x1d-capture based on traffic law enforcement, 0x1e-capture based on fire source detection, 0x1f-capture based on vandal-proof detection, 0x20-capture based on ship detection, 0x21-capture based on temperature pre-alarm, 0x22-capture based on temperature alarm, 0x23-capture based on temperature difference alarm, 0x24-capture based on illegal parking detection, 0x25-face capture, 0x26-capture based on offline thermometry alarm, 0x2a-capture based on getting up detection, 0x2b-capture based on climbing detection, 0x2c-capture based on in-toilet overtime alarm, 0x2d-capture based on hard hat detection, 0x2e-perimeter capture, 0x2f-capture human body, 0x30-face capture and modeling, 0x31-multi-target-type detection, 0x32-

zone alarm, 0x33-panic alarm, 0x34-business consulting, 0x35-capture based on unregistered street vendor detection, 0x36-capture based on people density detection (number of people pre-alarm), 0x37-capture based on absence detection, 0x38-capture based on number of people exception detection, 0x39-capture based on violent motion detection, 0x3a-capture based on illegal parking detection, 0x3b-capture based on wrong-way driving detection, 0x3c-capture based on driving on the lane line detection, 0x3d-capture based on motor vehicle on non-vehicle lane detection, 0x3e-capture based on illegal lane change detection, 0x3f-capture based on U-turning detection, 0x40-capture based on pedestrian detection, 0x41-capture based on roadblock detection, 0x42-capture based on thrown object detection, 0x43-capture based on fog detection, 0x44-capture based on construction detection, 0x45-capture based on congestion detection, 0x46-capture based on traffic accident detection, 0x47-capture based on parallel parking detection, 0x48-trigger alarm manually, 0x4d-dredger detection alarm, 0x52-capture based on people queuing-up detection, 0x53-capture based on waiting time detection, 0x54-capture based on vehicle arming (including manual arming, ordinary arming, and platform arming), 0x55-capture based on waking up by schedule, 0x56-capture based on exhaust emission, 0x57-capture based on grayscale alarm, 0x58-capture based on vibration detection, 0x59-people running detection, 0x5a-people overstay detection, 0x5b-abnormal distance, 0x5c-people falling down, 0x5d-smoke detection, 0x5e-checkpoint detection, 0x5f-prohibition violation, 0x60-occupy emergency lane, 0x61-large-sized vehicle on the lane, 0x62-blocklist, 0x63-occupy dedicated lane, 0x64-smoke detection, 0x65-ReID person track, 0x66-channel scheduled capture event, 0x67-AI open platform event, 0x68-methane density, 0x69-methane light intensity, 0x70-channel scheduled capture, 0xff-all types.

byNeedCard

Whether the card No. is required: 0-no, 1-yes

byProvince

Reserved, set to 0.

byEventType

Event types: 0-all, 1-traffic incident, 2-enforcement.

byRes1

Reserved.

sCardNum

Card No.

struStartTime

Start time of search, refer to the structure for details.

struStopTime

End time of search, refer to the structure for details.

dwTrafficType

Picture search conditions (for intelligent traffic devices only).

```
enum _VCA_OPERATE_TYPE_{
    VCA_LICENSE_TYPE      = 0x1,
    VCA_PLATECOLOR_TYPE   = 0x2,
    VCA_CARDNO_TYPE       = 0x4,
    VCA_PLATETYPE_TYPE    = 0x8,
    VCA_LISTTYPE_TYPE     = 0x10,
    VCA_INDEX_TYPE         = 0x20,
    VCA_OPERATE_INDEX_TYPE = 0x40
}VCA_OPERATE_TYPE
```

VCA_LICENSE_TYPE

License plate number.

VCA_PLATECOLOR_TYPE

License plate color.

VCA_CARDNO_TYPE

Card No.

VCA_PLATETYPE_TYPE

License plate type.

VCA_LISTTYPE_TYPE

Vehicle list type

VCA_INDEX_TYPE

Data serial No.

VCA_OPERATE_INDEX_TYPE

Operation No.

dwVehicleType

Vehicle type (for intelligent traffic devices only).

```
enum _VCA_VEHICLE_TYPE_{
    VCA_OTHER_TYPE        = 0x1,
    VCA_SMALLCAR_TYPE     = 0x2,
    VCA_BIGCAR_TYPE       = 0x4,
    VCA_BUS_TYPE          = 0x8,
    VCA_TRUCK_TYPE        = 0x10,
    VCA_CAR_TYPE          = 0x20,
    VCA_MINIBUS_TYPE      = 0x40,
    VCA_SMALL_TRUCK_TYPE  = 0x80
}VCA_VEHICLE_TYPE
```

VCA_OTHER_TYPE

Other

VCA_SMALLCAR_TYPE

Small-sized vehicle

VCA_BIGCAR_TYPE

Large-sized vehicle

VCA_BUS_TYPE

Coach

VCA_TRUCK_TYPE

Track

VCA_CAR_TYPE

Car

VCA_MINIBUS_TYPE

Minibus

VCA_SMALL_TRUCK_TYPE

Mini track

dwIllegalType

Violation detection type (for intelligent traffic devices only).

```
enum _VCA_ILLEGAL_TYPE_{
    VCA_POSTPIC_TYPE      = 0x1,
    VCA_LOWSPEED_TYPE     = 0x2,
    VCA_HIGHSPEED_TYPE    = 0x4,
    VCA_RETROGRADE_TYPE   = 0x8,
    VCA_RUSH_REDLIGHT_TYPE = 0x10,
    VCA_PRESS_LANE_TYPE   = 0x20,
    VCA_VIOLATE_GUIDE_TYPE = 0x40,
    VCA_ROAD_STRAND_TYPE  = 0x80,
    VCA_VEHICLE_ILLEGAL_LANE_TYPE = 0x100,
    VCA_ILLEGAL_LANE_CHANGE_TYPE = 0x200,
    VCA_ILLEGAL_LANE_DRIVE_TYPE = 0x400,
    VCA_VIOLATE_BAN_TYPE   = 0x800,
    VCA_CROSS_PARKING_TYPE = 0x1000,
    VCA_GREEN_PARKING_TYPE = 0x2000
}VCA_ILLEGAL_TYPE
```

VCA_POSTPIC_TYPE

Checkpoint picture

VCA_LOWSPEED_TYPE

Low speed

VCA_HIGHSPEED_TYPE

Overspeed

VCA_RETROGRADE_TYPE

Wrong-way driving

VCA_RUSH_REDLIGHT_TYPE

Red light running

VCA_PRESS_LANE_TYPE

Driving on lane line

VCA_VIOLATE_GUIDE_TYPE

Driving in wrong lane at intersection

VCA_ROAD_STRAND_TYPE

Intersection congestion

VCA_VEHICLE_ILLEGAL_LANE_TYPE

Motor vehicle on non-motor vehicle lane

VCA_ILLEGAL_LANE_CHANGE_TYPE

Illegal lane change

VCA_ILLEGAL_LANE_DRIVE_TYPE

Driving on wrong lane

VCA_VIOLATE_BAN_TYPE

Ban violation

VCA_CROSS_PARKING_TYPE

Parking at intersection

VCA_GREEN_PARKING_TYPE

Stopping at green light

byLaneNo

Lane No. (for intelligent traffic devices only), value range: [1, 99]

bySubHvtType

Sub search type of multi-target-type detection: 0-reserved, 1-motor vehicle, 2-non-motor vehicle, 3-pedestrian

byRes2

Reserved, set as 0.

sLicense

License plate number, it is valid when **byFileType** is "0x1b".

byRegion

Region No.: 0-reserved, 1-Europe, 2-Russian, 3-Europe and Russian (EU&CIS), 4-the Middle East, 0xff-all.

byCountry

Country index No., refer to **COUNTRY_INDEX** for details.

byArea

Area.

byISO8601

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.

cStartTimeDifferenceH

Time difference between start time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStartTimeDifferenceM

Time difference between start time and UTC time, unit: minute, the value is -30, 0, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStopTimeDifferenceH

Time difference between stop time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStopTimeDifferenceM

Time difference between stop time and UTC time, unit: minute, the value is -30, 0, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

Related API

[NET_DVR_FindPicture](#)

A.103 NET_DVR_FIND_PICTURE_V50

Picture search result structure.

Structure Definition

```
struct{
    char          sFileName[PICTURE_NAME_LEN/*64*/];
    NET_DVR_TIME    struTime;
    DWORD         dwFileSize;
    char          sCardNum[CARDNUM_LEN_V30/*40*/];
    BYTE          byPlateColor;
    BYTE          byVehicleLogo;
    BYTE          byFileType;
    BYTE          byRecogResult;
    char          sLicense[MAX_LICENSE_LEN/*16*/];
    BYTE          byEventSearchStatus;
    BYTE          byRes1[2];
    BYTE          byThermometryUnit;
    float         fFaceSnapTemperature;
    NET_DVR_ADDRESS   struAddr;
    BYTE          byISO8601;
    signed char     cTimeDifferenceH;
    signed char     cTimeDifferenceM;
    BYTE          byRes[68];
```

```
NET_DVR_PIC_EXTRA_INFO_UNION uPicExtraInfo;  
}NET_DVR_FIND_PICTURE_V50,*LPNET_DVR_FIND_PICTURE_V50;
```

Members

sFileName

Picture name.

struTime

Picture time, refer to the structure for details.

dwFileSize

Picture size.

sCardNum

Card No.

byPlateColor

Vehicle color (for intelligent traffic device only), refer to the enumeration definition

[VCA_PLATE_COLOR](#) for details.

byVehicleLogo

Vehicle brand (for intelligent traffic device only). For details, refer to the enumeration

[VLR_VEHICLE_CLASS](#).

byFileType

Picture capture types: 0-continuous capture, 1-capture based on motion detection, 2- capture based on alarm, 3-capture based on motion detection or alarm, 4-capture based on motion detection and alarm, 6-manual capture, 9-intelligent picture, 10-capture based on PIR alarm, 11-capture based on wireless alarm, 12-capture based on panic alarm, 0xa-capture during live view, 0xd-capture based on face detection, 0xe-captured based on line crossing detection, 0xf-captured based on intrusion detection, 0x10-capture based on scene change detection, 0x11-capture during playback, 0x12-capture based on VCA detection, 0x13-capture based on region entrance detection, 0x14-capture based on region exiting detection, 0x15-capture based on loitering detection, 0x16-capture based on people gathering detection, 0x17-capture based on fast moving detection, 0x18-capture based on parking detection, 0x19-captured based on unattended baggage detection, 0x1a-captured based on object removal detection, 0x1b-capture based on license plate detection, 0x1c-capture based on multi-target-type detection, 0x1d-capture based on traffic law enforcement, 0x1e-capture based on fire source detection, 0x20-capture based on ship detection, 0x21-capture based on temperature pre-alarm, 0x22-capture based on temperature alarm, 0x23-capture based on temperature difference alarm, 0x24-capture based on illegal parking detection, 0x25-face capture, 0x26-capture based on offline thermometry alarm, 0x2a-capture based on getting up detection, 0x2b-capture based on climbing detection, 0x2c-capture based on in-toilet overtime alarm, 0x2d-capture based on hardhat detection, 0xff-all types.

byRecogResult

Vehicle recognition results:

sLicense

License plate number

byEventSearchStatus

Search status, 0-no more picture, 1-searching.

byRes1

Reserved, set to 0.

byThermometryUnit

Temperature unit: 0-Celsius, 1-Fahrenheit, 2-Kelvin.

fFaceSnapTemperature

Temperature information of the face in the captured picture, which is accurate to one decimal place, default unit: Celsius. It is valid only when the value of **byFileType** is 0x25; otherwise, the value of this parameter is 0.

struAddr

Picture URL, which is for picture downloading.

byISO8601

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.

cTimeDifferenceH

Time difference between tag time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cTimeDifferenceM

Time difference between tag time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

byRes

Reserved, set to 0.

uPicExtraInfo

Additional information of picture, it is valid only when the **byFileType** is "0xd", refer to the structure NET_DVR_PIC_EXTRA_INFO_UNION in the HCNetSDK user manual of face applications for details.

Related API

[NET_DVR_FindNextPicture_V50](#)

A.104 NET_DVR_FOCUSMODE_CFG

Structure About Cocus Mode Configuration of Speed Dome

Member	Data Type	Description
dwSize	DWORD	Structure size.
byFocusMode	BYTE	Focus mode: 0-auto, 1-manual, 2-semiautomatic.
byAutoFocusMode	BYTE	Auto focus mode: 0-close, 1-mode A, 2-mode B, 3-mode AB, 4-mode C.
wMinFocusDistance	WORD	Minimum focusing distance, unit: CM, 0- auto, 0xffff- infinite.
byZoomSpeedLevel	BYTE	Zoom speed, ranges from 1 to 3.
byFocusSpeedLevel	BYTE	Focus speed, ranges from 1 to 3.
byOpticalZoom	BYTE	Optical zoom, ranges from 0 to 255.
byDigitalZoom	BYTE	Digital zoom, ranges from 0 to 255.
fOpticalZoomLevel	float	Optical zoom, ranges from 1 to 32, minimum interval is 0.5
dwFocusPos	DWORD	Focus value, range: [0x1000,0xC000].
byFocusDefinitionDisplay	BYTE	Whether to display focus definition value: 0-no, 1-yes.
byFocusSensitivity	BYTE	Focus sensitivity, range: [0,2], it is valid when byFocusMode is 0 or 2.
byRes1	Array of BYTE	Reserved, set to 0.
dwRelativeFocusPos	BYTE	Relative focus value, low 16 bytes indicate focus value (ranges from 0 to 4000), and high 16 bytes indicate temperature value under current focus.
byRes	Array of BYTE	Reserved, set to 0.

A.105 NET_DVR_GAIN

Parameter structure of CCD gain

```
struct{
    BYTE byGainLevel;
    BYTE byGainUserSet;
```

```
BYTE byRes[2];
DWORD dwMaxGainValue;
}NET_DVR_GAIN,*LPNET_DVR_GAIN;
```

Members

byGainLevel

Gain value, unit: dB, range: [0,100]

byGainUserSet

Customized gain value, unit: dB, range: [0,100]

byRes

Reserved, set to 0.

dwMaxGainValue

Maximum gain value, unit: dB.

A.106 NET_DVR_GAMMACORRECT

Gamma correction parameter structure

Structure Definition

```
struct{
    BYTE   byGammaCorrectionEnabled;
    BYTE   byGammaCorrectionLevel;
    BYTE   byRes[6];
}NET_DVR_GAMMACORRECT,*LPNET_DVR_GAMMACORRECT;
```

Members

byGammaCorrectionEnabled

Whether to enable Gamma correction: 0-no, 1-yes

byGammaCorrectionLevel

Correction level, which is between 0 and 100.

byRes

Reserved, set to 0.

A.107 NET_DVR_GENERIC_DATA_CFG

Data Information Structure

Member	Data Type	Description
sData	Array of BYTE	Start flag of transaction, the maximum size is 128 bytes (value of macro definition "MAX_POS_FILTER_DATA_LEN").
byFlagType	BYTE	Flag type: 0-string, 1-hexadecimal.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 3 bytes.

A.108 NET_DVR_GENERIC_END

Structure about End Information of Transaction

Member	Data Type	Description
sData	Array of BYTE	Start flag of transaction, the maximum size is 128 bytes (value of macro definition "MAX_POS_FILTER_DATA_LEN").
byExclusive	BYTE	Whether it contains the flag: 0-yes, 1-no.
byFlagType	BYTE	Flag type: 0-string, 1-hexadecimal.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 6 bytes.

A.109 NET_DVR_GENERIC_START

Structure about Start Information of Transaction

Member	Data Type	Description
sData	Array of BYTE	Start flag of transaction, the maximum size is 128 bytes (value of macro definition "MAX_POS_FILTER_DATA_LEN").
byExclusive	BYTE	Whether it contains the flag: 0-yes, 1-no.
byFlagType	BYTE	Flag type: 0-string, 1-hexadecimal.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 2 bytes.

A.110 NET_DVR_GET_STREAM_UNION

Union of streaming mode.

Structure Definition

```
union{
    NET_DVR_IPCHANINFO      struChanInfo;
    NET_DVR_PU_STREAM_CFG_V41 struPUStruct;
    NET_DVR_IPSERVER_STREAM  struIPServerStream;
    NET_DVR_DDNS_STREAM_CFG  struDDNSStream;
    NET_DVR_PU_STREAM_URL    struStreamUrl;
    NET_DVR_HKDDNS_STREAM    struHkDDNSStream;
    NET_DVR_IPCHANINFO_V40   struIPChan;
}NET_DVR_GET_STREAM_UNION,*LPNET_DVR_GET_STREAM_UNION;
```

Members

struChanInfo

Network channel information for getting stream directly, refer to the structure [NET_DVR_IPCHANINFO](#) for details.

struPUStruct

Get stream from stream media server, refer to the structure [NET_DVR_PU_STREAM_CFG_V41](#) for details.

struIPServerStream

Get stream after getting device IP address by IPServer, refer to the structure [NET_DVR_IPSERVER_STREAM](#) for details.

struDDNSStream

Get device IP address by IPServer, and then get stream from stream media server, refer to the structure [NET_DVR_DDNS_STREAM_CFG](#) for details.

struStreamUrl

Get stream from the device or stream media server by URL

struHkDDNSStream

Connect to the device by hkDDNS and then get stream from the device, refer to the structure [NET_DVR_HKDDNS_STREAM](#) for details.

struIPChan

Get stream directly from device (extended), refer to the structure [NET_DVR_IPCHANINFO_V40](#) for details.

See Also

[NET_DVR_STREAM_MODE](#)

A.111 NET_DVR_HANDLEEXCEPTION_V40

Structure about the linkage actions of alarm or exception.

Structure Definition

```
struct{
    DWORD dwHandleType;
    DWORD dwMaxRelAlarmOutChanNum;
    DWORD dwRelAlarmOutChanNum ;
    DWORD dwRelAlarmOut[MAX_CHANNUM_V30/*64*/];
    BYTE byRes[128];
}NET_DVR_HANDLEEXCEPTION_V40, *LPNET_DVR_HANDLEEXCEPTION_V40;
```

Members

dwHandleType

Alarm or exception linkage actions, see details below:

- 0x00: No response
- 0x01: Monitor warning
- 0x02: Audible warning
- 0x04: Upload to center
- 0x08: Trigger alarm output
- 0x10: Capture JPEG picture and send Email
- 0x20: Wireless combined aural and visual alarm linkage
- 0x40: E-map linkage (only supported by PCNVR)
- 0x200: Capture and upload to FTP
- 0x1000: Capture and upload to cloud storage
- 0x8000: message alarm

E.g. dwHandleType==0x01|0x04 indicates that the alarm linkage action is monitor warning and uploading to center.

dwMaxRelAlarmOutChanNum

The maximum number of alarm outputs can be triggered (read only)

dwRelAlarmOutChanNum

Number of triggered alarm outputs

dwRelAlarmOut

Triggered alarm outputs, the No. starts from 1, and ends at the value of **dwRelAlarmOutChanNum**. The initial value is 0xffffffff (not link with channel). E.g., if **dwRelAlarmOutChanNum** is 5, the alarm output can be configured and triggered ranges from **dwRelAlarmOut[0]** to **dwRelAlarmOut[4]**.

byRes

Reserved.

Remarks

- When the value of **dwRelAlarmOutChanNum** is larger than 64, the channels should be grouped, and maximum 64 channels can be in a group.
- The video channel No. is linked with the group No., e.g., if the group No. is 0, the channel No. is between 1 and 64; if the group No. is 1, the channel No. is between 65 and 128; if 0xffffffff appears, it indicates that the following channel No. is invalid.

A.112 NET_DVR_HANDLEEXCEPTION_V41

Exception Information Structure

Member	Data Type	Description
dwHandleType	DWORD	<p>Handling types, see details below:</p> <ul style="list-style-type: none"> 0x00: no response 0x01: display alarm on monitor screen 0x02: audio warning 0x04: upload to center 0x08: trigger alarm output 0x10: send picture with JPEG format by e-mail 0x20: trigger wireless sound and light alarm 0x40: trigger e-map (supported by PCNVR only) 0x200: capture picture and upload to FTP 0x400: focus mode linkage (for defocus detection) 0x800: PTZ linkage (speed dome tracks the target) 0x1000: capture picture and upload to cloud storage. 0x10000: message alarm <p>E.g., if dwHandleType is 0x01 0x04, it indicates that the alarm information will be displayed on monitor screen and uploaded to alarm center when the alarm is triggered.</p>
dwMaxAlarmOutChannelNum	DWORD	Manixmum number of alarm outputs (read only) supported by the device.

Member	Data Type	Description
dwRelAlarmOut	Array of DWORD	Alarm output No. triggered by alarm, which starts from 0, 0xffffffff-invalid. E.g. byRelAlarmOut[i]==3 indicates that the alarm output No.4 is triggered.
byRes	Array of BYTE	Reserved, set to 0.

A.113 NET_DVR_HDCFG_V50

HDD configuration structure

Structure Definition

```
struct{
    WORD        dwSize;
    WORD        dwHDCount;
    NET_DVR_SINGLE_HD_V50  struHDInfoV50[MAX_DISKNUM_V30/*33*/];
}NET_DVR_HDCFG_V50, *LPNET_DVR_HDCFG_V50;
```

Members

dwSize

Structure size.

dwHDCount

Number of local HDDs, this member is read-only.

struHDInfo

HDD parameters, refer to the structure ***NET_DVR_SINGLE_HD_V50*** for details.

Remarks

After setting HDD parameters, you should reboot the device to take effect.

A.114 NET_DVR_HIDDEN_INFORMATION_CFG

Structure about hidden parameter configuration

Hidden Parameter Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
byFuncType	BYTE	Functional type: 0-hide POS information
byRes1	Array of BYTE	Reserved, set to 0. The maximum size is 3 bytes.
struPosInfo	<u>NET_DVR_POS_HIDDE N_INFORMATION</u>	Hidden parameters of POS information, this member is valid when the value of byFuncType is 0.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 1024 bytes.

A.115 NET_DVR_HIDEALARM_V40

Structure about video tampering alarm parameters.

Structure Definition

```
struct{
    DWORD      dwEnableHideAlarm;
    WORD       wHideAlarmAreaTopLeftX;
    WORD       wHideAlarmAreaTopLeftY;
    WORD       wHideAlarmAreaWidth;
    WORD       wHideAlarmAreaHeight;
    DWORD      dwHandleType;
    DWORD      dwMaxRelAlarmOutChanNum;
    DWORD      dwRelAlarmOut[MAX_ALARMOUT_V40/*4128*/];
    NET_DVR_SCHEDTIME struAlarmTime[MAX_DAYS/*7*/][MAX_TIMESEGMENT_V30/*8*/];
    BYTE       byRes[64];
}NET_DVR_HIDEALARM_V40, *LPNET_DVR_HIDEALARM_V40;
```

Members

dwEnableHideAlarm

Whether to enable video tampering alarm: 0-no, 1-yes

wHideAlarmAreaTopLeftX

X-coordinate of video tampering region.

wHideAlarmAreaTopLeftY

Y-coordinate of video tampering region.

wHideAlarmAreaWidth

Width of video tampering region

wHideAlarmAreaHeight

Height of video tampering region

dwHandleType

Alarm linkage actions, see details below:

- 0x00: No response
- 0x01: Monitor warning
- 0x02: Audible warning
- 0x04: Upload to center
- 0x08: Trigger alarm output
- 0x10: Capture JPEG picture and send Email
- 0x20: Wireless combined aural and visual alarm linkage
- 0x40: E-map linkage (only supported by PCNVR)
- 0x200: Capture and upload to FTP
- 0x1000: Capture and upload to cloud storage

E.g. dwHandleType==0x01|0x04 indicates that the alarm linkage action is monitor warning and uploading to center.

dwMaxRelAlarmOutChanNum

The maximum number of alarm outputs can be triggered (read only)

dwRelAlarmOut

Triggered alarm output No., which starts from 0, 0xffffffff-the following is invalid. For example, if the start value of dwRelAlarmOut[8] is 0xffffffff, the alarm output No. (dwRelAlarmOut[0]+1),..., (dwRelAlarmOut[7]+1) will be triggered.

struAlarmTime

Arming time, up to 8 time periods can be set on a day, see details in the structure

[NET_DVR_SCHEDTIME](#).

byRes

Reserved, set to 0

Remarks

By default, the size of whole image is set to 704*576, so the coordinates, width, and height of privacy mask region should be converted to adapt to the size 704*576.

See Also

[NET_DVR_PICCFG_V40](#)

A.116 NET_DVR_HKDdns_STREAM

Structure about hkDDNS streaming configuration.

Structure Definition

```
struct{
    BYTE  byEnable;
    BYTE  byRes[3];
    BYTE  byDDNSDomain[64];
    WORD  wPort;
    WORD  wAliasLen;
    BYTE  byAlias[NAME_LEN/*32*/];
    WORD  wDVRSerialLen;
    BYTE  byRes1[2];
    BYTE  byDVRSerialNumber[SERIALNO_LEN/*48*/];
    BYTE  byUserName[NAME_LEN/*32*/];
    BYTE  byPassWord[PASSWD_LEN/*16*/];
    BYTE  byChannel;
    BYTE  byRes2[11];
}NET_DVR_HKDDNS_STREAM,*LPNET_DVR_HKDDNS_STREAM;
```

Members

byEnable

Whether to enable: 0-no, 1-yes

byRes

Reserved, set to 0

byDDNSDomain

hiDDNS server address

wPort

hkDDNS port No., default: 80

wAliasLen

Alias size

byAlias

Alias

wDVRSerialLen

Serial No.size

byRes1

Reserved, set to 0

byDVRSerialNumber

Serial nNo.

byUserName

Device user name

byPassWord

Device password

byChannel

Device channel No.

byRes2

Reserved, set to 0

See Also

[NET_DVR_GET_STREAM_UNION](#)

A.117 NET_DVR_I_FRAME

Force I frame parameter structure.

Structure Definition

```
struct{
    DWORD dwSize;
    BYTE sStreamID[STREAM_ID_LEN/*32*/];
    DWORD dwChan;
    BYTE byStreamType;
    BYTE byRes[63];
}NET_DVR_I_FRAME,*LPNET_DVR_I_FRAME;
```

Members

dwSize

Structure size

sStreamID

Stream ID

dwChan

Device channel No.

byStreamType

Stream type: 0-main stream, 1-sub-stream, 2-third stream, 3-virtual stream, ...

byRes

Reserved, set to 0.

Related API

[NET_DVR_RemoteControl](#)

A.118 NET_DVR_IGNORE_STRING

Structure about ignoring string information

Structure Definition

```
struct{
    BYTE sData[MAX_POS_FILTER_DATA_LEN/*128*/];
}NET_DVR_IGNORE_STRING, *LPNET_DVR_IGNORE_STRING;
```

Members

sData

Ignored string.

A.119 NET_DVR_IN_PARAM

Structure about input parameters.

Structure Definition

```
struct{
    NET_DVR_BUF_INFO
    struCondBuf;

    NET_DVR_BUF_INFO
    struInParamBuf;
    DWORD dwRecvTimeout;
    BYTE byRes[32];
}NET_DVR_IN_PARAM, *LPNET_DVR_IN_PARAM;
```

Members

struCondBuf

Buffer for storing conditions

struInParamBuf

Buffer for storing input parameters

dwRecvTimeout

Data receiving timeout, unit: ms

byRes

Reserved, set to 0

A.120 NET_DVR_INIT_CFG_ABILITY

Initialization Capability Structure

Member	Data Type	Description
enumMaxLoginUsersNum	INIT_CFG_MAX_NUM	<p>Maximum number of users can log in, 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>
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.121 NET_DVR_INQUEST_SECRET_INFO

Structure of stream encryption key information

Structure Definition

```
struct{
    BYTE sSecretKey[16];
```

```
BYTE byRes[64];  
}NET_DVR_INQUEST_SECRET_INFO,*LPNET_DVR_INQUEST_SECRET_INFO;
```

Members

sSecretKey

Stream encryption key.

byRes

Reserved

Related API

[NET_DVR_InquestSetSecretKey](#)

A.122 NET_DVR_INTRUSION_SEARCHCOND

Condition structure for searching videos recorded based on intrusion detection.

Structure Definition

```
struct{  
    NET_VCA_INTRUSION      struVcaIntrusion[MAX_INTRUSIONREGION_NUM/*8*/];  
    DWORD                  dwPreTime;  
    DWORD                  dwDelayTime;  
    NET_DVR_PTZPOS_INFO    struPTZPosInfo;  
    BYTE                   byAdvanceType;  
    BYTE                   byRes1[3];  
    NET_DVR_ADVANCE_COND_UNION uAdvanceCond;  
    BYTE                   byRes[5348];  
}NET_DVR_INTRUSION_SEARCHCOND,*LPNET_DVR_INTRUSION_SEARCHCOND;
```

Members

struVcaIntrusion

Intrusion detection region parameters, refer to the structure NET_VCA_INTRUSION in the HCNetSDK user manual of behavior analysis applications for details.

dwPreTime

Pre-alarm time, unit: second.

dwDelayTime

Post-alarm time, unit: second.

struPTZPosInfo

Camera's PTZ position information in intrusion detection, refer to the structure for details.

byAdvanceType

Group types: 0-without group No., 1-group with human body properties, 2-group with object's color share. Refer to the structure for details.

byRes1

Reserved, set to 0.

uAdvanceCond

Group attribute.

byRes

Reserved, set to 0.

See Also

A.123 NET_DVR_IPADDR_UNION

IP Address Union

Member	Data Type	Description
szIPv4	char[]	IPv4 address. The maximum length is 16 bytes.
szIPv6	char[]	IPv6 address. The maximum length is 256 bytes.

A.124 NET_DVR_IPCHANINFO

Structure about network device channel information.

Structure Definition

```
struct{
    BYTE  byEnable;
    BYTE  byIPID;
    BYTE  byChannel;
    BYTE  byIPIDHigh;
    BYTE  byRes[32];
}NET_DVR_IPCHANINFO, *LPNET_DVR_IPCHANINFO;
```

Members

byEnable

Network device channel online status, it is read only. 0-connecting to network device by HDVR or NVR failed, the channel is offline; 1-connected, the channel is online.

byIPID

Low 8-byte in device ID, **byIPID==iDevID**/256.

byChannel

Channel No. of network device, for example, if the network channel No.1 of device A corresponds to the channel No.4 of device B, the **byChannel** is 4.

byIPIDHigh

High 8-byte in device ID, **byIPIDHigh==iDevID** /256

byRes

Reserved, set to 0

Remarks

The **iDevID** is the device ID, and **iDevID==byIPIDHigh*256 + byIPID**. It is used to search for the device information (array parameter in **struIPDeviceInfo** in **NET_DVR_IPPARACFG_V40**). The relation between **iDevID** and **iDevInfoIndex** is: **iDevID==iDevInfoIndex+iGroupNO*64+1**.

A.125 NET_DVR_IPCHANINFO_V40

Structure about network device channel information (extended).

Structure Definition

```
struct{
    BYTE   byEnable;
    BYTE   byRes1;
    WORD   wIPID;
    DWORD  dwChannel;
    BYTE   byTransProtocol;
    BYTE   byTransMode;
    BYTE   byFactoryType;
    BYTE   byRes[241];
}NET_DVR_IPCHANINFO_V40, *LPNET_DVR_IPCHANINFO_V40;
```

Members

byEnable

Network device channel online status, read only. 0-connecting to network channel by HDVR or NVR failed, the channel is offline; 1-connected, the channel is online.

byRes1

Reserved, set to 0

wIPID

Network device ID

dwChannel

Network device channel No., e.g., if the network channel No.1 of device A corresponds to the channel No.4 of device B, the byChannel is 4.

byTransProtocol

Transmission protocol type: 0-TCP, 1-UDP, 2-multicast, 0xff-aslf-adaptive

byTransMode

Stream type: 0-main stream, 1-sub-stream

byFactoryType

Call [***NET_DVR_GetIPCProtoList_V41***](#) to get the network camera protocol list.

byRes

Reserved, set as 0

A.126 NET_DVR_IPC_PROTO_LIST_V41

Structure about network camera protocol list.

Structure Definition

```
struct{
    DWORD dwSize;
    DWORD dwProtoNum;
    BYTE *pBuffer;
    DWORD dwBufferLen;
    BYTE byRes[32];
}NET_DVR_IPC_PROTO_LIST_V41,*LPNET_DVR_IPC_PROTO_LIST_V41;
```

Members

dwSize

Structure size

dwProtoNum

Valid number of network camera protocols

pBuffer

Protocol list buffer, used to save structure [***NET_DVR_PROTO_TYPE***](#) (the number of structures is the value of **dwProtoNum**).

dwBufferLen

Buffer size, size sum of structures [***NET_DVR_PROTO_TYPE***](#) (the number of structures is the value of **dwProtoNum**).

byRes

Reserved, set to 0.

Related API

[NET_DVR_GetIPCProtoList_V41](#)

A.127 NET_DVR_IPDEVINFO_V31

Structure about network device information.

Structure Definition

```
struct{
    BYTE      byEnable;
    BYTE      byProType;
    BYTE      byEnableQuickAdd;
    BYTE      byCameraType;
    BYTE      sUserName[NAME_LEN/*32*/];
    BYTE      sPassword[PASSWD_LEN/*16*/];
    BYTE      byDomain[MAX_DOMAIN_NAME/*64*/];
    NET_DVR_IPADDR struIP;
    WORD      wDVRPort;
    BYTE      szDeviceID[DEV_ID_LEN/*1*/];
    BYTE      byEnableTiming;
    BYTE      byRes2;
}NET_DVR_IPDEVINFO_V31, *LPNET_DVR_IPDEVINFO_V31;
```

Members

byEnable

Whether the network device is enabled

byProType

Protocol type: 0-Hikvision private protocol (default), 1-Panasonic private protocol, 2-Sony private protocol,...For getting more protocol types, you can call

[NET_DVR_GetIPCProtoList_V41](#).

byEnableQuickAdd

Whether supports quick adding: 0-no, 1-yes. When quickly adding, the device IP address and protocol type are required.

byCameraType

Camera function: 0-none, 1-teacher tacking, 2-student tracking, 3-teacher panoramic view, 4-student panoramic view, 5-multimedia, 6-teacher location, 7-student location, 8-blackboard writing location, 9-blackboard writing

sUserName

User name

sPassword

Password

byDomain

 Domain name

struIP

 IP address, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

wDVRPort

 Port No.

szDeviceID

 Reserved.

byEnableTiming

 Whether enables automatic time synchronization between NVR and network camera: 0-reserved, 1-no, 2-yes

byRes2

 Reserved, set to 0

Remarks

- When all network channels of one network device are deleted, the "IPID-1" of all network channel parameters in network device channel does not correspond to the subscript value of this network device parameters, the local network device parameters will be deleted.
- If the domain name does not exist and IPv4 address is valid, the IPv4 address will be adopted to access; if both the domain name and IPv4 do not exist, but IPv6 is valid, the IPv6 address will be adopted to access.

A.128 NET_DVR_IPPARACFG_V40

Configuration parameter structure of network devices and channels.

Structure Definition

```
struct{
    DWORD      dwSize;
    DWORD      dwGroupNum;
    DWORD      dwAChanNum;
    DWORD      dwDChanNum;
    DWORD      dwStartDChan;
    BYTE       byAnalogChanEnable[MAX_CHANNUM_V30/*64*/];
    NET_DVR_IPDEVINFO_V31 struIPDevInfo[MAX_IP_DEVICE_V40/*64*/];
    NET_DVR_STREAM_MODE struStreamMode[MAX_CHANNUM_V30/*64*/];
    BYTE       byRes2[20];
}NET_DVR_IPPARACFG_V40, *LPNET_DVR_IPPARACFG_V40;
```

Members

dwSize

Structure size

dwGroupNum

Total number of groups supported by device, read-only

dwAChanNum

Maximum number of analog channels, read-only

dwDChanNum

Number of digital channels, read-only

dwStartDChan

Start No. of digital channel, read-only

byAnalogChanEnable

Whether to enable analog channel: 0-no, 1-yes. E.g., byAnalogChanEnable[i]=1 indicates that the channel No.(i+1) is enabled.

struIPDevInfo

Network device information, see details in [NET_DVR_IPDEVINFO_V31](#).

struStreamMode

Streaming types, see details in the structure [NET_DVR_STREAM_MODE](#).

byRes2

Reserved, set to 0.

A.129 NET_DVR_IPSERVER_STREAM

Structure about IP server mode configuration.

Structure Definition

```
struct{
    BYTE      byEnable;
    BYTE      byRes[3];
    NET_DVR_IPADDR struIPServer;
    WORD      wPort;
    WORD      wDvrNameLen;
    BYTE      byDVRName[NAME_LEN/*32*/];
    WORD      wDVRSerialLen;
    WORD      byRes1[2];
    BYTE      byDVRSerialNumber[SERIALNO_LEN/*48*/];
    BYTE      byUserName[NAME_LEN/*32*/];
    BYTE      byPassWord[PASSWD_LEN/*16*/];
    BYTE      byChannel;
```

```
    BYTE      byRes2[11];
}NET_DVR_IPSERVER_STREAM, *LPNET_DVR_IPSERVER_STREAM;
```

Members

byEnable

Whether to enable: 0-no, 1-yes

byRes

Reserved, set to 0

strIPServer

IPServer IP address, refer to the structure [NET_DVR_IPADDR_UNION](#) for details.

wPort

IPServer port No.

wDvrNameLen

DVR name size

byDVRName

DVR name

wDVRSerialLen

Serial No. size

byRes1

Reserved, set to 0

byDVRSerialNumber

DVR serial No.

byUserName

DVR user name

byPassWord

DVR password

byChannel

DVR channel No.

byRes2

Reserved, set to 0

A.130 NET_DVR_ISP_CAMERAPARAMCFG

Image signal processing (ISP) parameter structure of front-end device.

Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byWorkType;
    BYTE       byRes[3];
    NET_DVR_SCHEDULE_DAYTIME struDayNightScheduleTime;
    NET_DVR_CAMERAPARAMCFG_EX struSelfAdaptiveParam;
    NET_DVR_CAMERAPARAMCFG_EX struDayIspAdvanceParam;
    NET_DVR_CAMERAPARAMCFG_EX struNightIspAdvanceParam;
    BYTE       byRes1[512];
}NET_DVR_ISP_CAMERAPARAMCFG, *LPNET_DVR_ISP_CAMERAPARAMCFG;
```

Members

dwSize

Structure size.

byWorkType

Working mode: 0-self-adaptive, 1-scheduled

byRes

Reserved, set to 0.

struDayNightScheduleTime

Switching time interval, it is valid when **byWorkType** is 1. The configured time interval indicates the day, otherwise, it is the night. Refer to the structure [NET_DVR_SCHEDULE_DAYTIME](#) for details.

struSelfAdaptiveParam

Self-adaptive parameters, it is valid when **byWorkType** is 0. Refer to the structure [NET_DVR_CAMERAPARAMCFG_EX](#) for details.

struDayIspAdvanceParam

Day parameters, it is valid when **byWorkType** is 1. Refer to the structure [NET_DVR_CAMERAPARAMCFG_EX](#) for details.

struNightIspAdvanceParam

Night parameters, it is valid when **byWorkType** is 1. Refer to the structure [NET_DVR_CAMERAPARAMCFG_EX](#) for details.

byRes1

Reserved, set to 0.

Remarks

When **byWorkType** is 1, only a part of parameters in the structure [NET_DVR_CAMERAPARAMCFG_EX](#) support day and night switching.

A.131 NET_DVR_JPEGPARA

Structure about information of picture in JPEG format.

Structure Definition

```
struct{
    WORD    wPicSize;
    WORD    wPicQuality;
}NET_DVR_JPEGPARA,*LPNET_DVR_JPEGPARA;
```

Members

wPicSize

Picture size: 0-CIF(352 × 288/352 × 240), 1-QCIF(176 × 144/176 × 120), 2-4CIF(704 × 576/704 × 480) or D1(720 × 576/720 × 486), 3-UXGA(1600 × 1200), 4-SVGA(800 × 600), 5-HD720P(1280 × 720), 6-VGA(640 × 480), 7-XVGA(1280 × 960), 8-HD900P(1600 × 900), 9-HD1080P(1920 × 1080), 10-2560 × 1920, 11-1600 × 304, 12-2048 × 1536, 13-2448 × 2048, 14-2448 × 1200, 15-2448 × 800, 16-XGA(1024 × 768), 17-SXGA(1280 × 1024), 18-WD1(960 × 576/960 × 480), 19-1080I (1920 × 1080), 20-576 × 576, 21-1536 × 1536, 22-1920 × 1920, 23-320 × 240, 24-720 × 720, 25-1024 × 768, 26-1280 × 1280, 27-1600 × 600, 28-2048 × 768, 29-160 × 120, 75-336 × 256, 78-384 × 256, 79-384 × 216, 80-320 × 256, 82-320 × 192, 83-512 × 384, 127-480 × 272, 128-512 × 272, 161-288 × 320, 162-144 × 176, 163-480 × 640, 164-240 × 320, 165-120 × 160, 166-576 × 720, 167-720 × 1280, 168-576 × 960, 180-180 × 240, 181-360 × 480, 182-540 × 720, 183-720 × 960, 184-960 × 1280, 185-1080 × 1440, 215-1080 × 720 (reserved), 216-360 × 640 (reserved), 218-1440 × 1440, 500-384 × 288, 0xff-Auto.

wPicQuality

Picture quality: 0-high, 1-medium, 2-low

A.132 NET_DVR_JPEG_CAPTURE_CFG_V40

Structure about extended device capture parameters.

Structure Definition

```
struct{
    DWORD        dwSize;
    NET_DVR_TIMING_CAPTURE struTimingCapture;
    NET_DVR_EVENT_CAPTURE_V40 struEventCapture;
    BYTE         byStreamType;
    BYTE         byRes3[19];
}NET_DVR_JPEG_CAPTURE_CFG_V40,*LPNET_DVR_JPEG_CAPTURE_CFG_V40;
```

Members

dwSize

Structure size.

struTimingCapture

Scheduled capture configuration, see details in the structure [**NET_DVR_TIMING_CAPTURE**](#).

struEventCapture

Event capture configuration, see details in the structure [**NET_DVR_EVENT_CAPTURE_V40**](#).

byStreamType

Capture stream type: 0-main stream, 1-sub stream.

byRes3

Reserved, set to 0.

A.133 NET_DVR_LABEL_IDENTIFY

Video tag ID structure.

Structure Definition

```
struct{
    BYTE sLabelIdentify[LABEL_IDENTIFY_LEN/*64*/];
    BYTE byRes[8];
}NET_DVR_LABEL_IDENTIFY,*LPNET_DVR_LABEL_IDENTIFY;
```

Members

sLabelIdentify

Video tag ID.

byRes

Reserved, set to 0.

A.134 NET_DVR_LASER_PARAM_CFG

Laser parameter structure

Structure Definition

```
struct{
    BYTE byControlMode;
    BYTE bySensitivity;
    BYTE byTriggerMode;
    BYTE byBrightness;
```

```
BYTE byAngle;
BYTE byLimitBrightness;
BYTE byEnabled;
BYTE byIllumination;
BYTE byLightAngle;
BYTE byRes[7];
}NET_DVR_LASER_PARAM_CFG, *LPNET_DVR_LASER_PARAM_CFG;
```

Members

byControlMode

Control mode: 0-invalid, 1-auto (default), 2-manual.

bySensitivity

Laser light sensitivity, value range: [0,100], default: 50.

byTriggerMode

Laser light triggering mode: 0-invalid, 1-triggered by camera module (default), 2-triggered by photoresister.

byBrightness

Laser light brightness, value range: [0,255], default: 100; this member is available when the **byControlMode** is set to "2".

byAngle

Laser light range, the range is a circle, so this value is the angle between radiuses, value range: [1,36], 0-invalid, default: 12.

byLimitBrightness

Adjustment range of laser light brightness, value rang: [0,100]; this member is available when the **byControlMode** is set to "1".

byEnabled

Whether to enable manual control of laser light: 0-no, 1-yes.

byIllumination

Laser light strength, value range: [0,100].

byLightAngle

Supplement light angle, value range: [0,100].

byRes

Reserved, set to 0.

A.135 NET_DVR_LINK_ADDR

Structure about IP address and port information of the stream sender and recipient

Member	Data Type	Description
uLocalIP	<i>NET_DVR_IPADDR_UNION</i>	Local IP address.
wLocalPort	HPR_UINT16[]	Local port No. The maximum length is 10 bytes.
byLocalPortNum	HPR_UINT8	Number of local ports.
byRes1	HPR_UINT8[]	Reserved, the maximum length is 3 bytes.
uDevIP	<i>NET_DVR_IPADDR_UNION</i>	Device IP address.
wDevPort	HPR_UINT16[]	Device port No. The maximum length is 10 bytes.
byDevPortNum	HPR_UINT8	Number of device ports.
byRes2	HPR_UINT8[]	Reserved, the maximum length is 3 bytes.
byRes	HPR_UINT8[]	Reserved, the maximum length is 80 bytes.

A.136 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.137 NET_DVR_LOCAL_ASYNC_CFG

Structure about Asynchronous Configuration Parameter

Member	Data Type	Description
bEnable	BOOL	Whether to enable asynchronous configuration: "TRUE"-yes, "FALSE"-no (default).
byRes	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.138 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.139 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.140 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.141 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.142 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.143 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.144 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.145 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.146 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.147 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.148 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.149 NET_DVR_LOCAL_STREAM_CALLBACK_CFG

Key Parameter Structure

Member	Data Type	Description
byPlayBackEndFlag	BYTE	Whether to call back playback end flag:0-No, 1-Yes
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 255 bytes.

A.150 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.151 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.152 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.153 NET_DVR_LOG_V50

Structure about log file information.

Structure Definition

```
struct{
    NET_DVR_TIME_V50 struLogTime;
    DWORD      dwMajorType;
    DWORD      dwMinorType;
    BYTE       sPanelUser[MAX_NAMELEN/*16*/];
    BYTE       sNetUser[MAX_NAMELEN/*16*/];
    NET_DVR_IPADDR struRemoteHostAddr;
    DWORD      dwParaType;
    DWORD      dwChannel;
    DWORD      dwDiskNumber;
    DWORD      dwAlarmInPort;
    DWORD      dwAlarmOutPort;
    DWORD      dwInfoLen;
    char       sInfo[LOG_INFO_LEN/*11840*/];
}
```

```
BYTE      byRes[128];
}NET_DVR_LOG_V50, *LPNET_DVR_LOG_V50;
```

Members

strLogTime

Log time, refer to the structure for details.

dwMajorType

Major log type, see details in [HCNetSDK Log Types](#).

dwMinorType

Minor log type, see details in [HCNetSDK Log Types](#).

sPanelUser

Operator name.

sNetUser

User name.

struRemoteHostAddr

Remote host server address, refer to the structure for details.

dwParaType

This parameter is available for DS-9000 series devices, and it is valid only when the minor log type is "MINOR_START_VT"/"MINOR_STOP_VT" (the parameter is the two-way audio channel No.) and the major log type is "MAJOR_OPERATION", or the minor type is "MINOR_LOCAL_CFG_PARM", "MINOR_REMOTE_GET_PARM"/ "MINOR_REMOTE_CFG_PARM". The available parameter types are shown below:

Parameter Type	Value	Description
PARA_VIDEOOUT	0x1	Video output configuration
PARA_IMAGE	0x2	Image configuration
PARA_ENCODE	0x4	Encoding configuration
PARA_NETWORK	0x8	Network configuration
PARA_ALARM	0x10	Alarm configuration
PARA_EXCEPTION	0x20	Exception configuration
PARA_DECODER	0x40	Decoder configuration
PARA_RS232	0x80	RS232 configuration
PARA_PREVIEW	0x100	Live view configuration
PARA_SECURITY	0x200	User permission configuration

Parameter Type	Value	Description
PARA_DATETIME	0x400	Local system configuration
PARA_FRAMETYPE	0x800	Frame information configuration
PARA_DETECTION	0x1000	Detection configuration

dwChannel

Channel No.

dwDiskNumber

HDD No.

wAlarmInPort

Alarm input No.

dwAlarmOutPort

Alarm output No.

dwInfoLen

Size of log additional information.

sInfo

Log additional information.

byRes

Reserved, set to 0.

Related API

[NET_DVR_FindNextLog_V50](#)

A.154 NET_DVR_MACFILTER_CFG

Structure about configuration parameters of MAC address filter

Structure Definition

```
struct{
    DWORD dwSize;
    BYTE byEnabled;
    BYTE byPermissionType;
    BYTE byRes1[2];
    BYTE szMacAddress[MAC_ADDRESS_NUM/*48*/][MACADDR_LEN/*6*/];
    BYTE byRes[128];
}NET_DVR_MACFILTER_CFG,*LPNET_DVR_MACFILTER_CFG;
```

Members

dwSize

Structure size

byEnabled

Enable MAC address filter or not: 0- Disable, 1- Enable

byPermissionType

MAC address filter type: 0-forbidden, 1-allowed

byRes1

Reserved, and set to 0

szMacAddress

Filtered MAC address, up to 48 MAC address can be supported, and each address is with the length of 6 bytes.

byRes

Reserved, and set to 0

Remarks

The corresponding capability of MAC address filter configuration is [XML_Cap_MACFilter](#), API: [NET_DVR_GetSTDAbility](#), capability set type: NET_DVR_GET_MACFILTER_CAPABILITIES.

A.155 NET_DVR_MESSAGE_CALLBACK_PARAM_V51

Alarm Callback Configuration Parameters

Key Parameter Structure

Member	Data Type	Description
byVcaAlarmJsonType	BYTE	JSON format for alarm transmission (COMM_VCA_ALARM): 0-new JSON format, 1-old JSON format.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 63 bytes.

A.156 NET_DVR_MIME_UNIT

Input Content Details Structure of Message Transmission API (NET_DVR_STDXMLConfig)

Member	Data Type	Description
szContentType	Array of char	Content type (corresponds to Content-Type field in the message), e.g., text/json, text/xml, and so on. The content format must be supported by HTTP.
szName	Array of char	Content name (corresponds to name field in the message), e.g., name="upload".
szFilename	Array of char	Content file name (corresponds to filename field in the message), e.g., filename="C:\Users\test\Desktop\11.txt".
dwContentLen	DWORD	Content size
pContent	char*	Data point
bySelfRead	BYTE	0-External file, 1-Internal data, whose address is specified by szFilename .
byRes	Array of BYTE	Reserved. Set to 0. Maximum: 15 bytes.

See Also

[NET_DVR_XML_CONFIG_INPUT](#)

A.157 NET_DVR_MONITOR_INFO

Structure about network listening mode configuration

Structure Definition

```
struct{
    WORD      wPort;
    BYTE      byRes1[2];
    NET_DVR_IPADDR struRestrictRemoteIP;
    BYTE      byRes[164];
}NET_DVR_MONITOR_INFO,*LPNET_DVR_MONITOR_INFO;
```

Members

wPort

Listening port No.

byRes1

Reserved, set it to 0.

struRestrictRemoteIP

Remote access restriction IP.

byRes

Reserved, set it to 0.

A.158 NET_DVR_MOTION_MODE_PARAM

Parameter structure of motion detection region in different modes.

Structure Definition

```
struct{
    NET_DVR_MOTION_SINGLE_AREA
    struMotionSingleArea;

    NET_DVR_MOTION_MULTI_AREA
    struMotionMultiArea;
}NET_DVR_MOTION_MODE_PARAM, *LPNET_DVR_MOTION_MODE_PARAM;
```

Members

struMotionSingleArea

Single region mode, it is valid when **byConfigurationMode** is "0", refer to the structure **NET_DVR_MOTION_SINGLE_AREA** for details.

struMotionMultiArea

Multi-region mode, it is valid when **byConfigurationMode** is "1" or "2", refer to the structure **NET_DVR_MOTION_MULTI_AREA** for details.

See Also

NET_DVR_MOTION_V40

A.159 NET_DVR_MOTION_MULTI_AREA

Structure about multiple regions configuration of motion detection.

Structure Definition

```
struct{
    BYTE          byDayNightCtrl;
    BYTE          byAllMotionSensitive;
```

```
BYTE           byRes[2];

NET_DVR_SCHEDULE_DAYTIME
    struScheduleTime;

NET_DVR_MOTION_MULTI_AREAPARAM
    struMotionMultiAreaParam[MAX_MULTI_AREA_NUM/*24*/];
BYTE           byRes1[60];
}NET_DVR_MOTION_MULTI_AREA, *LPNET_DVR_MOTION_MULTI_AREA;
```

Members

byDayNightCtrl

Day/night switch: 0-disabled (default), 1-auto, 2-scheduled

byAllMotionSensitive

Motion detection sensitive, it ranges from 0 to 5, if it is 0xff, it indicates that the sensitive settings is disabled, and higher value corresponds to more sensitive.

byRes

Reserved, set to 0

struScheduleTime

Scheduled switching time, it is valid when **byDayNightCtrl** is 2, it is day when time is within period, otherwise, it is night.

struMotionMultiAreaParam

Motion detection region parameter, up to 24 regions are supported.

byRes1

Reserved, set to 0

See Also

NET_DVR_MOTION_MODE_PARAM

A.160 NET_DVR_MOTION_MULTI_AREAPARAM

Structure about motion detection region parameters.

Structure Definition

```
struct{
BYTE      byAreaNo;
BYTE      byRes[3];

NET_VCA_RECT
    struRect;

NET_DVR_DNMODE
```

```
    struDayNightDisable;

NET_DVR_DNMODE
    struDayModeParam;

NET_DVR_DNMODE
    struNightModeParam;
    BYTE      byRes1[8];
}NET_DVR_MOTION_MULTI_AREAPARAM, *LPNET_DVR_MOTION_MULTI_AREAPARAM;
```

Members

byAreaNo

Region ID, starts from 1

byRes

Reserved, set to 0

struRect

Coordinate information of single region (rectangle)

struDayNightDisable

Disable day/night mode, it is valid when **byDayNightCtrl** is 0

struDayModeParam

Day mode, it is valid when **byDayNightCtrl** is 1 or 2

struNightModeParam

Night mode, it is valid when **byDayNightCtrl** is 1 or 2

byRes1

Reserved, set to 0

See Also

[NET_DVR_MOTION_MODE_PARAM](#)

A.161 NET_DVR_MOTION_SINGLE_AREA

Region parameter structure about motion detection in normal mode.

Structure Definition

```
struct{
    BYTE  byMotionScope[64][96];
    BYTE  byMotionSensitive;
    BYTE  byRes[3];
}NET_DVR_MOTION_SINGLE_AREA, *LPNET_DVR_MOTION_SINGLE_AREA;
```

Members

byMotionScope

Whether to set the block as motion detection region: 0-no, 1-yes, other values-reserved, set to 0. In the 96*64 array, for P video standard, only 22*18 is valid; for N video standard, only 22*15 is valid.

byMotionSensitive

Motion detection sensitive, it ranges from 0 to 5, if it is 0xff, it indicates that the sensitive settings is disabled, and higher value corresponds to more sensitive.

byRes

Reserved, set to 0.

Remarks

For P video standard, the image size is 704*576, for N video standard, the image size is 704*480, so the image with the size of 704*576 or 704*480 will be divided in to 22*18 blocks. And then each block can be set as motion detection region. If the image size is larger than 704*576 in P video standard, the image will be narrowed to 704*576 when setting motion detection region.

See Also

[NET_DVR_MOTION_MODE_PARAM](#)

A.162 NET_DVR_MOTION_V40

Structure about motion detection parameters.

Structure Definition

```
struct{
    NET_DVR_MOTION_MODE_PARAM  struMotionMode;
    BYTE           byEnableHandleMotion;
    BYTE           byEnableDisplay;
    BYTE           byConfigurationMode;
    BYTE           byKeyingEnable;
    DWORD          dwHandleType;
    DWORD          dwMaxRelAlarmOutChanNum;
    DWORD          dwRelAlarmOut[MAX_ALARMOUT_V40/*4128*/];
    NET_DVR_SCHEDTIME   struAlarmTime[MAX_DAYS/*7*/][MAX_TIMESEGMENT_V30/*8*/];
    DWORD          dwMaxRecordChanNum;
    DWORD          dwRelRecordChan[MAX_CHANNUM_V40/*512*/];
    BYTE           byDiscardFalseAlarm;
    BYTE           byRes[127];
}NET_DVR_MOTION_V40, *LPNET_DVR_MOTION_V40;
```

Members

struMotionMode

Set motion detection region in the different modes (see the structure **NET_DVR_MOTION_MODE_PARAM** for details).

byEnableHandleMotion

Whether to handle the motion detection: 0-no, 1-yes

byEnableDisplay

Whether to enable highlight display of motion detection: 0-no, 1-yes

byConfigurationMode

Motion detection mode: 0-normal, 1-expert (only for network camera), 2-coordinate system mode (only for NVR)

byKeyingEnable

Whether to enable manual motion detection: 0-no, 1-yes

dwHandleType

Alarm linkage actions, see details below:

- 0x00: No response
- 0x01: Monitor warning
- 0x02: Audible warning
- 0x04: Upload to center
- 0x08: Trigger alarm output
- 0x10: Capture JPEG picture and send Email
- 0x20: Wireless combined aural and visual alarm linkage
- 0x40: E-map linkage (only supported by PCNVR)
- 0x200: Capture and upload to FTP
- 0x1000: Capture and upload to cloud storage

E.g. dwHandleType==0x01|0x04 indicates that the alarm linkage action is monitor warning and uploading to center.

dwMaxRelAlarmOutChanNum

The maximum output channels triggered by alarm (read only)

dwRelAlarmOut

Triggered alarm output No., which starts from 0, 0xffffffff-the following is invalid. For example, if the start value of dwRelAlarmOut[8] is 0xffffffff, the alarm output No. (dwRelAlarmOut[0]+1),..., (dwRelAlarmOut[7]+1) will be triggered.

struAlarmTime

Arming time, up to 8 time periods can be set on a day, see details in the structure **NET_DVR_SCHEDTIME**.

dwMaxRecordChanNum

The maximum number of linked recording channel (read only)

dwRelRecordChan

Actual recording triggered channel, 0xffffffff-the following is invalid.

byRes

Reserved, set to 0

See Also

[NET_DVR_PICCFG_V40](#)

A.163 NET_DVR_MRD_SEARCH_PARAM

Condition structure for searching videos by calendar.

Structure Definition

```
struct{
    DWORD      dwSize;
    NET_DVR_STREAM_INFO  struStreamInfo;
    WORD       wYear;
    BYTE       byMonth;
    BYTE       byDrawFrame;
    BYTE       byStreamType;
    BYTE       byLocalOrUTC;
    BYTE       byRes[30];
}NET_DVR_MRD_SEARCH_PARAM, *LPNET_DVR_MRD_SEARCH_PARAM;
```

Members

dwSize

Structure size.

struStreamInfo

Camera information, refer to the structure for details.

wYear

Year

byMonth

Month

byDrawFrame

Whether to extract the frame: 0-no, 1-yes.

byStreamType

Stream type: 0-main stream, 1-sub-stream

byLocalOrUTC

Time type: 0-device local time, 1-UTC time

byRes

Reserved, set to 0.

A.164 NET_DVR_MRД_SEARCH_RESULT

Result structure for searching videos by calendar.

Structure Definition

```
struct{
    DWORD dwSize;
    BYTE byRecordDistribution[32];
    BYTE byHasEventRecode[31];
    BYTE byRes;
}NET_DVR_MRД_SEARCH_RESULT, *LPNET_DVR_MRД_SEARCH_RESULT;
```

Members

dwSize

Structure size.

byRecordDistribution

Video distribution on the calendar, e.g., byRecordDistribution[0]==1: video exists in the 1st of the month, byRecordDistribution[0]==0: no video in the 1st of the month, byRecordDistribution[1]==1: video exists in the 2nd of the month, byRecordDistribution[2]==1: video exists in the 3rd of the month, and so on.

byHasEventRecode

Video recorded based on event, e.g., byHasEventRecode[0]==1: event video exists in the 1st of the month, byHasEventRecode[0]==0: no event video in the 1st of the month, byHasEventRecode[1]==1: event video exists in the 2nd of the month, byHasEventRecode[2]==1: event video exists in the 3rd of the month, and so on.

byRes

Reserved, set to 0.

A.165 NET_DVR_MULTI_STREAM_COMPRESSIONCFG

Encoding parameter structure of multi-stream.

Structure Definition

```
struct{
    DWORD dwSize;
```

```
DWORD          dwStreamType;
NET_DVR_COMPRESSION_INFO_V30 struStreamPara;
DWORD          dwResolution;
BYTE           byRes[76];
}NET_DVR_MULTI_STREAM_COMPRESSIONCFG,*LPNET_DVR_MULTI_STREAM_COMPRESSIONCFG;
```

Members

dwSize

Structure size.

dwStreamType

Stream type: 0-main stream, 1-sub-stream, 2-event stream, 3-stream 3, and so on.

struStreamPara

Stream encoding parameters, refer to the structure [NET_DVR_COMPRESSION_INFO_V30](#) for details.

dwResolution

The index number of the current resolution when the value of **byResolution** in [NET_DVR_COMPRESSION_INFO_V30](#) is 254 (the actual index number of the current resolution exceeds 254).

byRes

Reserved, set to 0.

A.166 NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND

Condition structure of multi-stream encoding parameter configuration.

Structure Definition

```
struct{
    DWORD      dwSize;
    NET_DVR_STREAM_INFO struStreamInfo;
    DWORD      dwStreamType;
    BYTE       byRes[32];
}NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND,*LPNET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND;
```

Members

dwSize

Structure size.

struStreamInfo

Stream information, i.e., stream ID or channel No., refer to the structure

[NET_DVR_STREAM_INFO](#) for details.

dwStreamType

Stream type: 0-main stream, 1-sub-stream, 2-event linked stream, 3-third-stream, 4-virtual stream, ...

byRes

Reserved, set to 0.

A.167 NET_DVR_NAT_CFG

UPnP configuration structure.

Structure Definition

```
struct{
    DWORD      dwSize;
    WORD       wEnableUpnp;
    WORD       wEnableNat;
    NET_DVR_IPADDR strulpAddr;
    NET_DVR_NAT_PORT struHttpPort;
    NET_DVR_NAT_PORT struCmdPort;
    NET_DVR_NAT_PORT struRtspPort;
    BYTE        byFriendName[64];
    BYTE        byNatType;
    BYTE        byRes1[3];
    NET_DVR_NAT_PORT struHttpsPort;
    NET_DVR_NAT_PORT struSDKOverTLSPort;
    NET_DVR_NAT_PORT struRtsspsPort;
    BYTE        byRes[44];
}NET_DVR_NAT_CFG, *LPNET_DVR_NAT_CFG;
```

Members

dwSize

Structure size

wEnableUpnp

Whether to enable UPnP: 0-no, 1-yes

wEnableNat

Whether to enable UPnP port mapping (NAT) function: 0-no, 1-yes

strulpAddr

IP address of NAT router in LAN, see details in the structure [NET_DVR_IPADDR_UNION](#).

struHttpPort

Mapping configuration of web server HTTP port, see details in the structure

[NET_DVR_NAT_PORT](#).

struCmdPort

Mapping parameters of command port (default: 8000), see details in the structure [**NET_DVR_NAT_PORT**](#).

struRtspPort

Mapping parameters of RTSP port, see details in the structure [**NET_DVR_NAT_PORT**](#).

byFriendName

Service name

byNatType

Mapping mode of UPNP port: 0-manual, 1-auto

byRes1

Reserved, set to 0

struHttpsPort

Mapping parameters of HTTP port, see details in the structure [**NET_DVR_NAT_PORT**](#).

struSDKOverTLSPort

Mapping parameters of SDKOverTLS port, see details in the structure [**NET_DVR_NAT_PORT**](#).

struRtspSPort

Mapping parameters of RTSPS port, see details in the structure [**NET_DVR_NAT_PORT**](#).

byRes

Reserved, set to 0

Remarks

If the mapping mode (**byNatType**) of UPnP port is "auto", the external port number and internal port number will be set as same automatically, and they cannot be edited. If the mapping mode (**byNatType**) of UPnP port is "manual", the external port number can be set as different with the internal port number.

A.168 NET_DVR_NAT_PORT

Port mapping configuration structure

Structure Definition

```
struct{
    WORD  wEnable;
    WORD  wExtPort;
    BYTE  byRes[12];
}NET_DVR_NAT_PORT, *LPNET_DVR_NAT_PORT;
```

Members

wEnable

Whether to enable mapping for this port.

wExtPort

Mapped external port number

byRes

Reserved

See Also

A.169 NET_DVR_NETCFG_V50

Network Configuration Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
struEtherNet	Array of <u>NET_DVR_ETHERNET_V30</u>	Ethernet interface
struRes1	Array of	Reserved, set to 0.
struAlarmHostIpAddr	<u>NET_DVR_IPADDR_UNION</u>	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	<u>NET_DVR_IPADDR_UNION</u>	IP address of domain name server 1
struDnsServer2IpAddr	<u>NET_DVR_IPADDR_UNION</u>	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).

Member	Data Type	Description
wIpResolverPort	WORD	IP resolver port No.
wHttpPortNo	WORD	HTTP port No.
struMulticastIpAddr	<u>NET_DVR_IPADDR_UN ION</u>	Multicast group address
struGatewayIpAddr	<u>NET_DVR_IPADDR_UN ION</u>	Gateway address
struPPPoE	<u>NET_DVR_PPPOECONFIG</u>	PPPoE parameters
byEnablePrivateMulticastDiscovery	BYTE	Private multicast search (SADP): 0-default, 1-enable, 2-disable
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_UN ION</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.170 NET_DVR_NET_RECEIVE

Structure about network receiving mode configuration

Structure Definition

```
struct{
    WORD wNetPort;
```

```
BYTE byRes[2];
}NET_DVR_NET_RECEIVE,*LPNET_DVR_NET_RECEIVE;
```

Members

wNetPort

Network port.

byRes

Reserved, set it to 0.

A.171 NET_DVR_NET_SNIFF

Structure about network detection mode configuration

Structure Definition

```
struct{
    BYTE byEnableSourcePort;
    BYTE byEnableDestAddr;
    BYTE byEnableDestPort;
    BYTE byRes1;
    BYTE bySourceIpAddr[MAX_DOMAIN_NAME/*64*/];
    BYTE byDestinationIpAddr[MAX_DOMAIN_NAME/*64*/];
    WORD wSourcePort;
    WORD wDestinationPort;
    BYTE byRes[16];
}NET_DVR_NET_SNIFF,*LPNET_DVR_NET_SNIFF;
```

Members

byEnableSourcePort

Whether to enable filtering data source port: 0-disable, 1-enable.

byEnableDestAddr

Whether to enable filtering target IP address: 0-disable, 1-enable.

byEnableDestPort

Whether to enable filtering port No. of target address: 0-disable, 1-enable.

byRes1

Reserved, set it to 0.

bySourceIpAddr

Data source IP address.

byDestinationIpAddr

Destination address where data is sent to.

wSourcePort

Port No. of data source.

wDestinationPort

Port No. of destination address where data is sent to.

byRes

Reserved.

A.172 NET_DVR_NOISEREMOVE

Digital noise reduction (DNR) parameter structure

Structure Definition

```
struct{
    BYTE    byDigitalNoiseRemoveEnable;
    BYTE    byDigitalNoiseRemoveLevel;
    BYTE    bySpectralLevel;
    BYTE    byTemporalLevel;
    BYTE    byDigitalNoiseRemove2DEnable;
    BYTE    byDigitalNoiseRemove2DLevel;
    BYTE    byRes[2];
}NET_DVR_NOISEREMOVE, *LPNET_DVR_NOISEREMOVE;
```

Members

byDigitalNoiseRemoveEnable

Whether to enable DNR: 0-no, 1-yes, enable normal mode, 2-yes, enable expert mode.

byDigitalNoiseRemoveLevel

DNR level in normal mode, which is between 0x0 and 0xF.

bySpectralLevel

Spatial strength in expert mode, its value is between 0 and 100.

byTemporalLevel

Temporal strength in expert mode, its value is between 0 and 100.

byDigitalNoiseRemove2DEnable

Whether to enable 2D noise reduction of captured frame: 0-no, 1-yes; this member is only available for intelligent traffic camera.

byDigitalNoiseRemove2DLevel

2D noise reduction level of captured frame, its value is between 0 and 100; this member is only available for intelligent traffic camera.

byRes

Reserved, set to 0.

A.173 NET_DVR_OBJECT_FEATURE

Object's color features structure.

Structure Definition

```
struct{
    BYTE  byColorRate;
    BYTE  byRed;
    BYTE  byGreen;
    BYTE  byBlue;
    BYTE  byRes[32];
}NET_DVR_OBJECT_FEATURE, *LPNET_DVR_OBJECT_FEATURE;
```

Members

byColorRate

The color share threshold of the motion object, unit: percentage, value range: from 0 to 100.

byRed

Red component value in RGB.

byGreen

Green component value in RGB.

byBlue

Blue component value in RGB.

byRes

Reserved, set to 0.

See Also

A.174 NET_DVR_ONLINEUPGRADE_SERVER

Structure of upgrade server status information

Structure Definition

```
struct{
    DWORD  dwSize;
    BYTE   byConnectStatus;
```

```
BYTE  byRes[1019];
}NET_DVR_ONLINEUPGRADE_SERVER, *LPNET_DVR_ONLINEUPGRADE_SERVER;
```

Members

dwSize

Structure size

byConnectStatus

Connection status, 1-unconnected, 2-connected

byRes

Reserved.

Related API

[NET_DVR_GetSTDConfig](#)

A.175 NET_DVR_ONLINEUPGRADE_VERSION_COND

Search condition structure of latest version information

Structure Definition

```
struct{
    DWORD  dwSize;
    BYTE   byCheckFromSvr;
    BYTE   byRes[59];
}NET_DVR_ONLINEUPGRADE_VERSION_COND, *LPNET_DVR_ONLINEUPGRADE_VERSION_COND;
```

Members

dwSize

Structure size

byCheckFromSvr

Whether server detection is required, 1-not required, the regular detection result of the device is returned, 1-required, whether new version is detected via connecting with server

byRes

Reserved.

Related API

[NET_DVR_GetSTDConfig](#)

A.176 NET_DVR_ONLINEUPGRADE_VERSION_RET

Structure of online upgrade version information

Structure Definition

```
struct{
    DWORD dwSize;
    BYTE byNewVersionAvailable;
    BYTE byNewVersion[MAX_VERSIIN_LEN/*64*/];
    BYTE byChangeLog[SDK_LEN_2048/*2048*/];
    BYTE byRes[971];
}NET_DVR_ONLINEUPGRADE_VERSION_RET, *LPNET_DVR_ONLINEUPGRADE_VERSION_RET;
```

Members

dwSize

Structure size

byNewVersionAvailable

Whether the latest version is available, 1- no, 2-yes

byNewVersion

Version No. of the latest version

byChangeLog

changes of the latest version

byRes

Reserved

Related API

[NET_DVR_GetSTDConfig](#)

A.177 NET_DVR_OPTICAL_DEHAZE

Optical defog parameter structure

Structure Definition

```
struct{
    BYTE byEnable;
    BYTE byRes[7];
}NET_DVR_OPTICAL_DEHAZE, *LPNET_DVR_OPTICAL_DEHAZE;
```

Members

byEnable

Whether to enable optical defog: 0-no, 1-yes.

byRes

Reserved, set to 0.

A.178 NET_DVR_OUT_PARAM

Structure about output parameters.

Structure Definition

```
struct{  
    NET_DVR_BUF_INFO  
    struOutBuf;  
    void*      lpStatusList;  
    BYTE       byRes[32];  
}NET_DVR_OUT_PARAM,*LPNET_DVR_OUT_PARAM;
```

Members

struOutBuf

Buffer for storing output parameters

pStatusList

Buffer for storing statuses

byRes

Reserved, set to 0

A.179 NET_DVR_PHONECFG_V50

Phone configuration structure

Structure Definition

```
struct{  
    BYTE  byAllowList[MAX_PHONE_NUM/*32*/];  
    BYTE  byPhonePerssion[32];  
    BYTE  byAlarmHandler[32];  
    BYTE  byAcsPassword[16];  
    BYTE  byName[NAME_LEN/*32*/];
```

```
BYTE byRes[112];
}NET_DVR_PHONECFG_V50,*LPNET_DVR_PHONECFG_V50;
```

Members

byAllowList

Allowlist number

byPhonePersion

Whether to enable certain function of this number, according to bit: 0x0: support receiving alarm sms; 0x1: support sms control online and offline; 0x2: support call control online; 0x3: support sms restart; 0x4: support controlling door operation

byAlarmHandler

whether to send a alarm in sms, values: 0- No, 1- Yes, according to bit: 0x0: full HDD; 0x1: HDD error; 0x2: network disconnected; 0x3: IP address conflict; 0x4: illegal access; 0x5: video signal exception; 0x6: input/ output video standard mismatched; 0x7: no video signal; 0x8: no audio signal; 0x9: alarm input 0xa: video tampering detection alarm; 0xb: motion detection; 0xc: recording exception; 0xd: PIR alarm; 0xe: wireless alarm; 0xf: emergency alarm; 0x10: audio exception detection alarm; 0x11: scene detection alarm; 0x12: defocus detection alarm; 0x13: face detection alarm; 0x14: line crossing detection alarm; 0x15: intrusion detection alarm; 0x16: region exiting detection alarm; 0x17: region entrance alarm; 0x18: loitering detection alarm; 0x19: people gathering detection alarm; 0x1a: fast moving detection alarm; 0x1b: parking detection alarm; 0x1c: unattended baggage detection alarm; 0x1d: object removal detection alarm; 0x1e: data usage exceeds limit, refer to other SDK programming manual

byAcsPassword

Command to open door

byName

Contact name

byRes

Reserved, set as 0

A.180 NET_DVR_PHY_DISK_INFO

Structure of single physical disk parameter.

Structure Definition

```
struct{
    WORD wPhySlot;
    BYTE byType;
    BYTE byStatus;
    BYTE byMode[40];
    DWORD dwHCapacity;
```

```
DWORD dwLCapacity;
BYTE byArryName[MAX_NAMELEN/*16*/];
WORD wArrayID;
BYTE byRes[102];
}NET_DVR_PHY_DISK_INFO, *LPNET_DVR_PHY_DISK_INFO;
```

Members

wPhySlot

Slot number of hard disk

byType

Hard disk information: 0- general, 1- global hot spare, 2- RAID hot spare, 3- RAID disk

byStatus

Hard disk state: 0- normal, 1- demotion, 2- deleted, 3- HDD loss, 4- offline, 5- subnormal, 6- extraneous HDD, 7- exception, 8- SMART status exception, 9- dormancy, 11- SMR disk does not support RAID, 0xff- not-exist

byMode

Hard disk type (string)

dwHCapacity

Higher 32 bits of disk capacity (Unit: kb)

dwLCapacity

Lower 32 bits of disk capacity

wArrayID

Respective RAID ID

byRes

Reserved, please set to 0

A.181 NET_DVR_PICCFG_V40

Image parameter structure.

Structure Definition

```
struct{
    WORD      dwSize;
    BYTE      sChanName[NAME_LEN/*32*/];
    WORD      dwVideoFormat;
    NET_DVR_VICOLOR struViColor;
    WORD      dwShowChanName;
    WORD      wShowNameTopLeftX;
    WORD      wShowNameTopLeftY;
    WORD      dwEnableHide;
```

```
NET_DVR_SHELTER    struShelter[MAX_SHELTERNUM/*4*/];
DWORD              dwShowOsd;
WORD               wOSDTOPLeftX;
WORD               wOSDTOPLeftY;
BYTE               byOSDType;
BYTE               byDispWeek;
BYTE               byOSDAttrib;
BYTE               byHourOsdType;
BYTE               byFontSize;
BYTE               byOSDColortype;
BYTE               byAlignment;
BYTE               byOSDMilliSecondEnable;
NET_DVR_VILOST_V40 struViLost;
NET_DVR_VILOST_V40 struAULost;
NET_DVR_MOTION_V40 struMotion;
NET_DVR_HIDEALARM_V40 struHideAlarm;
NET_DVR_RGB_COLOR  struOsdColor;
DWORD              dwBoundary;
BYTE               byRes[120];
}NET_DVR_PICCFG_V40,*LPNET_DVR_PICCFG_V40;
```

Members

dwSize

Structure size.

sChanName

Channel name.

dwVideoFormat

Video standard (read-only): 0-not support, 1-NTSC, 2-PAL.

struViColor

Image parameter set by time period, see details in the structure [**NET_DVR_VICOLOR**](#).

dwShowChanName

Whether to display the channel name on the image of live view: 0-no, 1-yes (area size is 704*576).

wShowNameTopLeftX

X-coordinate of the display position of channel name.

wShowNameTopLeftY

Y-coordinate of the display position of channel name.

dwEnableHide

Whether to enable the privacy mask: 0-no, 1-yes.

struShelter

Privacy mask parameter, see details in [**NET_DVR_SHELTER**](#).

dwShowOsd

Whether to enable OSD on the image of live view: 0-no, 1-yes (area size is 704*576).

wOSDTopLeftX

X-coordinate of OSD position.

wOSDTopLeftY

Y-coordinate of OSD position.

byOSDType

Displayed date format:

- 0: YYYY-MM-DD (year-month-day)
- 1: MM-DD-YYYY (month-day-year)
- 2: YYYY (year) MM (month) DD (day)
- 3: MM (month) DD (day) YYYY (year)
- 4: DD-MM-YYYY (day-month-year)
- 5: DD (day) MM (month) YYYY (year)
- 6: MM/DD/YYYY (month/day/year)
- 7: YYYY/MM/DD (year/month/day)
- 8: DD/MM/YYYY (day/month/year)

byDispWeek

Whether to display the day of the week: 0-no, 1-yes.

byOSDAttrib

OSD properties (transparent/non-transparent, flicker/non-flicker): 1-transparent, flicker; 2-transparent, non-flicker; 3-non-transparent, flicker; 4-non-transparent, non-flicker.

byHourOsdType

Displayed time format: 0-24-hour, 1-12-hour or am./pm.

byFontSize

Displayed font size: 0-8*16(EN), 1-16*32(EN), 2-32*64(EN), 3-24*48(EN), 4-12*24(EN), 0xff-self-adaptive.

byOSDColorType

Displayed font color: 0-black and white (default), 1-custom.

byAlignment

Alignment type: 0-self-adapted, 1-align to right, 2-align to left, 3-reserved.

byOSDMilliSecondEnable

Whether to display time duration of video: 0-no, 1-yes. Unit: millisecond, this function is only supported by the intelligent traffic camera with version 4.1 or above.

struVILost

Video loss alarm parameters, see details in the structure [NET_DVR_VILOST_V40](#).

struAULost

Audio loss alarm parameters, see details in the structure [NET_DVR_VILOST_V40](#).

struMotion

Motion detection alarm parameters, see details in the structure [NET_DVR_MOTION_V40](#).

struHideAlarm

Video tampering alarm parameters, see details in the structure [NET_DVR_HIDEALARM_V40](#).

struOsdColor

Custom OSD font color, see details in the structure [NET_DVR_RGB_COLOR](#).

dwBoundary

Aligned boundary value, unit: pixel, the default value is 0.

byRes

Reserved, set to 0.

A.182 NET_ITS_PICTURE_INFO

Captured picture information structure.

Structure Definition

```
struct{
    DWORD      dwDataLen;
    BYTE       byType;
    BYTE       byDataType;
    BYTE       byCloseUpType;
    BYTE       byPicRecogMode;
    DWORD      dwRedLightTime;
    BYTE       byAbsTime[32];
    NET_VCA_RECT struPlateRect;
    NET_VCA_RECT struPlateRecgRect;
    BYTE       *pBuffer;
    DWORD      dwUTCTime;
    BYTE       byCompatibleAbility;
    BYTE       byTimeDiffFlag;
    signed char cTimeDifferenceH;
    signed char cTimeDifferenceM;
    BYTE       byRes2[4];
}NET_ITS_PICTURE_INFO, *LPNET_ITS_PICTURE_INFO;
```

Members

dwDataLen

Size of media data.

byType

Data type: 0-license plate picture, 1-scene picture, 2-composite picture, 3-close-up picture, 4-binary picture, 5-stream, 6-driver's face thumbnail, 7-co-driver's face thumbnail, 8-non-motor vehicle, 9-pedestrian, 10-raw data, 11-target picture, 12-driver room picture, 13-co-driver room picture, 14-face thumbnail

byDataType

Data uploading type: 0-directly upload, 1-upload URL of cloud storage server

byCloseUpType

Close-up picture type: 0-reserved, 1-non-motor vehicle, 2-pedestrian

byPicRecogMode

Recognition mode: 0-recognize in forward direction, 1-recognize in backward direction

dwRedLightTime

Red light duration, unit: s

byAbsTime

Absolute time: yyyyymmddhhmmssxxx, e.g.20090810235959999, the last three bits are milliseconds.

struPlateRect

When **byType** is "1", this parameter indicates the position of license plate on the scene picture, when **byType** is "8" or "9", this parameter indicates the person coordinates. See details in the structure [NET_VCA_RECT](#).

struPlateRecgRect

ANPR region coordinates, when **byType** is "12" or "13", this parameter indicates the coordinates of driver. See details in the structure [NET_VCA_RECT](#).

pBuffer

Buffer for saving data.

dwUTCTime

UTC time

byCompatibleAbility

Compatible with capability filed, and it is represented by bit, value: 0-invalid, 1-valid. bit0-whether the parameter **dwUTCTime** is valid.

byTimeDiffFlag

Whether the time difference parameter is valid: 0-invalid, 1-valid.

cTimeDifferenceH

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 **byTimeDiffFlag** is "1".

cTimeDifferenceM

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 **byTimeDiffFlag** is "1".

byRes2

Reserved

Remarks

- If only the scene picture is uploaded, you can capture the close-up view from scene picture according to the parameter **struPlateRecgRect**, and you can also adjust the width and height as required.
- Picture URL format: http://CVMIP:Port/pic? did=DevID&bid=BlkID&pid=PictureID&pftime=PicTime. E.g., http://10.192.65.140:8009/pic? did=35b9cbd0-8ffa-1031-87e6-0025903c6a50&bid=387&pid=2952790009&pftime=1378106185

CVMIP

IP address of CVM (Cloud Video Management) server.

Port

Port number (default: 8009) of HTTP service provided by CVM (Cloud Video Management) server.

DevID

Device ID of CVS (Cloud Video Server).

BlkID

Device block ID of CVS (Cloud Video Server)

PictureID

Picture ID generated by CVS (Cloud Video Server)

PicTime

Picture timestamp.

A.183 NET_DVR_PIC_FEATURE_UNION

Picture features union.

Structure Definition

```
union{
    BYTE      byLen[256];
    NET_DVR_PLATE_INFO   struPlateInfo;
    NET_DVR_FACE_SUB_PICTURE_INFO struFaceSubInfo;
    NET_DVR_BEHAVIOR_INFO   struBehavior;
}NET_DVR_PIC_FEATURE_UNION, *LPNET_DVR_PIC_FEATURE_UNION;
```

Members

byLen

Union size, which is 256 bytes.

struPlateInfo

Vehicle detection picture information, it is valid only when the picture type (**wPicType**) in the structure is "0". For details, refer to the structure NET_DVR_PLATE_INFO in the HCNetSDK user manual of traffic capture and analysis applications.

struFaceSubInfo

Face picture information, it is valid only when the picture type (**wPicType**) in the structure is "1". For details, refer to the structure .

struBehavior

Abnormal event detection picture information, it is valid only when the picture type (**wPicType**) in the structure is from "3" to "12". For details, refer to the structure .

See Also

A.184 NET_DVR_PIC_PARAM

Picture parameters structure

Structure Definition

```
struct{
    char      *pDVRFileName;
    char      *pSavedFileBuf;
    DWORD     dwBufLen;
    DWORD     *lpdwRetLen;
    NET_DVR_ADDRESS struAddr;
    BYTE      byRes[256];
}NET_DVR_PIC_PARAM, *LPNET_DVR_PIC_PARAM;
```

Members

pDVRFileName

Picture name

pSavedFileBuf

The buffer to save the picture data

dwBufLen

The buffer size

IpdwRetLen

Actually received data length pointer, it cannot be NULL

struAddr

Picture address information, it will be returned when searching for pictures

byRes

Reserved

A.185 NET_DVR_PIRIS_PARAM

IR iris parameter structure

Structure Definition

```
struct{
    BYTE  byMode;
    BYTE  byPIrisAperture;
    BYTE  byRes[6];
}NET_DVR_PIRIS_PARAM, *LPNET_DVR_PIRIS_PARAM;
```

Members

byMode

IR iris mode: 0-auto, 1-manual.

byPIrisAperture

IR iris level (higher level corresponds to larger iris), value range: [1,100], the default value is 50; this member can be configured when in manual mode (**byMode** is set to 1).

byRes

Reserved, set to 0.

A.186 NET_DVR_PLAYCOND

Structure about Playback and Downloading Parameters

Member	Data Type	Description
dwChannel	DWORD	Channel No.
struStartTime	<u>NET_DVR_TIME</u>	Start time of playback or downloading.
struStopTime	<u>NET_DVR_TIME</u>	End time of playback or downloading.
byDrawFrame	BYTE	Whether to enable frame extract: 0-no, 1-yes

Member	Data Type	Description
byStreamType	BYTE	Stream type: 0-main stream, 1-sub-stream, 2-third stream
byStreamID	Array of BYTE	Stream ID, it is valid when dwChannel is set to 0xffffffff. The maximum size is 32 bytes (value of macro definition "STREAM_ID_LEN").
byCourseFile	BYTE	Whether to play or download course file: 0-no, 1-yes
byDownload	BYTE	Whether to download: 0-no, 1-yes
byOptimalStreamType	BYTE	Whether to play back stream with optimal type: 0-no, 1-yes. For dual-stream devices, if the stream type is not specified to the recorded file, the returned video will use the actual stream type.
byVODFileType	BYTE	Stream type of downloaded video file: 0-PS stream type, 1-3GP type. You can choose stream type of the saved video file if you download it by time. If the video is saved in 3GP type, you can play the video via the browser.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 29 bytes.

A.187 NET_DVR_POE_CFG

PoE Parameter Structure

Member	Data Type	Description
strlP	NET_DVR_IPADDR_UNION	IP address.
byRes	Array of char	Reserved, set to 0. The maximum length is 128 characters.

A.188 NET_DVR_POS_AVE

Structure about AVE Protocol Parameters

Member	Data Type	Description
dwPosAddr	DWORD	POS address No.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 948 bytes.

A.189 NET_DVR_POS_CONNECTMODE_UNION

Structure about Connection Mode Parameters of DVR and POS

Member	Data Type	Description
byLen	Array of BYTE	Union size, the maximum size is 312 bytes.
struNetRecv	<u>NET_DVR_NET_RECV_E</u>	Network receiving mode.
struTcpMonitor	<u>NET_DVR_MONITOR_INFO</u>	Network listening mode.
struRS232	<u>NET_DVR_SINGLE_RS232</u>	Serial port mode.
struUdpMonitor	<u>NET_DVR_MONITOR_INFO</u>	Network listening in UDP mode.
struSniff	<u>NET_DVR_NET_SNIFF</u>	Network sniffer mode.
struMcast	<u>NET_DVR_ADDRESS</u>	Multicast mode.
struUSBRS232	<u>NET_DVR_USB_RS232</u>	Mode of converting USB to RS-232 serial port.

A.190 NET_DVR_POS_FILTER_CFG

Structure about Filter Rule Parameters of POS Information

Member	Data Type	Description
dwSize	DWORD	Structure size.
byEnable	BYTE	Whether to enable filter rules: 0-no, 1-yes.

Member	Data Type	Description
byCharSetType	BYTE	Encoding type, see details in <u>NET_DVR_CHARSET_ENUM</u> .
byRes1	BYTE	One-byte reserved member, set it to 0.
byProtocolType	BYTE	Protocol type: 1-generic text protocol (Universal Protocol), 2-AVE VSI-ADD, 3-AVE Vnet, 4-EPSON, 5-NUCLEUS.
struPosProtocol	<u>NET_DVR_POS_PROTO_COL_UNION</u>	POS protocol parameters.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 32 bytes.

A.191 NET_DVR_POS_GENERIC

Structure about generic text protocol information

Structure Definition

```
struct{
    BYTE      byCaseSensitive;
    BYTE      byRes1[7];
    NET_DVR_GENERIC_START struTransactionStart;
    NET_DVR_GENERIC_END   struTransactionEnd;
    NET_DVR_GENERIC_DATA_CFG struLineDeli;
    NET_DVR_IGNORE_STRING  struIgnoreString[MAX_IGNORE_STRING_NUM];
    BYTE      byRes[40];
}NET_DVR_POS_GENERIC, *LPNET_DVR_POS_GENERIC;
```

Members

byCaseSensitive

Whether it is case-sensitive: 0-no, 1-yes

byRes1

Reserved.

struTransactionStart

Start transaction

struTransactionEnd

End transaction

struLineDeli

Line separator.

strIgnoreString

Number of ignored strings, the maximum is 4 currently.

byRes

Reserved.

A.192 NET_DVR_POS_HIDDEN_INFORMATION

Structure about Hidden Parameters of POS Information

Member	Data Type	Description
szKeyWord	Array of char	Keyword, up to three keywords are supported (value of macro definition "KEY_WORD_NUM"), and the maximum size of each keyword is 128 bytes (value of macro definition "KEY_WORD_LEN").
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 128 bytes.

Remarks

* will be used to replace and hide the POS information.

A.193 NET_DVR_POS_NUCLEUS

Structure about NUCLEUS protocol parameters

Structure Definition

```
struct{
    char szEmployeeNo[SERIAL_NUM_LEN/*8*/];
    char szTerminalNo[SERIAL_NUM_LEN/*8*/];
    char szShiftNo[SERIAL_NUM_LEN/*8*/];
    BYTE byRes[928];
}NET_DVR_POS_NUCLEUS,*LPNET_DVR_POS_NUCLEUS;
```

Members

szEmployeeNo

Employee No.

szTerminalNo

POS No.

szShiftNo

Shift No., corresponding to different shift period.

byRes

Reserved.

A.194 NET_DVR_POS OSD_REGION

Structure about POS information overlaying area.

Structure Definition

```
struct{
    NET_VCA_POINT  struStart;
    NET_VCA_POINT  struEnd;
}NET_DVR_POS OSD_REGION,*LPNET_DVR_POS OSD_REGION;
```

Members

struStart

Coordinates of start point.

struEnd

Coordinates of end point.

A.195 NET_DVR_POS_PROTOCOL_UNION

Union about POS protocol parameters

Structure Definition

```
union{
    BYTE      byLenth[952];
    NET_DVR_POS_GENERIC  struGeneric;
    NET_DVR_POS_AVE     struAve;
    NET_DVR_POS_NUCLEUS struNUCLEUS;
}NET_DVR_POS_PROTOCOL_UNION,*LPNET_DVR_POS_PROTOCOL_UNION;
```

Members

byLenth

Union size, 952 bytes.

struGeneric

Generic text protocol parameters, it is valid only when byProtocolType is set to 1.

struAve

AVE protocol parameters, it is valid only when byProtocolType is set to 2 or 3.

struNUCLEUS

NUCLEUS protocol parameters, it is valid only when byProtocolType is set to 5.

A.196 NET_DVR_POSINFO_OVERLAY

Text overlay parameter structure

Structure Definition

```
struct{
    DWORD dwSize;
    BYTE byEnable;
    BYTE byBackpackEnanble;
    BYTE bySexEnanble;
    BYTE byCarryEnanble;
    BYTE byRideEnanble;
    BYTE byMaskEnanble;
    BYTE byHatEnanble;
    BYTE bySleeveEnanble;
    BYTE byPantsTypeEnanble;
    BYTE byHairEnanble;
    BYTE byGlassesEnanble;
    BYTE byAgeEnanble;
    BYTE byHeightEnanble;
    BYTE byRes[511];
}NET_DVR_POSINFO_OVERLAY,*LPNET_DVR_POSINFO_OVERLAY;
```

Members

dwSize

Structure size.

byEnable

whether to enable text overlay: 0-no, 1-yes.

byBackpackEnanble

whether to enable backpack information overlay: 0-no, 1-yes.

bySexEnanble

whether to enable gender information overlay: 0-no, 1-yes.

byCarryEnanble

whether to enable carrying information overlay: 0-no, 1-yes.

byRideEnanble

whether to enable riding information overlay: 0-no, 1-yes.

byMaskEnanble

whether to enable wearing mask information overlay: 0-no, 1-yes.

byHatEnanble

whether to enable wearing hat information overlay: 0-no, 1-yes.

bySleeveEnanble

whether to enable tops information overlay: 0-no, 1-yes.

byPantsTypeEnanble

whether to enable overlay bottoms information overlay: 0-no, 1-yes.

byHairEnanble

whether to enable hair information overlay: 0-no, 1-yes.

byGlassesEnanble

Whether to enable wearing glasses information overlay: 0-no, 1-yes.

byAgeEnanble

whether to enable age information overlay: 0-no, 1-yes.

byHeightEnanble

whether to enable stature information overlay: 0-no, 1-yes.

byRes

Reserved, set to 0.

A.197 NET_DVR_PPPOECFG

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.198 NET_DVR_PRESETCHAN_INFO

Preset Information Structure

Member	Data Type	Description
dwEnablePresetChan	DWORD	Channel that called preset.
dwPresetPointNo	DWORD	Called preset No., 0xffffffff-not call preset.

A.199 NET_DVR_PRESET_NAME

Preset name and predefined PTZ information structure.

Structure Definition

```
struct{
    DWORD      dwSize;
    WORD       wPresetNum;
    BYTE       byRes1[2];
    BYTE       byName[NAME_LEN/*32%/];
    WORD       wPanPos;
    WORD       wTiltPos;
    WORD       wZoomPos;
    BYTE       byRes2;
    BYTE       byPTZPosExEnable;
    NET_PTZ_INFO_EX struPtzPosEx;
    BYTE       byRes[58];
}NET_DVR_PRESET_NAME, *LPNET_DVR_PRESET_NAME;
```

Members

dwSize

Structure size.

wPresetNum

Preset No.

byRes1

Reserved, set to 0.

byName

Preset name.

wPanPos

Panning parameter (setting is not available).

wTiltPos

Tilting parameter (setting is not available).

wZoomPos

Zooming parameter (setting is not available).

byRes2

Reserved, set to 0.

byPTZPosExEnable

Whether to enable PTZ parameters extension, 0-no, 1-yes. The extended PTZ coordinates are subject to the values of **struPtzPosEx**.

struPtzPosEx

Extended PTZ parameters, which supports high accuracy, see details in the structure .

byRes

Reserved, set to 0.

A.200 NET_DVR_PREVIEWINFO**Structure about Live View Parameters**

Member	Data Type	Description
lChannel	LONG	Channel No., the analog channel No. starts from 1, the start No. of digital channel is obtained by calling API NET_DVR_GetDVRConfig with the command of NET_DVR_GET_IPPARACFG_V40 (command No.1062), and the parameter is returned by the structure NET_DVR_IPPARACFG_V40 .
dwStreamType	DWORD	Stream type: 0-main stream, 1-sub-stream, 2-third stream, 3-virtual stream, and so on.
dwLinkMode	DWORD	Stream transmission mode: 0-TCP, 1-UDP, 2-multicast, 3-RTP, 4-RTP/RTSP, 5-RSTP/HTTP, 6-HRUDP, 7-RTSP/HTTPS, 8-NPQ (Network Protocol Quality).

Member	Data Type	Description
hPlayWnd	HWND	Handle of display window, "NULL"-not decode and display.
bBlocked	DWORD	0-non-blocking streaming (check the success once requesting for connection, if receiving stream failed or playing failed, the live view exception will be uploaded for notification, which can reduce the pausing duration during auto-switch), 1-blocking streaming (check the success until playing ended, if network exception, the connection failed message is returned after 5 second timeout, so it is not application for auto-switch).
bPassbackRecord	DWORD	Whether to enable ANR function: 0-no, 1-yes. The ANR function helps to synchronize the camera's data automatically after the network is restored, this function should be supported by device.
byPreviewMode	BYTE	Live view mode: :0-nomal mode, 1-delay mode
byStreamID	Array[BYTE]	Stream ID,which consist of letters, digits, and underlines. It is valid when IChannel equals to "0xffffffff". The maximum length is 32 bytes (macro definition: STREAM_ID_LEN).
byProtoType	BYTE	Application layer streaming protocol: 0-private protocol (default), 1-RTSP, 2-RTSP stream encryption. The supported protocol of main

Member	Data Type	Description
		<p>stream and sub-stream is returned by the structure <i>NET_DVR_DEVICEINFO_V30</i> (related parameters: byMainProto and bySubProto). This parameter is valid only when the device supports both the private protocol and RTSP. For RTSP stream encryption, if dwLinkMode is 0, the transmission mode is TCP, and the signaling and stream will be encrypted by TLS; if dwLinkMode is 1, the transmission mode is UDP, then the signaling will be encrypted by TLS and the stream will be encrypted by SRTP; if dwLinkMode is 2, the transmission mode is multicast, then the signaling will be encrypted by TLS and the stream will be encrypted by SRTP; other transmission mode is not supported.</p>
byRes1	BYTE	Reserved.
byVideoCodingType	BYTE	Encoding and decoding type of stream data: 0-general, 1-raw data from thermal detector (encryption information of temperature data)
dwDisplayBufNum	DWORD	Maximum frames can be buffered in the PlayCtrl library buffer, parameter value: 1, 6 (default, self-adaptive), 15. If this parameter is set to 0, the value will automatically change to 1.

Member	Data Type	Description
byNPQMode	BYTE	NPQ mode: 0-directly connected, 1-via stream media
byRes	Array[BYTE]	Reserved, set to 0. The maximum length is 215 bytes.

Remarks

- The parameters **dwStreamType** (stream type), **dwLinkMode** (connection type), **bPassbackRecord** (ANR), **byPreviewMode** (delay live view mode), **byStreamID** (stream ID), should be supported by device.
- The API **NET_DVR_RealPlay_V40** supports live view in multicast mode (set **dwLinkMode** to 2) without setting multicast group address. As the lower layer will automatically get the configured multicast group address (**struMulticastIpAddr** in the structure **NET_DVR_NETCFG_V50**) from the device .
- NPQ (Network Protocol Quality) is a service library to improve the transmission quality of video and audio data by network technology. The transmission is based on UDP, and the transmitted video and audio data is in RTP container format, so the network bandwidth usage can be estimated according to the packet loss rate and RTT of RTCP. And then the stream sender will change the stream encoding strategy and bit rate to meet the current network situation.
- When the value of **dwLinkMode** is 0, and the value of **byVideoCodingType** is 1, it indicates getting raw stream data. The supported types of raw data includes thermal raw data, temperature data, and real-time raw data. The default type is thermal raw data. The temperature data is uploaded in the format of **STREAM_FRAME_INFO_S**.

A.201 NET_DVR_PREVIEWINFO_SPECIAL

Live view parameter structure.

Structure Definition

```
typedef struct tagNET_DVR_PREVIEWINFO_SPECIAL{
    char    sURL[1024];
    DWORD   dwLinkMode;
    HWND    hPlayWnd;
    DWORD   bBlocked;
    DWORD   dwDisplayBufNum;
    BYTE    byRes[64];
}NET_DVR_PREVIEWINFO_SPECIAL, *LPNET_DVR_PREVIEWINFO_SPECIAL;
```

Members

sURL

Streaming URL returned by device, e.g., rtsp://admin:abcd12345@10.7.37.2:554/Streaming/Channels/101?transportmode=unicast

dwLinkMode

Connection mode: 0-TCP, 1-UDP, 2-, 3-RTP, 4-RTP/RTSP, 5-RTP/HTTP

hPlayWnd

Handle of display window, "NULL"-not decode and display.

Blocked

0-non-blocking streaming (check the success once requesting for connection, if receiving stream failed or playing failed, the live view exception will be uploaded for notification, which can reduce the pausing duration during auto-switch), 1-blocking streaming (check the success until playing ended, if network exception, the connection failed message is returned after 5 second timeout, so it is not application for auto-switch).

dwDisplayBufNum

Maximum frames can be buffered in the PlayCtrl library buffer, range: from 1 to 50 frames. If this parameter is set to 0, the value will automatically change to 1.

byRes

Reserved, set to 0.

See Also

[NET_DVR_RealPlaySpecial](#)

A.202 NET_DVR_PROTO_TYPE

Structure about protocol parameters.

Structure Definition

```
struct{
    DWORD dwType;
    BYTE byDescribe[DESC_LEN/*16*/];
}NET_DVR_PROTO_TYPE,*LPNET_DVR_PROTO_TYPE;
```

Members

dwType

Protocol value

byDescribe

Protocol description

Remarks

- The descriptions (**byDescribe**) of different protocol types (**dwType**) are shown in the table below.

```
enum tagNET_DVR_IPC_ENUM_UNIFY{
    ENUM_IPC_PROTOCOL_INVALID = -1,
    ENUM_IPC_PROTOCOL_HIKVISION = 0,      //HIKVISION
    ENUM_IPC_PROTOCOL_PANASONIC = 1,      //PANASONIC
    ENUM_IPC_PROTOCOL_SONY = 2,           //SONY
    ENUM_IPC_PROTOCOL_AXIS = 4,          //AXIS
    ENUM_IPC_PROTOCOL_SANYO = 5,          //SANYO
    ENUM_IPC_PROTOCOL_BOSCH = 6,          //BOSCH
    ENUM_IPC_PROTOCOL_ZAVIO = 7,          //ZAVIO
    ENUM_IPC_PROTOCOL_GRANDEYE,          //GRANDEYE
    ENUM_IPC_PROTOCOL_PROVIDEO = 16,     //PROVIDEO
    ENUM_IPC_PROTOCOL_ARECONT = 10,      //ARECONT
    ENUM_IPC_PROTOCOL_ACTI = 11,         //ACTI
    ENUM_IPC_PROTOCOL_PELCO = 12,        //PELCO
    ENUM_IPC_PROTOCOL_VIVOTEK = 13,      //VIVOTEK
    ENUM_IPC_PROTOCOL_DAHUA = 3,         //DAHUA
    ENUM_IPC_PROTOCOL_SAMSUNG = 15,      //SAMSUNG
    ENUM_IPC_PROTOCOL_PSIA = 17,         //PSIA
    ENUM_IPC_PROTOCOL_ONVIF = 18,        //ONVIF
    ENUM_IPC_PROTOCOL_BRICKCOM = 19,     //BRICKCOM
    ENUM_IPC_PROTOCOL_CANON = 23,        //CANON
    ENUM_IPC_PROTOCOL_HUINT = 32,        //HUINT
    ENUM_IPC_PROTOCOL_INFINOVA = 14,     //INFINOVA
    ENUM_IPC_PROTOCOL_HIK_STD_H264,      //HIKVISION STANDARD H.264
    ENUM_IPC_PROTOCOL_HIK_STD_MPEG4,     //HIKVISION STANDARD MPEG4
    ENUM_IPC_PROTOCOL_SUNELL,           //SUNELL
    ENUM_IPC_PROTOCOL_ATEME,            //ATEME
    ENUM_IPC_PROTOCOL_LAUNCH,          //LAUNCH
    ENUM_IPC_PROTOCOL_YAAN,             //YAAN
    ENUM_IPC_PROTOCOL_BLUESKY,          //BLUESKY
    ENUM_IPC_PROTOCOL_BLUESKYLIMIT,    //BLUESKYLIMIT
    ENUM_IPC_PROTOCOL_TDWY,             //TIANDY
    ENUM_IPC_PROTOCOL_HBGK,             //HBGK
    ENUM_IPC_PROTOCOL_SANTACHI,         //SANTACHI
    ENUM_IPC_PROTOCOL_HIGHEASY,         //HIGHEASY
    ENUM_IPC_PROTOCOL_HANBANG,          //HANBANG
    ENUM_IPC_PROTOCOL_SAMSUNG_3120,     //SAMSUNG 3120
    ENUM_IPC_PROTOCOL_SAMSUNG_3080,     //SAMSUNG 3080
    ENUM_IPC_PROTOCOL_SAMSUNG_2000,     //SAMSUNG 2000
    ENUM_IPC_PROTOCOL_SAMSUNG_5200,     //SAMSUNG 5200
    ENUM_IPC_PROTOCOL_JINGYUAN,          //JINGYUAN
    ENUM_IPC_PROTOCOL_VIDEOTREC,        //VIDEOTREC
    ENUM_IPC_PROTOCOL_CHENOVA,          //CHENOVA
    ENUM_IPC_PROTOCOL_FENGHUO,          //FENGHUO
    ENUM_IPC_PROTOCOL_ZB_5301,           //ZB_5301
    ENUM_IPC_PROTOCOL_ZB_5401,           //ZB_5401
    ENUM_IPC_PROTOCOL_HAIXIN,            //HISENSE
    ENUM_IPC_PROTOCOL_ZHONGYINGXIN,     //ZHONGYINGXIN
}
```

```

ENUM_IPC_PROTOCOL_AVUN,           //AVUN
ENUM_IPC_PROTOCOL_GOVTY,          //GOVTY
ENUM_IPC_PROTOCOL_SAE,            //SAE
ENUM_IPC_PROTOCOL_DONGFANGWANGLI, //NETPOSA
ENUM_IPC_PROTOCOL_CHANGHONG,      //CHANGHONG
ENUM_IPC_PROTOCOL_H3C,            //H3C
ENUM_IPC_PROTOCOL_BAIAN,          //BAIAN
ENUM_IPC_PROTOCOL_HAT,            //HAT
ENUM_IPC_PROTOCOL_YUANYE,          //YUANYE
ENUM_IPC_PROTOCOL_HIKCARD,         //HIKVISION BOARD CARD
ENUM_IPC_PROTOCOL_HAIXINCAP,       //HISENSE CAPTURE CAMERA
ENUM_IPC_PROTOCOL_WENANCAP,        //WENAN CAPTURE CAMERA
ENUM_IPC_PROTOCOL_XUNMEI,          //XUNMEI
ENUM_IPC_PROTOCOL_BAIWO,           //BAIWO
ENUM_IPC_PROTOCOL_APD,             //APD
ENUM_IPC_PROTOCOL_REACHDEV,        //REACHDEV
ENUM_IPC_PROTOCOL_XUNMEI_DAHUA,    //XUNMEI_DAHUA OEM
ENUM_IPC_PROTOCOL_HUANGHE,         //HUANGHE
ENUM_IPC_PROTOCOL_LIANCHEN,        //LIANCHEN
ENUM_IPC_PROTOCOL_CHENGYE,          //CHENGYE
ENUM_IPC_PROTOCOL_VISIONDIGI,       //VISIONDIGI
ENUM_IPC_PROTOCOL_HENGHE,           //HENGHE
ENUM_IPC_PROTOCOL_KODAK,            //KODAK
ENUM_IPC_PROTOCOL_AIRONIX,          //AIRONIX
ENUM_IPC_PROTOCOL_LG,               //LG
ENUM_IPC_PROTOCOL_HASEE,             //HASEE
ENUM_IPC_PROTOCOL_8000ME,            //8000ME
ENUM_IPC_PROTOCOL_POVITEL,          //POVITEL
ENUM_IPC_PROTOCOL_YIVIEW,            //YIVIEW
ENUM_IPC_PROTOCOL_TIANYUE,           //TIANYUE
ENUM_IPC_PROTOCOL_HOWELL,            //HOWELL
ENUM_IPC_PROTOCOL_WAPA,              //WAPA
ENUM_IPC_PROTOCOL_SANLE,             //SANLE
ENUM_IPC_PROTOCOL_HIKCARD_ENCRYPTION, //ENCRYPTED HIKVISION BOARD CARD
ENUM_IPC_PROTOCOL_JUNSDA,             //JUNSDA
ENUM_IPC_PROTOCOL_LIYUAN,              //LIYUAN
ENUM_IPC_PROTOCOL_XINCHAN,            //XINCHAN
ENUM_IPC_PROTOCOL_BITE,               //BITE
ENUM_IPC_PROTOCOL_MEIAN,              //MEIAN
ENUM_IPC_PROTOCOL_ROSEEK,             //ROSEEK
ENUM_IPC_PROTOCOL_AEBELL,              //AEBELL
ENUM_IPC_PROTOCOL_JSL_ST,              //JSL ST
ENUM_IPC_PROTOCOL_VIMICRO,             //VIMICRO
ENUM_IPC_PROTOCOL_TYPE,                //MAX MANUFACTURER TYPE
}NET_DVR_IPC_ENUM_UNIFY

enum _NET_DVR_IPC_ENUM_{
    ENUM_BUSINESS_INVALID = -1,
    ENUM_BUSINESS_HIKVISION = 0,
    ENUM_BUSINESS_PANASONIC,
    ENUM_BUSINESS_SONY,
    ENUM_BUSINESS_AXIS,

```

```

ENUM_BUSINESS_SANYO,
ENUM_BUSINESS_BOSCH,
ENUM_BUSINESS_ZAVIO,
ENUM_BUSINESS_GRANDEYE,
ENUM_BUSINESS_PROVIDEO,
ENUM_BUSINESS_ARECONT,    //9
ENUM_BUSINESS_ACTI,
ENUM_BUSINESS_PELCO,
ENUM_BUSINESS_VIVOTEK,
ENUM_BUSINESS_INFINOVA,
ENUM_BUSINESS_DAHUA,      //14
ENUM_BUSINESS_HIK_STD_H264 = 0x20,
ENUM_BUSINESS_HIK_STD_MPEG4,
ENUM_BUSINESS_SUNELL,      //SUNELL
ENUM_BUSINESS_ATEME,
ENUM_BUSINESS_LAUNCH,      //LAUNCH
ENUM_BUSINESS_YAAN,        //YAAN
ENUM_BUSINESS_BLUESKY,     //BLUESKY
ENUM_BUSINESS_BLUESKYLIMIT, //BLUESKYLIMIT
ENUM_BUSINESS_TDWFY,       //TIANDY
ENUM_BUSINESS_HBGK,        //HBGK
ENUM_BUSINESS_SANTACHI,    //SANTACHI
ENUM_BUSINESS_HIGHEASY,    //HIGHEASY
ENUM_BUSINESS_SAMSUNG,
ENUM_BUSINESS_URL_RTSP = 0x40,//Streaming via URL
ENUM_BUSINESS_ONVIF,
ENUM_MAX_BUSINESS_TYPE,    //Maximum manufacturer type
}NET_DVR_IPC_ENUM

```

- If the device supports unified network camera protocol, the supported protocol types are enumerated in NET_DVR_IPC_ENUM_UNIFY; otherwise, the supported protocol types are enumerated in NET_DVR_IPC_ENUM.

A.203 NET_DVR_RTSPCFG

RTSP protocol parameter structure.

Structure Definition

```

struct{
    DWORD dwSize;
    WORD wPort;
    WORD wRtspPort;
    BYTE byReserve[52];
}NET_DVR_RTSPCFG, *LPNET_DVR_RTSPCFG;

```

Members

dwSize

Structure size.

wPort

RTSP server listening port.

wRtspPort

RTSPS sever listening port.

byReserve

Reserved, set to 0.

A.204 NET_DVR_RTSP_PARAMS_CFG

RTSP Parameter Structure

Member	Data Type	Description
dwMaxBuffRoomNum	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.
byUseSort	BYTE	Whether to enable RTP over UDP sorting: 0-no, 1-yes.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 123 bytes.

A.205 NET_DVR_PTZTRACKCHAN_INFO

Pattern Information Structure

Member	Data Type	Description
dwEnablePtzTrackChan	DWORD	Channel that called the pattern.
dwPtzTrackNo	DWORD	Called pattern No., 0xffffffff-invalid.

A.206 NET_DVR_PTZPOS

PTZ position parameter structure.

Structure Definition

```
struct{
    WORD  wAction;
    WORD  wPanPos;
    WORD  wTiltPos;
    WORD  wZoomPos;
}NET_DVR_PTZPOS, *LPNET_DVR_PTZPOS;
```

Members

wAction

Operation type, it is valid only when setting parameters. 1-locate PTZ parameter, 2-locate Pan parameter, 3-locate Tilt parameter, 4-locate Zoom parameter, 5-locate Pan and Tilt parameter.

wPanPos

Pan parameter (horizontal)

wTiltPos

Tilt parameter (vertical)

wZoomPos

Zoom parameter (zoom in or out)

Remarks

The actual PTZ position value is the one-tenth of the received hexadecimal value. If the obtained Pan value is 0x1750, the actual Pan value is 175 degree; if the obtained Tilt value is 0x0789, the actual Tilt value is 78.9 degree; if the obtained Zoom value is 0x1100, the actual Zoom value is 110.

A.207 NET_DVR_PTZPOS_INFO

PTZ position parameter structure.

Structure Definition

```
struct{
    DWORD  dwPanPos;
    DWORD  dwTiltPos;
    DWORD  dwZoomPos;
}NET_DVR_PTZPOS_INFO, *LPNET_DVR_PTZPOS_INFO;
```

Members

dwPanPos

Panning parameter (horizontal)

dwTiltPos

Tilting parameter (vertical)

dwZoomPos

Zooming parameter

See Also

A.208 NET_DVR_PTZ_POSITION

Structure about scene configuration.

Structure Definition

```
struct{
    BYTE      byEnable;
    BYTE      byRes1[3];
    BYTE      byPtzPositionName[NAME_LEN/*32*/];

    NET_DVR_PTZPOS
    struPtzPos;

    BYTE      byRes2[40];
}NET_DVR_PTZ_POSITION, *LPNET_DVR_PTZ_POSITION;
```

Members

byEnable

Whether to enable scene: 0-no, 1-yes

byRes1

Reserved, set to 0

byPtzPositionName

Scene name

struPtzPos

PTZ coordinates

byRes2

Reserved, set to 0

Remarks

The **byEnable** is invalid when setting behavior analysis rule, but it is valid when setting local scene parameters.

A.209 NET_DVR_PU_STREAM_CFG_V41

Dynamic decoding parameter structure.

Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byStreamMode;
    BYTE       byStreamEncrypt;
    BYTE       byRes1[2];
    NET_DVR_DEC_STREAM_MODE uDecStreamMode;
    DWORD      dwDecDelayTime;
    BYTE       sStreamPassword[STREAM_PASSWD_LEN/*12*/];
    BYTE       byRes2[48];
}NET_DVR_PU_STREAM_CFG_V41,*LPNET_DVR_PU_STREAM_CFG_V41;
```

Members

dwSize

Structure size

byStreamMode

Streaming mode: 0-invalid, 1-get stream by IP address or domain name, 2-get stream by URL, 3-get stream from device by DDNS.

byStreamEncrypt

Whether to encrypt the stream: 0-no, 1-yes.

byRes1

Reserved, set to 0

uDecStreamMode

Streaming configuration parameters, see the structure [NET_DVR_DEC_STREAM_MODE](#) for details.

dwDecDelayTime

Decoding delay time duration, unit: millisecond.

sStreamPassword

Stream encryption password. The sensitive information should be encrypted.

byRes2

Reserved, set to 0.

A.210 NET_DVR_PU_STREAM_URL

Configuration parameter structure about getting stream by URL.

Structure Definition

```
struct{
    BYTE  byEnable;
    BYTE  strURL[240];
    BYTE  byTransPortocol;
    WORD  wIPID;
    BYTE  byChannel;
    BYTE  byRes[7];
}NET_DVR_PU_STREAM_URL,*LPNET_DVR_PU_STREAM_URL;
```

Members

byEnable

Enable/disable getting stream by URL: 0-disable, 1-enable.

strURL

Stream URL

byTransPortocol

Transfer protocol type: 0-TCP, 1-UDP

wIPID

Device ID= iDevInfoIndex + iGroupNO*64 +1

byChannel

Device channel No.

byRes

Reserved, set to 0.

Remarks

The stream URL format is {rtsp://ip[:port]/urlExtension}[?username=username][?password=password][?linkmode=linkmode]. You can also customize the URL format if the network camera supports custom URL.

A.211 NET_DVR_REL_CAPTURE_CHAN_V40

Capture channel information.

Structure Definition

```
struct{
    DWORD dwMaxRelCaptureChanNum;
    DWORD dwChanNo[MAX_CHANNUM_V40/*512*/];
    BYTE byRes[32];
}NET_DVR_REL_CAPTURE_CHAN_V40, *LPNET_DVR_REL_CAPTURE_CHAN_V40;
```

Members

dwMaxRelCaptureChanNum

Maximum number of channels that can be triggered by alarm input, which is read-only.

dwChanNo

No. of the channel that is triggered to capture, which is represented by value and permuted compactly. 0xffffffff indicates that the subsequent value is invalid.

byRes

Reserved, set to 0.

A.212 NET_DVR_RGB_COLOR

Structure about color parameters.

Structure Definition

```
struct{
    BYTE byRed;
    BYTE byGreen;
    BYTE byBlue;
    BYTE byRes;
}NET_DVR_RGB_COLOR, *LPNET_DVR_RGB_COLOR;
```

Members

byRed

Red component in RGB color model.

byGreen

Green component in RGB color model.

byBlue

Blue component in RGB color model.

byRes

Reserved

Remarks

The color component value is based on RGB888 standard.

A.213 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).
byRes1	BYTE	Reserved, set to 0. The size is 3 bytes.
struStreamInfo	<u>NET_DVR_STREAM_IN_FO</u>	Stream information structure. This member is valid when getting the task information by stream ID.
byRes	Array of BYTE	Reserved, set to 0. The size is 128 bytes.

A.214 NET_DVR_RECORD_PASSBACK_MANUAL_TASK_RET**Structure About Results of Getting Task of Manually Copying Back Videos**

Member	Data Type	Description
dwSize	DWORD	Structure size.
struStreamInfo	<u>NET_DVR_STREAM_IN_FO</u>	Stream information structure. This member is valid when getting the task information by stream ID.
dwTaskID	DWORD	Task ID
struStartTime	<u>NET_DVR_TIME_EX</u>	Start time of video copy-back
struStopTime	<u>NET_DVR_TIME_EX</u>	End time of video copy back
byTaskStatus	BYTE	Task status: 0 (not executed), 1 (pausing), 2 (executed), 3 (copying back), 4 (copy-back)

Member	Data Type	Description
		failed), 5 (succeeded, but only some videos are copied back), 6 (succeeded, but there is no video in the camera).
byRes1	Array of BYTE	Reserved, set to 0. The size is 3 bytes.
struExecuteStartTime	<u>NET_DVR_TIME_EX</u>	Actual start time of executing the task. This member is valid when the value of byTaskStatus is 1 or 2.
struExecuteStopTime	<u>NET_DVR_TIME_EX</u>	Actual end time of executing the task. This member is valid when the value of byTaskStatus is 1 or 2.
byRes	Array of BYTE	Reserved, set to 0. The size is 128 bytes.

A.215 NET_DVR_SCENE_INFO

Alarm scene information structure.

Structure Definition

```
struct {
    DWORD      dwSceneID;
    BYTE       bySceneName[NAME_LEN];
    BYTE       byDirection;
    BYTE       byRes1[3];
    NET_DVR_PTZPOS struPtzPos;
    BYTE       byRes2[64];
}NET_DVR_SCENE_INFO,*LPNET_DVR_SCENE_INFO;
```

Members

dwSceneID

Scene ID, 0 means that this scene is invalid.

bySceneName

Scene name.

byDirection

Detection direction: 1-upward, 2-downward, 3-bidirectional, 4-westward, 5-northward, 6-eastward, 7-southward, 8-other.

byRes1

Reserved.

struPtzPos

PTZ coordinates, see details in the structure .

byRes2

Reserved.

A.216 NET_DVR_SCENECHANGE_DETECTION_RESULT

Structure about the information of the scene change alarm.

Structure Definition

```
struct{
    DWORD      dwSize;
    NET_VCA_DEV_INFO struDevInfo;
    WORD       wDevInfolvmsChannelEx;
    BYTE       byRes[126];
}NET_DVR_SCENECHANGE_DETECTION_RESULT, *LPNET_DVR_SCENECHANGE_DETECTION_RESULT;
```

Members

dwSize

Structure size.

struDevInfo

Front-end device information, see details in the structure [NET_VCA_DEV_INFO](#) .

wDevInfolvmsChannelEx

Device channel No. for the HCNetSDK to access the device.

byRes

Reserved, set to 0.

A.217 NET_DVR_SCHEDTIME

Structure About Start and End Time Parameters

Member	Data Type	Description
byStartHour	BYTE	Start time: hour.
byStartMin	BYTE	Start time: minute.
byStopHour	BYTE	End time: hour.
byStopMin	BYTE	End time: minute.

A.218 NET_DVR_SCHEDULE_DAYTIME

Structure about auto-switch time interval parameters.

Structure Definition

```
struct{
    NET_DVR_DAYTIME
    struStartTime;
    NET_DVR_DAYTIME
    struStopTime;
}NET_DVR_SCHEDULE_DAYTIME, *LPNET_DVR_SCHEDULE_DAYTIME;
```

Members

struStartTime

Start time

struStopTime

Stop time

A.219 NET_DVR_SEARCH_EVENT_PARAM_V50

Condition structure about searching videos by event.

Structure Definition

```
struct{
    WORD          wMajorType;
    WORD          wMinorType;
    NET_DVR_TIME_SEARCH_COND  struStartTime;
    NET_DVR_TIME_SEARCH_COND  struEndTime;
    BYTE          byLockType;
    BYTE          byRes[255];
    union{
        BYTE      byLen[SEARCH_EVENT_INFO_LEN_V40/*800*/];
        struct{
            WORD      wAlarmInNo[128];
            BYTE      byRes[544];
        }struAlarmParam;
        struct{
            WORD      wMotDetChanNo[MAX_CHANNUM_V30/*64*/];
            BYTE      byRes[672];
        }struMotionParam;
        struct{
            WORD      wChanNo[MAX_CHANNUM_V30/*64*/];
        }struChanParam;
    };
};
```

```

    BYTE      byRuleID;
    BYTE      byRes[671];
}struVcaParam;
struct{
    BYTE      byRoomIndex;
    BYTE      byRes[799];
}struInquestParam;
struct{
    BYTE      byAll;
    BYTE      byRes1[3];
    WORD      wChanNo[MAX_CHANNUM_V30/*64*/];
    BYTE      byRes1[668];
}struVCADetect;
struct{
    NET_DVR_STREAM_INFO struIDInfo;
    DWORD      dwCmdType;
    BYTE       byBackupVolumeNum;
    BYTE       byRes[723];
}struStreamIDParam;
struct{
    WORD      wChannel[MAX_CHANNUM_V30/*64*/];
    BYTE      byAllChan;
    BYTE      byCaseSensitive;
    BYTE      byCombinatMode;
    BYTE      byRes1;
    char      sKeyWord[MAX_POS_KEYWORDS_NUM/*3*/][MAX_POS_KEYWORD_LEN/*128*/];
    BYTE      byRes[284];
}struPosAlarm;
struct{
    BYTE      byCaseNo[SEARCH_CASE_NO_LEN/*56*/];
    BYTE      byCaseName[SEARCH_CASE_NAME_LEN/*100*/];
    BYTE      byLitigant1[SEARCH_LITIGANT_LEN/*32*/];
    BYTE      byLitigant2[SEARCH_LITIGANT_LEN/*32*/];
    BYTE      byChiefJudge[SEARCH_CHIEF_JUDGE_LEN/*32*/];
    BYTE      byCaseType;
    BYTE      byRes[547];
}struTrialParam;
struct{
    DWORD      dwMajor;
    DWORD      dwMinor;
    BYTE       byCardNo[ACS_CARD_NO_LEN/*32*/];
    BYTE       byName[NAME_LEN/*32*/];
    BYTE       byMACAddr[MACADDR_LEN/*6*/];
    BYTE      byRes[722];
}struACSAAlarm;
struct{
    WORD      wDeviceType;
    WORD      wEventType;
    WORD      wChannel[MAX_CHANNUM_V30/*64*/];
    BYTE      byAllChan;
    BYTE      byCaseSensitive;
    BYTE      byCombinatMode;
}

```

```

BYTE      bySearchType;
char      sKeyWord[MAX_POS_KEYWORDS_NUM/*3*/][MAX_POS_KEYWORD_LEN/*128*/];
WORD      wZoneNo;
BYTE      byRes[278];
}struIOTAlarm;
}uSeniorParam;
}NET_DVR_SEARCH_EVENT_PARAM_V50,*LPNET_DVR_SEARCH_EVENT_PARAM_V50;

```

Members

wMajorType

Major alarm/event types, see details below:

```

enum _MAIN_EVENT_TYPE_{
    EVENT_MOT_DET      = 0,
    EVENT_ALARM_IN     = 1,
    EVENT_VCA_BEHAVIOR = 2,
    EVENT_INQUEST      = 3,
    EVENT_VCA_DETECTION = 4,
    EVENT_POS          = 5,
    EVENT_TRIAL_CASE   = 6,
    EVENT_ACS_CASE     = 7,
    EVENT_IOT_CASE     = 8,
    EVENT_STREAM_INFO  = 100,
}MAIN_EVENT_TYPE

```

EVENT_MOT_DET

Motion detection

EVENT_ALARM_IN

Alarm input

EVENT_VCA_BEHAVIOR

Behavior analysis

EVENT_INQUEST

Inquest event (not support)

EVENT_VCA_DETECTION

VCA detection

EVENT_POS

POS information

EVENT_TRIAL_CASE

Trial case information (not support)

EVENT_ACS_CASE

Access control event

EVENT_IOT_CASE

IoT device event

EVENT_STREAM_INFO

Stream ID information

wMinorType

Minor alarm/event types, which vary with the major types. 0xffff-all types. For motion detection, alarm input, and POS recording, there is no minor type, so this parameter is set to "0xffff"; for access control event, refer to the integration manual of access control applications; for the major types of behavior analysis, VCA detection, and stream ID information, the corresponding minor types are shown below:

Table A-15 EVENT_VCA_BEHAVIOR

Minor Type	Value	Description
EVENT_TRAVERSE_PLANE	0	Line crossing
EVENT_ENTER_AREA	1	Entering the area,support regional rule
EVENT_EXIT_AREA	2	Leaving the area, support regional rule
EVENT_INTRUSION	3	Perimeter intrusion, support regional rule
EVENT_LOITER	4	Loitering, support regional rule
EVENT_LEFT_TAKE	5	Droping/picking up, support regional rule
EVENT_PARKING	6	Parking, support regional rule
EVENT_RUN	7	Running, support regional rule
EVENT_HIGH_DENSITY	8	People density in the area, support regional rule
EVENT_STICK_UP	9	Sticking a note, support regional rule
EVENT_INSTALL_SCANNER	10	Installing card reader, support regional rule
EVENT_OPERATE_OVER_TIME	11	Operation timeout
EVENT_FACE_DETECT	12	Abnormal face detection
EVENT_LEFT	13	Unattended baggage

Minor Type	Value	Description
EVENT_TAKE	14	Object removal
EVENT_LEAVE_POSITION	15	Absence event
EVENT_TRAIL_INFO	16	Tailing
EVENT_HUMAN_ENTER	18	Human entrance
EVENT_FALL_DOWN_INFO	19	Falling down
EVENT_OBJECT_PASTE	20	Sticking script area
EVENT_FACE_CAPTURE_INFO	21	Normal face
EVENT_MULTI_FACES_INFO	22	Multiple faces
EVENT_AUDIO_ABNORMAL_INFO	23	Sudden change of sound intensity
EVENT_DETECT	24	VCA detection
EVENT_SUNGASSES_FACE_INFO	25	Face with sunglasses
EVENT_CALLING_FACE_INFO	26	Person is making call
EVENT_VIOLENT_MOTION	27	Violent motion
EVENT_SAFETY_HELMET	28	Hart hat detection

Table A-16 EVENT_VCA_DETECTION

Minor Type	Value	Description
EVENT_VCA_TRAVERSE_PLANE	1	Line crossing
EVENT_FIELD_DETECTION	2	Intrusion detection
EVENT_AUDIO_INPUT_ALARM	3	Audio loss detection
EVENT_SOUND_INTENSITY_ALARM	4	Sudden increase of sound increase/decrease detection
EVENT_FACE_DETECTION	5	Face detection
EVENT_VIRTUAL_FOCUS_ALARM	6	Defocus detection
EVENT_SCENE_CHANGE_ALARM	7	Scene change detection

Minor Type	Value	Description
EVENT_PIR_ALARM	8	PIR alarm
EVENT_ENTER_REGION	9	Region entrance
EVENT_EXIT_REGION	10	Region exiting
EVENT_LOITERING	11	Loitering
EVENT_GROUPDETECTION	12	People gathering in the area, support regional rule
EVENT_RAPIDMOVE	13	Fast moving
EVENT_PARK	14	Parking
EVENT_UNATTENDED_BAGGAGE	15	Unattended baggage
EVENT_ATTENDED_BAGGAGE	16	Object removal
EVENT_VEHICLE_DETECTION	17	Vehicle detection

Table A-17 EVENT_STREAM_INFO

Minor Type	Value	Description
EVENT_STREAM_ID	0	Stream ID
EVENT_TIMING	1	Timing record
EVENT_MOTION_DETECT	2	Motion detection
EVENT_ALARM	3	Alarm record
EVENT_ALARM_OR_MOTION_DETECT	4	Alarm or motion detection
EVENT_ALARM_AND_MOTION_DETECT	5	Alarm and motion detection
EVENT_COMMAND_TRIGGER	6	Command triggering
EVENT_MANUAL	7	Manual record
EVENT_BACKUP_VOLUME	8	Storage volume record
STREAM_EVENT_SEMAPHORE	9	Sensor alarm
STREAM_EVENT_HIDE	10	Video tempering
STREAM_EVENT_INVERSE	11	Driving in opposite direction

Minor Type	Value	Description
STREAM_EVENT_VIDEO_LOST	. 12	Video loss
STREAM_EVENT_WIRELESS_ALARM	13	Wireless alarm
STREAM_EVENT_PIR_ALARM	14	PIR alarm
STREAM_EVENT_CALLHELP_ALARM	15	Call for help alarm
STREAM_EVENT_FACESNAP	16	Face capture
STREAM_EVENT_FACE_DETECTION	17	Face detection
STREAM_EVENT_ITS_PLATE	18	Vehicle detection
STREAM_EVENT_PDC	19	People counting
STREAM_EVENT_SCENECHANGE_DETECTION	20	Scene change
STREAM_EVENT_DEFOCUS_DETECTION	21	Defocus detection
STREAM_EVENT_AUDIOEXCEPTION	22	Audio exception
VCA_EVENT_TRAVERSE_PLANE	23	Line crossing
VCA_EVENT_INTRUSION	24	Intrusion
VCA_EVENT_ENTER_AREA	25	Region entrance
VCA_EVENT_EXIT_AREA	26	Region exiting
VCA_EVENT_LOITER	27	Loitering detection
VCA_EVENT_HIGH_DENSITY	28	People gathering
VCA_EVENT_RUN	29	Fast moving
VCA_EVENT_PARKING	30	Illegal parking
VCA_EVENT_LEFT	31	Unattended baggage
VCA_EVENT_TAKE	32	Object removal

struStartTime

Start time of search, refer to the structure for details.

struEndTime

Stop time of search, refer to the structure for details.

byLockType

Whether to lock: 0xff-all, 0-no, 1-yes.

byRes

Reserved, set to 0.

uSeniorParam

Condition structure of searching for video files based on event/alarm, see details below:

byLen

Union size, which is 800 bytes.

struAlarmParam

Condition structure of searching for video files based on alarm input alarm, see details below:

wAlarmInNo

Alarm input No., "0xffff"-the following are invalid. For example, wAlarmInNo[0]==1&&wAlarmInNo[1]==2 indicates the alarm input event of alarm input No.1 and alarm input No.2.

byRes

Reserved, set to 0.

struMotionParam

Condition structure of searching for video files based on motion detection, see details below:

wMotDetChanNo

The channel No., "0xffff"-the following are invalid. For example,wMotDetChanNo[0]==1&&wMotDetChanNo[1]==2 indicates the motion detection event of channel No.1 and channel No.2.

byRes

Reserved, set to 0.

struVcaParam

Condition structure of searching for video files based on behavior analysis, see details below:

wChanNo

The event triggered channel No., "0xffff"-the following are invalid. For example, wChanNo[0]==1&&wChanNo[1]==2 indicates the behavior analysis event of channel No.1 and channel No.2.

byRuleID

Rule ID, 0xff-all

byRes

Reserved, set to 0.

struInquestParam

Condition structure of searching for video files based on inquest event, see details below:

byRoomIndex

The inquest room index No., starts from 1.

byRes

Reserved, set to 0.

struVCADetect VCA detection

Condition structure of searching for video files based on VCA detection, see details below:

byAll

0-Search by channel No., 1-Search in all channels.

byRes1

Reserved, set to 0.

wChanNo

VCA detection channel No., "0xffff"-the following are invalid. For example, wChanNo[0]==1&&wChanNo[1]==2 indicates the VCA detection of channel No.1 and channel No.2.

byRes

Reserved, set to 0.

struStreamIDParam

Condition structure of searching for video files with stream ID information, see details below:

struIDInfo

Stream ID information, and it is 72 bytes, refer to for details.

dwCmdType

The external triggering type, which is used for NVR accessing to the cloud storage.

byBackupVolumeNum

The storage volume No., which is available for CVR.

byRes

Reserved, set to 0.

struTrialParam

Condition structure of searching for video files based on trial information, see details below:

byCaseNo

Case No.

byCaseName

Case name.

byLitigant1

Litigant No.1.

byLitigant2

Litigant No.2

byChiefJudge

Chief judge

byCaseType

Case type: 0-all, 1criminal case, 2-civil case.

byRes

Reserved, set to 0

struPosAlarm

Condition structure of searching for video files with POS information, see details below:

wChannel

Channel No., "0xffff"-the following are invalid. For example,

wChannel[0]==1&&wChannel[1]==2 indicates the video with POS information of channel No.1 and channel No.2.

byAllChan

Whether to search in all channels: 0-no (**wChannel** is valid), 1-all channels (**wChannel** is invalid).

byCaseSensitive

Whether to enable case sensitive: 0-no, 1-yes.

byCombindateMode

The keyword combination mode: 0-or, 1-and.

byRes1

Reserved, set to 0.

sKeyWord

Search by keyword.

byRes

Reserved, set to 0.

struACSAalarm

Condition structure of searching for video files based on access control event, see details below:

dwMajor

Alarm/event major type, 0-all.

dwMinor

Alarm/event minor type, 0-all.

byCardNo

Card No.

byName

Name

byMACAddr

Physical MAC address.

byRes

Reserved, set to 0.

struIOTAlarm

Condition structure of searching for video files based on IoT device event, see details below:

wDeviceType

Device type: 0-access controller, 1-video intercom, 2-security control panel, 3-GJD security control panel, 4-Luminite security control panel, 5-OPTEX security control panel, 6-Detector

wEventType

Event searching sub type, it varies according to main type, 0xffff-all

wChannel

Channel No.

byAllChan

Search all channel: 0-no, the **wChannel** is valid, 1-all channel, the **wChannel** is invalid

byCaseSensitive

Case sensitivity or not: 1-no, 1-yes

byCombinantMode

Keyword combination mode: 0-or, 1-and

bySearchType

Search method: 1-according to video source, now the channel is video channel.

sKeyWord

Keyword

wZoneNo

Zone No., it is valid only when the **wDeviceType** values "2-security control panel" and the sub type **wEventType** values "1"

byRes

Reserved, set to 0.

Related API

[NET_DVR_FindFileByEvent_V50](#)

A.220 NET_DVR_SEARCH_EVENT_RET_V50

Information structure of video files searched by event.

Structure Definition

```
struct{
    WORD      wMajorType;
    WORD      wMinorType;
    NET_DVR_TIME_SEARCH struStartTime;
    NET_DVR_TIME_SEARCH struEndTime;
    NET_DVR_ADDRESS struAddr;
    WORD      wChan[MAX_CHANNUM_V40/*512*/];
    BYTE     byRes[256];
union{
    BYTE     byLen[800];
    struct{
        DWORD     dwAlarmInNo;
        BYTE      byRes[796];
    }struAlarmRet;
    struct{
        DWORD     dwMotDetNo;
        BYTE      byRes[796];
    }struMotionRet;
    struct{
        DWORD     dwChanNo;
        BYTE      byRuleID;
        BYTE      byRes1[3];
        BYTE      byRuleName[NAME_LEN/*32*/];
        NET_VCA_EVENT_UNION uEvent;
        BYTE      byRes[668];
    }struVcaRet;
    struct{
        BYTE      byRoomIndex;
        BYTE      byDriveIndex;
        BYTE      byRes1[6];
        DWORD     dwSegmentNo;
        WORD      wSegmetSize;
        WORD      wSegmentState;
        BYTE      byRes2[784];
    }
}
```

```

}struInquestRet;
struct{
    DWORD      dwRecordType;
    DWORD      dwRecordLength;
    BYTE       byLockFlag;
    BYTE       byDrawFrameType;
    BYTE       byRes1[2];
    BYTE       byFileName[NAME_LEN/*32*/];
    DWORD      dwFileIndex;
    BYTE       byRes[752];
}struStreamIDRet;
struct{
    DWORD      dwChanNo;
    BYTE       byRes[796];
}struPosRet;
struct{
    BYTE       byRoomIndex;
    BYTE       byDriveIndex;
    WORD       wSegmetSize;
    DWORD      dwSegmentNo;
    BYTE       bySegmentState;
    BYTE       byCaseType;
    BYTE       byRes[2];
    BYTE       byCaseNo[CASE_NO_RET_LEN/*52*/];
    BYTE       byCaseName[CASE_NAME_RET_LEN/*64*/];
    BYTE       byLitigant1[LITIGANT_RET_LEN/*24*/];
    BYTE       byLitigant2[LITIGANT_RET_LEN/*24*/];
    BYTE       byChiefJudge[CHIEF_JUDGE_RET_LEN/*24*/];
    BYTE       byRes1[600];
}struTrialRet;
}uSeniorRet;
}NET_DVR_SEARCH_EVENT_RET_V50,*LPNET_DVR_SEARCH_EVENT_RET_V50;

```

Members

wMajorType

Major types, see details below:

```

enum _MAIN_EVENT_TYPE_{
    EVENT_MOT_DET      = 0,
    EVENT_ALARM_IN     = 1,
    EVENT_VCA_BEHAVIOR = 2,
    EVENT_INQUEST      = 3,
    EVENT_VCA_DETECTION = 4,
    EVENT_TRIAL_CASK   = 6,
    EVENT_STREAM_INFO   = 100,
}MAIN_EVENT_TYPE

```

EVENT_MOT_DET

Motion detection

EVENT_ALARM_IN

Alarm input

EVENT_VCA_BEHAVIOR

Behavior analysis

EVENT_INQUEST

Inquest event (not support)

EVENT_VCA_DETECTION

VCA detection

EVENT_POS

POS information

EVENT_TRIAL_CASK

Trial case information (not support)

EVENT_ACS_CASK

Access control event

EVENT_STREAM_INFO

Stream ID information

wMinorType

Minor alarm/event types, which vary with the major types. 0xffff-all types. For motion detection, alarm input, and POS recording, there is no minor type, so this parameter is set to "0xffff"; for access control event, refer to the integration manual of access control applications; for the major types of behavior analysis, VCA detection, and stream ID information, the corresponding minor types are shown below:

Table A-18 EVENT_VCA_BEHAVIOR

Minor Type	Value	Description
EVENT_TRAVERSE_PLANE	0	Line crossing
EVENT_ENTER_AREA	1	Entering the area,support regional rule
EVENT_EXIT_AREA	2	Leaving the area, support regional rule
EVENT_INTRUSION	3	Perimeter intrusion, support regional rule
EVENT_LOITER	4	Loitering, support regional rule

Minor Type	Value	Description
EVENT_LEFT_TAKE	5	Dropping/picking up, support regional rule
EVENT_PARKING	6	Parking, support regional rule
EVENT_RUN	7	Running, support regional rule
EVENT_HIGH_DENSITY	8	People density in the area, support regional rule
EVENT_STICK_UP	9	Sticking a note, support regional rule
EVENT_INSTALL_SCANNER	10	Installing card reader, support regional rule
EVENT_OPERATE_OVER_TIME	11	Operation timeout
EVENT_FACE_DETECT	12	Abnormal face detection
EVENT_LEFT	13	Unattended baggage
EVENT_TAKE	14	Object removal
EVENT_LEAVE_POSITION	15	Absence event
EVENT_TRAIL_INFO	16	Tailing
EVENT_FALL_DOWN_INFO	19	Falling down
EVENT_OBJECT_PASTE	20	Sticking script area
EVENT_FACE_CAPTURE_INFO	21	Normal face
EVENT_MULTI_FACES_INFO	22	Multiple faces
EVENT_AUDIO_ANORMAL_INFO	23	Sudden change of sound intensity
EVENT_SUNGASSES_FACE_INFO	25	Face with sunglasses
EVENT_CALLING_FACE_INFO	26	Person is making call

Table A-19 EVENT_VCA_DETECTION

Minor Type	Value	Description
EVENT_VCA_TRAVERSE_PLANE	1	Line crossing detection
EVENT_FIELD_DETECTION	2	Intrusion detection
EVENT_AUDIO_INPUT_ALARM	3	Audio loss detection
EVENT_SOUND_INTENSITY_ALARM	4	Sudden increase of sound increase/decrease detection
EVENT_FACE_DETECTION	5	Face detection
EVENT_VIRTUAL_FOCUS_ALARM	6	Defocus detection
EVENT_SCENE_CHANGE_ALARM	7	Scene change detection
EVENT_PIR_ALARM	8	PIR alarm
EVENT_ENTER_REGION	9	Region entrance
EVENT_EXIT_REGION	10	Region exiting
EVENT_LOITERING	11	Loitering
EVENT_GROUPDETECTION	12	People gathering in the area, support regional rule
EVENT_RAPIDMOVE	13	Fast moving
EVENT_PARK	14	Parking
EVENT_UNATTENDED_BAGGAGE	15	Unattended baggage
EVENT_ATTENDED_BAGGAGE	16	Object removal
EVENT_VEHICLE_DETECTION	17	Vehicle detection

Table A-20 EVENT_STREAM_INFO

Minor Type	Value	Description
EVENT_STREAM_ID	0	Stream ID
EVENT_TIMING	1	Timing record
EVENT_MOTION_DETECT	2	Motion detection

Minor Type	Value	Description
EVENT_ALARM	3	Alarm record
EVENT_ALARM_OR_MOTION_DETECT	4	Alarm or motion detection
EVENT_ALARM_AND_MOTION_DETECT	5	Alarm and motion detection
EVENT_COMMAND_TRIGGER	6	Command triggering
EVENT_MANUAL	7	Manual record
EVENT_BACKUP_VOLUME	8	Storage volume record
STREAM_EVENT_SEMAPHORE	9	Sensor alarm
STREAM_EVENT_HIDE	10	Video tempering
STREAM_EVENT_INVERSE	11	Driving in opposite direction
STREAM_EVENT_VIDEO_LOST	12	Video loss
STREAM_EVENT_WIRELESS_ALARM	13	Wireless alarm
STREAM_EVENT_PIR_ALARM	14	PIR alarm
STREAM_EVENT_CALLHELP_ALARM	15	Call for help alarm
STREAM_EVENT_FACENAP	16	Face capture
STREAM_EVENT_FACE_DETECTION	17	Face detection
STREAM_EVENT_ITS_PLATE	18	Vehicle detection
STREAM_EVENT_PDC	19	People counting
STREAM_EVENT_SCENECHANGE_DETECTION	20	Scene change
STREAM_EVENT_DEFOCUS_DETECTION	21	Defocus detection
STREAM_EVENT_AUDIOEXCEPTION	22	Audio exception
VCA_EVENT_TRAVERSE_PLANE	23	Line crossing
VCA_EVENT_INTRUSION	24	Intrusion

Minor Type	Value	Description
VCA_EVENT_ENTER_AREA	25	Region entrance
VCA_EVENT_EXIT_AREA	26	Region exiting
VCA_EVENT_LOITER	27	Loitering detection
VCA_EVENT_HIGH_DENSITY	28	People gathering
VCA_EVENT_RUN	29	Fast moving
VCA_EVENT_PARKING	30	Illegal parking
VCA_EVENT_LEFT	31	Unattended baggage
VCA_EVENT_TAKE	32	Objet removal

struStartTime

Start time of search, refer to the structure for details.

struEndTime

Stop time of search, refer to the structure for details.

struAddr

Address information of video segment, which for cluster playback, refer to for details.

wChan

Alarm triggered or event occurred channel No. 0xffff-the followings are invalid.

byRes

Reserved, set to 0.

uSeniorRet

Result union of searching for video files based on event/alarm, see details below:

byLen

Union size, it is 800 bytes.

struAlarmRet

Result structure of searching for video files based on alarm input alarm, see details below:

dwAlarmInNo

Alarm input No..

byRes

Reserved.

struMotionRet

Result structure of searching for video files based on motion detection, see details below:

dwMotDetNo

Motion detection channel No.

byRes

Reserved, set to 0.

struVcaRet

Result structure of searching for video files based on behavior analysis alarm, see details below:

dwChanNo

Behavior analysis channel No.

byRuleID

Rule ID, 0xff-all rules

byRes1

Reserved, set to 0.

byRuleName

Rule name.

uEvent

Behavior analysis parameters, which depends on the parameter **wMinorType**, refer to the integration manual of behavior analysis applications for details.

byRes

Reserved, set to 0.

struInquestRet

Result structure of searching for video files based on inquest event, see details below:

byRoomIndex

Inquest room No., starts from 1.

byDriveIndex

Recorder No., starts from 1.

byRes1

Reserved, set to 0.

dwSegmentNo

Video segment No. of this inquest, starts from 1.

wSegmetSize

Video segment size, unit: MB

wSegmentState

Recording status: 0-normal, 1-exception, 2-unrecorded

byRes2

Reserved, set to 0.

struStreamIDRet

Result structure of searching for video files with stream ID information, see details below:

dwRecordType

Recording types: 0-scheduled recording, 1-based on motion detection, 2-based on alarm input alarm, 3-based on alarm input alarm or motion detection, 4-based on alarm input alarm and motion detection, 5-based on command, 6-manual recording, 7-based on vibration alarm, 8-based on environment alarm, 9-based on VCA alarm (including driving in the opposite direction, line crossing, unattended baggage, object removal and so on), 10-based on video tampering alarm, 13-based on event (motion detection, PIR, wireless panic alarm, and so on), 24-by video montage.

dwRecordLength

Video file size.

byLockFlag

Whether to lock: 0-no, 1-yes.

byDrawFrameType

Whether to extract the frame when recording: 0-no, 1-yes.

byRes1

Reserved, set to 0.

byFileName

File name.

dwFileIndex

File index No. in storage volume.

byRes

Reserved, set to 0.

struPosRet

Result structure of searching for video files with POS information, see details below:

dwChanNo

POS event channel No.

byRes

Reserved, set to 0.

struTrialRet

Result structure of searching for video files with trial information, see details below:

byRoomIndex

Inquest room No., starts from 1 .

byDriveIndex

Recorder No., starts from 1.

wSegmetSize

Video segment size, unit: MB.

dwSegmentNo

Video segment No. in this inquest, starts from 1.

bySegmentState

Recording status: 0-normal, 1-exception, 2-unrecorded.

byCaseType

Case type: 0-all, 1-criminal case, 2-civil case

byRes

Reserved, set to 0.

byCaseNo

Case No.

byCaseName

Case name.

byLitigant1

Litigant No.1.

byLitigant2

Litigant No.2.

byChiefJudge

Chief judge

byRes1

Reserved, set to 0.

Related API

[NET_DVR_FindNextEvent_V50](#)

A.221 NET_DVR_SECURITY_CFG_FILE_COND

Input parameter structure of importing/exporting configuration file

Structure Definition

```
struct{
    DWORD dwSize;
    char szSecretKey[NET_SDK_SECRETKEY_LEN/*128*/];
```

```
BYTE    byRes[128];
}NET_DVR_SECURITY_CFG_FILE_COND,*LPNET_DVR_SECURITY_CFG_FILE_COND;
```

Members

dwSize

Structure size

szSecretKey

Verification key of configuration file

byRes

Reserved, set to 0

A.222 NET_DVR_SEND_PARAM_IN

Structure about input parameters for sending data.

Structure Definition

```
struct{
    BYTE          *pSendData;
    DWORD         dwSendDataLen;
    NET_DVR_TIME_V30 struTime;
    BYTE          byPicType;
    BYTE          byPicURL;
    BYTE          byRes1[2];
    DWORD         dwPicMangeNo;
    BYTE          sPicName[NAME_LEN/*32*/];
    DWORD         dwPicDisplayTime;
    BYTE          *pSendAppendData;
    DWORD         dwSendAppendDataLen;
    BYTE          byRes[192];
}NET_DVR_SEND_PARAM_IN,*LPNET_DVR_SEND_PARAM_IN;
```

Members

ipSendData

Picture buffer

dwSendDataLen

Picture data size

struTime

Time displayed on the picture

byPicType

Picture format: 1-JPEG, 2-BMP, 3-PNG

byPicURL

Picture data mode: 0-binary, 1-URL

byRes1

Reserved, set to 0

dwPicMangeNo

Picture management No.

sPicName

Picture name

dwPicDisplayTime

Picture display time, unit: second

pSendAppendData

Additional information buffer of sent picture, set it to NULL if not supported

dwSendAppendDataLen

Additional information data length of sent picture, set it to 0 if not supported

byRes

Reserved

A.223 NET_DVR_SETUPALARM_PARAM

Arming parameter structure.

Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byLevel;
    BYTE       byAlarmInfoType;
    BYTE       byRetAlarmTypeV40;
    BYTE       byRetDevInfoVersion;
    BYTE       byRetVQDAlarmType;
    BYTE       byFaceAlarmDetection;
    BYTE       bySupport;
    BYTE       byBrokenNetHttp;
    WORD       wTaskNo;
    BYTE       byDeployType;
    BYTE       bySubSCRIPTION;
    BYTE       byRes1[2];
    BYTE       byAlarmTypeURL;
    BYTE       byCustomCtrl;
}NET_DVR_SETUPALARM_PARAM, *LPNET_DVR_SETUPALARM_PARAM;
```

Members

dwSize

Structure size

byLevel

Arming priority: 0-high, 1-medium, 2-low.

byAlarmInfoType

Intelligent traffic alarm information type: 0-old (NET_DVR_PLATE_RESULT),1-new (NET_ITS_PLATE_RESULT)

byRetAlarmTypeV40

0-motion detection, video loss, video tampering, and alarm input alarm information is uploaded in normal mode (alarm type: COMM_ALARM_V30, alarm information structure:

NET_DVR_ALARMINFO_V30), 1-alarm information is uploaded in variable size (alarm type: COMM_ALARM_V40, alarm information structure: **NET_DVR_ALARMINFO_V40**)

byRetDevInfoVersion

Alarm types of CVR: 0-COMM_ALARM_DEVICE (alarm information structure: NET_DVR_ALARMINFO_DEV), 1-COMM_ALARM_DEVICE_V40 (alarm information structure: NET_DVR_ALARMINFO_DEV_V40).

byRetVQDAlarmType

VQD alarm types: 0-COMM_ALARM_VQD (alarm information structure: NET_DVR_VQD_DIAGNOSE_INFO), 1-COMM_ALARM_VQD_EX (alarm information structure: NET_DVR_VQD_ALARM, including camera information and captured pictures)

byFaceAlarmDetection

Face detection alarm types: 1-face detection alarm (alarm type: COMM_ALARM_FACE_DETECTION, alarm information structure: NET_DVR_FACE_DETECTION), 0-face capture alarm (alarm type: COMM_UPLOAD_FACESNAP_RESULT, alarm information structure: NET_VCA_FACESNAP_RESULT).

bySupport

Capabilities, which is represented by bit:

- bit0-whether to upload picture: 0-yes, 1-no
- bit1-whether to enable ANR: 0-no, 1-yes
- 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.
- 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.

byBrokenNetHttp

ANR type, which is represented by bit and should be supported by device:

- bit0-whether to enable ANPR: 0-no, 1-yes.
- bit1-whether to enable people counting: 0-no, 1-yes.
- bit2-whether to enable heat map: 0-no, 1-yes.
- bit3-whether to enable face capture: 0-no, 1-yes.
- bit4-whether to enable face picture comparison: 0-no, 1-yes.

wTaskNo

Task No.

byDeployType

Arming type: 0-arm via client software, 1-real-time arming

bySubSubscription

Subscription parameters, which is represent by bit.

Bit7-whether to upload picture after subscribing motion detection alarm by person or vehicle:
0-no, 1-yes.

byRes1

Reserved, set to 0.

byAlarmTypeURL

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.

- bit0-type of captured face pictures: 0-binary data, 1-URL
- bit1-type of picture uploaded in message: 0-binary, 1-URL
- bit2-type of picture uploaded for face picture comparison: 0-binary, 1-URL

byCustomCtrl

Custom control type, which is represented by bit, bit0-whether to upload the face thumbnail of co-driver: 0-no, 1-yes

Remarks

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.

A.224 NET_DVR_SETUPALARM_PARAM_V50

Arming Parameter Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
byLevel	BYTE	Arming priority: 0-high, 1-medium, 2-low.
byAlarmInfoType	BYTE	Intelligent traffic alarm information type: 0-old (NET_DVR_PLATE_RESULT),1-new (NET_ITS_PLATE_RESULT).
byRetAlarmTypeV40	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: <u>NET_DVR_ALARMINFO_V30</u>); 1-alarm information is uploaded in variable size (alarm type: COMM_ALARM_V40, alarm information structure: <u>NET_DVR_ALARMINFO_V40</u>).
byRetDevInfoVersion	BYTE	Alarm types of CVR: 0-COMM_ALARM_DEVICE (alarm information structure: <u>NET_DVR_ALARMINFO_DEV</u>), 1-COMM_ALARM_DEVICE_V40 (alarm information structure: <u>NET_DVR_ALARMINFO_DEV_V40</u>).
byRetVQDAlarmType	BYTE	VQD alarm types: 0-COMM_ALARM_VQD (alarm information structure: NET_DVR_VQD_DIAGNOSE_INFO), 1-COMM_ALARM_VQD_EX (alarm information structure: NET_DVR_VQD_ALARM, including camera information and captured pictures)
byFaceAlarmDetection	BYTE	Face detection alarm types: 1-face detection alarm (alarm type: COMM_ALARM_FACE_DETECTION, alarm information structure: NET_DVR_FACE_DETECTION), 0-face capture alarm (alarm type: COMM_UPLOAD_FACESNAP_RESULT, alarm information structure: NET_VCA_FACESNAP_RESULT).
bySupport	BYTE	Capabilities, which is represented by bit: <ul style="list-style-type: none"> • bit0-whether to upload picture: 0-yes, 1-no • bit1-whether to enable ANR: 0-no, 1-yes

Member	Data Type	Description
		<ul style="list-style-type: none"> bit4-whether to upload abnormal event detection 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. 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.
byBrokenNetHttp	BYTE	<p>ANR type, which is represented by bit and should be supported by device:</p> <ul style="list-style-type: none"> bit0-whether to enable ANR for ANPR: 0-no, 1-yes. bit1-whether to enable ANR for people counting: 0-no, 1-yes. bit2-whether to enable ANR for heat map: 0-no, 1-yes. bit3-whether to enable ANR for face capture: 0-no, 1-yes. bit4-whether to enable ANR for face picture comparison: 0-no, 1-yes. bit5-whether to enable ANR for JSON message transmission: 0-no, 1-yes. bit6: whether to enable ANR for uploading heat map data by dwell time duration and by people quantity: 0-no, 1-yes. bit7: whether to enable ANR for uploading intersection analysis result: 0-no, 1-yes.
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.

Member	Data Type	Description
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"> bit0-type of captured face pictures: 0-binary data, 1-URL bit1-type of picture uploaded in message: 0-binary, 1-URL bit2-type of picture uploaded for face picture comparison: 0-binary, 1-URL
byCustomCtrl	BYTE	Custom control type, which is represented by bit, bit0-whether to upload the face thumbnail of the front passenger: 0-no, 1-yes
byRes4	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.225 NET_DVR_SHELTER

Structure about privacy mask region parameters.

Structure Definition

```
struct{
    WORD    wHideAreaTopLeftX;
    WORD    wHideAreaTopLeftY;
    WORD    wHideAreaWidth;
    WORD    wHideAreaHeight;
}NET_DVR_SHELTER, *LPNET_DVR_SHELTER;
```

Members

wHideAreaTopLeftX

X-coordinate of privacy mask region

wHideAreaTopLeftY

Y-coordinate of privacy mask region

wHideAreaWidth

Privacy mask area width

wHideAreaHeight

Privacy mask area height

Remarks

By default, the size of whole image is set to 704*576, so the coordinates, width, and height of privacy mask region should be converted to adapt to the size 704*576.

See Also

[NET_DVR_PICCFG_V40](#)

A.226 NET_DVR_SHOWSTRINGINFO

Sub structure about text overlay parameters.

Structure Definition

```
struct{
    WORD    wShowString;
    WORD    wStringSize;
    WORD    wShowStringTopLeftX;
    WORD    wShowStringTopLeftY;
    char    sString[44];
}NET_DVR_SHOWSTRINGINFO,*LPNET_DVR_SHOWSTRINGINFO;
```

Members

wShowString

Whether to enable OSD during live view: 0-no, 1-yes. The display region size is 704*576 , with single character size of 32*32

wStringSize

String size, up to 44 characters.

wShowStringTopLeftX

X-coordinate of text displayed position

wShowStringTopLeftY

Y-coordinate of text displayed position

sString

Text content.

See Also

[NET_DVR_SHOWSTRING_V30](#)

A.227 NET_DVR_SHOWSTRING_V30

Text overlay parameter structure.

Structure Definition

```
struct{
    DWORD      dwSize;
    NET\_DVR\_SHOWSTRINGINFO
    struStringInfo[MAX_STRINGNUM_V30/*8*/];
}NET_DVR_SHOWSTRING_V30,*LPNET_DVR_SHOWSTRING_V30;
```

Member

dwSize

Structure size

struStringInfo

Text to be displayed, see details in the structure [NET_DVR_SHOWSTRINGINFO](#) .

A.228 NET_DVR_SIMXML_LOGIN

Structure about Complement Fields by Stimulation Capability

Member	Data Type	Description
byLoginWithSimXml	BYTE	Whether to complement fields by stimulation capability: 0-no, 1-yes.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 127 bytes.

A.229 NET_DVR_SINGLE_HD_V50

Configuration parameter structure of a HDD.

Structure Definition

```
struct{
    DWORD dwHDNo;
    DWORD dwCapacity;
    DWORD dwFreeSpace;
    DWORD dwHdStatus;
    BYTE byHDArr;
    BYTE byHDTpe;
    BYTE byDiskDriver;
    BYTE byGenusGruop;
    DWORD dwHdGroup;
    BYTE byRecycling;
    BYTE bySupportFormatType;
    BYTE byFormatType;
    BYTE byRes2;
    DWORD dwStorageType;
    DWORD dwPictureCapacity;
    DWORD dwFreePictureSpace;
    BYTE byDiskLocation[NET_SDK_DISK_LOCATION_LEN/*16*/];
    BYTE bySupplierName[NET_SDK_SUPPLIER_NAME_LEN/*32*/];
    BYTE byDiskModel[NET_SDK_DISK_MODEL_LEN /*64*/];
    char szHDLocateIP[SDK_MAX_IP_LEN];
    BYTE byRes3[80];
}NET_DVR_SINGLE_HD_V50, *LPNET_DVR_SINGLE_HD_V50;
```

Members

dwHDNo

HDD No., which is between 0 and 32.

dwCapacity

HDD capacity, it is read-only.

dwFreeSpace

HDD free space, it is read-only.

dwHdStatus

HDD status (read-only): 0-normal, 1-unformatted, 2-error, 3-smart status, 4-mismatched, 5-sleepy, 6-offline (for network HDD), 7-Virtual HDD supports expansion, 10-HDD is repairing, 11-HDD is formatting, 12-HDD is waiting for formatting, 13-HDD is uninstalled, 14-the local HDD does not exist, 15-deleting (for network HDD), 16-locked, 17-warning, 18-bad HDD, 19-exception HDD, 20-unauthenticated, 21-unformatted for education sharing system.

byHDArr

HDD property: 0-normal, 1-redundant, 2-read-only, 3-archive, 4-cannot read and write.

byHDTypr

HDD type: 0-local HDD, 1-ESATA, 2-NFS, 3-iSCSI, 4-RAID, 5-SD card, 6-minSAS

byDiskDriver

HDD value, represented by ASCII characters.

byGenusGruop

Group: 0-reserved, 1-array, 2-storage pool, 3-HDD out of array, 4-uninitialized HDD, 5-invalid HDD, 6-local hot spare, 7-global hot spare, read-only.

dwHdGroup

HDD group, value range: [1, 16].

byRecycling

Whether to recycle: 0-no, 1-yes.

bySupportFormatType

Formatting type supported by HDD, read-only, it is valid only when **byHDTypr** is set to "5". This member is represented by bit, bit0: whether supports FAT32 formatting type, bit1: whether supports EXT4 formatting type. Value: 0-format HDD according to default type/not support, 1-support;

byFormatType

The selected formatting type, read-only. Value: 0-default formatting type, 1-FAT32 (valid when **byHDTypr** is set to "5"), 2-EXT4 (valid when **byHDTypr** is set to "5").

byRes2

Reserved.

dwStorageType

Storage type, which is represented by bit, and the bit value is 0 (not support) or others (support).

- dwStorageType & 0x1: HDD for saving normal video files.
- dwStorageType & 0x2: HDD for saving video files recorded by extracting frame.
- dwStorageType & 0x4: HDD for saving picture files.

dwPictureCapacity

HDD capacity for saving pictures (read-only), unit: MB

dwFreePictureSpace

Free space for saving picture (read-only), unit: MB

byDiskLocation

HDD location, read-only. The location description format is x/y-z, in which,

x: interface No. of host computer for connecting with extension module, i.e., 0-host computer, 1-extension module that connects to interface No.1, 2-extension module that connects to interface No.2;

y: extension module level No., i.e., 0-host computer, 1-level 1 extension module, 2-level 2 extension module;

z: slot No., e.g., if there are 48 slots on computer, the slot No. is between 1 and 48.

For example, 0/0-1 represents the location of slot No.1 on host computer; 1/1-24 represents the location of slot No.1 on level 1 extension module that connects to interface No.1 of host computer.

bySupplierName

HDD supplier name, read-only

byDiskModel

HDD model, read-only

szHDLocateIP

HDD IP address.

byRes3

Reserved, set to 0.

Remarks

As the members **dwCapacity**, **dwFreeSpace**, and **dwHdStatus** are the properties of local HDD, so they are read-only.

A.230 NET_DVR_SINGLE_RS232

RS232 Configuration Structure (extended for 9000) Sub Structure

Structure Definition

```
struct{
    DWORD    dwBaudRate;
    BYTE     byDataBit;
    BYTE     byStopBit;
    BYTE     byParity;
    BYTE     byFlowcontrol;
```

```
    DWORD dwWorkMode;
}NET_DVR_SINGLE_RS232, *LPNET_DVR_SINGLE_RS232;
```

Members

dwBaudRate

baud rate (bps): 0-50; 1-75; 2-110; 3-150; 4-300; 5-600; 6-1200; 7-2400; 8-4800; 9-9600;
10-19200; 11-38400; 12-57600; 13-76800; 14-115.2k

byDataBit

data bit: 0-5 digit; 1-6 digit; 2-7 digit; 3-8 digit

byStopBit

stop bit: 0-1 digit; 1-2 digit

byParity

parity: 0- disable; 1- odd parity; 2- even parity

byFlowcontrol

flow control: 0-disable; 1-software flow control; 2- hardware flow control

dwWorkMode

work mode: 0-narrowband transmission (232 is used for PPP dailing); 1- control center (232 works as paramter configuration); 2- transparent channel

A.231 NET_DVR_SIP_CFG_V50

SIP (Session Initiation Protocol) parameter structure.

Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byEnableAutoLogin;
    BYTE       byLoginStatus;
    BYTE       byRes1[2];
    NET_DVR_IPADDR stuServerIP;
    WORD       wServerPort;
    BYTE       byRes2[2];
    BYTE       byUserName[NAME_LEN/*32*/];
    BYTE       byPassWord[PASSWD_LEN/*16*/];
    BYTE       byLocalNo[MAX_NUMBER_LEN/*32*/];
    BYTE       byDispalyName[MAX_NAME_LEN/*128*/];
    WORD       wLocalPort;
    BYTE       byLoginCycle;
    BYTE       byRes3;
    BYTE       bySIPServerDomain[MAX_DOMAIN_NAME/*64*/];
    NET_DVR_IPADDR stuSTUNServerIP;
    BYTE       bySTUNServerDomain[MAX_DOMAIN_NAME/*64*/];
```

```
WORD      wSTUNServerPort;
BYTE     byRes4[2]
NET_DVR_IPADDR stuProxyServerIP;
BYTE     byProxyServerDomain[MAX_DOMAIN_NAME/*64*/];
WORD      wProxyServerPort;
BYTE     byNetWork;
BYTE     byRes5;
BYTE     byCalledTargetName[NET_SDK_MAX_CALLEDTARGET_NAME/*32*/];
BYTE     byRes[224];
}NET_DVR_SIP_CFG_V50, *LPNET_DVR_SIP_CFG_V50;
```

Members

dwSize

Structure size.

byEnableAutoLogin

Whether to enable registering automatically: 0-disable, 1-enable.

byLoginStatus

Login status: 0-unregistered,1-registered. This parameter can only be obtained.

byRes1

Reserved, set to 0.

stuServerIP

IP address of the SIP server.

wServerPort

Port No. of the SIP server.

byRes2

Reserved, set to 0.

byUserName

User name.

byPassWord

Password.

byLocalNo

Local No.

byDispalyName

Displayed name.

wLocalPort

Local port.

byLoginCycle

Registration period, value range: [1, 99], unit: minute.

byRes3

Reserved, set to 0.

bySIPServerDomain

SIP server domain name

Domain name of the SIP server. Only one member of **stuServerIP** and **bySIPServerDomain** should be configured; if both are configured, **stuServerIP** will be preferred.

stuSTUNServerIP

IP address of the STUN server.

bySTUNServerDomain

Domain name of the STUN server.

wSTUNServerPort

Port No. of the STUN server.

byRes4

Reserved, set to 0.

stuProxyServerIP

IP address of the proxy server.

byProxyServerDomain

Domain name of the proxy server. Only one member of **stuProxyServerIP** and **byProxyServerDomain** should be configured; if both are configured, the **stuProxyServerIP** will be preferred.

wProxyServerPort

Port No. of the proxy server.

byNetWork

Network type: 0-invalid, 1-wired network 1, 2-wired network 2, 3-wireless network. If this member is set to wired network, only the wired network will be used no matter whether the wireless network is normal or not; if this member is set to wireless network, only the wireless network will be used.

byRes5

Reserved, set to 0.

byCalledTargetName

User name of the called person.

byRes

Reserved, set to 0.

A.232 NET_DVR_SMARTIR_PARAM

SMART IR parameter structure

Structure Definition

```
struct{
    BYTE  byMode;
    BYTE  byIRDistance;
    BYTE  byShortIRDistance;
    BYTE  byLongIRDistance;
}NET_DVR_SMARTIR_PARAM, *LPNET_DVR_SMARTIR_PARAM;
```

Members

byMode

SMART IR mode: 0-auto, 1-manual.

byIRDistance

IR distance level (higher level corresponds to further IR distance), value range: [1,100], the default value is 50; this member can be configured in manual SMART IR mode (**byMode** is set to "1").

byShortIRDistance

Near light distance level, value range: [1,100].

byLongIRDistance

Far light distance level, value range: [1,100].

A.233 NET_DVR_SMARTSEARCH_PIC_UNION

Dual-VCA condition structure.

Structure Definition

```
union{
    BYTE          byLen[256];
    NET_DVR_VEHICLE_PARA   struVehiclePara;
    NET_VCA_HUMAN_FEATURE   struHumaFeature;
    NET_DVR_FACE_PIC_DATA_INFO struHumaPic;
    NET_DVR_BEHAVIOUR_COND   struBehaviourCond;
}NET_DVR_SMARTSEARCH_PIC_UNION,*LPNET_DVR_SMARTSEARCH_PIC_UNION;
```

Members

byLen

Union size, which is 256 bytes.

struVehiclePara

Conditions for vehicle analysis, it is valid when dual-VCA type (**wSearchType**) in the structure is "0". For details, refer to the structure .

struHumaFeature

Conditions for face picture analysis, it is valid when dual-VCA type (**wSearchType**) in the structure is "1". For details, refer to the structure NET_VCA_HUMAN_FEATURE in the HCNetSDK user manual of face applications.

struHumaPic

Conditions for searching by face picture, it is valid when dual-VCA type (**wSearchType**) in the structure is "2". For details, refer to the structure .

struBehaviourCond

Conditions for behavior analysis, it is valid when dual-VCA type (**wSearchType**) in the structure is "3". For details, refer to the structure .

A.234 NET_DVR_SMART_SEARCH_PARAM_V40

VCA search parameter structure.

Structure Definition

```
struct{
    DWORD          dwSize;
    NET_DVR_STREAM_INFO struIDInfo;
    BYTE           bySearchCondType;
    BYTE           bySensitivity;
    BYTE           byRes1[2];
    NET_DVR_TIME_EX   struStartTime;
    NET_DVR_TIME_EX   struEndTime;
    NET_DVR_AREA_SMARTSEARCH_COND_UNION uSmartSearchCond;
    BYTE           byISO8601;
    signed char      cStartTimeDifferenceH;
    signed char      cStartTimeDifferenceM;
    signed char      cStopTimeDifferenceH;
    signed char      cStopTimeDifferenceM;
    BYTE           byRes2[251];
}NET_DVR_SMART_SEARCH_PARAM_V40,*LPNET_DVR_SMART_SEARCH_PARAM_V40;
```

Members

dwSize

Structure size.

struIDInfo

Stream information, including stream ID or channel No., refer to the structure for details.

bySearchCondType

VCA search union index No.: 0-detection region, 1-line crossing detection, 2-intrusion detection, 3-face detection, 4-license plate detection.

bySensitivity

Dynamic analysis sensitivity of detection region, its value is between 1 and 100, and it is valid only when the member **bySearchCondType** is "0".

byRes1

Reserved, set to 0.

struStartTime

Start time of video, refer to the structure for details.

struEndTime

End time of video, refer to the structure for details.

uSmartSearchCond

VCA search condition union, refer to for details.

byISO8601

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.

cStartTimeDifferenceH

Time difference between start time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStartTimeDifferenceM

Time difference between start time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStopTimeDifferenceH

Time difference between stop time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStopTimeDifferenceM

Time difference between stop time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

byRes2

Reserved, set to 0.

Related API

[NET_DVR_SmartSearch_V40](#)

A.235 NET_DVR_SMART_SEARCH_PIC_PARA

Condition structure of picture dual-VCA.

Structure Definition

```
struct{
    WORD          dwChanNo;
    BYTE          byStreamID[STREAM_ID_LEN/*32*/];
    NET_DVR_TIME_EX struStartTime;
    NET_DVR_TIME_EX struEndTime;
    WORD          wSearchType;
    BYTE          byRes1[2];
    NET_DVR_SMARTSEARCH_PIC_UNION  uSmartSearchCond;
    BYTE          byISO8601;
    signed char   cStartTimeDifferenceH;
    signed char   cStartTimeDifferenceM;
    signed char   cStopTimeDifferenceH;
    signed char   cStopTimeDifferenceM;
    BYTE          byRes[59];
}NET_DVR_SMART_SEARCH_PIC_PARA,*LPNET_DVR_SMART_SEARCH_PIC_PARA;
```

Members

dwChanNo

Channel No.

byStreamID

Stream ID, it is prior to dwChanNo. Only when **byStreamID** is set to "null", **dwChanNo** will be used.

struStartTime

Start time of search, refer to the structure for details.

struEndTime

End time of search, refer to the structure for details.

wSearchType

Dual-VCA type: 0-vehicle feature analysis, 1-face features analysis, 2-picture analysis for searching by picture, 3-behavior analysis.

byRes1

Reserved.

uSmartSearchCond

Dual-VCA conditions, refer to the structure for details.

byISO8601

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.

cStartTimeDifferenceH

Time difference between start time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStartTimeDifferenceM

Time difference between start time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStopTimeDifferenceH

Time difference between stop time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStopTimeDifferenceM

Time difference between stop time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

byRes

Reserved.

Remarks

To support searching picture by picture, the picture modeling function should be enabled in device.

Related API

[NET_DVR_SmartSearchPicture](#)

A.236 NET_DVR_SMART_SEARCH_PIC_RET_V50

Picture dual-VCA results structure.

Structure Definition

```
struct{
    char          sFileName[PICTURE_NAME_LEN/*64*/];
    NET_DVR_TIME_EX      struTime;
    DWORD         dwFileSize;
    WORD          wPicType;
    BYTE          byRes1[2];
    NET_DVR_PIC_FEATURE_UNION  uPicFeature;
    BYTE          byISO8601;
    signed char    cTimeDifferenceH;
    signed char    cTimeDifferenceM;
    BYTE          byRes[29];
}NET_DVR_SMART_SEARCH_PIC_RET_V50,*LPNET_DVR_SMART_SEARCH_PIC_RET_V50;
```

Members

sFileName

Picture name.

struTime

Picture captured time, refer to the structure for details.

dwFileSize

Picture size.

wPicType

Picture type: 0-vehicle detection picture; 1-face picture; 2-picture for searching by picture; 3-line crossing detection picture; 4-intrusion detection picture; 5-unattended baggage detection picture; 6-object removal detection picture; 7-region entrance detection picture; 8-region exiting detection picture; 9-illegal parking detection picture; 10-loitering detection picture; 11-people gathering detection picture; 12-fast moving detection picture; 13-people running; 14-violent motion; 15-people falling down; 16-people number changed; 17-absence detection; 18-overstay detection; 19-situation analysis.

byRes1

Reserved.

uPicFeature

Picture features, refer to the structure for details.

byISO8601

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.

cTimeDifferenceH

Time difference between tag time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cTimeDifferenceM

Time difference between tag time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

byRes

Reserved, set to 0.

Related API

[NET_DVR_FindNextSmartPicture_V50](#)

A.237 NET_DVR_SMART_SEARCH_RET

VCA search results structure.

Structure Definition

```
struct{
    NET_DVR_TIME struStartTime;
    NET_DVR_TIME struEndTime;
    BYTE         byISO8601;
    signed char   cStartTimeDifferenceH;
    signed char   cStartTimeDifferenceM;
    signed char   cStopTimeDifferenceH;
    signed char   cStopTimeDifferenceM;
    BYTE         byRes[59];
}NET_DVR_SMART_SEARCH_RET,*LPNET_DVR_SMART_SEARCH_RET;
```

Members

struStartTime

Start time of video, refer to the structure for details.

struEndTime

End time of video, refer to the structure for details.

byISO8601

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.

cStartTimeDifferenceH

Time difference between start time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStartTimeDifferenceM

Time difference between start time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStopTimeDifferenceH

Time difference between stop time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStopTimeDifferenceM

Time difference between stop time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

byRes

Reserved, set to 0.

Related API[NET_DVR_SearchNextInfo](#)**A.238 NET_DVR_SMSRELATIVEPARAM_V50****Structure About SMS Configuration**

Member	Data Type	Description
dwSize	DWORD	Structure size
bEnableSmsAlarm	BYTE	Whether to enable SMS alarm: 0-disable, 1-enable
byRes1	Array of BYTE	Reserved.
struAllowList	Array of <u>NET_DVR_PHONECFG_V5</u> <u>0</u>	Contact allowlist
byRes2	Array of BYTE	Reserved.

A.239 NET_DVR_SNAP_CAMERAPARAMCFG

CCD parameter structure of capture camera

Structure Definition

```
struct{
    BYTE      byWDRMode;
    BYTE      byWDRTYPE;
    BYTE      byWDRLevel;
    BYTE      byRes1;
    NET_DVR_TIME_EX struStartTime;
    NET_DVR_TIME_EX struEndTime;
    BYTE      byDayNightBrightness;
    BYTE      byMCEEnabled;
    BYTE      byMCELevel;
    BYTE      byAutoContrastEnabled;
    BYTE      byAutoContrastLevel;
    BYTE      byLSEDEnabled;
    BYTE      byLSEDDetailLevel;
    BYTE      byLPDEEnabled;
    BYTE      byLPDELevel;
```

```
BYTE      byRes[35];
}NET_DVR_SNAP_CAMERAPARAMCFG, *LPNET_DVR_SNAP_CAMERAPARAMCFG;
```

Members

byWDRMode

WDR (Wide Dynamic Range) mode: 0-disabled, 1-DWDR, 2-WDR.

byWDRTypE

WDR enabling mode: 0-force enable, 1-enable by time, 2-enable by brightness.

byWDRLevel

WDR level, 0-level 1, 1-level 2, ..., 6-level 7; the default value is "2" (level 3).

byRes1

Reserved, set to 0.

struStartTime

Start time of WDR, refer to the structure [**NET_DVR_TIME_EX**](#) for details.

struEndTime

End time of WDR, refer to the structure [**NET_DVR_TIME_EX**](#) for details.

byDayNightBrightness

Brightness threshold of day/night auto switch, value range: [0,100], default value: 50.

byMCEEnabled

Whether to enable memory color enhancement: 0-no, 1-yes.

byMCELevel

Memory color enhancement level, value range: [0,100], default value: 50.

byAutoContrastEnabled

Whether to enable auto contrast: 0-no, 1-yes.

byAutoContrastLevel

Auto contrast level, value range: [0,100], default value: 50.

byLSEDetailEnabled

Whether to enable detail enhancement: 0-no, 1-yes.

byLSEDetailLevel

Detail enhancement level, value range: [0,100], default value: 50.

byLPDEEnabled

Whether to enable license plate enhancement: 0-no, 1-yes.

byLPDELevel

License plate enhancement level, value range: [0,100], default value: 50.

byRes

Reserved, set to 0.

A.240 NET_DVR_SPECIAL_FINDINFO_UNION

Specific search condition union.

Union Definition

```
union{
    BYTE      byLenth[8];
    NET_DVR_ATMFINDINFO  struATMFindInfo;
}NET_DVR_SPECIAL_FINDINFO_UNION,*LPNET_DVR_SPECIAL_FINDINFO_UNION;
```

Members

byLenth

Union size, it is 8 bytes.

struATMFindInfo

Condition for searching file with ATM information, see details in the structure of [**NET_DVR_ATMFINDINFO**](#).

A.241 NET_DVR_STD_ABILITY

Input and Output Parameter Structure for Getting Capabilities

Member	Data Type	Description
lpCondBuffer	LPVOID	Condition parameters (ASCII character format), e.g., the channel No., it can be set to "null".
dwCondSize	DWORD	Buffer size of condition parameters.
lpOutBuffer	LPVOID	Output parameters buffer (the parameter is returned in the message with XML format), it cannot be set to "null".
dwOutSize	DWORD	Output buffer size.
lpStatusBuffer	LPVOID	Get the returned status parameters (<u>XMLResponseStatus</u>) when getting capabilities failed. It can be set to null.
dwStatusSize	DWORD	Status buffer size.

Member	Data Type	Description
dwRetSize	DWORD	Obtained data size (if the capability is obtained, the value refers to the size of IpOutBuffer ; if getting failed, the value refers to the size of IpStatusBuffer).
byRes	Array [BYTE]	Reserved. The maximum size is 32 bytes.

Remarks

For different capability types (which depend on the parameter **dwAbilityType** in the API [**NET_DVR_GetSTDAbility**](#)), the condition parameter **IpCondBuffer** and output parameter **IpOutBuffer** are different. For details, refer to the typical applications.

A.242 NET_DVR_STD_CONFIG

Structure About Configuring Input and Output Parameters

Member	Data Type	Description
IpCondBuffer	LPVOID	Condition parameters, e.g., channel No., it can be set to "NULL".
dwCondSize	DWORD	Size of buffer for storing condition parameters
IpInBuffer	LPVOID	Input parameters (a structure)
dwInSize	DWORD	Size of buffer for storing input parameters
IpOutBuffer	LPVOID	Output parameters (a structure)
dwOutSize	DWORD	Size of buffer for storing output parameters
IpStatusBuffer	LPVOID	Returned status parameters in XML format, it can be set to NULL.
dwStatusSize	DWORD	Size of buffer for storing status parameters
IpXmlBuffer	LPVOID	Request or response message in XML format, it is valid when byDataType is 1.
dwXmlSize	DWORD	Size of memory pointed by IpXmlBuffer .
byDataType	BYTE	Input or output parameter type: 0-valid when the input or output parameters is a structure; 1-

Member	Data Type	Description
		valid when the input or output parameters is a XML message.
byRes	Array [BYTE]	Reserved, set to 0. The maximum size is 32 bytes.

A.243 NET_DVR_STD_CONTROL

Input and output parameters structure of remote control.

Structure Definition

```
struct{
    LPVOID      lpCondBuffer;
    DWORD       dwCondSize;
    LPVOID      lpStatusBuffer;
    DWORD       dwStatusSize;
    BYTE        byRes[64];
}NET_DVR_STD_CONTROL, *LPNET_DVR_STD_CONTROL;
```

Members

lpCondBuffer

Condition parameters (returned in structure or message), such as channel No., it can be set to null.

dwCondSize

Buffer size of condition parameters.

lpStatusBuffer

Status parameters (returned in [XMLResponseStatus](#)), if control completed, this parameter will not be assigned, and it can be set to null if not required.

dwStatusSize

Buffer size of status parameter.

lpXmlBuffer

Data with XML format, it is valid only when **byDataType** is 1.

dwXmlSize

Size of data with XML format, when the control command is getting, it is the input and output parameters, and after getting completed, actual size will be returned; when setting, the actual data size equals to strlen((char*) lpXmlBuffer).

byDataType

Input or output parameter type: 0-structure, 1-message

byRes

Reserved, set to 0.

Remarks

For different control function (refers to the parameter **dwCommand** in the API [NET_DVR_STDControl](#)), the **lpCondBuffer** in this structure are different, see details in the typical applications.

A.244 NET_DVR_STOP_LINE_PARAM

Structure about stop line information.

Structure Definition

```
struct{
    BYTE byStatus;
    BYTE byRes[39];
}NET_DVR_STOP_LINE_PARAM, *LPNET_DVR_STOP_LINE_PARAM;
```

Members

byStatus

Stop line status: 0-unknown, 1-entered, 1-exited.

byRes

Reserved, set to 0.

A.245 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.246 NET_DVR_STREAMING_COND

Condition structure about stream control.

Structure Definition

```
struct{
    DWORD dwSize;
    DWORD dwChannel;
    BYTE byStreamType;
    BYTE byRes[127];
}NET_DVR_STREAMING_COND,*LPNET_DVR_STREAMING_COND;
```

Members

dwSize

Structure size

dwChannel

Channel No.

byStreamType

Stream type: 0-main stream, 1-sub-stream, 2-third stream.

byRes

Reserved, set to 0.

A.247 NET_DVR_STREAM_MEDIA_SERVER

Structure about stream media server parameters.

Structure Definition

```
struct{
    BYTE  byValid;
    BYTE  byRes1[3];
    BYTE  byAddress[MAX_DOMAIN_NAME/*64*/];
    WORD   wDevPort;
    BYTE  byTransmitType;
    BYTE  byRes2[5];
}NET_DVR_STREAM_MEDIA_SERVER,*LPNET_DVR_STREAM_MEDIA_SERVER;
```

Members

byValid

Enable/disable stream media server to get stream: 0-disable, 1-enable.

byRes1

Reserved, set to 0.

byAddress

IP address or domain name of stream media server

wDevPort

Port number of stream media server

byTransmitType

Transfer protocol type: 0-TCP, 1-UDP

byRes2

Reserved, set to 0.

See Also

[NET_DVR_DEC_STREAM_DEV_EX](#)

[NET_DVR_DEC_DDNS_DEV](#)

A.248 NET_DVR_STREAM_MODE

Structure about steaming mode configuration.

Structure Definition

```
struct{
    BYTE      byGetStreamType;
    BYTE      byRes[3];
    NET_DVR_GET_STREAM_UNION uGetStream;
}NET_DVR_STREAM_MODE,*LPNET_DVR_STREAM_MODE;
```

Members

byGetStreamType

Streaming mode:

- 0-get stream from device directly
- 1-get stream from stream media server
- 2-get stream from device after getting the IP address by IPServer
- 3-get the device IP address by IPServer, and then get stream from stream media server
- 4-get stream from stream media server by URL
- 5-get steam from device after connecting the device via hkDDNS
- 6-get steam from device directly (extended)

byRes

Reserved, set to 0

uGetStream

Union of streaming modes, refer to [**NET_DVR_GET_STREAM_UNION**](#) for details.

See Also

[**NET_DVR_IPPARACFG_V40**](#)

A.249 NET_DVR_THERMOMETRY_AGC

AGC parameter structure of temperature measurement.

Structure Definition

```
struct{
    BYTE  byMode;
    BYTE  byRes1[3];
    int   iHighTemperature;
    int   iLowTemperature;
    BYTE  byRes[8];
}NET_DVR_THERMOMETRY_AGC,*LPNET_DVR_THERMOMETRY_AGC;
```

Members

byMode

AGC mode: 0-invalid, 1-auto, 2-manual.

byRes1

Reserved, set to 0.

iHighTemperature

Highest temperature, value range: [-273,9999], unit: °C; it is valid when **byMode** is set to 1.

iLowTemperature

Lowest temperature, value range: [-273,9999], unit: °C; it is valid when **byMode** is set to 1.

byRes

Reserved, set to 0.

A.250 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.251 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

Member	Data Type	Description
bySecond	BYTE	Second
byRes	BYTE	Reserved.

A.252 NET_DVR_TIME_SEARCH

OSD time information structure.

Structure Definition

```
struct{
    WORD  wYear;
    BYTE  byMonth;
    BYTE  byDay;
    BYTE  byHour;
    BYTE  byMinute;
    BYTE  bySecond;
    char   cTimeDifferenceH;
    char   cTimeDifferenceM;
    BYTE  byRes[3];
}NET_DVR_TIME_SEARCH, *LPNET_DVR_TIME_SEARCH;
```

Members

wYear

Year

byMonth

Month

byDay

Day

byHour

Hour

byMinute

Minute

bySecond

Second

cTimeDifferenceH

Time difference (hour) between OSD time and UTC, which ranges from -12 to +14.

cTimeDifferenceM

Time difference (minute) between OSD time and UTC, the values can be -30, 0, 30, and 45.

byRes

Reserved.

A.253 NET_DVR_TIME_SEARCH_COND

Time data structure of search condition.

Structure Definition

```
struct{
    WORD  wYear;
    BYTE  byMonth;
    BYTE  byDay;
    BYTE  byHour;
    BYTE  byMinute;
    BYTE  bySecond;
    BYTE  byLocalOrUTC;
    BYTE  byRes[4];
}NET_DVR_TIME_SEARCH_COND, *LPNET_DVR_TIME_SEARCH_COND;
```

Members

wYear

Year, device OSD time.

byMonth

Month, device OSD time.

byDay

Day, device OSD time.

byHour

Hour, device OSD time.

byMinute

Minute, device OSD time.

bySecond

Second, device OSD time.

byLocalOrUTC

Time type: 0-device local time (device's OSD time), 1-UTC time.

byRes

Reserved, set to 0.

See Also

[NET_DVR_FILECOND_V50](#)

A.254 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 zone), it is valid when byISO8601 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 byISO8601 is "1".

A.255 NET_DVR_TIME_V50

Time parameters structure.

Structure Definition

```
struct{
    WORD      wYear;
    BYTE      byMonth;
    BYTE      byDay;
    BYTE      byHour;
    BYTE      byMinute;
    BYTE      bySecond;
    BYTE      byISO8601;
    WORD      wMilliSec;
    signed char cTimeDifferenceH;
    signed char cTimeDifferenceM;
}NET_DVR_TIME_V50, *LPNET_DVR_TIME_V50;
```

Members

wYear

Year

byMonth

Month

byDay

Day

byHour

Hour

byMinute

Minute

bySecond

Second

byISO8601

ISO8601 format, whether time differences are valid, 0-no, it is local time, 1-yes

wMillisecond

Millisecond, it is 0 by default

cTimeDifferenceH

Time offset (hours) from UTC, e.g., -12 ... +14, positive offset indicates eastern time zone

cTimeDifferenceM

Time offset (minutes) from UTC, e.g., -30, 0, 30, 45, positive offset indicates eastern time zone

A.256 NET_DVR_TIMING_CAPTURE

Structure about scheduled capture parameters.

Structure Definition

```
struct{
    NET_DVR_JPEGPARA    struJpegPara;
    DWORD                dwPicInterval;
    BYTE                 byRes[12];
}NET_DVR_TIMING_CAPTURE, *LPNET_DVR_TIMING_CAPTURE;
```

Members

struJpegPara

Quality of pictured captured by schedule, see details in the structure [NET_DVR_JPEGPARA](#).

dwPicInterval

Time interval of scheduled capture which is between 500 and 604800000, unit: ms.

byRes

Reserved, set to 0.

A.257 NET_DVR_TRAVERSE_PLANE_SEARCHCOND

Condition structure for searching videos recorded based on line crossing detection.

Structure Definition

```
struct{
    NET_VCA_TRAVERSE_PLANE    struVcaTraversePlane[MAX_ALERTLINE_NUM/*8*/];
    DWORD                      dwPreTime;
    DWORD                      dwDelayTime;
    NET_DVR_PTZPOS_INFO       struPTZPosInfo;
    BYTE                       byAdvanceType;
    BYTE                       byRes1[3];
    NET_DVR_ADVANCE_COND_UNION uAdvanceCond;
    BYTE                       byRes[5604];
}NET_DVR_TRAVERSE_PLANE_SEARCHCOND,*LPNET_DVR_TRAVERSE_PLANE_SEARCHCOND;
```

Members

struVcaTraversePlane

Line crossing detection region parameters, refer to the structure NET_VCA_TRAVERSE_PLANE in the HCNetSDK user manual of behavior analysis applications.

dwPreTime

Pre-alarm time, unit: second.

dwDelayTime

Post-alarm time, unit: second.

struPTZPosInfo

Camera's PTZ position information in line crossing detection, refer to the structure for details.

byAdvanceType

Group types: 0-without group No., 1-group with human body properties, 2-group with object's color share. Refer to the structure for details.

byRes1

Reserved, set to 0.

uAdvanceCond

Group attribute.

byRes

Reserved, set to 0.

See Also

A.258 NET_DVR_UPGRADE_PARAM

Structure about parameters of remotely upgrading device.

Structure Definition

```
typedef struct{
    DWORD      dwUpgradeType;
    char const *sFilename;
    void       *pInbuffer;
    DWORD      dwBufferLen;
    char       *pUnitIdList[64];
    BYTE       byRes[112];
}NET_DVR_UPGRADE_PARAM,*LPNET_DVR_UPGRADE_PARAM;
```

Members

dwUpgradeType

Upgrade types, see details below:

```
enum _ENUM_UPGRADE_TYPE{
    ENUM_UPGRADE_DVR      = 0,
    ENUM_UPGRADE_ADAPTER   = 1,
    ENUM_UPGRADE_VCALIB    = 2,
```

```
ENUM_UPGRADE_OPTICAL    = 3,  
ENUM_UPGRADE_ACS        = 4,  
ENUM_UPGRADE_AUXILIARY_DEV = 5,  
ENUM_UPGRADE_LED         = 6,  
ENUM_UPGRADE_INTELLIGENT = 7,  
}ENUM_UPGRADE_TYPE;
```

ENUM_UPGRADE_DVR

Upgrade normal device.

ENUM_UPGRADE_ADAPTER

Upgrade the adapter of DVR.

ENUM_UPGRADE_VCALIB

Upgrade intelligent library.

ENUM_UPGRADE_OPTICAL

Upgrade optical transceiver.

ENUM_UPGRADE_ACS

Upgrade access control system.

ENUM_UPGRADE_AUXILIARY_DEV

Upgrade auxiliary device.

ENUM_UPGRADE_LED

Upgrade LED sending card and receiving card.

ENUM_UPGRADE_INTELLIGENT

Upgrade data analysis server, it is valid when login based on ISAPI protocol.

sFileName

Path of upgrade file (including file name). The path size relates to the operating system, for Windows, the default path size is smaller than or equal to 256 bytes.

pInbuffer

Upgrade condition buffer, different upgrade types correspond to different upgrade conditions, see details in the table below:

dwUpgradeType	pInbuffer
ENUM_UPGRADE_DVR	None.
ENUM_UPGRADE_ADAPTER	None.
ENUM_UPGRADE_VCALIB	4-byte channel No.
ENUM_UPGRADE_OPTICAL	None.

dwUpgradeType	pInbuffer
ENUM_UPGRADE_ACSS	4-bytes device No., 0: upgrade access control server, 1-8: upgrade card readers, 10001: upgrade distributed access controller No.1, 20001: upgrade main lane controller, 20002: upgrade sub-lane controller, 20003: upgrade extension module, 30001: upgrade fingerprint algorithm program of the card reader (1024 reserved), 400001: upgrade uboot.
ENUM_UPGRADE_AUXILIARY_DEV	<u>NET_DVR_AUXILIARY_DEV_UPGRADE_PARAM</u>
ENUM_UPGRADE_LED	None.
ENUM_UPGRADE_INTELLIGENT	None.

dwBufferLen

Buffer size.

pUnitIdList

Data analysis server name, which is obtained by calling the request URL: POST /ISAPI/SDT/Management/AnalysisUnit/search?format=json.

byRes

Reserved, set to 0.

A.259 NET_DVR_UPLOAD_RECORD_INFO

Structure about the video file information to be uploaded.

Structure Definition

```
struct{
    WORD      dwSize;
    WORD      dwRecordType;
    BYTE     sCameraID[MAX_CAMERAID_LEN/*64*/];
    NET_DVR_TIME_EX struStartTime;
    NET_DVR_TIME_EX struStopTime;
    WORD      dwStoragePoolID;
    BYTE     byFormatType;
    BYTE     byVideoEncType;
    BYTE     byAudioEncType;
    BYTE     byISO8601;
    signed char cStartTimeDifferenceH;
    signed char cStartTimeDifferenceM;
    signed char cStopTimeDifferenceH;
    signed char cStopTimeDifferenceM;
```

```
BYTE      byRes[120];
}NET_DVR_UPLOAD_RECORD_INFO,*LPNET_DVR_UPLOAD_RECORD_INFO;
```

Members

dwSize

Structure size.

dwRecordType

Recording type: 0-scheduled recording, 1-recording based on motion detection, 2-recording based on alarm, 3-recording based on alarm or motion detection, 4-recording based on alarm and motion detection, 5-recording based on command, 6-manual recording, 7-recording based on vibration alarm, 8-recording based on environment alarm, 9-recording based on VCA alarm, 10-recording based on ANR.

sCameraID

Camera ID.

struStartTime

Start time of recording, refer to the structure for details.

struStopTime

End time of recording, refer to the structure for details.

dwStoragePoolID

Storage pool ID.

byFormatType

Container format: 1-raw stream, 2-RTP, 3-PS, 4-TS, 5-private, 6-FLV, 7-ASF, 8-3GP, 9-RTP+PS, 0xff-invalid.

byVideoEncType

Video encoding type: 0-private H.264, 1-standard H.264, 2-standard MPEG4, 7-MJPEG, 8-MPEG2, 0xfe-auto (same with the source), 0xff-invalid.

byAudioEncType

Audio encoding type: 0-G.722, 1-G.711_U, 2-G.711_A, 5-MP2L2, 6-G.726, 7-AAC, 0xfe-auto (same with the source), 0xff-invalid.

byISO8601

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.

cStartTimeDifferenceH

Time difference between start time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStartTimeDifferenceM

Time difference between start time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStopTimeDifferenceH

Time difference between stop time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

cStopTimeDifferenceM

Time difference between stop time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".

byRes

Reserved, set to 0.

Related API

[NET_DVR_UploadFile V40](#)

A.260 NET_DVR_UPNP_NAT_STATE

UPnP port mapping status structure

Structure Definition

```
struct{
    NET_DVR_UPNP_PORT_STATE strUpnpPort[12];
    BYTE                 byRes[200];
}NET_DVR_UPNP_NAT_STATE,*LPNET_DVR_UPNP_NAT_STATE;
```

Members

strUpnpPort

Port mapping status: array0-web server port, array1-management port, array2-RTSP port, array3-HTTPs port, array4-SDKOverTLS port. See details in the structure .

byRes

Reserved.

Related API

[NET_DVR_GetUpnpNatState](#)

A.261 NET_DVR_UPNP_PORT_STATE

Port mapping status structure

Structure Definition

```
struct{
    DWORD      dwEnabled;
    WORD       wInternalPort;
    WORD       wExternalPort;
    DWORD      dwStatus;
    NET_DVR_IPADDR struNatExternalIp;
    NET_DVR_IPADDR struNatInternalIp;
    BYTE       byRes[16];
}NET_DVR_UPNP_PORT_STATE, *LPNET_DVR_UPNP_PORT_STATE;
```

Members

dwEnabled

Whether to enable mapping for this port.

wInternalPort

Port before mapping.

wExternalPort

Port after mapping.

dwStatus

Port mapping status: 0-unmapped; 1-unmapped: the mapping source port and target port should be same; 2-unmapped: the mapping port number is occupied; 3-mapped.

struNatExternalIp

External address after mapping.

struNatInternalIp

IP address of NAT router in LAN.

byRes

Reserved.

See Also

A.262 NET_DVR_USB_RS232

Structure about USB to port mode configuration

Structure Definition

```
struct{
    DWORD  dwBaudRate;
    BYTE   byDataBit;
```

```

BYTE byStopBit;
BYTE byParity;
BYTE byFlowcontrol;
BYTE byVirtualSerialPort;
BYTE byRes[3];
}NET_DVR_USB_RS232, *LPNET_DVR_USB_RS232;
```

Members

dwBaudRate

Baud rate(bps), 0-50, 1-75, 2-110, 3-150, 4-300, 5-600, 6-1200, 7-2400, 8-4800, 9-9600, 10-19200, 11-38400, 12-57600, 13-76800, 14-115.2k

byDataBit

Number of data bits: 0-5 bits, 1-6 bits, 2-7 bits, 3-8 bits

byStopBit

Number of end bits: 0-1 bit, 1-2 bits

byParity

Whether to check parity: 0-no, 1-odd parity, 2-even parity

byFlowcontrol

Whether to enable flow control: 0-no, 1-software flow control, 2-hardware flow control.

byVirtualSerialPort

Virtual serial port No.

byRes

Reserved, set it to 0.

A.263 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.

Member	Data Type	Description
sPassword	char	Login password.
cbLoginResult	fLoginResultCallBack	Callback function used to return login status, it is valid only when bUseAsynLogin 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 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 byLoginMode is 0 and byHttps is 1); when byVerifyMode is 0, CA certificate is not required, when byVerifyMode 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.264 NET_DVR_USER_INFO_V52

User information structure (V52)

Structure Definition

```

struct{
    BYTE      sUserName[NAME_LEN/*32*/];
    BYTE      sPassword[PASSWD_LEN/*16*/];
    BYTE      dwLocalRight[MAX_RIGHT/*32*/];
    BYTE      dwRemoteRight[MAX_RIGHT/*32*/];
    DWORD     dwNetPreviewRight[MAX_CHANNUM_V40/*512*/];
    DWORD     dwLocalRecordRight[MAX_CHANNUM_V40/*512*/];
    DWORD     dwNetRecordRight[MAX_CHANNUM_V40/*512*/];
    DWORD     dwLocalPlaybackRight[MAX_CHANNUM_V40/*512*/];
    DWORD     dwNetPlaybackRight[MAX_CHANNUM_V40/*512*/];
    DWORD     dwLocalPTZRight[MAX_CHANNUM_V40/*512*/];
    DWORD     dwNetPTZRight[MAX_CHANNUM_V40/*512*/];
    DWORD     dwLocalBackupRight[MAX_CHANNUM_V40/*512*/];
    DWORD     dwLocalPreviewRight[MAX_CHANNUM_V40/*512*/];
    DWORD     dwLocalDoubleVerificationPlaybackRight[MAX_CHANNUM_V40/*512*/];
    DWORD     dwLocalDoubleVerificationBackupRight[MAX_CHANNUM_V40/*512*/];
    DWORD     dwNetDoubleVerificationPlaybackRight[MAX_CHANNUM_V40/*512*/];
    NET_DVR_IPADDR struUserIP;
    BYTE      byMACAddr[MACADDR_LEN/*6*/];
    BYTE      byPriority;
    BYTE      byAlarmOnRight;
    BYTE      byAlarmOffRight;
    BYTE      byBypassRight;
    BYTE      byRes1[2];
    BYTE      byPublishRight[MAX_RIGHT/*32*/];
    DWORD     dwPasswordValidity;
    BYTE      byRes[1024];
}NET_DVR_USER_INFO_V52,*LPNET_DVR_USER_INFO_V52;

```

Members

sUserName

User name, up to 16 bytes are allowed.

sPassword

Password. For getting user information, this member will not be returned; for setting user information, if this member is empty, it indicates that the password will not be changed.

byLocalRight

Local permissions:

- 0: PTZ control
- 1: Manual recording
- 2: Playback
- 3: Parameter settings
- 4: Status and log search

- 5: Advanced operations (upgrade, format, restart, shutdown)
- 6: View parameters
- 7: Analog and network camera management
- 8: Backup
- 9: Shut down or restart
- 10: Live view
- 11: Double verification for playback
- 12: Double verification for backup

byRemoteRight

- Remote permissions:
- 0: PTZ control
- 1: Manual recording
- 2: Playback
- 3: Parameter settings
- 4: Status and log search
- 5: Advanced operations (upgrade, format, restart, shutdown)
- 6: Start two-way audio
- 7: View parameters
- 8: Request for alarm uploading and alarm output
- 9: Local output control
- 10: Serial port control
- 11: View parameters
- 12: Analog and network camera management
- 13: Shut down or restart
- 14: Double verification for playback

dwNetPreviewRight

Channels support remote live view, 0xffffffff-invalid

dwLocalRecordRight

Channels support local recording

dwNetRecordRight

Channels support remote recording

dwLocalPlaybackRight

Channels support local playback

dwNetPlaybackRight

Channels support remote playback

dwLocalPTZRight

Channels support local PTZ control

dwNetPTZRight

Channels support remote PTZ control

dwLocalBackupRight

Channels support local backup

dwLocalPreviewRight

Channels support local live view

dwLocalDoubleVerificationPlaybackRight

Channels support double verification for local playback

dwLocalDoubleVerificationBackupRight

Channels support double verification for local backup

dwNetDoubleVerificationPlaybackRight

Channels support double verification for remote playback

struUserIP

User IP address, 0-any IP address is allowed.

byMACAddr

Physical address, 00:00:00:00:00:00-any addresses is allowed.

byPriority

Priority: 0xff-priority is not configured, 0-low (default permission, including local and remote playback, local and remote log and status search, local and remote shutdown or restart), 1-medium (including local and remote PTZ control, local and remote manual recording, local and remote playback, two-way audio, remote live view, local backup, local and remote shutdown or restart), 2-high (administrator)

byAlarmOnRight

Whether contains alarm input arming permission: 1-yes, 0-no

byAlarmOffRight

Whether contains alarm input disarming permission: 1-yes, 0-no

byBypassRight

Whether contains alarm input bypass permission: 1-yes, 0-no

byRes1

Reserved, set to 0.

byPublishRight

Information release permissions.

dwPasswordValidity

Password expiry date, it can only be edited by admin user, unit: day, 0-permanent password.

byRes

Reserved, set to 0.

A.265 NET_DVR_USER_V52

User parameters structure (V52)

Structure Definition

```
struct{
    DWORD      dwSize;
    DWORD      dwMaxUserNum;
    NET_DVR_USER_INFO_V51 struUser[MAX_USERNUM_V30/*32*/];
    char       sloginPassword[PASSWD_LEN/*16*/];
    BYTE       byRes[240];
}NET_DVR_USER_V51,*LPNET_DVR_USER_V51;
```

Members

dwSize

Structure size.

dwMaxUserNum

Maximum number of users that can log in to device (read-only).

struUser

User information, see details in structure [NET_DVR_USER_INFO_V52](#).

sloginPassword

Set password for confirmation, the password should be same with the **sPassword** in [NET_DVR_USER_INFO_V52](#).

byRes

Reserved, set to 0

A.266 NET_DVR_VCA_VERSION

Structure of intelligent algorithm version parameter.

Structure Definition

```
struct{
    WORD  wMajorVersion;
    WORD  wMinorVersion;
    WORD  wRevisionNumber;
```

```
WORD wBuildNumber;
WORD wVersionYear;
BYTE byVersionMonth;
BYTE byVersionDay;
BYTE byType;
BYTE byRes[7];
}NET_DVR_VCA_VERSION, *LPNET_DVR_VCA_VERSION;
```

Members

wMajorVersion

Major version

wMinorVersion

Minor version

wRevisionNumber

Revision number

wBuildNumber

Build number

wVersionYear

Version: Year

byVersionMonth

Version: Month

byVersionDay

Version: Day

byType

Algorithms library type name: 0-reserved, 1-face attribute, 2-behavior analysis, 3-body attribute, 4-face capture, 5-face recognition, 6-face comparison, 7-student stood up detection, 8-people counting, 9-face picture library, 10-hard hat detection, 11-elevator detection algorithms library, 12-people counting statistics, 13-HMS, 14-teacher behavior detection, 15-people density detection, 16-mismatched number of people, 17-absence detection, 18-overstay detection

byRes

Reserved

A.267 NET_DVR_VCA_VERSION_LIST

Algorithms library version information structure.

Structure Definition

```
struct{
    DWORD      dwSize;
    NET_DVR_VCA_VERSION  struVcaVersion[NET_SDK_VERSION_LIST_LEN/*64*/];
    BYTE       byRes[128];
}NET_DVR_VCA_VERSION_LIST, *LPNET_DVR_VCA_VERSION_LIST;
```

Members

dwSize

Structure size

struVcaVersion

Algorithms library version information list, each array indicates one type, see details in the structure [**NET_DVR_VCA_VERSION**](#).

byRes

Reserved.

A.268 NET_DVR_VEHICLE_PARA

Structure about the vehicle detection information.

Structure Definition

```
struct{
    BYTE  sLicense[MAX_LICENSE_LEN/*16*/];
    BYTE  byCountry;
    BYTE  byRes[239];
}NET_DVR_VEHICLE_PARA, *LPNET_DVR_VEHICLE_PARA;
```

Members

sLicense

License plate number.

byCountry

Country index No.: 0-not support, 1-Czech Republic, 2-France, 3-Germany, 4-Spain, 5-Italy, 6-Netherlands, 7-Poland, 8-Slovakia, 9-Belorussia, 10-Moldova, 11-Russia, 12-Ukraine, 169-Canada, 172-US, 227-Australia, 0xfe-unknown.

byRes

Reserved

See Also

A.269 NET_DVR_VICOLOR

Structure about image parameters in a time period.

Structure Definition

```
struct{
    NET_DVR_COLOR      struColor[MAX_TIMESEGMENT_V30/*8*/];
    NET_DVR_SCHEDTIME  struHandleTime[MAX_TIMESEGMENT_V30/*8*/];
}NET_DVR_VICOLOR,*LPNET_DVR_VICOLOR;
```

Members

struColor

Image parameter, only the brightness (**byBrightness**) is valid, see details in the structure [**NET_DVR_COLOR**](#).

struHandleTime

Processing time duration, see details in the structure [**NET_DVR_SCHEDTIME**](#)

See Also

[**NET_DVR_PICCFG_V40**](#)

A.270 NET_DVR_VIDEOEFFECT

Video parameter structure

Structure Definition

```
struct{
    BYTE  byBrightnessLevel;
    BYTE  byContrastLevel;
    BYTE  bySharpnessLevel;
    BYTE  bySaturationLevel;
    BYTE  byHueLevel;
    BYTE  byEnableFunc;
    BYTE  byLightInhibitLevel;
    BYTE  byGrayLevel;
}NET_DVR_VIDEOEFFECT,*LPNET_DVR_VIDEOEFFECT;
```

Members

byBrightnessLevel

Brightness, range: [0,100]

byContrastLevel

Contrast, range: [0,100]

bySharpnessLevel

Sharpness, range: [0,100]

bySaturationLevel

Saturation, range: [0,100]

byHueLevel

Hue, range: [0,100], reserved.

byEnableFunc

Bit: bit0-enable/disable Smart IR, bit1-enable/disable low illumination, bit2-enable/disable HLC, bit3-sharpness adjusting mode (manual or auto). Value: 0-disable/auto, 1-enable/manual. E.g., byEnableFunc&0x2==1, it indicates that the low illumination function is enabled.

byLightInhibitLevel

HLC level, range: [1,3]

byGrayLevel

Gray scale, 0-[0,255], 1-[16,235]

A.271 NET_DVR_VILOST_V40

Structure about video loss alarm parameters.

Structure Definition

```
struct{
    DWORD      dwEnableVILostAlarm;
    DWORD      dwHandleType;
    DWORD      dwMaxRelAlarmOutChanNum;
    DWORD      dwRelAlarmOut[MAX_ALARMOUT_V40/*4128*/];
    NET_DVR_SCHEDTIME struAlarmTime[MAX_DAYS/*7*/][MAX_TIMESEGMENT_V30/*8*/];
    BYTE       byVILostAlarmThreshold;
    BYTE       byRes[64];
}NET_DVR_VILOST_V40, *LPNET_DVR_VILOST_V40;
```

Members

dwEnableVILostAlarm

Whether to enable video loss alarm: 0-no, 1-yes.

dwHandleType

Alarm linkage actions, see details below:

- 0x00: No response
- 0x01: Monitor warning

- 0x02: Audible warning
- 0x04: Upload to center
- 0x08: Trigger alarm output
- 0x10: Capture JPEG picture and send Email
- 0x20: Wireless combined aural and visual alarm linkage
- 0x40: E-map linkage (only supported by PCNVR)
- 0x200: Capture and upload to FTP
- 0x1000: Capture and upload to cloud storage

E.g. dwHandleType==0x01|0x04 indicates that the alarm linkage action is monitor warning and uploading to center.

dwMaxRelAlarmOutChanNum

The maximum number of alarm outputs can be triggered (read only)

dwRelAlarmOut

Triggered alarm output No., which starts from 0, 0xffffffff-the following is invalid. For example, if the start value of dwRelAlarmOut[8] is 0xffffffff, the alarm output No. (dwRelAlarmOut[0]+1), ..., (dwRelAlarmOut[7]+1) will be triggered.

struAlarmTime

Arming time, up to 8 time periods can be set on a day, see details in the structure

[NET_DVR_SCHEDTIME](#).

byVILostAlarmThreshold

Video loss alarm threshold, when the value is lower than the threshold, the alarm is triggered, the threshold ranges from 0 to 99.

byVILostAlarmThreshold

Threshold of video loss alarm, it ranges from 0 and 99. When the value is smaller than the threshold, it indicates that video loss occurred.

byRes

Reserved, set to 0

See Also

[NET_DVR_PICCFG_V40](#)

A.272 NET_DVR_VOD_DRAWFRAME_PARA

Condition structure for extracting frame during playback.

Structure Definition

```
struct{
    NET_DVR_TIME_EX      struTime;
    DWORD                 dwDrawType
```

```
    BYTE           byRes[128];
}NET_DVR_VOD_DRAWFRAME_PARA, *LPNET_DVR_VOD_DRAWFRAME_PARA;
```

Members

struTime

Time to start extracting frame.

dwDrawType

Frame extraction mode (it should be supported by device): 0: transmit I frame only.

- 0: Transmit I frame only.
- 1: Drop 1/2 P frame (only supported by SVC stream)
- 2: Drop 3/4 P frame (only supported by SVC stream)
- 3: Transmit 1/2 I frame (I frame only, transmit once per two I frames)
- 4: Transmit 1/4 I frame (I frame only, transmit once per four I frames)
- 5: Transmit 1/8 I frame (I frame only, transmit once per eight 8 I frames)

byRes

Reserved, set to 0.

Related API

[NET_DVR_PlayBackControl_V40](#)

A.273 NET_DVR_VOD_PARA_V50

Video playback parameter structure.

Structure Definition

```
struct{
    WORD          dwSize;
    NET_DVR_STREAM_INFO   struIDInfo;
    NET_DVR_TIME_V50     struBeginTime;
    NET_DVR_TIME_V50     struEndTime;
    HWND           hWnd;
    BYTE           byDrawFrame;
    BYTE           byVolumeType;
    BYTE           byVolumeNum;
    BYTE           byStreamType;
    BYTE           dwFileIndex;
    BYTE           byAudioFile;
    BYTE           byCourseFile;
    BYTE           byPlayMode;
    BYTE           byLinkMode;
    BYTE           byDownload;
    BYTE           byOptimalStreamType;
    BYTE           byDisplayBufNum;
    BYTE           byNPQMode;
```

```
BYTE          sUserName[NAME_LEN/*32*/];
BYTE          sPassword[PASSWD_LEN/*16*/];
BYTE          byRes2[202];
BYTE          byHls;
char*        pSavedFileName;
}NET_DVR_VOD_PARA_V50, *LPNET_DVR_VOD_PARA_V50;
```

Members

dwSize

Structure size.

struIDInfo

Stream information. This parameter is not supported when logging in based on ISAPI protocol.
Refer to the structure for details.

struBeginTime

Start time of playback, refer to the structure .

struEndTime

End time of playback, refer to the structure .

hWnd

Handle of playback window. If it is set to NULL, the stream data can still be received, but it will not be decoded and displayed .

byDrawFrame

Whether to extract frame: 0-no, 1-yes. This parameter is not supported when login based on ISAPI protocol.

byVolumeType

0-video volume, 1-storage volume (used for backup, and it is available for CVR). This parameter is not supported when login based on ISAPI protocol.

byVolumeNum

Storage volume No. This parameter is not supported when login based on ISAPI protocol.

byStreamType

Stream type: 0-main stream, 1-sub-stream, 2-third stream. This parameter is not supported when login based on ISAPI protocol.

dwFileIndex

Video file No. in storage volume, which is returned when searching for videos in the storage volume. This parameter is not supported when login based on ISAPI protocol.

byAudioFile

Whether to play audio: 0-no, 1-yes. This parameter is not supported when login based on ISAPI protocol.

byCourseFile

Whether to play course file: 0-no, 1-yes. This parameter is not supported when login based on ISAPI protocol.

byPlayMode

Playbakc mode: 0-playback, 1-reverse playback.

byLinkMode

Device connection mode:0-TCP, 1-Adaptive UDP

byDownload

Whether to download: 0-no, 1-yes

byOptimalStreamType

Whether plays back by optimal stream type: 0-no, 1-yes. (For dual-stream device, if the video file in a certain time period does not match the specified stream type, then the actual stream type video will be returned.)

byDisplayBufNum

The number of frames in buffer, it is valid when **hWnd** not values "NULL". By default it values "3"

byNPQMode

NPQ (Network Protocol Quality) mode: 0-direct connect, 1-stream media

sUserName

Double verification user name, the sensitive information should be encrypted.

sPassword

Double verification user password, the sensitive information should be encrypted.

byRes2

Reserved.

byHls

Whether to enable HLS playback: 0-no, 1-yes.

pSavedFileName

Path for saving downloaded recording files, it is valid when **byDownload** is "1". If the path length is longer than 256 bytes, the path will be cut off.

Related API

NET_DVR_PlayBackByTime_V50

A.274 NET_DVR_WDR

Wide dynamic range (WDR) parameter structure

Structure Definition

```
struct{
    BYTE   byWDREnabled;
    BYTE   byWDRLevel1;
    BYTE   byWDRLevel2;
    BYTE   byWDRContrastLevel;
    BYTE   byRes[16];
}NET_DVR_WDR, *LPNET_DVR_WDR;
```

Members

byWDREnabled

Whether to enable WDR: 0-no, 1-yes.

byWDRLevel1

WDR level 1: form 0 to F.

byWDRLevel2

WDR level 2: form 0 to F.

byWDRContrastLevel

Contrast level, which is between 0 and 100.

byRes

Reserved, set to 0.

A.275 NET_DVR_WHITEBALANCE

White balance parameter structure

Structure Definition

```
struct{
    BYTE   byWhiteBalanceMode;
    BYTE   byWhiteBalanceModeRGain;
    BYTE   byWhiteBalanceModeBGain;
    BYTE   byRes[5];
}NET_DVR_WHITEBALANCE, *LPNET_DVR_WHITEBALANCE;
```

Members

byWhiteBalanceMode

0-manual white balance (MWB), 1-auto white balance 1 (AWB1, small scale), 2-auto white balance 2 (AWB2, large scale, 2200K-15000K), 3-lock the white balance, 4-outdoor, 5-indoor, 6-daylight lamp, 7-sodium lamp, 8-auto-tracking, 9-white balance for once, 10-auto outdoor, 11-

auto sodium lamp, 12-mercury lamp, 13-auto white balance, 14-incandescent lamp, 15-warm light, 16-natural light.

byWhiteBalanceModeRGain

R gain of manual white balance, this member is valid only when **byWhiteBalanceMode** is set to "0".

byWhiteBalanceModeBGain

B gain of manual white balance, this member is valid only when **byWhiteBalanceMode** is set to "0".

byRes

Reserved, set to 0.

A.276 NET_DVR_WIRELESSDIAL_CFG

Wireless dial-up connection configuration structure.

Structure Definition

```
struct{
    DWORD dwSize;
    BYTE byEnabled;
    BYTE byDialMode;
    BYTE byNetworkMode;
    BYTE byRes1;
    BYTE byDialNum[MAX_DIALNUM_LENGTH];
    BYTE byUserName[NAME_LEN];
    BYTE byPassword[NAME_LEN];
    BYTE byAPNName[NAME_LEN];
    BYTE byUIMCardNum[NAME_LEN];
    BYTE byVerifProtocol;
    BYTE byRes2;
    WORD wMTU;
    DWORD dwOfflineTime;
    BYTE byNetAPN[NAME_LEN];
    BYTE byEnabled4G;
    BYTE byEnabledDNS;
    BYTE byRes3[62];
}NET_DVR_WIRELESSDIAL_CFG,*LPNET_DVR_WIRELESSDIAL_CFG;
```

Members

dwSize

Structure size

byEnabled

Enable wireless dial-up? 0-No, 1- Yes

byDialMode

Dial-up mode: 0-Auto, 1-Manual, The defult setting is auto, you can set the dial-up schedule, off-time time, manual dial-up under the manual mode.

byNetworkMode

Network mode: 0-Auto, 1-4G precedence, 2-3G precedence, 3- auto switch 2G, 4- auto switch 3G, 5- auto switch 4G, 6-wired connection precedence

byRes1

Reserved

byDialNum

Dial-up number

byUserName

User name

byPassword

Password

byAPNName

APN

byUIMCardNum

Mobile phone number

byVerifProtocol

Verification protocol:0- Auto, 1- CHAP, 2- PAP

byRes2

Reserved

wMTU

MTU Value

dwOfflineTime

Offline time, value range: 30 to 65535, unit: second, valid under the manual mod

byNetAPN

APN field configuration of LAN

byEnabled4G

Whether to enable 4G: 0-no, 1-yes

byEnabledDNS

Whether to enable DNS manual configuration: 0-no, 1-yes

byRes3

Reserved, please set to 0

Remarks

This function corresponded wireless dial-up capability(Dial), API:
NET_DVR_GetSTDAbility(Capability Set: NET_DVR_GET_WIRELESSDIAL_CAPABILITIES).

A.277 NET_DVR_WIRELESSDIAL_CONNECT_PARAM

Structure of wireless dialing connection control parameters.

Structure Definition

```
struct{
    DWORD dwSize;
    DWORD dwInterface;
    BYTE byEnableConnect;
    BYTE byRes[255];
}NET_DVR_WIRELESSDIAL_CONNECT_PARAM,*LPNET_DVR_WIRELESSDIAL_CONNECT_PARAM;
```

Members

dwSize

Structure size

dwInterface

Network card No.

byEnableConnect

Be connected to the internet? 0- Offline, 1- Online

byRes

Reserved, please set to 0

Remarks

After the online/offline control, you can refresh and get the current status via the API of
NET_DVR_GetSTDConfig(Command: NET_DVR_GET_WIRELESSDIAL_STATUS)

A.278 NET_DVR_WIRELESSDIAL_SCHEDULE

Wireless dial-up schedule structure.

Structure Definition

```
struct{
    DWORD      dwSize;
    NET_DVR_SCHETIME struAlarmTime[MAX_DAYS][MAX_TIMESEGMENT_V30];
    NET_DVR_SCHETIME struHolidayAlarmTime[MAX_TIMESEGMENT_V30]
```

```
    BYTE      byRes[160];
}NET_DVR_WIRELESSDIAL_SCHEDULE,*LPNET_DVR_WIRELESSDIAL_SCHEDULE;
```

Members

dwSize

Structure size

struAlarmTime

Arming time, 7 days a week, and max. 8 time segments each day

struHolidayAlarmTime

Holiday arming period

byRes

Reserved, please set to 0

Remarks

This function corresponded dial-up schedule capability (Schedule), API: NET_DVR_GetSTDAbility (Capability Set: NET_DVR_GET_WIRELESSDIAL_SCHEDULE_CAPABILITIES).

A.279 NET_DVR_WIRELESSDIAL_STATUS

Wireless connection status structure

Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE      byRealtimeMode[MAX_LENGTH_32];
    BYTE      byUIMStatus[MAX_LENGTH_32];
    DWORD      dwSignalQuality;
    BYTE      byDialStatus[MAX_LENGTH_32];
    NET_DVR_IPADDR struIpAddr;
    NET_DVR_IPADDR struIPMask;
    NET_DVR_IPADDR struGatewayIPMask;
    NET_DVR_IPADDR struDnsServerIpAddr;
    NET_DVR_IPADDR struSubDnsServerIpAddr;
    BYTE      byOperator[MAX_LENGTH_32];
    BYTE      byEnabledMessageConfig;
    BYTE      byRegistrationStatus;
    BYTE      byRes[256];
}NET_DVR_WIRELESSDIAL_CONNECT_PARAM,*LPNET_DVR_WIRELESSDIAL_CONNECT_PARAM;
```

Members

dwSize

Structure size

byRealtimeMode

Real-time mode (CDMA 1x, EVDO, HYBRID, GSM, GPRS, EDGE, WCDMA, HSDPA, HSUPA, HSPA, TDSCDMA, TD-LTE, FDD-LTE, LTE, UNKNOWN)

byUIMStatus

UIM status: "UNKNOWN"-unknown, "VALID"-valid, "NOVALID"-invalid, "ROAM"-roam, "NOEXIST"-not exist,"WRONGPINCODE"-PIN code error,"NONETWORKSERVICE"-connectionless network service, "ONLYSUPPORTEMERGENCYCALL"-the current network only supports emergency calls, "PINCODELOCKED"—PIN code locked

dwSignalQuality

Signal quality, value range: 1 to 100

byDialStatus

Dial-up status code: "disconnect"-disconnected, "dialing"-dialing, "success"-dialed, "shmError"-dialing exception, "certificationFailure"-authentication failed, "invalidDialingCharacters"-invalid dialing characters, "networkTimeout"-network timeout, "wirelessModemError"-wireless module error,"SIMError"-SIM card error

strulpAddr

IP address

strulPMask

Mask address

struGatewayIPMask

Gateway address

struDnsServerIpAddr

DNS address

struSubDnsServerIpAddr

Sub DNS address

byOperator

Operator name

byEnabledMessageConfig

Whether to enable SMS: 0-invalid, 1-enable, 2-disable.

byRegistrationStatus

Registration status: 0-invalid, 1-registered, 2-unregistered.

byRes

Reserved, please set to 0

A.280 NET_DVR_WIRELESSERVER

Hotspot service parameter structure

Structure Definition

```
struct{
    DWORD  dwSize;
    BYTE   byWlanShare;
    BYTE   byBroadcastSSID;
    BYTE   bySecurityMode;
    BYTE   byAlgorithmType;
    char   szSSID[MAX_SSID_LEN/*32*/];
    char   szPassWord[MAX_WS_PASSWD_LEN/*64*/];
    BYTE   byDefaultPassword;
    BYTE   byWifiApModeType;
    BYTE   byRes[254];
}NET_DVR_WIRELESSERVER,*LPNET_DVR_WIRELESSERVER;
```

Members

dwSize

Structure size

byWlanShare

Whether to enable wireless hotspot: 0-no, 1-yes.

byBroadcastSSID

Whether to enable SID broadcast: 0-no, 1-yes.

bySecurityMode

Safety mode: 0-unencrypted, 1-WPA, 2-WPA2. Currently, only 2 is supported.

byAlgorithmType

Encryption mode: 1-TKIP, 2- AES.

szSSID

SSID information.

szPassWord

Password.

byDefaultPassword

Whether it is the default password: 0-no, 1-yes.

byWifiApModeType

Wireless hotspot mode: 0-disabled, 1-enabled, 2-auto.

byRes

Reserved, set to 0.

A.281 NET_DVR_WORKSTATE_V40

Device Working Status Structure

Table A-21 Device Status Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
dwDeviceStatic	DWORD	Device status: 0-normal; 1-too high CPU usage, it exceeds 85%; 2-hardware error (e.g., serial port exception).
struHardDiskStatic	<u>NET_DVR_DISKSTATE</u> []	HDD status, up to 33 HDDs' information can be obtained for once.
struChanStatic	<u>NET_DVR_CHANNELSTATE_V30</u> []	Channel status, sorting from front to end. The maximum length is 512 bytes.
dwHasAlarmInStatic	DWORD[]	Alarm input with alarm triggered, indicated by bit, sort by the subscript value, 0xffffffff-the current or following value is invalid. The maximum length is 4128 bytes.
dwHasAlarmOutStatic	DWORD[]	Alarm output with alarm triggered, indicated by bit, sort by the subscript value, 0xffffffff-the current or following value is invalid. The maximum length is 4128 bytes.
dwLocalDisplay	DWORD	Local display status: 0-normal, 1-exception.
byAudioInChanStatus	BYTE[]	Audio channel status, bit0-audio channel 1, bit1-audio channel 2, array element value: 0-unused, 1-using, 0xff-invalid. The maximum length is 2 bytes.
byRes1	BYTE[]	Reserved, set to 0. The maximum length is 2 bytes.
fHumidity	float	Humidity obtained by the sensor, value range: [0.0,100.0].

Member	Data Type	Description
fTemperature	float	Temperature obtained by the sensor, value range: [-20.0,90.0].
byRes	BYTE[]	Reserved, set to 0. The maximum length is 116 bytes.

A.282 NET_DVR_XML_CONFIG_INPUT

Input Parameter Structure of Message Transmission API (NET_DVR_STDXMLConfig)

Member	Data Type	Description
dwSize	DWORD	Structure size.
lpRequestUrl	void*	Request URL (command) for implement different functions, and it is in string format.
dwRequestUrlLen	DWORD	Request URL size.
lpInBuffer	void*	Buffer for storing input parameters (request messages), see the input content details structure in <u>NET_DVR_MIME_UNIT</u> .
dwInBufferSize	DWORD	Input buffer size.
dwRecvTimeOut	DWORD	Receiving timeout, unit: ms, 0-5000ms (default).
byForceEncrpt	BYTE	Whether to enable force encryption (the messages will be encrypted by AES algorithm for transmission): 0-no, 1-yes.
byNumOfMultiPart	BYTE	Number of message segments: 0-invalid; other values-number of message segments, which is transmitted by the parameter lpInBuffer in the structure <u>NET_DVR_MIME_UNIT</u> .
byRes	Array of BYTE	Reserved, set to 0.

Related API

[NET_DVR_STDXMLConfig](#)

A.283 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 byNumOfMultiPart 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.284 NET_IPC_AUXALARM_RESULT

Structure about auxiliary (PIR or wireless) alarm parameters.

Structure Definition

```
struct{
    DWORD      dwSize;
    DWORD      dwChannel;
    BYTE       byAlarmType;
    BYTE       byRes1[3];
    NET_IPC_AUXALARM_UPLOAD_UNION struAuxAlarm;
```

```
    BYTE           byRes[64];  
}NET_IPC_AUXALARM_RESULT, *LPNET_IPC_AUXALARM_RESULT;
```

Members

dwSize

Structure size.

dwChannel

Channel No.

byAlarmType

Alarm device type, see the definition below:

```
enum _IPC_AUX_ALARM_TYPE_ {  
    IPC_AUXALARM_UNKNOW = 0,  
    IPC_AUXALARM_PIR = 1,  
    IPC_AUXALARM_WIRELESS = 2,  
    IPC_AUXALARM_CALLHELP = 3,  
}IPC_AUX_ALARM_TYPE
```

IPC_AUXALARM_UNKNOW

Unknown.

IPC_AUXALARM_PIR

PIR alarm.

IPC_AUXALARM_WIRELESS

Wireless alarm.

IPC_AUXALARM_CALLHELP

Emergency alarm.

byRes1

Reserved.

struAuxAlarm

Alarm result, see details in the structure [NET_IPC_AUXALARM_UPLOAD_UNION](#).

byRes

Reserved.

A.285 NET_IPC_AUXALARM_UPLOAD_UNION

Alarm result parameter union.

Structure Definition

```
union{  
    DWORD           uLen[66];
```

```
NET_IPC_PIR_ALARMCFG      struPIRAlarm;
NET_IPC_SINGLE_WIRELESS_ALARMCFG  struWirelessAlarm;
NET_IPC_CALLHELP_ALARMCFG      struCallHelpAlarm;
}NET_IPC_AUXALARM_UPLOAD_UNION, *LPNET_IPC_AUXALARM_UPLOAD_UNION;
```

Members

uLen

Union size.

struPIRAlarm

PIR alarm parameter, see details in the structure [NET_IPC_PIR_ALARMCFG](#).

struWirelessAlarm

Wireless alarm parameter, see details in the structure

[NET_IPC_SINGLE_WIRELESS_ALARMCFG](#).

struCallHelpAlarm

Emergency alarm parameter, see details in the structure [NET_IPC_CALLHELP_ALARMCFG](#).

See Also

[NET_IPC_AUXALARM_RESULT](#)

A.286 NET_IPC_CALLHELP_ALARMCFG

Emergency alarm configuration structure.

Structure Definition

```
struct{
    BYTE          byAlarmHandle;
    BYTE          byRes1[3];
    NET_DVR_HANDLEEXCEPTION_V41  struAlarmHandleType;
    BYTE          byRelRecordChan[MAX_CHANNUM_V30/*64*/];
    BYTE          byRes[64];
}NET_IPC_CALLHELP_ALARMCFG, *LPNET_IPC_CALLHELP_ALARMCFG;
```

Members

byAlarmHandle

Whether to handle the alarm: 0-no, 1-yes.

byRes1

Reserved, set to 0.

struAlarmHandleType

Handling type, see details in the structure [NET_DVR_HANDLEEXCEPTION_V41](#).

byRes

Reserved, set to 0.

A.287 NET_IPC_PIR_ALARMCFG

PIR alarm parameter structure.

Structure Definition

```
struct{
    BYTE          byAlarmName[NAME_LEN/*32*/];
    BYTE          byAlarmHandle;
    BYTE          byRes1[3];
    NET_DVR_HANDLEEXCEPTION_V41  struAlarmHandleType;
    BYTE          byRelRecordChan[MAX_CHANNUM_V30/*64*/];
    BYTE          byRes[64];
}NET_IPC_PIR_ALARMCFG, *LPNET_IPC_PIR_ALARMCFG;
```

Members

byAlarmName

Alarm name.

byAlarmHandle

Whether to handle the alarm: 0-no, 1-yes.

byRes1

Reserved.

struAlarmHandleType

Handling type, see details in the structure [NET_DVR_HANDLEEXCEPTION_V41](#).

byRelRecordChan

Recording channel triggered by the alarm. The channel will be triggered to start recording when **byRelRecordChan** is set to 1. For example, if **byRelRecordChan[0]** is set to 1, it indicates that channel 1 is triggered to record; if **byRelRecordChan[1]** is set to 1, it indicates that channel 2 is triggered to record, and so on.

byRes

Reserved.

A.288 NET_IPC_SINGLE_WIRELESS_ALARMCFG

Single wireless alarm parameter structure.

Structure Definition

```
struct{
    BYTE      byAlarmName[NAME_LEN/*32*/];
    BYTE      byAlarmHandle;
    BYTE      byID;
    BYTE      byRes1[2];
    NET_DVR_HANDLEEXCEPTION_V41 struAlarmHandleType;
    BYTE      byRelRecordChan[MAX_CHANNUM_V30/*64*/];
    BYTE      byRes[32];
}NET_IPC_SINGLE_WIRELESS_ALARMCFG, *LPNET_IPC_SINGLE_WIRELESS_ALARMCFG;
```

Members

byAlarmName

Alarm name.

byAlarmHandle

Whether to handle the alarm: 0-no, 1-yes.

byID

Wireless alarm ID, ranging from 1 to 8.

byRes1

Reserved.

struAlarmHandleType

Handling type, see details in the structure [NET_DVR_HANDLEEXCEPTION_V41](#).

byRelRecordChan

Recording channel triggered by the alarm. The channel will be triggered to start recording when **byRelRecordChan** is set to 1. For example, if **byRelRecordChan[0]** is set to 1, it indicates that channel 1 is triggered to record; if **byRelRecordChan[1]** is set to 1, it indicates that channel 2 is triggered to record, and so on.

byRes

Reserved.

A.289 NET_PTZ_INFO_EX

Extended PTZ parameters structure.

Structure Definition

```
struct{
    float fPan;
    float fTilt;
    float fVisibleZoom;
```

```

DWORD dwVisibleFocus;
float fThermalZoom;
DWORD dwThermalFocus;
}NET_PTZ_INFO_EX, *LPNET_PTZ_INFO_EX;

```

Members

fPan

Value of panning parameter, range: [0,360.000], which is accurate to 3 decimal places.

fTilt

Value of tilting parameter, range: [-90.000,90.000], which is accurate to 3 decimal places.

fVisibleZoom

Value of zooming parameter for visible light, range: [0, 10000.0], which is accurate to 1 decimal place.

dwVisibleFocus

Value of focus parameter for visible light.

fThermalZoom

Value of zooming parameter for thermal imaging, range: [0, 10000.0], which is accurate to 1 decimal place.

dwThermalFocus

Value of focus parameter for thermal imaging.

A.290 NET_VCA_AUDIO_ABNORMAL

Table A-22 Structure about Detection Parameters for Sudden Increase of Sound Intensity

Member	Data Type	Description
wDecibel	WORD	Sound intensity.
bySensitivity	BYTE	Sensitivity, value range: [1,100].
byAudioMode	BYTE	Sound detection mode: 0-enable sensitivity detection, 1-enable decibel threshold detection, 2-enable both.
byEnable	BYTE	Whether to enable sudden increase of sound intensity detection: 0-no, 1-yes.

Member	Data Type	Description
byThreshold	BYTE	Sound intensity threshold, value range: [0,100].
byRes	Array [BYTE]	Reserved, set to 0. The maximum size is 54 bytes.

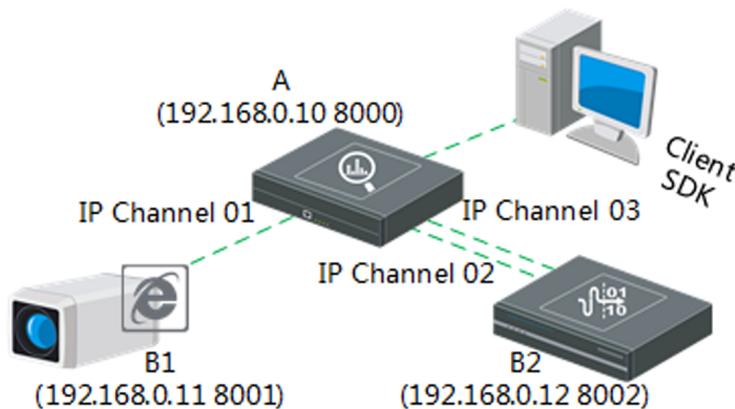
A.291 NET_VCA_DEV_INFO

Structure About Camera Information

Member	Data Type	Description
struDevIP	NET_DVR_IPADDR_UNION	Device IP address
wPort	WORD	Device port No.
byChannel	BYTE	Device channel No.
bylvmsChannel	BYTE	Device channel No. for the HCNetSDK to access the device.

Remarks

When accessing to device A, see the following figure for details.



- When the analog channel alarm of connected device (IPC, DVR, DVS, iVMS) is triggered, **struDevIP** and **wPort** are the IP address and port No. of connected device; **byChannel** and **bylvmsChannel** are both the alarm analog channel No.
- When the digital channel (IP channel) alarm of connected device (HDVR, NVR, iVMS) is triggered, **struDevIP**, **wPort** and **byChannel** are the IP address, port No., and channel No. of connected

device, respectively; **bylvmsChannel** is the digital channel. In the above figure, the channel No.1 of device B1 and the channel No.1, 2 of device B2 are used as channel No.1, 2, 3 of the connected device A; **struDevIP**, **wPort**, and **byChannel** are the IP address, port No. and channel No. of B1 or B2; **bylvmsChannel** is the digital channel No. of device A. E.g., if the behavior analysis alarm of channel No.2 of device B2 is triggered, **struDevIP** is 192.168.0.12, **wPort** is 8002, **byChannel** is 2 and **bylvmsChannel** is 3 in the received alarm message.

A.292 NET_VCA_MASK_REGION

Shielded area structure.

Structure Definition

```
struct{
    BYTE      byEnable;
    BYTE      byRes[3];
    NET_VCA_POLYGON struPolygon;
}NET_VCA_MASK_REGION, *LPNET_VCA_MASK_REGION;
```

Members

byEnable

Enable or not: 0-no, non 0-yes

byRes

Reserved, and set to 0

struPolygon

Shielded polygon, see details in

A.293 NET_VCA_MASK_REGION_LIST

Shielded areas parameters structure

Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE      byRes[4];
    NET_VCA_MASK_REGION struMask[MAX_MASK_REGION_NUM/*4*/];
}NET_VCA_MASK_REGION_LIST, *LPNET_VCA_MASK_REGION_LIST;
```

Members

dwSize

Structure size

byRes

Reserved, and set to 0

struMask

Shielded area array, see details in

A.294 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.295 NET_VCA_POLYGON

Polygon Coordinate Parameter Structure

Member	Data Type	Description
dwPointNum	DWORD	Valid point (larger than or equal 3), if 3 points are in the same line, as well as line-cross, region is invalid.
struPos	Array of <u>NET_VCA_POINT</u>	Boundary point of polygon, up to 10 points

A.296 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.

A.297 STREAM_FRAME_INFO_S

Stream Frame Data Structure

Member	Data Type	Description
u32MagicNo	UINT	ASCLL code of "FRMI".
u32HeaderSize	UINT	Structure size.
u32StreamType	UINT	Data types, which are enumerated in STREAM_TYPE_E .
u32StreamLen	UINT	Data size.
/	union	Real-time data union.
/	union	Additional information union.
res	Array[UINT]	Reserved. The maximum length is 12 bytes.
u32CrcVal	UINT	Verification code of structure, which is used to verify the data before the structure.

Table A-23 RT Data Union

Member	Data Type	Description
stRTDataInfo;	STREAM_RT_DATA_INFO_S	Real-time data details.
u32Res1	Array[UINT]	Reserved. The maximum length is 12 bytes.

Table A-24 Additional Information Union

Member	Data Type	Description
stFsSuppleInfo	STREAM_FS_SUPPLE_INFO_S	Additional details.
u32Res2	Array[UINT]	Reserved. The maximum length is 4 bytes.

A.298 STREAM_FS_SUPPLE_INFO_S

Additional Details Structure

Member	Data Type	Description
u32TmDataMode	UINT	Data size mode: 0: 4-byte, 1: 2-byte.
u32TmScale	UINT	Temperature scale.
u32TmOffset	UINT	Temperature offset. Currently, the value is 0.
res	UINT	One byte reserved.

A.299 STREAM_RT_DATA_INFO_S

Structure about Real-Time Data Details

Member	Data Type	Description
u32RTDataType	UINT	Data type, 1: 14-bit raw data, 2: temperature data, 3: YUV data.
u32FrmNum	UINT	Number of frames.

Member	Data Type	Description
u32StdStamp	UINT	Relative time stamp.
stTime	<u>DATE_TIME</u>	Absolute time stamp.
u32Width	UINT	Width.
u32Height	UINT	Height.
u32Len	UINT	Length.
u32Fps	UINT	Frame rate.
u32Chan	UINT	Channel No.

Appendix B. Enumeration

B.1 BITRATE_ENCODE_INDEX

Enumerate the audio encoding bit rates.

Enumeration Definition

```
typedef enum _BITRATE_ENCODE_INDEX_{
    BITRATE_ENCODE_def      = 0,
    BITRATE_ENCODE_8kps     = 1,
    BITRATE_ENCODE_16kps    = 2,
    BITRATE_ENCODE_32kps    = 3,
    BITRATE_ENCODE_64kps    = 4,
    BITRATE_ENCODE_128kps   = 5,
    BITRATE_ENCODE_192kps   = 6,
    BITRATE_ENCODE_40kps    = 7,
    BITRATE_ENCODE_48kps    = 8,
    BITRATE_ENCODE_56kps    = 9,
    BITRATE_ENCODE_80kps    = 10,
    BITRATE_ENCODE_96kps    = 11,
    BITRATE_ENCODE_112kps   = 12,
    BITRATE_ENCODE_144kps   = 13,
    BITRATE_ENCODE_160kps   = 14,
    BITRATE_ENCODE_224kps   = 15,
    BITRATE_ENCODE_256kps   = 16,
    BITRATE_ENCODE_320kps   = 17
}BITRATE_ENCODE_INDEX;
```

Members

BITRATE_ENCODE_def

The default audio encoding bit rate, for network camera with version 5.1.0, it is 64Kbps.

BITRATE_ENCODE_8kps

The audio encoding bit rate is 8Kbps.

BITRATE_ENCODE_16kps

The audio encoding bit rate is 16Kbps.

BITRATE_ENCODE_32kps

The audio encoding bit rate is 32Kbps.

BITRATE_ENCODE_64kps

The audio encoding bit rate is 64Kbps.

BITRATE_ENCODE_128kps

The audio encoding bit rate is 128Kbps.

BITRATE_ENCODE_192kps

The audio encoding bit rate is 192Kbps.

BITRATE_ENCODE_40kps

The audio encoding bit rate is 40Kbps.

BITRATE_ENCODE_48kps

The audio encoding bit rate is 48Kbps.

BITRATE_ENCODE_56kps

The audio encoding bit rate is 56Kbps.

BITRATE_ENCODE_80kps

The audio encoding bit rate is 80Kbps.

BITRATE_ENCODE_96kps

The audio encoding bit rate is 96Kbps.

BITRATE_ENCODE_112kps

The audio encoding bit rate is 112Kbps.

BITRATE_ENCODE_144kps

The audio encoding bit rate is 144Kbps.

BITRATE_ENCODE_160kps

The audio encoding bit rate is 160Kbps.

BITRATE_ENCODE_224kps

The audio encoding bit rate is 224Kbps.

BITRATE_ENCODE_256kps

The audio encoding bit rate is 256Kbps.

BITRATE_ENCODE_320kps

The audio encoding bit rate is 320Kbps.

B.2 COUNTRY_INDEX

Enumerate country names.

Enumeration Definition

```
enum{
    COUNTRY_NONSUPPORT = 0,
    COUNTRY_CZE        = 1,
    COUNTRY_FRA        = 2,
    COUNTRY_DEU        = 3,
    COUNTRY_ESP        = 4,
    COUNTRY_ITA        = 5,
    COUNTRY_NLD        = 6,
```

```
COUNTRY_POL      = 7,  
COUNTRY_SVK      = 8,  
COUNTRY_BLR      = 9,  
COUNTRY_MDA      = 10,  
COUNTRY_RUS      = 11,  
COUNTRY_UKR      = 12,  
COUNTRY_BEL      = 13,  
COUNTRY_BGR      = 14,  
COUNTRY_DNK      = 15,  
COUNTRY_FIN      = 16,  
COUNTRY_GBR      = 17,  
COUNTRY_GRC      = 18,  
COUNTRY_HRV      = 19,  
COUNTRY_HUN      = 20,  
COUNTRY_ISR      = 21,  
COUNTRY_LUX      = 22,  
COUNTRY_MKD      = 23,  
COUNTRY_NOR      = 24,  
COUNTRY_PRT      = 25,  
COUNTRY_ROU      = 26,  
COUNTRY_SRБ      = 27,  
COUNTRY_AZE      = 28,  
COUNTRY_GEO      = 29,  
COUNTRY_KAZ      = 30,  
COUNTRY_LTU      = 31,  
COUNTRY_TKM      = 32,  
COUNTRY_UZB      = 33,  
COUNTRY_LVA      = 34,  
COUNTRY_EST      = 35,  
COUNTRY_ALB      = 36,  
COUNTRY_AUT      = 37,  
COUNTRY_BIH      = 38,  
COUNTRY_IRL      = 39,  
COUNTRY_ISL      = 40,  
COUNTRY_VAT      = 41,  
COUNTRY_MLT      = 42,  
COUNTRY_SWE      = 43,  
COUNTRY_CHE      = 44,  
COUNTRY_CYP      = 45,  
COUNTRY_TUR      = 46,  
COUNTRY SVN     = 47,  
COUNTRY_MTG      = 48,  
COUNTRY_KOV      = 49,  
COUNTRY_ADR      = 50,  
COUNTRY_ARM      = 51,  
COUNTRY_MON      = 52,  
COUNTRY_LIE      = 53,  
COUNTRY_SMO      = 54,  
COUNTRY_RES1     = 55,  
COUNTRY_RES2     = 56,  
COUNTRY_RES3     = 57,  
COUNTRY_RES4     = 58,
```

COUNTRY_CHI	= 59,
COUNTRY_IBN	= 60,
COUNTRY_SKR	= 61,
COUNTRY_LEB	= 62,
COUNTRY_NEP	= 63,
COUNTRY_THA	= 64,
COUNTRY_PAK	= 65,
COUNTRY_EMI	= 66,
COUNTRY_BHU	= 67,
COUNTRY_OMA	= 68,
COUNTRY_KOR	= 69,
COUNTRY_PHI	= 70,
COUNTRY_CAM	= 71,
COUNTRY_QAT	= 72,
COUNTRY_KYR	= 73,
COUNTRY_MAL	= 74,
COUNTRY_MLY	= 75,
COUNTRY_MOG	= 76,
COUNTRY_ARA	= 77,
COUNTRY_BRU	= 78,
COUNTRY_LA0	= 79,
COUNTRY_JAP	= 80,
COUNTRY_RES19	= 81,
COUNTRY_PAS	= 82,
COUNTRY_TAJ	= 83,
COUNTRY_KUW	= 84,
COUNTRY_SYR	= 85,
COUNTRY_IND	= 86,
COUNTRY_ISA	= 87,
COUNTRY_AFG	= 88,
COUNTRY_LAN	= 89,
COUNTRY_IRQ	= 90,
COUNTRY_VIE	= 91,
COUNTRY_IRA	= 92,
COUNTRY_YEM	= 93,
COUNTRY_JOR	= 94,
COUNTRY_BUR	= 95,
COUNTRY_SIK	= 96,
COUNTRY_BAN	= 97,
COUNTRY_SGA	= 98,
COUNTRY_EAT	= 99,
COUNTRY_RES5	= 100,
COUNTRY_RES6	= 101,
COUNTRY_RES7	= 102,
COUNTRY_RES8	= 103,
COUNTRY_EGT	= 104,
COUNTRY_liy	= 105,
COUNTRY_SUA	= 106,
COUNTRY_TUN	= 107,
COUNTRY_ALG	= 108,
COUNTRY_MCC	= 109,
COUNTRY_ETH	= 110,

```
COUNTRY_ERI      = 111,
COUNTRY_SDE      = 112,
COUNTRY_DJI      = 113,
COUNTRY_KEN      = 114,
COUNTRY_TAI      = 115,
COUNTRY_UGA      = 116,
COUNTRY_RWA      = 117,
COUNTRY_BUD      = 118,
COUNTRY_SEY      = 119,
COUNTRY_CHA      = 120,
COUNTRY_CEA      = 121,
COUNTRY_CON      = 122,
COUNTRY_EQG      = 123,
COUNTRY_GAB      = 124,
COUNTRY_TCO      = 125,
COUNTRY_DRC      = 126,
COUNTRY_STP      = 127,
COUNTRY_MAN      = 128,
COUNTRY_WSA      = 129,
COUNTRY_SEL      = 130,
COUNTRY_TGA      = 131,
COUNTRY_MAI      = 132,
COUNTRY_BUF      = 133,
COUNTRY_GUI      = 134,
COUNTRY_GUB      = 135,
COUNTRY_CAV      = 136,
COUNTRY_SLE      = 137,
COUNTRY_LIR      = 138,
COUNTRY_IVC      = 139,
COUNTRY_GHA      = 140,
COUNTRY_TGO      = 141,
COUNTRY_BEN      = 142,
COUNTRY_NIG      = 143,
COUNTRY_ZAB      = 144,
COUNTRY_ANG      = 145,
COUNTRY_ZBE      = 146,
COUNTRY_MAW      = 147,
COUNTRY_MOQ      = 148,
COUNTRY_BOT      = 149,
COUNTRY_NAM      = 150,
COUNTRY_SAF      = 151,
COUNTRY_SWD      = 152,
COUNTRY_LES      = 153,
COUNTRY_MAG      = 154,
COUNTRY_UOC      = 155,
COUNTRY_MAT      = 156,
COUNTRY_NGE      = 157,
COUNTRY_SSD      = 158,
COUNTRY_SAH      = 159,
COUNTRY_MYT      = 160,
COUNTRY_REN      = 161,
COUNTRY_CAI      = 162,
```

```
COUNTRY_AZO      = 163,
COUNTRY_MAD      = 164,
COUNTRY_RES9      = 165,
COUNTRY_RES10     = 166,
COUNTRY_RES11     = 167,
COUNTRY_RES12     = 168,
COUNTRY_CAD      = 169,
COUNTRY_GRE      = 170,
COUNTRY_PIE      = 171,
COUNTRY_USA      = 172,
COUNTRY_BER      = 173,
COUNTRY_MEX      = 174,
COUNTRY_GUA      = 175,
COUNTRY_BLI      = 176,
COUNTRY_SAR      = 177,
COUNTRY_HOR      = 178,
COUNTRY_NIC      = 179,
COUNTRY_COR      = 180,
COUNTRY_PAN      = 181,
COUNTRY_TBM      = 182,
COUNTRY_TCI      = 183,
COUNTRY_CUB      = 184,
COUNTRY_JAM      = 185,
COUNTRY_CAY      = 186,
COUNTRY_HAT      = 187,
COUNTRY_TDO      = 188,
COUNTRY_PUR      = 189,
COUNTRY_VIL      = 190,
COUNTRY_BVI      = 191,
COUNTRY_ATV      = 192,
COUNTRY_ANB      = 193,
COUNTRY_CSM      = 194,
COUNTRY_ACY      = 195,
COUNTRY_SBY      = 196,
COUNTRY_SKN      = 197,
COUNTRY_MOT      = 198,
COUNTRY_GLP      = 199,
COUNTRY_DOM      = 200,
COUNTRY_MTE      = 201,
COUNTRY_LUC      = 202,
COUNTRY_SVG      = 203,
COUNTRY_GRD      = 204,
COUNTRY_BAR      = 205,
COUNTRY_TRT      = 206,
COUNTRY_CUR      = 207,
COUNTRY_ARB      = 208,
COUNTRY_NEA      = 209,
COUNTRY_COL      = 210,
COUNTRY_VEN      = 211,
COUNTRY_GUY      = 212,
COUNTRY_SUR      = 213,
COUNTRY_FRN      = 214,
```

```
COUNTRY_ECU      = 215,
COUNTRY_PER      = 216,
COUNTRY_BOL      = 217,
COUNTRY_PAR      = 218,
COUNTRY_CLE      = 219,
COUNTRY_BRA      = 220,
COUNTRY_UGY      = 221,
COUNTRY_ARG      = 222,
COUNTRY_RES13     = 223,
COUNTRY_RES14     = 224,
COUNTRY_RES15     = 225,
COUNTRY_RES16     = 226,
COUNTRY_ATN      = 227,
COUNTRY_NED      = 228,
COUNTRY_PNG      = 229,
COUNTRY_SAN      = 230,
COUNTRY_VAU      = 231,
COUNTRY_NCN      = 232,
COUNTRY_PAU      = 233,
COUNTRY_FSM      = 234,
COUNTRY_MRI      = 235,
COUNTRY_CNM      = 236,
COUNTRY_TEG      = 237,
COUNTRY_NUR      = 238,
COUNTRY_KIB      = 239,
COUNTRY_FID      = 240,
COUNTRY_TNG      = 241,
COUNTRY_TUV      = 242,
COUNTRY_WEF      = 243,
COUNTRY_TIS      = 244,
COUNTRY_EAS      = 245,
COUNTRY_TOE      = 246,
COUNTRY_NUE      = 247,
COUNTRY_TCD      = 248,
COUNTRY_PFP      = 249,
COUNTRY_PID      = 250,
COUNTRY_HAW      = 251,
COUNTRY_RES17     = 252,
COUNTRY_RES18     = 253,
COUNTRY_UNRECOGNIZED = 0xfe,
COUNTRY_ALL       = 0xff,
COUNTRY_INVALID   = 0xfd
}COUNTRY_INDEX
```

Members

COUNTRY_NONSUPPORT

Not support

COUNTRY_CZE

Czech Republic

COUNTRY_DEU

Germany

COUNTRY_ESP

Spain

COUNTRY_ITA

Italy

COUNTRY_NLD

Netherlands

COUNTRY_POL

Poland

COUNTRY_SVK

Slovakia

COUNTRY_BLR

Belorussia

COUNTRY_MDA

Moldova

COUNTRY_RUS

Russia

COUNTRY_UKR

Ukraine

COUNTRY_BEL

Belgium

COUNTRY_BGR

Bulgaria

COUNTRY_DNK

Denmark

COUNTRY_FIN

Finland

COUNTRY_GBR

United Kingdom

COUNTRY_GRC

Greece

COUNTRY_HRV

Croatia

COUNTRY_HUN

Hungary

COUNTRY_ISR

Israel

COUNTRY_LUX

Luxembourg

COUNTRY_MKD

Macedonia

COUNTRY_NOR

Norway

COUNTRY_PRT

Portugal

COUNTRY_ROU

Romania

COUNTRY_SRB

Serbia

COUNTRY_AZE

Azerbaijan

COUNTRY_GEO

Georgia

COUNTRY_KAZ

Kazakhstan

COUNTRY_LTU

Lithuania

COUNTRY_TKM

Turkmenistan

COUNTRY_UZB

Uzbekistan

COUNTRY_LVA

Latvia

COUNTRY_EST

Estonia

COUNTRY_ALB

Albania

COUNTRY_AUT

Austria

COUNTRY_BIH

Bosnia and Herzegovina

COUNTRY_IRL

Ireland

COUNTRY_ISL

Iceland

COUNTRY_VAT

Vatican

COUNTRY_MLT

Malta

COUNTRY_SWE

Sweden

COUNTRY_CHE

Switzerland

COUNTRY_CYP

Cyprus

COUNTRY_TUR

Turkey

COUNTRY SVN

Slovenia

COUNTRY_MTG

Montenegro

COUNTRY_KOV

Kosovo

COUNTRY_ADR

Andorra

COUNTRY_ARM

Armenia

COUNTRY_MON

Monaco

COUNTRY_LIE

Liechtenstein

COUNTRY_SMO

San Marino

COUNTRY_RES1

Reserved

COUNTRY_RES2

Reserved

COUNTRY_RES3

Reserved

COUNTRY_RES4

Reserved /* Asia, 48 countries, in which Cyprus is located on the border of Europe and Asia*/

COUNTRY_CHI

China

COUNTRY_IBN

In bahrain

COUNTRY_SKR

South Korea

COUNTRY_LEB

Lebanon

COUNTRY_NEA

Nepal

COUNTRY_THA

Thailand

COUNTRY_PAK

Pakistan

COUNTRY_EMI

The united Arab emirates

COUNTRY_BHU

Bhutan

COUNTRY_OMA

Oman

COUNTRY_KOR

North Korea

COUNTRY_PHI

The Philippines

COUNTRY_CAM

Cambodia

COUNTRY_QAT

Qatar

COUNTRY_KYR

Kyrgyzstan

COUNTRY_MAL

The maldives

COUNTRY_MLY

Malaysia

COUNTRY_MOG

Mongolia

COUNTRY_ARA

Saudi Arabia

COUNTRY_BRU

brunei

COUNTRY_LA0

Laos

COUNTRY_JAP

Japan

COUNTRY_RES19

Reserved

COUNTRY_PAS

Palestinian state

COUNTRY_TAJ

Tajikistan

COUNTRY_KUW

Kuwait

COUNTRY_SYR

Syria

COUNTRY_IND

India

COUNTRY_ISA

Indonesia

COUNTRY_AFG

Afghanistan

COUNTRY_LAN

Sri Lanka

COUNTRY_IRQ

Iraq

COUNTRY_VIE

Vietnam

COUNTRY_IRA

Iran

COUNTRY_YEM

yemen

COUNTRY_JOR

Jordan

COUNTRY_BUR

Burma

COUNTRY_SIK

Sikkim

COUNTRY_BAN

Bangladesh

COUNTRY_SGA

Singapore

COUNTRY_EAT

East Timor

COUNTRY_RES5

Reserved

COUNTRY_RES6

Reserved

COUNTRY_RES7

Reserved

COUNTRY_RES8

Reserved /*Africa, 60 countries and regions*/

COUNTRY_EGT

Egypt

COUNTRY_LIY

Libya

COUNTRY_SUA

Sudan

COUNTRY_TUN

Tunisia

COUNTRY_ALG

Algeria

COUNTRY_MCC

Morocco

COUNTRY_ETH

Ethiopia

COUNTRY_ERI

Eritrea

COUNTRY_SDE

Somalia Democratic

COUNTRY_DJI

Djibouti

COUNTRY_KEN

Kenya

COUNTRY_TAI

Tanzania

COUNTRY_UGA

Uganda

COUNTRY_RWA

Rwanda

COUNTRY_BUD

Burundi

COUNTRY_SEY

Seychelles

COUNTRY_CHA

Chad

COUNTRY_CEA

Central African

COUNTRY_CON

Cameroon

COUNTRY_EQG

Equatorial Guinea

COUNTRY_GAB

Gabon

COUNTRY_TCO

the Congo

COUNTRY_DRC

Democratic Republic of the Congo

COUNTRY_STP

Sao Tome and Principe

COUNTRY_MAN

Mauritania

COUNTRY_WSA

Western Sahara

COUNTRY_SEL

Senega

COUNTRY_TGA

the Gambia

COUNTRY_MAI

Mali

COUNTRY_BUF

Burkina Faso

COUNTRY_GUI

Guinea

COUNTRY_GUB

Guinea-Bissau

COUNTRY_CAV

Cape Verde

COUNTRY_SLE

Sierra Leone

COUNTRY_LIR

Liberia

COUNTRY_IVC

Ivory Coast

COUNTRY_GHA

Ghana

COUNTRY_TGO

Togo

COUNTRY_BEN

Benin

COUNTRY_NIG

Niger

COUNTRY_ZAB

Zambia

COUNTRY_ANG

Angola

COUNTRY_ZBE

Zimbabwe

COUNTRY_MAW

Malawi

COUNTRY_MOQ

Mozambique

COUNTRY_BOT

Botswana

COUNTRY_NAM

Namibia

COUNTRY_SAF

South Africa

COUNTRY_SWD

Swaziland

COUNTRY_LES

Lesotho

COUNTRY_MAG

Madagascar

COUNTRY_UOC

Union of Comoros

COUNTRY_MAT

Mauritius

COUNTRY_NGE

Nigeria

COUNTRY_SSD

South Sudan

COUNTRY_SAH

Saint Helena

COUNTRY_MYT

Mayotte

COUNTRY_REN

Reunion

COUNTRY_CAI

Canary Islands

COUNTRY_AZO

AZORES

COUNTRY_MAD

Madeira

COUNTRY_RES9

Reserved

COUNTRY_RES10

Reserved

COUNTRY_RES11

Reserved

COUNTRY_RES12

Reserved /*America, 55 countries and regions*/

COUNTRY_CAD

Canada

COUNTRY_GRE

Greenland Nuuk

COUNTRY_PIE

/Pierre and Miquelon

COUNTRY_USA

United States

COUNTRY_BER

Bermuda

COUNTRY_MEX

Mexico

COUNTRY_GUA

Guatemala

COUNTRY_BLI

Belize

COUNTRY_SAR

El Salvador

COUNTRY_HOR

Honduras

COUNTRY_NIC

Nicaragua

COUNTRY_COR

Costa Rica

COUNTRY_PAN

Panama

COUNTRY_TBM

The Bahamas

COUNTRY_TCI

The Turks and Caicos Islands

COUNTRY_CUB

Cuba

COUNTRY_JAM

Jamaica

COUNTRY_CAY

Cayman Islands

COUNTRY_HAT

Haiti

COUNTRY_TDO

The Dominican

COUNTRY_PUR

Puerto Rico

COUNTRY_VIL

The United States Virgin Islands

COUNTRY_BVI

The British Virgin Islands

COUNTRY_ATV

Anguilla The Valley

COUNTRY_ANB

Antigua and Barbuda

COUNTRY_CSM

Collectivite de Saint-Martin

COUNTRY_ACY

Autonomous country

COUNTRY_SBY

Saint-Barthelemy

COUNTRY_SKN

Saint Kitts and Nevis

COUNTRY_MOT

Montserrat

COUNTRY_GLP

Guadeloupe

COUNTRY_DOM

Dominica

COUNTRY_MTE

Martinique

COUNTRY_LUC

St. Lucia

COUNTRY_SVG

Saint Vincent and the Grenadines

COUNTRY_GRD

Grenada

COUNTRY_BAR

Barbados

COUNTRY_TRT

Trinidad and Tobago

COUNTRY_CUR

Curaao

COUNTRY_ARB

Aruba

COUNTRY_NEA

Netherlands Antilles

COUNTRY_COL

Colombia

COUNTRY_VEN

Venezuela

COUNTRY_GUY

Guyana

COUNTRY_SUR

Suriname

COUNTRY_FRN

Guyane Francaise

COUNTRY_ECU

Ecuador

COUNTRY_PER

Peru

COUNTRY_BOL

Bolivia

COUNTRY_PAR

Paraguay

COUNTRY_CLE

Chile

COUNTRY_BRA

Brazil

COUNTRY_UGY

Uruguay

COUNTRY_ARG

Argentina

COUNTRY_RES13

Reserved

COUNTRY_RES14

Reserved

COUNTRY_RES15

Reserved

COUNTRY_RES16

Reserved /*Oceania, 25 countries and regions*/

COUNTRY_ATN

Australien

COUNTRY_NED

Neuseeland

COUNTRY_PNG

Papua New Guinea

COUNTRY_SAN

Salomonen

COUNTRY_VAU

Vanuatu

COUNTRY_NCN

New Caledonia

COUNTRY_PAU

Palau

COUNTRY_FSM

Federated States of Micronesia

COUNTRY_MRI

Marshall Island

COUNTRY_CNM

Commonwealth of the Northern Mariana Islands

COUNTRY_TEG

The Territory of Guahan

COUNTRY_NUR

Nauru

COUNTRY_KIB

Kiribati

COUNTRY_FID

Fidschi

COUNTRY_TNG

Tonga

COUNTRY_TUV

Tuvalu

COUNTRY_WEF

Wallis et Futuna

COUNTRY_TIS

The Independent State of Samoa

COUNTRY_EAS

Eastern Samoa

COUNTRY_TOE

Tokelau

COUNTRY_NUE

Niue

COUNTRY_TCD

The Cook Islands

COUNTRY_PFP

Polynesie franaise French Polynesia

COUNTRY_PID

Pitcairn Islands

COUNTRY_HAW

Hawaii State

COUNTRY_RES17

Reserved

COUNTRY_RES18

Reserved

COUNTRY_UNRECOGNIZED

Unrecognized

COUNTRY_ALL

ALL

COUNTRY_INVALID

byCountry is invalid and you should use **CRIndex**, see [CR_INDEX](#) for details.

B.3 CR_INDEX

Enumeration about country or region index.

Enumeration Definition

```
enum{
    CR_NONSUPPORT = 0,
    CR_CZE = 1,
    CR_FRA = 2,
    CR_DEU = 3,
    CR_ESP = 4,
    CR_ITA = 5,
    CR_NLD = 6,
    CR_POL = 7,
    CR_SVK = 8,
    CR_BLR = 9,
    CR_MDA = 10,
    CR_RUS = 11,
    CR_UKR = 12,
    CR_BEL = 13,
    CR_BGR = 14,
    CR_DNK = 15,
    CR_FIN = 16,
    CR_GBR = 17,
    CR_GRC = 18,
    CR_HRV = 19,
    CR_HUN = 20,
    CR_ISR = 21,
    CR_LUX = 22,
    CR_MKD = 23,
    CR_NOR = 24,
    CR_PRT = 25,
    CR_ROU = 26,
    CR_SRБ = 27,
    CR_AZE = 28,
    CR_GEO = 29,
    CR_KAZ = 30,
    CR_LTU = 31,
    CR_TKM = 32,
    CR_UZB = 33,
    CR_LVA = 34,
    CR_EST = 35,
    CR_ALB = 36,
    CR_AUT = 37,
    CR_BIH = 38,
    CR_IRL = 39,
    CR_ISL = 40,
    CR_VAT = 41,
    CR_MLT = 42,
```

```
CR_SWE      = 43,  
CR_CHE      = 44,  
CR_CYP      = 45,  
CR_TUR      = 46,  
CR SVN     = 47,  
CR_MTG      = 48,  
CR_KOV      = 49,  
CR_ADR      = 50,  
CR_ARM      = 51,  
CR_MON      = 52,  
CR_LIE      = 53,  
CR_SMO      = 54,  
CR_RES1     = 55,  
CR_RES2     = 56,  
CR_RES3     = 57,  
CR_RES4     = 58,  
CR_CHI      = 59,  
CR_IBN      = 60,  
CR_SKR      = 61,  
CR_LEB      = 62,  
CR_NEPE     = 63,  
CR_THA      = 64,  
CR_PAK      = 65,  
CR_EMI      = 66,  
CR_BHU      = 67,  
CR_OMA      = 68,  
CR_KOR      = 69,  
CR_PHI      = 70,  
CR_CAM      = 71,  
CR_QAT      = 72,  
CR_KYR      = 73,  
CR_MAL      = 74,  
CR_MLY      = 75,  
CR_MOG      = 76,  
CR_ARA      = 77,  
CR_BRU      = 78,  
CR_LAOS     = 79,  
CR_JAP      = 80,  
CR_RES19    = 81,  
CR_PAS      = 82,  
CR_TAJ      = 83,  
CR_KUW      = 84,  
CR_SYR      = 85,  
CR_IND      = 86,  
CR_ISA      = 87,  
CR_AFG      = 88,  
CR_LAN      = 89,  
CR_IRQ      = 90,  
CR_VIE      = 91,  
CR_IRA      = 92,  
CR_YEM      = 93,  
CR_JOR      = 94,
```

```
CR_BUR      = 95,
CR_SIK      = 96,
CR_BAN      = 97,
CR_SGA      = 98,
CR_EAT      = 99,
CR_RES5     = 100,
CR_RES6     = 101,
CR_RES7     = 102,
CR_RES8     = 103,
CR_EGT      = 104,
CR_LIY      = 105,
CR_SUA      = 106,
CR_TUN      = 107,
CR_ALG      = 108,
CR_MCC      = 109,
CR_ETH      = 110,
CR_ERI      = 111,
CR_SDE      = 112,
CR_DJI      = 113,
CR_KEN      = 114,
CR_TAI      = 115,
CR_UGA      = 116,
CR_RWA      = 117,
CR_BUD      = 118,
CR_SEY      = 119,
CR_CHA      = 120,
CR_CEA      = 121,
CR_CON      = 122,
CR_EQG      = 123,
CR_GAB      = 124,
CR_TCO      = 125,
CR_DRC      = 126,
CR_STP      = 127,
CR_MAN      = 128,
CR_WSA      = 129,
CR_SEL      = 130,
CR_TGA      = 131,
CR_MAI      = 132,
CR_BUF      = 133,
CR_GUI      = 134,
CR_GUB      = 135,
CR_CAV      = 136,
CR_SLE      = 137,
CR_LIR      = 138,
CR_IVC      = 139,
CR_GHA      = 140,
CR_TGO      = 141,
CR_BEN      = 142,
CR_NIG      = 143,
CR_ZAB      = 144,
CR_ANG      = 145,
CR_ZBE      = 146,
```

CR_MAW	= 147,
CR_MOQ	= 148,
CR_BOT	= 149,
CR_NAM	= 150,
CR_SAF	= 151,
CR_SWD	= 152,
CR_LES	= 153,
CR_MAG	= 154,
CR_UOC	= 155,
CR_MAT	= 156,
CR_NGE	= 157,
CR_SSD	= 158,
CR_SAII	= 159,
CR_MYT	= 160,
CR_REN	= 161,
CR_CAI	= 162,
CR_AZO	= 163,
CR_MAD	= 164,
CR_RES9	= 165,
CR_RES10	= 166,
CR_RES11	= 167,
CR_RES12	= 168,
CR_CAD	= 169,
CR_GRE	= 170,
CR_PIE	= 171,
CR_USA	= 172,
CR_BER	= 173,
CR_MEX	= 174,
CR_GUA	= 175,
CR_BLI	= 176,
CR_SAR	= 177,
CR_HOR	= 178,
CR_NIC	= 179,
CR_COR	= 180,
CR_PAN	= 181,
CR_TBM	= 182,
CR_TCI	= 183,
CR_CUB	= 184,
CR_JAM	= 185,
CR_CAY	= 186,
CR_HAT	= 187,
CR_TDO	= 188,
CR_PUR	= 189,
CR_VIL	= 190,
CR_BVI	= 191,
CR_ATV	= 192,
CR_ANB	= 193,
CR_CSM	= 194,
CR_ACY	= 195,
CR_SBY	= 196,
CR_SKN	= 197,
CR_MOT	= 198,

```
CR_GLP      = 199,  
CR_DOM      = 200,  
CR_MTE      = 201,  
CR_LUC      = 202,  
CR_SVG      = 203,  
CR_GRD      = 204,  
CR_BAR      = 205,  
CR_TRT      = 206,  
CR_CUR      = 207,  
CR_ARB      = 208,  
CR_NEA      = 209,  
CR_COL      = 210,  
CR_VEN      = 211,  
CR_GUY      = 212,  
CR_SUR      = 213,  
CR_FRN      = 214,  
CR_ECU      = 215,  
CR_PER      = 216,  
CR_BOL      = 217,  
CR_PAR      = 218,  
CR_CLE      = 219,  
CR_BRA      = 220,  
CR_UGY      = 221,  
CR_ARG      = 222,  
CR_RES13    = 223,  
CR_RES14    = 224,  
CR_RES15    = 225,  
CR_RES16    = 226,  
CR_ATN      = 227,  
CR_NED      = 228,  
CR_PNG      = 229,  
CR_SAN      = 230,  
CR_VAU      = 231,  
CR_NCN      = 232,  
CR_PAU      = 233,  
CR_FSM      = 234,  
CR_MRI      = 235,  
CR_CNM      = 236,  
CR_TEG      = 237,  
CR_NUR      = 238,  
CR_KIB      = 239,  
CR_FID      = 240,  
CR_TNG      = 241,  
CR_TUV      = 242,  
CR_WEF      = 243,  
CR_TIS      = 244,  
CR_EAS      = 245,  
CR_TOE      = 246,  
CR_NUE      = 247,  
CR_TCD      = 248,  
CR_PFP      = 249,  
CR_PID      = 250,
```

```
CR_HAW      = 251,  
CR_RES17    = 252,  
CR_RES18    = 253,  
CR_UNRECOGNIZED = 0xfe,  
CR_ALL      = 0xff,  
CR_TAIWAN   = 256  
}CR_INDEX
```

Members

CR_NONSUPPORT

Not support

CR_CZE

Czech Republic

CR_DEU

Germany

CR_ESP

Spain

CR_ITA

Italy

CR_NLD

Netherlands

CR_POL

Poland

CR_SVK

Slovakia

CR_BLR

Belorussia

CR_MDA

Moldova

CR_RUS

Russia

CR_UKR

Ukraine

CR_BEL

Belgium

CR_BGR

Bulgaria

CR_DNK

Denmark

CR_FIN

Finland

CR_GBR

United Kingdom

CR_GRC

Greece

CR_HRV

Croatia

CR_HUN

Hungary

CR_ISR

Israel

CR_LUX

Luxembourg

CR_MKD

Macedonia

CR_NOR

Norway

CR_PRT

Portugal

CR_ROU

Romania

CR_SRB

Serbia

CR_AZE

Azerbaijan

CR_GEO

Georgia

CR_KAZ

Kazakhstan

CR_LTU

Lithuania

CR_TKM

Turkmenistan

CR_UZB

Uzbekistan

CR_LVA

Latvia

CR_EST

Estonia

CR_ALB

Albania

CR_AUT

Austria

CR_BIH

Bosnia and Herzegovina

CR_IRL

Ireland

CR_ISL

Iceland

CR_VAT

Vatican

CR_MLT

Malta

CR_SWE

Sweden

CR_CHE

Switzerland

CR_CYP

Cyprus

CR_TUR

Turkey

CR SVN

Slovenia

CR_MTG

Montenegro

CR_KOV

Kosovo

CR_ADR

Andorra

CR_ARM

Armenia

CR_MON

Monaco

CR_LIE

Liechtenstein

CR_SMO

San Marino

CR_RES1

Reserved

CR_RES2

Reserved

CR_RES3

Reserved

CR_RES4

Reserved /* Asia, 48 countries, in which Cyprus is located on the border of Europe and Asia*/

CR_CHI

China

CR_IBN

In bahrain

CR_SKR

South Korea

CR_LEB

Lebanon

CR_NEП

Nepal

CR_THA

Thailand

CR_PAK

Pakistan

CR_EMI

The united Arab emirates

CR_BHU

Bhutan

CR_OMA

Oman

CR_KOR

North Korea

CR_PHI

The Philippines

CR_CAM

Cambodia

CR_QAT

Qatar

CR_KYR

Kyrgyzstan

CR_MAL

The maldives

CR_MLY

Malaysia

CR_MOG

Mongolia

CR_ARA

Saudi Arabia

CR_BRU

brunei

CR_LA0

Laos

CR_JAP

Japan

CR_RES19

Reserved

CR_PAS

Palestinian state

CR_TAJ

Tajikistan

CR_KUW

Kuwait

CR_SYR

Syria

CR_IND

India

CR_ISA

Indonesia

CR_AFG

Afghanistan

CR_LAN

Sri Lanka

CR IRQ

Iraq

CR_VIE

Vietnam

CR_IRA

Iran

CR_YEM

yemen

CR_JOR

Jordan

CR_BUR

Burma

CR_SIK

Sikkim

CR_BAN

Bangladesh

CR_SGA

Singapore

CR_EAT

East Timor

CR_RES5

Reserved

CR_RES6

Reserved

CR_RES7

Reserved

CR_RES8

Reserved /*Africa, 60 countries and regions*/

CR_EGT

Egypt

CR_LIY

Libya

CR_SUA

Sudan

CR_TUN

Tunisia

CR_ALG

Algeria

CR_MCC

Morocco

CR_ETH

Ethiopia

CR_ERI

Eritrea

CR_SDE

Somalia Democratic

CR_DJI

Djibouti

CR_KEN

Kenya

CR_TAI

Tanzania

CR_UGA

Uganda

CR_RWA

Rwanda

CR_BUD

Burundi

CR_SEY

Seychelles

CR_CHA

Chad

CR_CEA

Central African

CR_CON

Cameroon

CR_EQG

Equatorial Guinea

CR_GAB

Gabon

CR_TCO

the Congo

CR_DRC

Democratic Republic of the Congo

CR_STP

Sao Tome and Principe

CR_MAN

Mauritania

CR_WSA

Western Sahara

CR_SEL

Senega

CR_TGA

the Gambia

CR_MAI

Mali

CR_BUF

Burkina Faso

CR_GUI

Guinea

CR_GUB

Guinea-Bissau

CR_CAV

Cape Verde

CR_SLE

Sierra Leone

CR_LIR

Liberia

CR_IVC

Ivory Coast

CR_GHA

Ghana

CR_TGO

Togo

CR_BEN

Benin

CR_NIG

Niger

CR_ZAB

Zambia

CR_ANG

Angola

CR_ZBE

Zimbabwe

CR_MAW

Malawi

CR_MOQ

Mozambique

CR_BOT

Botswana

CR_NAM

Namibia

CR_SAF

South Africa

CR_SWD

Swaziland

CR_LES

Lesotho

CR_MAG

Madagascar

CR_UOC

Union of Comoros

CR_MAT

Mauritius

CR_NGE

Nigeria

CR_SSD

South Sudan

CR_SAH

Saint Helena

CR_MYT

Mayotte

CR_REN

Reunion

CR_CAI

Canary Islands

CR_AZ0

AZORES

CR_MAD

Madeira

CR_RES9

Reserved

CR_RES10

Reserved

CR_RES11

Reserved

CR_RES12

Reserved /*America, 55 countries and regions*/

CR_CAD

Canada

CR_GRE

Greenland Nuuk

CR_PIE

/Pierre and Miquelon

CR_USA

United States

CR_BER

Bermuda

CR_MEX

Mexico

CR_GUA

Guatemala

CR_BLI

Belize

CR_SAR

El Salvador

CR_HOR

Honduras

CR_NIC

Nicaragua

CR_COR

Costa Rica

CR_PAN

Panama

CR_TBM

The Bahamas

CR_TCI

The Turks and Caicos Islands

CR_CUB

Cuba

CR_JAM

Jamaica

CR_CAY

Cayman Islands

CR_HAT

Haiti

CR_TDO

The Dominican

CR_PUR

Puerto Rico

CR_VIL

The United States Virgin Islands

CR_BVI

The British Virgin Islands

CR_ATV

Anguilla The Valley

CR_ANB

Antigua and Barbuda

CR_CSM

Collectivite de Saint-Martin

CR_ACY

Autonomous CR

CR_SBY

Saint-Barthelemy

CR_SKN

Saint Kitts and Nevis

CR_MOT

Montserrat

CR_GLP

Guadeloupe

CR_DOM

Dominica

CR_MTE

Martinique

CR_LUC

St. Lucia

CR_SVG

Saint Vincent and the Grenadines

CR_GRD

Grenada

CR_BAR

Barbados

CR_TRT

Trinidad and Tobago

CR_CUR

Curaao

CR_ARB

Aruba

CR_NEA

Netherlands Antilles

CR_COL

Colombia

CR_VEN

Venezuela

CR_GUY

Guyana

CR_SUR

Suriname

CR_FRN

Guyane Francaise

CR_ECU

Ecuador

CR_PER

Peru

CR_BOL

Bolivia

CR_PAR

Paraguay

CR_CLE

Chile

CR_BRA

Brazil

CR_UGY

Uruguay

CR_ARG

Argentina

CR_RES13

Reserved

CR_RES14

Reserved

CR_RES15

Reserved

CR_RES16

Reserved /*Oceania, 25 countries and regions*/

CR_ATN

Australien

CR_NED

Neuseeland

CR_PNG

Papua New Guinea

CR_SAN

Salomonen

CR_VAU

Vanuatu

CR_NCN

New Caledonia

CR_PAU

Palau

CR_FSM

Federated States of Micronesia

CR_MRI

Marshall Island

CR_CNM

Commonwealth of the Northern Mariana Islands

CR_TEG

The Territory of Guahan

CR_NUR

Nauru

CR_KIB

Kiribati

CR_FID

Fidschi

CR_TNG

Tonga

CR_TUV

Tuvalu

CR_WEF

Wallis et Futuna

CR_TIS

The Independent State of Samoa

CR_EAS

Eastern Samoa

CR_TOE

Tokelau

CR_NUE

Niue

CR_TCD

The Cook Islands

CR_PFP

Polynesie franaise French Polynesia

CR_PID

Pitcairn Islands

CR_HAW

Hawaii State

CR_RES17

Reserved

CR_RES18

Reserved

CR_UNRECOGNIZED

Unrecognized

CR_ALL

ALL

CR_TAIWAN

Taiwan (China)

B.4 FIRE_ALARM

Enumeration Types of Fire Point Detection Information

Enumeration Type	Macro Definition Value	Description
FIRE_FRAME_DIS	0x00000001	Information of frame for marking the file point
FIRE_MAX_TEMP	0x00000002	Highest temperature value
FIRE_MAX_TEMP_POSITION	0x00000004	Position information of highest temperature point
FIRE_DISTANCE	0x00000008	Distance between target and highest temperature point

B.5 NET_DVR_CHARSET_ENUM

Enumeration about encoding types.

Enumeration Definition

```
enum _NET_DVR_CHARSET_ENUM_{
    ENUM_UTF8 =0,
    ENUM_GB2312,
    ENUM_LATIN1,
    ENUM_LATIN2,
    ENUM_LATIN3,
    ENUM_LATIN4,
    ENUM_CYRILLIC,
    ENUM_ARABIC,
    ENUM_GREEK,
    ENUM_HEBREW,
```

```
ENUM_TURKISH,  
ENUM_NORDIC,  
ENUM_THAI,  
ENUM_LATIN7,  
ENUM_LATIN8,  
ENUM_LATIN9,  
ENUM_LATIN10,  
ENUM_CENTRAL_EUROPEAN,  
ENUM_CYRILLIC_SCRIPT,  
ENUM_HEBREW_DOS862  
}NET_DVR_CHARSET_ENUM
```

Members

ENUM_UTF8

0- UTF8

ENUM_GB2312

1- GB2312

ENUM_LATIN1

2- LATIN1 (IOS-8859-1)

ENUM_LATIN2

3- LATIN2 (IOS-8859-2)

ENUM_LATIN3

4- LATIN3 (IOS-8859-3)

ENUM_LATIN4

5- LATIN4 (IOS-8859-4)

ENUM_CYRILLIC

6- CYRILLIC (IOS-8859-5)

ENUM_ARABIC

7- ARABIC (IOS-8859-6)

ENUM_GREEK

8- GREEK (IOS-8859-7)

ENUM_HEBREW

9- HEBREW (IOS-8859-8)

ENUM_TURKISH

10- TURKISH (IOS-8859-9)

ENUM_NORDIC

11- NORDIC (IOS-8859-10)

ENUM_THAI

12- THAI (IOS-8859-11)

ENUM_LATIN7

13- LATIN7 (IOS-8859-13)

ENUM_LATIN8

14- LATIN8 (IOS-8859-14)

ENUM_LATIN9

15- LATIN9 (IOS-8859-15)

ENUM_LATIN10

16- LATIN10 (IOS-8859-16)

ENUM_CENTRAL_EUROPEAN

17- CENTRAL EUROPEAN (windows-1250)

ENUM_CYRILLIC_SCRIPT

18- CYRILLIC SCRIPT (windows-1251)

ENUM_HEBREW_DOS862

19- HEBREW DOS862 (dos-862)

See Also

[NET_DVR_POS_FILTER_CFG](#)

B.6 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 IpBuffer is 4-byte status.
NET_SDK_CALLBACK_STATUS_FAILED	1002	Failed. The IpBuffer is the value of 4-byte status and 4-byte error code.

B.7 NET_DVR_LINK_KIND

Enumeration about stream types

Enumeration Type	Macro Definition Value	Description
ENUM_LINK_PREVIEW	1	Live view stream.
ENUM_LINK_PLAYBACK	2	Playback stream.
ENUM_LINK_VOICEPLAY	3	Stream of two-way audio or audio forward.

B.8 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_PROXYNS         =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_TYPE_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_MULTIPLEX

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](#).

B.9 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_TEMPLET = 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_PICTURES

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.

B.10 NET_SDK_UPLOAD_TYPE

Enumeration about File Types to Be Uploaded

Enumeration Type	Macro Definition Value	Description
UPGRADE_CERT_FILE	0	Certificate file to be upgraded.
UPLOAD_CERT_FILE	1	Certificate file to be uploaded.
TRIAL_CERT_FILE	2	Trial license file.
CONFIGURATION_FILE	3	Configuration file.
UPLOAD_RECORD_FILE	4	Video file.
SCENE_CONFIGURATION_FILE	5	Scene configuration file.
UPLOAD_PICTURE_FILE	6	Picture file.

Enumeration Type	Macro Definition Value	Description
UPLOAD_VIOLATION_FILE	7	Violation dictionary file.
UPLOAD_TG_FIL	8	Timing generator file.
UPLOAD_DATA_TO_DB	9	File to be uploaded to picture and video library.
UPLOAD_BACKGROUND_PIC	10	Background picture.
UPLOAD_CALIBRATION_FILE	11	Calibration file.
UPLOAD_TME_FILE	12	Entrance and exiting management file.
UPLOAD_VEHICLE_BLOCKALLOWLST_FILE	13	Vehicle blocklist file.
UPLOAD_PICTURE_TO_CLOUD	15	Picture file to be uploaded to cloud storage.
UPLOAD_VIDEO_FILE	16	Video file.
UPLOAD_SCREEN_FILE	17	Screen server file.
UPLOAD_PUBLISH_MATERIAL	18	Local material file of information release system.
UPLOAD_PUBLISH_UPGRADE_FILE	19	Upgrade file of information release system.
UPLOAD_RING_FILE	20	Ringtone file.
UPLOAD_ENCRYPT_CERT	21	Encryption certificate.
UPLOAD_THERMOMETRIC_FILE	22	Calibration file for temperature measurement.
UPLOAD_SUBBRAND_FILE	23	Vehicle sub brand file.
UPLOAD_LED_CHECK_FILE	24	LED correction file.
BATCH_UPLOAD_PICTURE_FILE	25	Picture files for uploading in batch.
UPLOAD_EDID_CFG_FILE	26	EDID configuration file.
UPLOAD_PANORAMIC_STITCH	27	Panorama stitching configuration file.
UPLOAD_BINOCULAR_COUNTING	28	Binocular counting correction sheet.
UPLOAD_AUDIO_FILE	29	Audio file.

Enumeration Type	Macro Definition Value	Description
UPLOAD_PUBLISH_THIRD_PARTY_FILE	30	Third-party file.
UPLOAD_DEEPEYES_BINOCULAR	31	TX1 binocular correction sheet.
UPLOAD_CERTIFICATE_BLOCKLIST	32	ID card blocklist.
UPLOAD_HD_CAMERA_CORRECT_TABLE	33	HD camera correction sheet (CAL format).
UPLOAD_FD_DATA	35	Face data file to be imported to face picture library.
UPLOAD_FACE_DATA	36	Face picture file to be imported to face picture library.
UPLOAD_FACE_ANALYSIS_DATA	37	Picture file to be imported to picture recognition target.
UPLOAD_FILEVOLUME_DATA	38	File volume file
IMPORT_DATA_TO_FACELIB	39	Face data (face picture and picture additional information) to be imported to face picture library of device.
UPLOAD_LEFTEYE_4K_CALIBFILE	40	Camera calibration parameter file.
UPLOAD_SECURITY_CFG_FILE	41	Configuration file to be securely imported.
UPLOAD_RIGHT_CONTROLLER_AUDIO	42	Audio file of main controller.
UPLOAD_MODBUS_CFG_FILE	43	Configuration file of Modbus protocol.
UPLOAD_NOTICE_VIDEO_DATA	44	Bulletin video file.
UPLOAD_RS485_PROTOCOL_DLL_FILE	45	Dynamic library file of RS485 protocol.
UPLOAD_PIC_BY_BUF	46	Picture file for importing by picture cache.
UPLOAD_CLIENT_CALIBFILE	47	User calibration file (PTO format).
UPLOAD_HD_CAMERA_CORRECT_TABLE_3200W	48	HD camera correction sheet (CAL format).

Enumeration Type	Macro Definition Value	Description
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.

B.11 PRIDATA_RENDER

Enumeration Types of Intelligent Information

Enumeration Type	Macro Definition Value	Description
RENDER_ANA_INTEL_DATA	0x00000001	Intelligent analysis information
RENDER_MD	0x00000002	Motion detection information
RENDER_ADD_POS	0x00000004	POS information
RENDER_ADD_PIC	0x00000008	Picture
RENDER_FIRE_DETCET	0x00000010	Fire point detection information, its sub types are enumerated in FIRE_ALARM .
RENDER_TEM	0x00000020	Rule information of temperature measurement, its

Enumeration Type	Macro Definition Value	Description
		sub types are enumerated in <i>TEM_FLAG</i> .

B.12 STREAM_TYPE_E

Stream Type Enumeration

Enumeration Type	Macro Definition Value	Description
ENUM_STREAM_H264_TYPE	0	H.264 video stream data
ENUM_STREAM_H265_TYPE	1	H.265 video stream data.
ENUM_STREAM_JPEG_TYPE	2	JPEG picture data
ENUM_STREAM_AUD_TYPE	3	Audio data.
ENUM_STREAM_META_TYPE	4	Metadata.
ENUM_STREAM_UPDATE_TYPE	5	Change data.
ENUM_STREAM_RTDATA_TYPE	6	Real-time data.

B.13 TEM_FLAG

Enumeration Types of Temperature Measurement Rule Information

Enumeration Type	Macro Definition Value	Description
TEM_REGION_BOX	0x00000001	Rule information of measuring temperature by frame
TEM_REGION_LINE	0x00000002	Rule information of measuring temperature by line
TEM_REGION_POINT	0x00000004	Rule information of measuring temperature by point

B.14 VCA_PLATE_COLOR

Enumerate the license plate colors.

Enumeration Definition

```
enum{
    VCA_BLUE_PLATE =0,
    VCA_YELLOW_PLATE =1,
    VCA_WHITE_PLATE =2,
    VCA_BLACK_PLATE =3,
    VCA_GREEN_PLATE =4,
    VCA_BKAIR_PLATE =5,
    VCA_RED_PLATE,
    VCA_ORANGE_PLATE,
    VCA_OTHER      =0xff
}VCA_PLATE_COLOR
```

Member

VCA_BLUE_PLATE

Blue

VCA_YELLOW_PLATE

Yellow

VCA_WHITE_PLATE

White

VCA_BLACK_PLATE

Black

VCA_GREEN_PLATE

Green

VCA_BKAIR_PLATE

Black (for special use)

VCA_RED_PLATE

Red

VCA_ORANGE_PLATE

Orange

VCA_OTHER

Other

B.15 VLR_VEHICLE_CLASS

Enumerate the vehicle parent brands.

Enumeration Definition

```
typedef enum _VLR_VEHICLE_CLASS{
    VLR_OTHER      = 0, //Other
    VLR_VOLKSWAGEN = 1, //Volkswagen
    VLR_BUICK      = 2, //Buick
    VLR_BMW        = 3, //BMW
    VLR_HONDA     = 4, //Honda
    VLR_PEUGEOT    = 5, //Peugeot
    VLR_TOYOTA     = 6, //Toyota
    VLR_FORD       = 7, //Ford
    VLR_NISSAN     = 8, //Nissan
    VLR_AUDI       = 9, //Audi
    VLR_MAZDA      = 10, //Mazda
    VLR_CHEVROLET = 11, //Chevrolet
    VLR_CITROEN    = 12, //Citroen
    VLR_HYUNDAI    = 13, //Hyundai
    VLR_CHERY      = 14, //Chery
    VLR_KIA        = 15, //Kia
    VLR_ROEWE      = 16, //Roewe
    VLR_MITSUBISHI = 17, //Mitsubishi
    VLR_SKODA      = 18, //Skoda
    VLR_GEELY      = 19, //Geely
    VLR_ZHONGHUA  = 20, //Zhonghua
    VLR_VOLVO      = 21, //Volvo
    VLR_LEXUS      = 22, //Lexus
    VLR_FIAT       = 23, //Fiat
    VLR_EMGRAND    = 24, //Emgrand (Geely)
    VLR_DONGFENG   = 25, //Dongfeng
    VLR_BYD        = 26, //BYD
    VLR_SUZUKI     = 27, //Suzuki
    VLR_JINBEI     = 28, //Jinbei
    VLR_HAIMA      = 29, //Haima
    VLR_SGMW       = 30, //SGMW
    VLR_JAC        = 31, //JAC
    VLR_SUBARU     = 32, //Subaru
    VLR_ENGLON     = 33, //Englon (Geely)
    VLR_GREATWALL  = 34, //Great Wall
    VLR_HAFEI      = 35, //Hafei
    VLR_ISUZU      = 36, //Isuzu
    VLR_SOUEAST    = 37, //Soueast
    VLR_CHANA      = 38, //Changan
    VLR_FOTON      = 39, //Foton
    VLR_XIALI      = 40, //Xiali (FAW)
    VLR_BENZ       = 41, //Benz
    VLR_FAW        = 42, //FAW
```

VLR_NAVECO	= 43, //Iveco
VLR_LIFAN	= 44, //Lifan
VLR_BESTURN	= 45, //FAW Besturn (FAW)
VLR_CROWN	= 46, //Crown (Toyota)
VLR_RENAULT	= 47, //Renault
VLR_JMC	= 48, //JMC
VLR_MG	= 49, //MG
VLR_KAMA	= 50, //Kama
VLR_ZOTYE	= 51, //Zotye
VLR_CHANGHE	= 52, //Changhe
VLR_XMKINGLONG	= 53, //Xiamen King Long (Golden Dragon)
VLR_HUIZHONG	= 54, //Shanghai Huizhong
VLR_SZKINGLONG	= 55, //Suzhou Jinlong
VLR_HIGER	= 56, //Higer
VLR_YUTONG	= 57, //Yutong
VLR_CNHTC	= 58, //CNHTC
VLR_BEIBEN	= 59, //Beiben Truck
VLR_XINGMA	= 60, //Hualing Xingma
VLR_YUEJIN	= 61, //Yuejin
VLR_HUANGHAI	= 62, //Huanghai
VLR_OLDWALL	= 63, //Great Wall (Old Version)
VLR_CHANACOMMERCIAL	= 64, //Chang'an Business
VLR_PORSCHE	= 65, //Porsche
VLR_CADILLAC	= 66, //Cadillac
VLR_INFINITI	= 67, //Infiniti
VLR_GLEAGLE	= 68, //Gleagle (Geely)
VLR_JEEP	= 69, //Jeep
VLR_LANDROVER	= 70, //Land Rover
VLR_CHANGFENG	= 71, //Changfeng
VLR_BENNI	= 72, //Chang'an Benni
VLR_EERA	= 73, //Foton Forland
VLR_TAURUS	= 74, //Chana Tauri Star (Chang'an Business)
VLR_EADO	= 75, //Chang'an Yidong
VLR_SHANQI	= 76, //Shanqi
VLR_HONGYAN	= 77, //Hongyan Auto (SAIC IVECO HONGYAN)
VLR_DRAGON	= 78, //Balong Motor (Dongfeng Liuqi)
VLR_JACTEXT	= 79, //Jianghuai JAC
VLR_JACBUS	= 80, //Jianghuai Xiandai Bus
VLR_ANKAI	= 81, //Ankai Bus
VLR_SHENLONG	= 82, //Shenlong Bus
VLR_DAEWOO	= 83, //Daewoo Bus
VLR_WUZHENG	= 84, //Wuzheng Motor
VLR_MAN	= 85, //MAN Motor
VLR_ZHONGTONG	= 86, //Zhongtong Bus
VLR_BAOJUN	= 87, //Baojun
VLR_BQWEIWANG	= 88, //BAIC Weiwang
VLR_TRUMPCHE	= 89, //Trumpchi
VLR_LUFENG	= 90, //Landwind
VLR_HMZHENGHOU	= 91, //Zhengzhou Hippocampus
VLR_BEIJING	= 92, //BAIC Motor
VLR_ZHONGSHUN	= 93, //Zhongshun
VLR_WEILIN	= 94, //Weiling Motor

VLR_OPEL	= 95, //Opel
VLR_KARRY	= 96, //Karry
VLR_SMA	= 97, //Huapu Motor
VLR_SMATEXT	= 98, //Huapu Motor Wenzi SMA
VLR_YUWIN	= 99, //JMC Yusheng
VLR_MINI	= 100, //BMW MINI
VLR_MARK	= 101, //Toyota MARKX
VLR_HAVAL	= 102, //HAVAL
VLR_OGA	= 103, //Acura
VLR_VENUCIA	= 104, //Venucia
VLR_BYD2	= 105, //BYD Style 2
VLR_SMART	= 106, //Benz SMART
VLR_BAW	= 107, //Beijing Vehicle Manufacture/BAW
VLR_LUXGEN	= 108, //Luxgen
VLR_YEMA	= 109, //Yema
VLR_ZTE	= 110, //ZXAUT
VLR_EVERUS	= 111, //Linian
VLR_CHRYSLER	= 112, //Chrysler
VLR_GONOW	= 113, //Ji'ao
VLR_SHJIANG	= 114, //Songhua River
VLR_RUILIN	= 115, //Chrey
VLR_FORTA	= 116, //Fuda
VLR_GAGUAR	= 117, //Jaguar
VLR_HEIBAO	= 118, //Heibao
VLR_TKING	= 119, //TKING
VLR_TKINGTEXT	= 120, //Tangjun Wenzi
VLR_FODAY	= 121, //Foday
VLR_LOTUS	= 122, //Lianhua Motor
VLR_NANJUN	= 123, //CNJ
VLR_SHUANGHUAN	= 124, //Shuanghuan Motor
VLR_SAIBAO	= 125, //HAFEI Saibao
VLR_HAWTAI	= 126, //Hawtai
VLR_LIFO	= 127, //Yongyuan Feidie
VLR_JONWAY	= 128, //Yongyuan Motor
VLR_FULONGMA	= 129, //Fulongma
VLR_WEILI	= 130, //Huaili
VLR_ANCHI	= 131, //Jianghuai Anchi
VLR_PAIXI	= 132, //Splash
VLR_HIGERTEXT	= 133, //HIGER Wenzi
VLR_RIYECAR	= 134, //Hino Light Truck
VLR_RIYETRUCK	= 135, //Hino Heavy Truck
VLR_JIANGNAN	= 136, //Jiangnan
VLR_OLDZOTYE	= 137, //Zhongtai (Old Version)
VLR_OLDXIALI	= 138, //Xiali (Old Version)
VLR_NEWAOCHI	= 139, //New Aochi
VLR_CDW	= 140, //Zhongqi Wangpai
VLR_CDWTEXT	= 141, //Zhongqi Wangpai Wenzi
VLR_CIIMO	= 142, //Honda CIIMO
VLR_CHANADS	= 143, //Chang'an Di Ai Shi
VLR_DS	= 144, //Dodge
VLR_ROHENS	= 145, //Hyundai Rohens Coupe
VLR_YANTAI	= 146, //Yantai

```
VLR_SHUANGLONG    = 147, //Shuanglong
VLR_FENGLING      = 148, //Shidai Fengling
VLR_XINKAI        = 149, //Xinkai
VLR_GMC           = 150, //GMC
VLR_DATONG        = 151, //MAXUS
VLR_BQYINXIANG   = 152, //BAIC Yinxiang
VLR_NEWCHERY      = 153, //New Chery
VLR_MUDAN         = 154, //Mudan
VLR_DAYUN          = 155, //Dayun Motor
VLR_DONGWO        = 156, //Dongwo Motor
VLR_UNION          = 157, //Union Motor
VLR_CHUNZHOU      = 158, //Chunzhou Bus
VLR_SANY           = 159, //Sany
VLR_YAXING         = 160, //Asiastar Bus
VLR_HENGTONG       = 161, //Hengtong Bus
VLR_SHAOLIN        = 162, //Shaolin Bus
VLR_YOUNG          = 163, //Young Man Bus
VLR_STOM           = 164, //Shitong
VLR_SANHUAN        = 165, //Tri-Ring
VLR_XUGONG         = 166, //XCMG
VLR_BEIFANG        = 167, //Beifang Motor
VLR_JIANGHUAN      = 168, //Jianghuan Truck
VLR_BJFARM          = 169, //Beijing Agricultural
VLR_NEWDADI        = 170, //Xin Dadi Motor
VLR_SUNWIN          = 171, //Sunwin Bus
VLR_YINGTIAN        = 172, //Yingtian
VLR_QINGQI          = 173, //Qingqi
VLR_CHUFENG         = 174, //Chufeng Motor
VLR_SCANIA          = 175, //Scania
VLR_JIULONG         = 176, //Jiulong Bus
VLR_YOUYI           = 177, //Youyi Bus
VLR_SHANGRAO        = 178, //Shangrao Bus
VLR_JIJIANG         = 179, //Ji Jiang
VLR_YANGZI          = 180, //Yangzi Bus
VLR_XIWO            = 181, //Seewo Bus
VLR_CHANGJIANG      = 182, //Changjiang Bus
VLR_WUYI             = 183, //Wuyi
VLR_CHENGDU         = 184, //Chengdu Bus
VLR_TIANMA          = 185, //Tianma
VLR_BAOLONG         = 186, //Baolong
VLR_NEWYATU         = 187, //Soyat
VLR_BARUI           = 188, //Kia Borrego
VLR_GUANZHI         = 189, //Qoros
VLR_XIYATE          = 190, //Seat
VLR_BINLI           = 191, //Bentley
VLR_DADI            = 192, //Dadi
VLR_FUQI             = 193, //Fuqi
VLR_HANGTIAN        = 194, //Hangtian Motor
VLR_HENGTEAN        = 195, //Hi-tech
VLR_JMCCAR          = 196, //JMC
VLR_KAERSEN          = 197, //Carlson Motor
VLR_KAWEI            = 198, //Kawei Motor
```

```
VLR_LANBOJINI    = 199, //Lamborghini  
VLR_MASHALADI   = 200, //Maserati  
VLR_SHUCHI      = 201, //Shuchi Bus  
VLR_SHILI       = 202, //Shili Bus  
VLR_HUABEI      = 203, //Zhongke Huabei  
VLR_YIZHENG     = 204, //SAIC Yizheng  
VLR_CHUNLAN     = 205, //Chunlan  
VLR_DAFA        = 206, //Daihatsu Motor  
VLR_SHENYE      = 207, //Shenyne Motor  
VLR_FALALI      = 208, //Ferrari  
VLR_FUXING      = 209, //Fuxing Motor  
VLR_ANYUAN      = 210, //Anyuan-Bus  
VLR_JINGHUA     = 211, //Jinghua Bus  
VLR_TAIHU        = 212, //Taihu Bus  
VLR_WUZHOULONG  = 213, //Wuzhoulong  
VLR_CHANGLONG    = 214, //Changlong-Bus  
VLR_YUEXI        = 215, //Yuexi Bus  
VLR_SHENMA      = 216, //Shenma Bus  
VLR_LUSHAN       = 217, //Lushan  
VLR_WANFENG      = 218, //Wanfeng  
VLR_GZYUNBAO     = 219, //Guangzhou Yunbao  
VLR_ZHONGDA      = 220, //Zhongda Motor  
VLR_THREEWHEEL   = 221, //Tricycle  
VLR_TWOWHEEL     = 222, //Two Wheeler  
VLR_JBC          = 223, //JBC  
VLR_YZJIANG      = 224, //Yangtze Bus  
VLR_CNJ          = 225, //CNJ  
VLR_FORLAND      = 226, //Futian Shidai Wensi  
VLR_FARMCAR      = 227, //Agricultural Vehicle  
VLR_DONGFANGHONG = 228, //Dong Fang Hong  
VLR_STEYR        = 229, //Steyr  
VLR_HONGQI       = 230, //Hongqi  
VLR_USER1         = 231, //User 1  
VLR_USER2         = 232, //User 2  
VLR_USER3         = 233, //User 3  
VLR_USER4         = 234, //User 4  
VLR_USER5         = 235, //User 5  
VLR_USER6         = 236, //User 6  
VLR_USER7         = 237, //User 7  
VLR_USER8         = 238 //User 8  
}VLR_VEHICLE_CLASS;
```

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 C-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

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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_FACENAP_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

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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 C-2 Minor Types of Exception Log

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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 C-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)

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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)

Log Minor Type	Value	Description
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)

Log Minor Type	Value	Description
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)

Log Minor Type	Value	Description
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)

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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)

Log Minor Type	Value	Description
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)

Log Minor Type	Value	Description
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)

Log Minor Type	Value	Description
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)

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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_SPECTIME	0x3006	Delete events by specified time remotely
MIMOR_LOCAL_DELETE_EVENTS_AT_SPECTIME	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 C-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

Log Minor Type	Value	Description
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

Log Minor Type	Value	Description
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 C-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.

Log Minor Type	Value	Description
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_IOABILITY	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 MulitRs485 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, abnormal event detection, 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_DESCRIBEREVTIMEOUT	413	Receiving "RTSP DECRIBE" is timeout.
NET_DVR_RTSP_DESCRIBEREVDATALOST	414	Receiving data of "RTSP DECRIBE" error.
NET_DVR_RTSP_DESCRIBERECCVERROR	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_SETUPRECVDATALOST	424	Receiving data of "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRECVERROR	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/AUXInfo/attributes/Channels

Get the attributes of channels.

Request URI Definition

Table C-1 GET /ISAPI/AUXInfo/attributes/Channels

Method	GET
Description	Get the attributes of channels.
Query	None.
Request	None.
Response	<u>XML_ChannelInfoList</u>

E.2 /ISAPI/AUXInfo/attributes/Channels/<ID>

Get the attributes of a specific channel according to channel ID.

Request URI Definition

Table C-2 GET /ISAPI/AUXInfo/attributes/Channels/<ID>

Method	GET
Description	Get the attributes of a specific channel according to channel ID.
Query	None.

Request	None.
Response	<u>XML_ChannelInfo</u>

Remarks

The <ID> in the request URI is the channel ID.

E.3 /ISAPI/ContentMgmt/capabilities

Get storage capability.

Request URI Definition

Table C-3 GET /ISAPI/ContentMgmt/capabilities

Method	GET
Description	Get storage capability.
Query	None.
Request	None.
Response	Succeeded: <u>XML_RacmCap</u> Failed: <u>XML_ResponseStatus</u>

E.4 /ISAPI/ContentMgmt/download

Download the file via plug-in.

Request URI Definition

Table C-4 POST /ISAPI/ContentMgmt/download

Method	POST
Description	Download the file.
Query	None.
Request	<u>XML_downloadRequest</u>
Response	File

E.5 /ISAPI/ContentMgmt/download/capabilities

Get the downloading capability sets.

Request URI Definition

Table C-5 GET /ISAPI/ContentMgmt/download/capabilities

Method	GET
Description	Get the downloading capability sets.
Query	None.
Request	None.
Response	Succeeded: <i>XML_DownloadAbility</i> Failed: <i>XML_ResponseStatus</i>

E.6 /ISAPI/ContentMgmt/InputProxy/channels/<ID>

Operations about management of a specific digital channel.

Request URI Definition

Table C-6 GET /ISAPI/ContentMgmt/InputProxy/channels/<ID>

Method	GET
Description	Get parameter of a specific digital channel.
Query	None.
Request	None.
Response	Succeeded: <i>XML_InputProxyChannel</i> Failed: <i>XML_ResponseStatus</i>

Table C-7 PUT /ISAPI/ContentMgmt/InputProxy/channels/<ID>

Method	PUT
Description	Set parameters of a specific digital channel.
Query	None.

Request	<u>XML_InputProxyChannel</u>
Response	<u>XML_ResponseStatus</u>

Table C-8 DELETE /ISAPI/ContentMgmt/InputProxy/channels/<ID>

Method	DELETE
Description	Delete a specific digital channel.
Query	None.
Request	None.
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the digital channel ID.

E.7 /ISAPI/ContentMgmt/logSearch?format=json

Search log information.

Request URI Definition**Table C-9 POST /ISAPI/ContentMgmt/logSearch?format=json**

Method	POST
Description	Search log information.
Query	format: determine the format of request or response message.
Request	<u>JSON_SearchCondition</u>
Response	Succeeded: <u>JSON_MatchList</u> Failed: <u>JSON_ResponseStatus</u>

E.8 /ISAPI/ContentMgmt/record/control/locks/capabilities

Get video locking or unlocking capability.

Request URI Definition

Table C-10 GET /ISAPI/ContentMgmt/record/control/locks/capabilities

Method	GET
Description	Get video locking or unlocking capability.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	None
Response	Succeeded: <u>XML_LockCap</u> Failed: <u>XML_ResponseStatus</u>

E.9 /ISAPI/ContentMgmt/record/control/locks/name

Lock or unlock videos by file name.

Request URI Definition

Table C-11 PUT /ISAPI/ContentMgmt/record/control/locks/name

Method	PUT
Description	Lock or unlock videos by file name.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	<u>XML_LockByName</u>
Response	<u>XML_ResponseStatus</u>

E.10 /ISAPI/ContentMgmt/record/control/remark/capabilities? format=json

Get capability of adding remarks for videos by video ID.

Request URI Definition

Table C-12 GET /ISAPI/ContentMgmt/record/control/remark/capabilities?format=json

Method	GET
Description	Get capability of adding remarks for videos by video ID.
Query	format: determine the format of request or response message.
Request	None
Response	Succeeded: <u>JSON_Cap_RemarkList</u> Failed: <u>XML_ResponseStatus</u>

E.11 /ISAPI/ContentMgmt/record/control/remark?format=json

Add remarks for videos by video ID.

Request URI Definition

Table C-13 PUT /ISAPI/ContentMgmt/record/control/remark?format=json

Method	PUT
Description	Add remarks for videos by video ID.
Query	format: determine the format of request or response message.
Request	<u>JSON_RemarkList</u>
Response	<u>XML_ResponseStatus</u>

E.12 /ISAPI/ContentMgmt/record/tracks

Operations about recording schedule configuration.

Request URI Definition

Table C-14 GET /ISAPI/ContentMgmt/record/tracks

Method	GET
Description	Get all recording schedules.
Query	None.
Request	None.
Response	Succeeded: <i>XML TrackList</i> Failed: <i>XML ResponseStatus</i>

Table C-15 PUT /ISAPI/ContentMgmt/record/tracks

Method	PUT
Description	Set all recording schedules.
Query	None.
Request	<i>XML TrackList</i>
Response	<i>XML ResponseStatus</i>

Table C-16 POST /ISAPI/ContentMgmt/record/tracks

Method	POST
Description	Add a recording schedule.
Query	None.
Request	<i>XML Track</i>
Response	<i>XML ResponseStatus</i>

E.13 /ISAPI/ContentMgmt/record/tracks/<ID>

Operations about a recording schedule.

Request URI Definition

Table C-17 GET /ISAPI/ContentMgmt/record/tracks/<ID>

Method	GET
Description	Get a recording schedule.
Query	None.
Request	None.
Response	Succeeded: <i>XML Track</i> Failed: <i>XML ResponseStatus</i>

Table C-18 PUT /ISAPI/ContentMgmt/record/tracks/<ID>

Method	PUT
Description	Set a recording schedule.
Query	None.
Request	<i>XML Track</i>
Response	<i>XML ResponseStatus</i>

Table C-19 DELETE /ISAPI/ContentMgmt/record/tracks/<ID>

Method	DELETE
Description	Delete a recording schedule.
Query	None.
Request	None.
Response	<i>XML ResponseStatus</i>

Remarks

The <ID> in the request URI refers to the recording schedule ID.

E.14 /ISAPI/ContentMgmt/record/tracks/<ID>/capabilities

Get the configuration capability of the recording schedule.

Request URI Definition

Table C-20 GET /ISAPI/ContentMgmt/record/tracks/<ID>/capabilities

Method	GET
Description	Get the configuration capability of the recording schedule.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_Track</i> Failed: <i>XML_ResponseStatus</i>

Remarks

The <ID> in the request URI refers to the recording schedule ID.

E.15 /ISAPI/ContentMgmt/search

Search for specified resources.

Request URI Definition

Table C-21 GET or POST /ISAPI/ContentMgmt/search

Method	GET or POST
Description	Search for specific resources.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2. TimeType : string, time type: UTC (search according to UTC) and local time (search according to local time).
Request	<i>XML_VideoPic_CMSearchDescription</i>
Response	Succeeded: <i>XML_VideoPic_CMSearchResult</i> Failed: <i>XML_ResponseStatus</i>

Example

Sample Code for Searching Video Files Stored in Device

```
POST /ISAPI/ContentMgmt/search HTTP/1.1
Host: 10.17.132.49
Content-Length: 493
Connection: Keep-Alive
Authorization: Digest username="admin",
realm="DS-2CD2F12FWD-IWS",
nonce="4e7a46474e305a454d5445365a4445314e6a51345a54413d",
uri="/ISAPI/ContentMgmt/search",
cnonce="ce22590094d2f2bb352fc3c4cd2a1ca3",
nc=00000019,
response="ad6f2c23636f25c6db5911a113375ea9",
qop="auth"
```

```
<?xml version="1.0" encoding="utf-8"?>
<CMSearchDescription>
  <searchID>C77384AD-66A0-0001-E7C2-1151F04F90B0</searchID>
  <trackIDList>
    <trackID>101</trackID>
  </trackIDList>
  <timeSpanList>
    <timeSpan>
      <startTime>2017-03-13T16:00:00Z</startTime>
      <endTime>2017-03-16T15:59:59Z</endTime>
    </timeSpan>
  </timeSpanList>
  <maxResults>40</maxResults>
  <searchResultPostion>0</searchResultPostion>
  <metadataList>
    <metadataDescriptor>//recordType.meta.std-cgi.com</metadataDescriptor>
  </metadataList>
</CMSearchDescription>
```

HTTP/1.1 200 OK
Date: Wed, 15 Mar 2017 09:40:02 GMT
Connection: close
Content-Length: 1649
Content-Type: application/xml

```
<?xml version="1.0" encoding="UTF-8"?>
<CMSearchResult version="2.0" xmlns="http://www.isapi.com/ver20/XMLSchema">
  <searchID>{C77384AD-66A0-0001-E7C2-1151F04F90B0}</searchID>
  <responseStatus>true</responseStatus>
  <responseStatusStrg>OK</responseStatusStrg>
  <numOfMatches>29</numOfMatches>
  <matchList>
    <searchMatchItem>
      <sourcID>{0000000000-0000-0000-000000000000}</sourcID>
      <trackID>101</trackID>
      <timeSpan>
```

```

<startTime>2017-03-14T10:32:01Z</startTime>
<endTime>2017-03-14T10:40:42Z</endTime>
</timeSpan>
<mediaSegmentDescriptor>
<contentType>video</contentType>
<codecType>H.264-BP</codecType>
<playbackURI>rtsp://10.17.132.49/Streaming/tracks/101/?startime=20170314T103201Z&endtime=20170314T104042Z&name=ch01_0800000001600000&size=260358144</playbackURI>
</mediaSegmentDescriptor>
<metadataMatches>
<metadataDescriptor>recordType.meta.isapi.com/timing</metadataDescriptor>
</metadataMatches>
</searchMatchItem>
<searchMatchItem>
<sourceID>{0000000000-0000-0000-0000-000000000000}</sourceID>
<trackID>101</trackID>
<timeSpan>
<startTime>2017-03-14T10:40:42Z</startTime>
<endTime>2017-03-14T10:53:14Z</endTime>
</timeSpan>
<mediaSegmentDescriptor>
<contentType>video</contentType>
<codecType>H.264-BP</codecType>
<playbackURI>rtsp://10.17.132.49/Streaming/tracks/101/?startime=20170314T104042Z&endtime=20170314T105314Z&name=ch01_0800000001700000&size=260603904</playbackURI>
</mediaSegmentDescriptor>
<metadataMatches>
<metadataDescriptor>recordType.meta.isapi.com/timing</metadataDescriptor>
</metadataMatches>
</searchMatchItem>
</matchList>
</CMSearchResult>

```

E.16 /ISAPI/ContentMgmt/search/profile

Get video search conditions.

Request URI Definition

Table C-22 GET /ISAPI/ContentMgmt/search/profile

Method	GET
Description	Get video search conditions.
Query	None.

Request	None.
Response	Succeeded: <i>XML_CMSearchProfile</i> Failed: <i>XML_ResponseStatus</i>

E.17 /ISAPI/ContentMgmt/storage

Get or set the current working mode of storage.

Request URI Definition

Table C-23 GET /ISAPI/ContentMgmt/storage

Method	GET
Description	Get the current working mode of storage.
Query	None.
Request	None.
Response	Succeeded: <i>XML_storage</i> Failed: <i>XML_ResponseStatus</i>

Table C-24 PUT /ISAPI/ContentMgmt/storage

Method	PUT
Description	Set the current working mode of storage.
Query	None.
Request	<i>XML_storage</i>
Response	<i>XML_ResponseStatus</i>

E.18 /ISAPI/ContentMgmt/Storage/capabilities

Get the configuration capability of current storage working mode.

Request URI Definition

Table C-25 GET /ISAPI/ContentMgmt/Storage/capabilities

Method	GET
Description	Get the configuration capability of current storage working mode.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_storage</i> Failed: <i>XML_ResponseStatus</i>

E.19 /ISAPI/ContentMgmt/Storage/dataReconstruction/progress? format=json

Get the disk data reconstruction progress.

Request URI Definition

Table C-26 GET /ISAPI/ContentMgmt/Storage/dataReconstruction/progress?format=json

Method	GET
Description	Get the disk data reconstruction progress.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_Reconstruction_ProgressLists</i> Failed: <i>JSON_ResponseStatus</i>

E.20 /ISAPI/ContentMgmt/Storage/dataReconstruction?format=json

Start disk data reconstruction.

Request URI Definition

Table C-27 PUT /ISAPI/ContentMgmt/Storage/dataReconstruction?format=json

Method	PUT
Description	Start disk data reconstruction.
Query	format: determine the format of request or response message.
Request	<u>JSON DataReconstruction</u>
Response	<u>JSON ResponseStatus</u>

E.21 /ISAPI/ContentMgmt/Storage/extension

Get or set parameters of storage strategy.

Request URI Definition

Table C-28 GET /ISAPI/ContentMgmt/Storage/extension

Method	GET
Description	Get parameters of storage strategy.
Query	None.
Request	None.
Response	Succeeded: <u>XML storageExtension</u> Failed: <u>XML ResponseStatus</u>

Table C-29 PUT /ISAPI/ContentMgmt/Storage/extension

Method	PUT
Description	Set parameters of storage strategy.
Query	None.
Request	<u>XML storageExtension</u>
Response	<u>XML ResponseStatus</u>

E.22 /ISAPI/ContentMgmt/Storage/extension/capabilities

Get configuration capability of log storage.

Request URI Definition

Table C-30 GET /ISAPI/ContentMgmt/Storage/extension/capabilities

Method	GET
Description	Get configuration capability of log storage.
Query	None.
Request	None
Response	Succeeded: <u>XML_Cap_storageExtension</u> Failed: <u>JSON_ResponseStatus</u>

E.23 /ISAPI/ContentMgmt/Storage/hdd

Get configuration parameters of all HDDs.

Request URI Definition

Table C-31 GET /ISAPI/ContentMgmt/Storage/hdd

Method	GET
Description	Get configuration parameters of all HDDs.
Query	None.
Request	None.
Response	Succeeded: <u>XML_hddList</u> Failed: <u>XML_ResponseStatus</u>

E.24 /ISAPI/ContentMgmt/Storage/hdd/<ID>

Get and set parameters for a specific HDD.

Request URI Definition

Table C-32 GET /ISAPI/ContentMgmt/Storage/hdd/<ID>

Method	GET
Description	Get parameters for a specific HDD.
Query	None.
Request	None.
Response	Succeeded: <u>XML_hdd</u> Failed: <u>XMLResponseStatus</u>

Table C-33 PUT /ISAPI/ContentMgmt/Storage/hdd/<ID>

Method	PUT
Description	Set parameters for a specific HDD.
Query	None.
Request	<u>XML_hdd</u>
Response	<u>XMLResponseStatus</u>

Remarks

The <ID> in the request URI refers to the HDD ID.

E.25 /ISAPI/ContentMgmt/Storage/hdd/<ID>/encryptFormat?format=json

Format an encrypted HDD.

Request URI Definition

Table C-34 PUT /ISAPI/ContentMgmt/Storage/hdd/<ID>/encryptFormat?format=json

Method	PUT
Description	Format an encrypted HDD.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates

	that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2. format : determine the format of request or response message.
Request	<u>JSON_EncryptFormat</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the HDD ID.

E.26 /ISAPI/ContentMgmt/Storage/hdd/<ID>/encryptVerfy?format=json

Verify the HDD encryption password.

Request URI Definition

Table C-35 PUT /ISAPI/ContentMgmt/Storage/hdd/<ID>/encryptVerfy?format=json

Method	PUT
Description	Verify the HDD encryption password.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2. format : determine the format of request or response message.
Request	<u>JSON_EncryptVerfy</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the HDD ID.

E.27 /ISAPI/ContentMgmt/Storage/hdd/<ID>/logProTest?format=json

Configure preverification for log HDDs.

Request URI Definition

Table C-36 PUT /ISAPI/ContentMgmt/Storage/hdd/<ID>/logProTest?format=json

Method	PUT
Description	Configure preverification for log HDD.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_LogProTestResult</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

The <ID> in the request URI refers to the HDD No.

E.28 /ISAPI/ContentMgmt/Storage/hdd/<ID>/syncStatus?format=json

Get the HDD data sync status. Now only supports data sync from HDD to SSD.

Request URI Definition

Table C-37 GET /ISAPI/ContentMgmt/Storage/hdd/<ID>/syncStatus?format=json

Method	GET
Description	Get the HDD data sync status.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_SyncStatus</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the request URI refers to the HDD ID.
- You can call this URI only when the value of sub node **status** (**hddList**>**hdd**>**status**) in XML_storage is "synching" (related URI: /ISAPI/ContentMgmt/Storage).

E.29 /ISAPI/ContentMgmt/Storage/hdd/capabilities

Get HDD management capability.

Request URI Definition

Table C-38 GET /ISAPI/ContentMgmt/Storage/hdd/capabilities

Method	GET
Description	Get HDD management capability.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_hddList</i> Failed: <i>XML_ResponseStatus</i>

E.30 /ISAPI/ContentMgmt/storage/hdd/clearingSpace/capabilities?format=json

Get configuration capability of clearing HDD.

Request URI Definition

Table C-39 GET /ISAPI/ContentMgmt/storage/hdd/clearingSpace/capabilities?format=json

Method	GET
Description	Get configuration capability of clearing HDD.
Query	format: determine the format of request or response message.
Request	None
Response	Succeeded: <i>JSON_Cap_ClearingSpaceConfig</i> Failed: <i>JSON_ResponseStatus</i>

E.31 /ISAPI/ContentMgmt/storage/hdd/clearingSpace?format=json

Get or set parameters of clearing HDD.

Request URI Definition

Table C-40 GET /ISAPI/ContentMgmt/storage/hdd/clearingSpace?format=json

Method	GET
Description	Get parameters of clearing HDD.
Query	format: determine the format of request or response message.
Request	None
Response	Succeeded: <i>JSON_ClearingSpaceConfig</i> Failed: <i>JSON_ResponseStatus</i>

Table C-41 PUT /ISAPI/ContentMgmt/storage/hdd/clearingSpace?format=json

Method	PUT
Description	Set parameters of clearing HDD.
Query	format: determine the format of request or response message.
Request	<i>JSON_ClearingSpaceConfig</i>
Response	<i>JSON_ResponseStatus</i>

E.32 /ISAPI/ContentMgmt/Storage/hdd/specifyHddFormat?format=json

Specify multiple HDDs to be formatted.

Request URI Definition

Table C-42 PUT /ISAPI/ContentMgmt/Storage/hdd/specifyHddFormat?format=json

Method	PUT
Description	Specify multiple HDDs to format.
Query	format: determine the format of request or response message.
Request	<i>JSON_HddFormatList</i>
Response	<i>JSON_ResponseStatus</i>

E.33 /ISAPI/ContentMgmt/Storage/ssd

Get the information of all SSDs.

Request URI Definition

Table C-43 GET /ISAPI/ContentMgmt/Storage/ssd

Method	GET
Description	Get the information of all SSDs.
Query	None.
Request	None.
Response	Succeeded: <u>XML_ssdlList</u> Failed: <u>XML_ResponseStatus</u>

E.34 /ISAPI/ContentMgmt/Storage/ssd/<ID>

Get the specified SSD information.

Request URI Definition

Table C-44 GET /ISAPI/ContentMgmt/Storage/ssd/<ID>

Method	GET
Description	Get the specified SSD information.
Query	None.
Request	None.
Response	Succeeded: <u>XML_ssdl</u> Failed: <u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the SSD ID.

E.35 /ISAPI/ContentMgmt/Storage/ssd/<ID>/capabilities?format=json

Get the SSD management capability.

Request URI Definition

Table C-45 GET /ISAPI/ContentMgmt/Storage/ssd/<ID>/capabilities?format=json

Method	GET
Description	Get the SSD management capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_Cap_SSD</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the request URI refers to the SSD ID.
- If the node **isSupportSSDManagement** is returned in the message *XML_Cap_storage* (related URI: [/ISAPI/ContentMgmt/Storage/capabilities](#)) and its value is true, it indicates that the device supports this function.

E.36 /ISAPI/ContentMgmt/Storage/ssd/<ID>/SMARTTest/start

Run SSD S.M.A.R.T (self-monitoring, analysis, and reporting technology) detection.

Request URI Definition

Table C-46 PUT /ISAPI/ContentMgmt/Storage/ssd/<ID>/SMARTTest/start

Method	PUT
Description	Run SSD S.M.A.R.T (self-monitoring, analysis, and reporting technology) detection.
Query	None.
Request	<i>XML_SSDSMARTTest</i>
Response	<i>XML_ResponseStatus</i>

Remarks

- The <ID> in the request URI refers to the SSD ID.
- If the node **isSupportSSDSMARTTest** is returned in the message *XML_RacmCap* (related URI: [/ISAPI/ContentMgmt/capabilities](#)) and its value is true, it indicates that the device supports this function.

E.37 /ISAPI/ContentMgmt/Storage/ssd/<ID>/SMARTTest/status

Get the SSD S.M.A.R.T (self-monitoring, analysis, and reporting technology) detection status.

Request URI Definition

Table C-47 GET /ISAPI/ContentMgmt/Storage/ssd/<ID>/SMARTTest/status

Method	GET
Description	Get the SSD S.M.A.R.T (self-monitoring, analysis, and reporting technology) detection status.
Query	None.
Request	None.
Response	Succeeded: <u>XML_SSD_SmartTestStatus</u> Failed: <u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the SSD ID.

E.38 /ISAPI/ContentMgmt/Storage/ssd/<ID>/upgrade/status? format=json

Get the SSD firmware upgrade progress.

Request URI Definition

Table C-48 GET /ISAPI/ContentMgmt/Storage/ssd/<ID>/upgrade/status?format=json

Method	GET
Description	Get the SSD firmware upgrade progress.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_SSD_UpgradeStatus</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the SSD ID.

E.39 /ISAPI/ContentMgmt/Storage/ssd/<ID>/upgrade?format=json

Upgrade the SSD firmware.

Request URI Definition**Table C-49 POST /ISAPI/ContentMgmt/Storage/ssd/<ID>/upgrade?format=json**

Method	POST
Description	Upgrade the SSD firmware.
Query	format: determine the format of request or response message.
Request	Binary data in form format
Response	<i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the request URI refers to the SSD ID.
- If the node **isSupportSSDUpgrade** is returned in the message ***JSON_Cap_SSD*** (related URI: **/ISAPI/ContentMgmt/Storage/ssd/<ID>/capabilities?format=json**) and its value is true, it indicates that the device supports this function.

E.40 /ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/capabilities

Get the capability of digital video channel proxy.

Request URI Definition**Table C-50 GET /ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/capabilities**

Method	GET
Description	Get the capability of digital video channel proxy.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_StreamingChannel</i> Failed: <i>XML_ResponseStatus</i>

Remarks

The <ID> in the request URI refers to streaming channel ID, which is equal to (channel No. × 100 + stream type), the stream type can be: 1 (main stream), 2 (sub-stream), and so on. For example,

when **ID** is 101, it indicates the main stream of channel 1, when **ID** is 302, it indicates the sub-stream of channel 3.

E.41 /ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/PictureByUrl/capabilities?format=json

Get the capability of capturing picture in URL format

Request URI Definition

Table C-51 GET /ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/PictureByUrl/capabilities?format=json

Method	GET
Description	Get the capability of capturing picture in URL format
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_Cap_CaptureDescription</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to streaming channel ID, which is equal to (channel No. × 100 + stream type), the stream type can be: 1 (main stream), 2 (sub-stream), and so on. For example, when **ID** is 101, it indicates the main stream of channel 1, when **ID** is 302, it indicates the sub-stream of channel 3.

E.42 /ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/PictureByUrl?format=json

Capture the picture in URL format.

Request URI Definition

Table C-52 POST /ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/PictureByUrl?format=json

Method	POST
Description	Capture the picture in URL format.

Query	format: determine the format of request or response message.
Request	<u>JSON_CaputreDescription</u>
Response	Succeeded: <u>JSON_CaputreResult</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

- The <ID> in the request URI refers to streaming channel ID, which is equal to (channel No. × 100 + stream type), the stream type can be: 1 (main stream), 2 (sub-stream), and so on. For example, when **ID** is 101, it indicates the main stream of channel 1, when **ID** is 302, it indicates the sub-stream of channel 3.
- If the node **isSupportPictureByUrl** is returned in the message **XML_Cap_StreamingChannel** (related URI: [/ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/capabilities](#)) and its value is true, it indicates that the device supports this function.

E.43 /ISAPI/ContentMgmt/time/search/capabilities?format=json

Get the capability of searching for recording start and end time.

Request URI Definition

Table C-53 GET /ISAPI/ContentMgmt/time/search/capabilities?format=json

Method	GET
Description	Get the capability of searching for recording start and end time.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_Cap_TimeSearchCond</u> Failed: <u>JSON_ResponseStatus</u>

E.44 /ISAPI/ContentMgmt/time/search?format=json

Search for the recording start and end time.

Request URI Definition

Table C-54 POST /ISAPI/ContentMgmt/time/search?format=json

Method	POST
Description	Search for the recording start and end time.
Query	format: determine the format of request or response message.
Request	<i>JSON_TimeSearchCond</i>
Response	Succeeded: <i>JSON_TimeSearchResult</i> Failed: <i>JSONResponseStatus</i>

E.45 /ISAPI/Event/capabilities

Get the device event capability.

Request URI Definition

Table C-55 GET /ISAPI/Event/capabilities

Method	GET
Description	Get the device event capability set.
Query	None.
Request	None.
Response	Succeeded: <i>XML_EventCap</i> Failed: <i>XMLResponseStatus</i>

E.46 /ISAPI/Event/channels/<ID>/capabilities

Get event capabilities supported by the channel.

Request URI Definition

Table C-56 GET /ISAPI/Event/channels/<ID>/capabilities

Method	GET
Description	Get event capabilities supported by the channel.

Query	None.
Request	None.
Response	Succeeded: <i>XML_ChannelEventCap</i> Failed: <i>XML_ResponseStatus</i>

Remarks

The <ID> in the URI refers to the channel ID.

E.47 /ISAPI/Event/channels/capabilities

Get the event capability of all channels.

Request URL Definition**Table C-57 GET /ISAPI/Event/channels/capabilities**

Method	GET
Description	Get the event capability of all channels.
Query	None.
Request	None.
Response	Succeeded: <i>XML_ChannelEventCapList</i> Failed: <i>XML_ResponseStatus</i>

E.48 /ISAPI/Event/IOT/channels/<ID>/capabilities?format=json

Get the event capabilities supported by IoT device channel.

Request URI Definition**Table C-58 GET /ISAPI/Event/IOT/channels/<ID>/capabilities?format=json**

Method	GET
Description	Get the event capabilities supported by IoT device channel.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_IOTChannelEventCap</i>

	Filed: <u>JSON_ResponseStatus</u>
--	---

E.49 /ISAPI/Event/notification/subscribeEventCap

Get event/alarm subscription capability.

Request URI Definition

Table C-59 GET /ISAPI/Event/notification/subscribeEventCap

Method	GET
Description	Get event/alarm subscription capability.
Query	None.
Request	None.
Response	Succeeded: <u>XML_SubscribeEventCap</u> Failed: <u>XML_ResponseStatus</u>

E.50 /ISAPI/Event/schedules/vibrationDetection/<ID>

Get or set arming schedule of vibration detection by channel.

Request URI Definition

Table C-60 GET /ISAPI/Event/schedules/vibrationDetection/<ID>

Method	GET
Description	Get arming schedule of vibration detection by channel.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Schedule</u> Failed: <u>XML_ResponseStatus</u>

Table C-61 PUT /ISAPI/Event/schedules/vibrationDetection/<ID>

Method	PUT
Description	Set arming schedule of vibration detection by channel.

Query	None.
Request	<u>XML_Schedule</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI should be set in the format of "vibrationDetection-<channelID>", e.g., /ISAPI/Event/schedules/vibrationDetection/vibrationDetection-101.

E.51 /ISAPI/Event/schedules/<EventType>/<ID>

Operations about arming schedule of specified event type.

Request URI Definition

Table C-62 GET /ISAPI/Event/schedules/<EventType>/<ID>

Method	GET
Description	Get the arming schedule of specified event type.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Schedule</u> Failed: <u>XML_ResponseStatus</u>

Table C-63 PUT /ISAPI/Event/schedules/<EventType>/<ID>

Method	PUT
Description	Set the arming schedule of specified event type.
Query	None.
Request	<u>XML_Schedule</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <EventType> in the URI refers to the specified event type.

The <ID> in the URI refers to the ID of channel, which is triggered by alarm.

E.52 /ISAPI/Event/triggers/hdBadBlock

Operations about the linkage configuration of the HDD bad sector detection.

Request URI Definition

Table C-64 GET /ISAPI/Event/triggers/hdBadBlock

Method	GET
Description	Get the linkage parameters of the HDD bad sector detection.
Query	None.
Request	None.
Response	Succeeded: <i>XML_EventTrigger</i> Failed: <i>XML_ResponseStatus</i>

Table C-65 PUT /ISAPI/Event/triggers/hdBadBlock

Method	PUT
Description	Set the linkage parameters of the HDD bad sector detection.
Query	None.
Request	<i>XML_EventTrigger</i>
Response	<i>XML_ResponseStatus</i>

E.53 /ISAPI/Event/triggers/hdImpact

Operations about the linkage configuration of the HDD impact detection.

Request URI Definition

Table C-66 GET /ISAPI/Event/triggers/hdImpact

Method	GET
Description	Get the linkage parameters of the HDD impact detection.
Query	None.
Request	None.
Response	Succeeded: <i>XML_EventTrigger</i>

	Failed: <i>XMLResponseStatus</i>
--	----------------------------------

Table C-67 PUT /ISAPI/Event/triggers/hdImpact

Method	PUT
Description	Set the linkage parameters of the HDD impact detection.
Query	None.
Request	<i>XML_EventTrigger</i>
Response	<i>XMLResponseStatus</i>

E.54 /ISAPI/Event/triggers/highHDTemperature

Operations about the linkage configuration of the HDD high temperature detection.

Request URI Definition

Table C-68 GET /ISAPI/Event/triggers/highHDTemperature

Method	GET
Description	Get the linkage parameters of the HDD high temperature detection.
Query	None.
Request	None.
Response	Succeeded: <i>XML_EventTrigger</i> Failed: <i>XMLResponseStatus</i>

Table C-69 PUT /ISAPI/Event/triggers/highHDTemperature

Method	PUT
Description	Set the linkage parameters of the HDD high temperature detection.
Query	None.
Request	<i>XML_EventTrigger</i>
Response	<i>XMLResponseStatus</i>

E.55 /ISAPI/Event/triggers/lowHDTemperature

Operations about the linkage configuration of the HDD low temperature detection.

Request URI Definition

Table C-70 GET /ISAPI/Event/triggers/lowHDTemperature

Method	GET
Description	Get the linkage parameters of the HDD low temperature detection.
Query	None.
Request	None.
Response	Succeeded: <i>XML_EventTrigger</i> Failed: <i>XML_ResponseStatus</i>

Table C-71 PUT /ISAPI/Event/triggers/lowHDTemperature

Method	PUT
Description	Set the linkage parameters of the HDD low temperature detection.
Query	None.
Request	<i>XML_EventTrigger</i>
Response	<i>XML_ResponseStatus</i>

E.56 /ISAPI/Event/triggers/severeHDFailure

Operations about the linkage configuration of the HDD severe fault detection.

Request URI Definition

Table C-72 GET /ISAPI/Event/triggers/severeHDFailure

Method	GET
Description	Get the linkage parameters of the HDD severe fault detection.
Query	None.
Request	None.
Response	Succeeded: <i>XML_EventTrigger</i>

	Failed: <i>XMLResponseStatus</i>
--	----------------------------------

Table C-73 PUT /ISAPI/Event/triggers/severeHDFailure

Method	PUT
Description	Set the linkage parameters of the HDD severe fault detection.
Query	None.
Request	<i>XML_EventTrigger</i>
Response	<i>XMLResponseStatus</i>

E.57 /ISAPI/Event/triggers/<eventType>-<channelID>

Get, set, or delete the alarm linkage action by channel.

Request URI Definition**Table C-74 GET /ISAPI/Event/triggers/<eventType>-<channelID>**

Method	GET
Description	Get the alarm linkage action by channel.
Query	None
Request	None
Response	Succeeded: <i>XML_EventTrigger</i> Failed: <i>XMLResponseStatus</i>

Table C-75 PUT /ISAPI/Event/triggers/<eventType>-<channelID>

Method	PUT
Description	Set the alarm linkage action by channel.
Query	None
Request	<i>XML_EventTrigger</i>
Response	<i>XMLResponseStatus</i>

Table C-76 DELETE /ISAPI/Event/triggers/<eventType>-<channelID>

Method	DELETE
Description	Delete the alarm linkage action by channel.
Query	None
Request	None
Response	<u>XMLResponseStatus</u>

Remarks

The <eventType> in the request URI refers to the predefined event or alarm type name, and the <channelID> is the No. of the event detection channel. For example, if the No. of the face capture channel is 101, the "<eventType>-<channelID>" is "faceSnap-101".

E.58 /ISAPI/Event/triggers/<eventType>-<channelID>/notifications

Operations about configurations of alarm/event linkage actions.

Request URI Definition**Table C-77 GET /ISAPI/Event/triggers/<eventType>-<channelID>/notifications**

Method	GET
Description	Get the configuration parameters of alarm/event linkage actions.
Query	None.
Request	None.
Response	Succeeded: <u>XML_EventTriggerNotificationList</u> Failed: <u>XMLResponseStatus</u>

Table C-78 PUT /ISAPI/Event/triggers/<eventType>-<channelID>/notifications

Method	PUT
Description	Set the configuration parameters for alarm/event linkage actions.
Query	None.
Request	<u>XML_EventTriggerNotificationList</u>
Response	<u>XMLResponseStatus</u>

Table C-79 DELETE /ISAPI/Event/triggers/<eventType>-<channelID>/notifications

Method	DELETE
Description	Delete the configuration parameters of alarm/event linkage actions.
Query	None.
Request	None.
Response	<u>XML_ResponseStatus</u>

Remarks

The <eventType> in the request URI refers to the predefined event or alarm type name, and the <channelID> is the No. of the event detection channel. For example, if the No. of the face capture channel is 101, the "<eventType>-<channelID>" is "faceSnap-101".

E.59 /ISAPI/Event/triggers/<ID>/zone/<ID>

Operations about linkage configurations of an alarm in a specified zone.

Request URL Definition**Table C-80 GET /ISAPI/Event/triggers/<ID>/zone/<ID>**

Method	GET
Description	Get the linkage configurations of an alarm in specified zone.
Query	None.
Request	None.
Response	<u>XML_EventTrigger</u>

Table C-81 PUT /ISAPI/Event/triggers/<ID>/zone/<ID>

Method	PUT
Description	Set the linkage configurations of an alarm in specified zone.
Query	None.
Request	<u>XML_EventTrigger</u>
Response	<u>JSON_ResponseStatus</u>

Table C-82 DELETE /ISAPI/Event/triggers/<ID>/zone/<ID>

Method	DELETE
Description	Delete the linkage configurations of an alarm in specified zone.
Query	None.
Request	<u>XML_EventTrigger</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The first <ID> in the URL is defined as the No. of alarm triggered video input channel. For example, if the video input channel No. with alarm type emergencyCallhelp triggered is 1, the first <ID> is "emergencyCallhelp-1". The second <ID> in the URL is defined as the zone No.

E.60 /ISAPI/Event/triggers/notifications/AudioAlarm/capabilities?format=json

Get audible warning configuration capability.

Request URI Definition

Table C-83 GET /ISAPI/Event/triggers/notifications/AudioAlarm/capabilities?format=json

Method	GET
Description	Get audible warning configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_AudioAlarmCap</u> Failed: <u>JSON_ResponseStatus</u>

E.61 /ISAPI/Event/triggers/notifications/AudioAlarm?format=json

Get or set audible warning parameters.

Request URI Definition

Table C-84 GET /ISAPI/Event/triggers/notifications/AudioAlarm?format=json

Method	GET
Description	Get parameters of audible warning.
Query	format: determine the format of request or response message. alarmType: alarm type, including "behavior"-behavior analysis, "thermometry"-temperature measurement, "dynamicFire"-fire detection, "smokingMode"-smoke detection, "noMaskDetection"-no wearing mask detection.
Request	None.
Response	Succeeded: <u>JSON_AudioAlarm</u> Failed: <u>JSON_ResponseStatus</u>



Note

For example, if you want to get the audible warning parameters of thermometry, the request URI is: GET /ISAPI/Event/triggers/notifications/AudioAlarm?format=json&alarmType=thermometry

Table C-85 PUT /ISAPI/Event/triggers/notifications/AudioAlarm?format=json

Method	PUT
Description	Set parameters of audible warning.
Query	format: determine the format of request or response message.
Request	<u>JSON_AudioAlarm</u>
Response	<u>JSON_ResponseStatus</u>

E.62 /ISAPI/Event/triggers/notifications/AudioAlarm/customAudio? format=json

Import custom alarm audio file.

Request URI Definition

Table C-86 POST /ISAPI/Event/triggers/notifications/AudioAlarm/customAudio?format=json

Method	POST
Description	Import custom alarm audio file.
Query	format : determine the format of request or response message. type : audio file type: url, bianry.
Request	<u>JSON_ImportCustomAudio</u>
Response	Succeeded: <u>JSON_CustomAudioID</u> Failed: <u>JSON_ResponseStatus</u>

E.63 /ISAPI/Event/triggers/notifications/AudioAlarm/customAudioInfo?format=json

Get the custom alarm audio file.

Request URI Definition

Table C-87 GET /ISAPI/Event/triggers/notifications/AudioAlarm/customAudioInfo?format=json

Method	GET
Description	Get the custom alarm audio file.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_GetCustomAudio</u> Failed: <u>JSON_ResponseStatus</u>

E.64 /ISAPI/Event/triggers/notifications/AudioAlarm/customAudioTypeInfo/channels?format=json

Get encoding format information of custom audio file in a batch.

Request URI Definition**Table C-88 GET /ISAPI/Event/triggers/notifications/AudioAlarm/customAudioTypeInfo/channels?format=json**

Method	GET
Description	Get encoding format information of custom audio file in a batch.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_CustomAudioTypeInfoList</u> Failed: <u>JSON_ResponseStatus</u>

E.65 /ISAPI/Event/triggers/notifications/whiteLightAlarm/capabilities?format=json

Get the configuration capability of supplement light alarm linkage.

Request URI Definition**Table C-89 GET /ISAPI/Event/triggers/notifications/whiteLightAlarm/capabilities?format=json**

Method	GET
Description	Get the configuration capability of supplement light alarm linkage.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_WhiteLightAlarmCap</u>

E.66 /ISAPI/Event/triggers/notifications/whiteLightAlarm?format=json

Operations about configuration of supplement light alarm linkage.

Request URI Definition**Table C-90 GET /ISAPI/Event/triggers/notifications/whiteLightAlarm?format=json**

Method	GET
Description	Get parameters of supplement light alarm linkage.

Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_WhiteLightAlarm</u> Failed: <u>JSON_ResponseStatus</u>

Table C-91 PUT /ISAPI/Event/triggers/notifications/whiteLightAlarm?format=json

Method	PUT
Description	Set parameters of supplement light alarm linkage.
Query	format: determine the format of request or response message.
Request	<u>JSON_WhiteLightAlarm</u>
Response	<u>JSON_WhiteLightAlarm</u>

E.67 /ISAPI/Event/triggersCap

Get alarm linkage capability.

Request URI Definition

Table C-92 GET /ISAPI/Event/triggersCap

Method	GET
Description	Get alarm linkage capability.
Query	None
Request	None
Response	Succeeded: <u>XML_EventTriggersCap</u> Failed: <u>XML_ResponseStatus</u>

E.68 /ISAPI/Event/triggersCap/IOT

Get the alarm linkage configuration capability of IoT device.

Request URI Definition

Table C-93 GET /ISAPI/Event/triggersCap/IOT

Method	GET
Description	Get the alarm linkage configuration capability of IoT device.
Query	None.
Request	None.
Response	<u>XML_IOTTriggersCap</u>

E.69 /ISAPI/Image/channels/<ID>

Operations about image configuration of a specific channel.

Request URI Definition

Table C-94 GET /ISAPI/Image/channels/<ID>

Method	GET
Description	Get image parameters of a specific channel.
Query	None.
Request	None.
Response	Succeeded: <u>XML_ImageChannel</u> Failed: <u>XML_ResponseStatus</u>

Table C-95 PUT /ISAPI/Image/channels/<ID>

Method	PUT
Description	Set image parameters of a specific channel.
Query	None.
Request	<u>XML_ImageChannel</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the channel ID.

E.70 /ISAPI/Image/channels/<ID>/capabilities

Get image configuration capability of a specific channel.

Request URI Definition

Table C-96 GET /ISAPI/Image/channels/<ID>/capabilities

Method	GET
Description	Get image configuration capability of a specific channel.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Cap_ImageChannel</u> Failed: <u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.71 /ISAPI/Image/channels/<ID>/color

Operations about the image adjustment parameters of a specific channel.

Request URI Definition

Table C-97 GET /ISAPI/Image/channels/<ID>/color

Method	GET
Description	Get the image adjustment parameters of a specific channel.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Color</u> Failed: <u>XML_ResponseStatus</u>

Table C-98 PUT /ISAPI/Image/channels/<ID>/color

Method	PUT
Description	Set the image adjustment parameters of a specific channel.

Query	None.
Request	<u>XML_Color</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.72 /ISAPI/Image/channels/<ID>/color/capabilities

Get the image adjustment capability of a specified channel.

Request URI Definition

Table C-99 GET /ISAPI/Image/channels/<ID>/color/capabilities

Method	GET
Description	Get the image adjustment capability of a specified channel.
Query	None.
Request	None.
Response	<u>XML_Cap_Color</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.73 /ISAPI/Image/channels/<ID>/EPTZ

Get or set the e-PTZ configuration.

Request URI Definition

Table C-100 GET /ISAPI/Image/channels/<ID>/EPTZ

Method	GET
Description	Get the e-PTZ parameters.
Query	None.

Request	None.
Response	Succeeded: <u>XML_EPTZ</u> Failed: <u>XML_ResponseStatus</u>

Table C-101 PUT /ISAPI/Image/channels/<ID>/EPTZ

Method	PUT
Description	Set the e-PTZ parameters.
Query	None.
Request	<u>XML_EPTZ</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.74 /ISAPI/Image/channels/<ID>/EPTZ/mode/capabilities?format=json

Get the capability of switching e-PTZ mode.

Request URI Definition**Table C-102 GET /ISAPI/Image/channels/<ID>/EPTZ/mode/capabilities?format=json**

Method	GET
Description	Get the capability of switching e-PTZ mode.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_EPTZModeCap</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.75 /ISAPI/Image/channels/<ID>/EPTZ/mode?format=json

Get or set the configuration for switching e-PTZ mode.

Request URI Definition

Table C-103 GET /ISAPI/Image/channels/<ID>/EPTZ/mode?format=json

Method	GET
Description	Get the parameters of switching e-PTZ mode.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_EPTZMode</i> Failed: <i>JSONResponseStatus</i>

Table C-104 PUT /ISAPI/Image/channels/<ID>/EPTZ/mode?format=json

Method	PUT
Description	Set the parameters for switching e-PTZ mode.
Query	format: determine the format of request or response message.
Request	<i>JSON_EPTZMode</i>
Response	<i>JSONResponseStatus</i>

Remarks

The <ID> in the request URI refers to the channel No.

E.76 /ISAPI/Image/channels/<ID>/focusConfiguration

Get or set the focus parameters of a specified channel.

Request URI Definition

Table C-105 GET /ISAPI/Image/channels/<ID>/focusConfiguration

Method	GET
Description	Get the focus parameters of a specified channel.
Query	None.
Request	None.
Response	<i>XML_FocusConfiguration</i>

Table C-106 PUT /ISAPI/Image/channels/<ID>/focusConfiguration

Method	PUT
Description	Set the focus parameters of a specified channel.
Query	None.
Request	<i>XML_FocusConfiguration</i>
Response	<i>XMLResponseStatus</i>

E.77 /ISAPI/Image/channels/<ID>/focusConfiguration/capabilities

Get the focus configuration capability.

Request URI Definition

Table C-107 GET /ISAPI/Image/channels/<ID>/focusConfiguration/capabilities

Method	GET
Description	Get the focus configuration capability.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_FocusConfiguration</i> Failed: <i>XML_ResponseStatus</i>

Remarks

The <ID> in the request URI refers to the channel No.

E.78 /ISAPI/Image/channels/<ID>/focusConfiguration/status? format=json

Get the focus status.

Request URI Definition

Table C-108 GET /ISAPI/Image/channels/<ID>/focusConfiguration/status?format=json

Method	GET
Description	Get the focus status.

Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_FocusStatus</u> Failed: <u>JSON_ResponseStatus</u>

E.79 /ISAPI/Image/channels/<ID>/lensDistortionCorrection

Get or set parameters of image distortion correction.

Request URI Definition

Table C-109 GET /ISAPI/Image/channels/<ID>/lensDistortionCorrection

Method	GET
Description	Get parameters of image distortion correction.
Query	None.
Request	None.
Response	Succeeded: <u>XML_LensDistortionCorrection</u> Failed: <u>XML_ResponseStatus</u>

Table C-110 PUT /ISAPI/Image/channels/<ID>/lensDistortionCorrection

Method	PUT
Description	Set parameters of image distortion correction.
Query	None.
Request	<u>XML_LensDistortionCorrection</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the video input channel ID.

E.80 /ISAPI/Image/channels/<ID>/lensDistortionCorrection/capabilities

Get capability of image distortion correction.

Request URI Definition

Table C-111 GET /ISAPI/Image/channels/<ID>/lensDistortionCorrection/capabilities

Method	GET
Description	Get capability of image distortion correction.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_LensDistortionCorrection</i> Failed: <i>XML_ResponseStatus</i>

Remarks

The <ID> in the request URI refers to the video input channel ID.

E.81 /ISAPI/Image/channels/<ID>/mountingScenario

Operations about the mounting scenario mode of a camera.

Request URI Definition

Table C-112 GET /ISAPI/Image/channels/<ID>/mountingScenario

Method	GET
Description	Get the mounting scenario mode of a camera.
Query	None.
Request	None.
Response	<i>XML_MountingScenario</i>

Table C-113 PUT /ISAPI/Image/channels/<ID>/mountingScenario

Method	PUT
Description	Set the mounting scenario mode of a camera.
Query	None.
Request	<i>XML_MountingScenario</i>
Response	<i>XML_ResponseStatus</i>

Remarks

- The <ID> in the request URI refers to channel ID.
- The image parameters should be obtained again when mounting scenario mode changes.

E.82 /ISAPI/Image/channels/<ID>/mountingScenario/capabilities

Get the capability of configurations of mounting scenario mode.

Request URI Definition

Table C-114 GET /ISAPI/Image/channels/<ID>/mountingScenario/capabilities

Method	GET
Description	Get the capability of configurations of mounting scenario mode.
Query	None.
Request	None.
Response	<i>XML_Cap_MountingScenario</i>

Remarks

The <ID> in the request URI refers to channel ID.

E.83 /ISAPI/Image/channels/<ID>/supplementLight

Get or set supplement light parameters by channel.

Request URI Definition

Table C-115 GET /ISAPI/Image/channels/<ID>/supplementLight

Method	GET
Description	Get supplement light parameters by channel.
Query	None.
Request	None.
Response	Succeeded: <i>XML_SupplementLight</i> Failed: <i>XML_ResponseStatus</i>

Table C-116 PUT /ISAPI/Image/channels/<ID>/supplementLight

Method	PUT
Description	Set supplement light parameters by channel.
Query	None.
Request	<i>XML_SupplementLight</i>
Response	<i>XML_ResponseStatus</i>

Remarks

The <ID> in the request URI refers to the channel ID.

E.84 /ISAPI/Image/channels/<ID>/supplementLight/capabilities

Get supplement light configuration capability by channel.

Request URI Definition**Table C-117 GET /ISAPI/Image/channels/<ID>/supplementLight/capabilities**

Method	GET
Description	Get supplement light configuration capability by channel.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_SupplementLight</i> Failed: <i>XML_ResponseStatus</i>

Remarks

The <ID> in the request URI refers to the channel ID.

E.85 /ISAPI/Image/channels/<ID>/targetEnhancement

Get or set the target enhancement parameters.

Request URI Definition

Table C-118 GET /ISAPI/Image/channels/<ID>/targetEnhancement

Method	GET
Description	Get the target enhancement parameters.
Query	None.
Request	None.
Response	Succeeded: <i>XML_TargetEnhancement</i> Failed: <i>XML_ResponseStatus</i>

Table C-119 PUT /ISAPI/Image/channels/<ID>/targetEnhancement

Method	PUT
Description	Set the target enhancement parameters.
Query	None.
Request	<i>XML_TargetEnhancement</i>
Response	<i>XML_ResponseStatus</i>

Remarks

- This URI is used in combination with URI [*/ISAPI/Image/channels/<ID>*](#).
- The <ID> in the request URI refers to the channel No.

E.86 /ISAPI/Image/channels/<ID>/targetEnhancement/capabilities

Get the target enhancement capability.

Request URI Definition

Table C-120 GET /ISAPI/Image/channels/<ID>/targetEnhancement/capabilities

Method	GET
Description	Get the target enhancement capability.
Query	None.
Request	None.
Response	Succeeded: <i>XML_TargetEnhancementCap</i>

	Failed: <i>XMLResponseStatus</i>
--	----------------------------------

Remarks

The <ID> in the request URI refers to the channel No.

E.87 /ISAPI/Image/channels/<ID>/tempRange

Operation about temperature range parameters of the specified channel.

Request URI Definition

Table C-121 GET /ISAPI/Image/channels/<ID>/tempRange

Method	GET
Description	Get the temperature range parameters.
Query	None.
Request	None.
Response	Succeeded: <i>XML_tempRange</i> Failed: <i>XMLResponseStatus</i>

Table C-122 PUT /ISAPI/Image/channels/<ID>/tempRange

Method	PUT
Description	Set the temperature range parameters.
Query	None.
Request	None.
Response	Succeeded: <i>XML_tempRange</i> Failed: <i>XMLResponseStatus</i>

Remarks

The <ID> in the request URI refers to the channel No.

E.88 /ISAPI/Image/channels/<ID>/tempRange/capabilities

Get the temperature range configuration capability of a specified channel.

Request URI Definition

Table C-123 GET

Method	GET
Description	Get the temperature range configuration capability of a specified channel.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Cap_tempRange</u> Failed: <u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.89 /ISAPI/Intelligent/channels/<ID>/Shield/EventType

Get and set the shielded area of a specified channel according to event type.

Request URI Definition

Table C-124 GET /ISAPI/Intelligent/channels/<ID>/Shield/EventType

Method	GET
Description	Get the shielded area of a specified channel according to event type.
Query	None.
Request	None.
Response	<u>XML_Shield</u>

Table C-125 PUT /ISAPI/Intelligent/channels/<ID>/Shield/EventType

Method	PUT
Description	Set the shielded area of a specified channel according to event type.
Query	None.
Request	<u>XML_Shield</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the URI indicates the channel ID.

The EventType in the URI indicates the event type, e.g., behavior, faceSnap, humanRecognition, etc. The supported event types of this function refer to node <**shieldEventType**> in [XML_ChannelEventCap](#).

E.90 /ISAPI/MasterSlaveTracking/capabilities

Get the smart linkage capability.

Request URI Definition

Table C-126 GET /ISAPI/MasterSlaveTracking/capabilities

Method	GET
Description	Get the smart linkage capability.
Query	None.
Request	None.
Response	<u>XML_MasterSlaveTrackingCap</u>

E.91 /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/<ID>

Operations about the basic configuration of the PTZ camera.

Request URI Definition

Table C-127 GET /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/<ID>

Method	GET
Description	Get the basic configuration parameters of the PTZ camera.
Query	None.
Request	None.
Response	<u>XML_SlaveCamera</u>

Table C-128 PUT /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/<ID>

Method	PUT
Description	Set the basic configuration parameters for the PTZ camera.
Query	None.
Request	<u>XML_SlaveCamera</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The first <ID> in the request URI refers to the channel No. The second <ID> in the request URI refers to the PTZ camera ID which ranges from 1 to 4.

E.92 /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/<ID>/tracking

Operations about the tracking configuration of the PTZ camera.

Request URI Definition

Table C-129 GET /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/<ID>/tracking

Method	GET
Description	Get the tracking configuration parameters of the PTZ camera.
Query	None.
Request	None.
Response	<u>XML_Tracking</u>

Table C-130 PUT /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/<ID>/tracking

Method	PUT
Description	Set the tracking configuration parameters for the PTZ camera.
Query	None.
Request	<u>XML_Tracking</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The first <ID> in the request URI refers to the channel No. The second <ID> in the request URI refers to the PTZ camera ID.

E.93 /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/<ID>/trackingRatio

Set the tracking zoom ratio for the PTZ camera.

Request URI Definition

Table C-131 PUT /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/<ID>/trackingRatio

Method	PUT
Description	Set the tracking zoom ratio for the PTZ camera.
Query	None.
Request	None.
Response	<u>XML ResponseStatus</u>

E.94 /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/capabilities

Get the configuration capability of basic parameters for the PTZ camera.

Request URI Definition

Table C-132 GET /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCamera/capabilities

Method	GET
Description	Get the configuration capability of basic parameters for the PTZ camera.
Query	None.
Request	None.
Response	<u>XML Cap_SlaveCamera</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.95 /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCameraStatus

Get the connection status of the PTZ camera.

Request URI Definition

Table C-133 GET /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCameraStatus

Method	GET
Description	Get the connection status of the PTZ camera.
Query	None.
Request	None.
Response	<i>XML_SlaveCameraStatus</i>

Remarks

The <ID> in the request URI refers to the PTZ camera ID.

E.96 /ISAPI/MasterSlaveTracking/channels/<ID>/tracking/capabilities

Get the tracking configuration capability of the PTZ camera.

Request URI Definition

Table C-134 GET /ISAPI/MasterSlaveTracking/channels/<ID>/tracking/capabilities

Method	GET
Description	Get the tracking configuration capability of the PTZ camera.
Query	None.
Request	None.
Response	<i>XML_Cap_Tracking</i>

E.97 /ISAPI/MasterSlaveTracking/linkedTracking/capabilities? format=json

Get capability of configuring advanced parameters of linked tracking capture.

Request URI Definition

Table C-135 GET /ISAPI/MasterSlaveTracking/linkedTracking/capabilities?format=json

Method	GET
Description	Get capability of configuring advanced parameters of linked tracking capture.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_LinkedTrackingCap</u> Failed: <u>JSONResponseStatus</u>

E.98 /ISAPI/MasterSlaveTracking/linkedTracking?format=json

Get or set the advanced parameters of linked tracking capture.

Request URI Definition

Table C-136 GET /ISAPI/MasterSlaveTracking/linkedTracking?format=json

Method	GET
Description	Get the advanced parameters of linked tracking capture.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_LinkedTracking</u> Failed: <u>JSONResponseStatus</u>

Table C-137 PUT /ISAPI/MasterSlaveTracking/linkedTracking?format=json

Method	PUT
Description	Set the advanced parameters of linked tracking capture.
Query	format: determine the format of request or response message.
Request	<u>JSON_LinkedTracking</u>
Response	<u>JSONResponseStatus</u>

E.99 /ISAPI/PTZCtrl/channels/<ID>

Get or set the single PTZ control parameters.

Request URI Definition

Table C-138 GET /ISAPI/PTZCtrl/channels/<ID>

Method	GET
Description	Get the single PTZ control parameters.
Query	None
Request	None
Response	Succeeded: <u>XML_PTZChannel</u> Failed: <u>XML_ResponseStatus</u>

Table C-139 PUT /ISAPI/PTZCtrl/channels/<ID>

Method	PUT
Description	Set the single PTZ control parameters.
Query	None
Request	<u>XML_PTZChannel</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the URI refers to the channel No.

E.100 /ISAPI/PTZCtrl/channels/<ID>/auxcontrols/<ID>

Get or set the specified PTZ auxiliary status.

Request URI Definition

Table C-140 GET /ISAPI/PTZCtrl/channels/<ID>/auxcontrols/<ID>

Method	GET
Description	Get the specified PTZ auxiliary status.
Query	None.

Request	None.
Response	Succeeded: <u>XML_PTZAux</u> Failed: <u>XML_ResponseStatus</u>

Table C-141 PUT /ISAPI/PTZCtrl/channels/<ID>/auxcontrols/<ID>

Method	PUT
Description	Set the specified PTZ auxiliary status.
Query	None.
Request	<u>XML_PTZAux</u>
Response	<u>XML_ResponseStatus</u>

Remarks

- The first <ID> in the URI refers to the channel number.
- The second <ID> in refers to the auxiliary number, e.g., when there is only one wiper, the second <ID> is "1", when there are multiple wipers, the second <ID> varies according to the requirement.

E.101 /ISAPI/PTZCtrl/channels/<ID>/capabilities

Get the PTZ control capabilities.

Request URI Definition**Table C-142 GET /ISAPI/PTZCtrl/channels/ID/capabilities**

Method	GET
Description	Get the PTZ control capabilities.
Query	None
Request	None.
Response	Succeeded: <u>XML_PTZChanelCap</u> Failed: <u>XML_ResponseStatus</u>

E.102 /ISAPI/PTZCtrl/channels/<ID>/EagleFocusing/auto/capabilities?format=json

Get the capability of rapid focus automatic calibration.

Request URI Definition

Table C-143 GET /ISAPI/PTZCtrl/channels/<ID>/EagleFocusing/auto/capabilities?format=json

Method	GET
Description	Get the capability of rapid focus automatic calibration.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_AutoEagleFocusingCap</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URI refers to the channel ID.

E.103 /ISAPI/PTZCtrl/channels/<ID>/EagleFocusing/auto?format=json

Automatically calibrate for rapid focus.

Request URI Definition

Table C-144 GET /ISAPI/PTZCtrl/channels/<ID>/EagleFocusing/auto?format=json

Method	GET
Description	Automatically calibrate for rapid focus.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_AutoEagleFocusing</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URI refers to the channel ID.

E.104 /ISAPI/PTZCtrl/channels/<ID>/EagleFocusing/capabilities

Get the rapid focus capability.

Request URI Definition**Table C-145 GET /ISAPI/PTZCtrl/channels/<ID>/EagleFocusing/capabilities**

Method	GET
Description	Get the rapid focus capability.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_EagleFocusing</i> Failed: <i>XML_ResponseStatus</i>

Remarks

The <ID> in the URI refers to the channel ID.

E.105 /ISAPI/PTZCtrl/channels/<ID>/GPS?format=json

Control the device turn to corresponding place according to applied GPS information.

Request URI Definition**Table C-146 PUT /ISAPI/PTZCtrl/channels/<ID>/GPS?format=json**

Method	PUT
Description	Control the device turn to corresponding place according to applied GPS information.
Query	format: determine the format of request or response message.
Request	<i>JSON_GPSInfo</i>
Response	<i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URI refers to channel No.

E.106 /ISAPI/PTZCtrl/channels/<ID>/gyroVerify/capabilities?format=json

Get the gyroscope calibration capability.

Request URI Definition

Table C-147 GET /ISAPI/PTZCtrl/channels/<ID>/gyroVerify/ capabilities?format=json

Method	GET
Description	Get the gyroscope calibration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_GyroVerifyCap</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

The <ID> in the URI refers to channel No.

E.107 /ISAPI/PTZCtrl/channels/<ID>/gyroVerify?format=json

Set the gyroscope calibration parameters.

Request URI Definition

Table C-148 PUT /ISAPI/PTZCtrl/channels/<ID>/gyroVerify?format=json

Method	PUT
Description	Set the gyroscope calibration parameters.
Query	format: determine the format of request or response message.
Request	<u>JSON_GyroVerify</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the URI refers to channel No.

E.108 /ISAPI/PTZCtrl/channels/<ID>/maxelevation

Get or set the max. tilt-angle parameters.

Request URI Definition

Table C-149 GET /ISAPI/PTZCtrl/channels/<ID>/maxelevation

Method	GET
Description	Get the max. tilt-angle configuration parameters
Query	None.
Request	None.
Response	<i>XML_MaxElevation</i>

Table C-150 PUT /ISAPI/PTZCtrl/channels/<ID>/maxelevation

Method	PUT
Description	Set the max. tilt-angle
Query	None.
Request	<i>XML_MaxElevation</i>
Response	<i>XML_ResponseStatus</i>

Remarks

The <ID> in the URI indicates the channel ID.

E.109 /ISAPI/PTZCtrl/channels/<ID>/maxelevation/capabilities

Get the max. tilt-angle capability.

Request URI Definition

Table C-151 GET /ISAPI/PTZCtrl/channels/<ID>/maxelevation/capabilities

Method	GET
Description	Get the max. tilt-angle capability.
Query	None.

Request	None.
Response	<u>XML_Cap_MaxElevation</u>

Remarks

The <ID> in the URI indicates the channel ID.

E.110 /ISAPI/PTZCtrl/channels/<ID>/moveAutoTracking?format=json

Get or set moving object tracking status.

Request URI Definition

Table C-152 GET /ISAPI/PTZCtrl/channels/<ID>/moveAutoTracking?format=json

Method	GET
Description	Get moving object tracking status.
Query	format: determine the format of request or response message.
Request	None
Response	Succeeded: <u>JSON_MoveAutoTracking</u> Failed: <u>JSON_ResponseStatus</u>

Table C-153 PUT /ISAPI/PTZCtrl/channels/<ID>/moveAutoTracking?format=json

Method	PUT
Description	Set moving object tracking status.
Query	format: determine the format of request or response message.
Request	<u>JSON_MoveAutoTracking</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the PTZ channel ID.

E.111 /ISAPI/PTZCtrl/channels/<ID>/PTZOSDDisplay

Get or set the on-screen display parameters of the PTZ status.

Request URI Definition

Table C-154 GET /ISAPI/PTZCtrl/channels/<ID>/PTZOSDDisplay

Method	GET
Description	Get the on-screen display parameters of the PTZ status.
Query	None.
Request	None.
Response	Succeeded: <u>XML_PTZOSDDisplay</u> Failed: <u>XML_ResponseStatus</u>

Table C-155 PUT /ISAPI/PTZCtrl/channels/<ID>/PTZOSDDisplay

Method	PUT
Description	Set the on-screen display parameters of the PTZ status.
Query	None.
Request	<u>XML_PTZOSDDisplay</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the PTZ channel No.

E.112 /ISAPI/PTZCtrl/channels/<ID>/zoomFocus

Operations about zoom and focus coordinates configuration for zoom camera module.

Request URI Definition

Table C-156 GET /ISAPI/PTZCtrl/channels/<ID>/zoomFocus

Method	GET
Description	Get parameters of zoom and focus coordinates of zoom camera module.
Query	None.
Request	None.
Response	<u>XML_ZoomFocus</u>

Table C-157 PUT /ISAPI/PTZCtrl/channels/<ID>/zoomFocus

Method	PUT
Description	Set parameters of zoom and focus coordinates of zoom camera module.
Query	None.
Request	<u>XML_ZoomFocus</u>
Response	<u>XMLResponseStatus</u>

Remarks

The <ID> in the request URI refers to the channel ID.

E.113 /ISAPI/SDT/Management/capabilities?format=json

Get the intelligent management capability of the device.

Request URI Definition**Table C-158 GET /ISAPI/SDT/Management/capabilities?format=json**

Method	GET
Description	Get the intelligent management capability of the device to check the intelligent functions supported by the device.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_Cap_IntelliManagement</u> Failed: <u>JSON_ResponseStatus</u>

E.114 /ISAPI/SDT/Management/EventSearch?format=json

Search for event.

Request URI Definition

Table C-159 POST /ISAPI/SDT/Management/EventSearch?format=json

Method	POST
Description	Search for event.
Query	format: determine the format of request or response message.
Request	<u>JSON_EventSearchCond</u>
Response	Succeeded: <u>JSON_EventSearchResult</u> Failed: <u>JSON_ResponseStatus</u>

E.115 /ISAPI/SDT/Management/EventSearch/capabilities?format=json

Get the event search capability.

Request URI Definition

Table C-160 GET /ISAPI/SDT/Management/EventSearch/capabilities?format=json

Method	GET
Description	Get the event search capability.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_EventSearchCap</u>

E.116 /ISAPI/SDT/Management/IntelligentSearch/capabilities?format=json

Get intelligent search capability.

Request URI Definition

Table C-161 GET /ISAPI/SDT/Management/IntelligentSearch/capabilities?format=json

Method	GET
Description	Get intelligent search capability.
Query	format: determine the format of request or response message.

Request	None
Response	Succeeded: <i>JSON_IntelligentSearchCap</i> Failed: <i>JSON_ResponseStatus</i>

E.117 /ISAPI/SDT/Management/IntelligentSearch?format=json

Start intelligent search.

Request URI Definition

Table C-162 POST /ISAPI/SDT/Management/IntelligentSearch?format=json

Method	POST
Description	Start intelligent search.
Query	format: determine the format of request or response message.
Request	<i>JSON_IntelligentSearchCondition</i>
Response	Succeeded: <i>JSON_IntelligentSearchResult</i> Failed: <i>JSON_ResponseStatus</i>

E.118 /ISAPI/Security/adminAccesses

Get or set parameters of all protocols supported by device.

Request URI Definition

Table C-163 GET /ISAPI/Security/adminAccesses

Method	GET
Description	Get parameters of all protocols supported by device.
Query	None.
Request	None.
Response	Succeeded: <i>XML_AdminAccessProtocolList</i> Failed: <i>XML_ResponseStatus</i>

Table C-164 PUT /ISAPI/Security/adminAccesses

Method	PUT
Description	Set parameters of all protocols supported by device.
Query	None.
Request	<u>XML_AdminAccessProtocolList</u>
Response	<u>XMLResponseStatus</u>

E.119 /ISAPI/Security/adminAccesses/capabilities

Get device protocol capability.

Request URI Definition

Table C-165 GET /ISAPI/Security/adminAccesses/capabilities

Method	GET
Description	Get device protocol capability.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Cap_AdminAccessProtocolList</u> Failed: <u>XMLResponseStatus</u>

E.120 /ISAPI/Security/adminAccesses/<ID>

Get or set the parameters of a specific protocol that supported by device.

Request URI Definition

Table C-166 GET /ISAPI/Security/adminAccesses/<ID>

Method	GET
Description	Get the parameters of a specific protocol that supported by device.
Query	None.

Request	None.
Response	Succeeded: <u>XML_AdminAccessProtocol</u> Failed: <u>XML_ResponseStatus</u>

Table C-167 PUT /ISAPI/Security/adminAccesses/<ID>

Method	PUT
Description	Set the parameters of a specific protocol that supported by device.
Query	None.
Request	<u>XML_AdminAccessProtocol</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the protocol ID.

E.121 /ISAPI/Security/capabilities

Get the security capability of the device.

Request URI Definition**Table C-168 GET /ISAPI/Security/capabilities**

Method	GET
Description	Get the security capability of the device.
Query	username: user name, string, it should be encrypted.
Request	None.
Response	Succeeded: <u>XML_SecurityCap</u> Failed: <u>XML_ResponseStatus</u>

E.122 /ISAPI/Security/CloudUserManage/users/<ID>?format=json

Get, set, or delete the information of a cloud user.

Request URI Definition

Table C-169 GET /ISAPI/Security/CloudUserManage/users/<ID>?format=json

Method	GET
Description	Get the information of a cloud user.
Query	format: determine the format of request or response message.
Request	None
Response	Succeeded: <i>JSON_CloudUserManage</i> Failed: <i>JSONResponseStatus</i>

Table C-170 PUT /ISAPI/Security/CloudUserManage/users/<ID>?format=json

Method	PUT
Description	Set the information of a cloud user.
Query	format: determine the format of request or response message.
Request	<i>JSON_CloudUserManage</i>
Response	<i>JSONResponseStatus</i>

Table C-171 DELETE /ISAPI/Security/CloudUserManage/users/<ID>?format=json

Method	DELETE
Description	Delete the information of a cloud user.
Query	format: determine the format of request or response message.
Request	None
Response	<i>JSONResponseStatus</i>

Remarks

The <ID> in the request URI refers to user ID.

E.123 /ISAPI/Security/CloudUserManage/users/byType?format=json

Get information of cloud users by type.

Request URI Definition

Table C-172 POST /ISAPI/Security/CloudUserManage/users/byType?format=json

Method	POST
Description	Get information of cloud users by type.
Query	format: determine the format of request or response message.
Request	<u>JSON_GetUserInfoByType</u>
Response	Succeeded: <u>JSON_CloudUserManageList</u> Failed: <u>JSON_ResponseStatus</u>

E.124 /ISAPI/Security/CloudUserManage/users/capabilities? format=json

Get the capability of managing cloud users.

Request URI Definition

Table C-173 GET /ISAPI/Security/CloudUserManage/users/capabilities?format=json

Method	GET
Description	Get the capability of managing cloud users.
Query	format: determine the format of request or response message.
Request	None
Response	Succeeded: <u>JSON_CloudUserManageCap</u> Failed: <u>JSON_ResponseStatus</u>

E.125 /ISAPI/Security/CloudUserManage/users?format=json

Get information of all cloud users or add information of a cloud user.

Request URI Definition

Table C-174 GET /ISAPI/Security/CloudUserManage/users?format=json

Method	GET
Description	Get the information of all users.
Query	format: determine the format of request or response message.
Request	None
Response	Succeeded: <i>JSON_CloudUserManageList</i> Failed: <i>JSON_ResponseStatus</i>

Table C-175 POST /ISAPI/Security/CloudUserManage/users?format=json

Method	POST
Description	Add information of a cloud user.
Query	format: determine the format of request or response message.
Request	<i>JSON_CloudUserManage</i>
Response	<i>JSON_ResponseStatus</i> +id (optional, int, user ID).

E.126 /ISAPI/Security/deviceCertificate/certificateRevocation/capabilities?format=json

Get the configuration capability of the certificate expiry alarm.

Request URI Definition

Table C-176 GET /ISAPI/Security/deviceCertificate/certificateRevocation/capabilities?format=json

Method	GET
Description	Get the configuration capability of the certificate expiry alarm.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_CertificateRevocationCap</i> Failed: <i>JSON_ResponseStatus</i>

E.127 /ISAPI/Security/deviceCertificate/certificateRevocation?format=json

Get or set the configuration of certificate expiry alarm.

Request URI Definition

Table C-177 GET /ISAPI/Security/deviceCertificate/certificateRevocation?format=json

Method	GET
Description	Get the parameters of certificate expiry alarm.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_CertificateRevocation</i> Failed: <i>JSON_ResponseStatus</i>

Table C-178 PUT /ISAPI/Security/deviceCertificate/certificateRevocation?format=json

Method	PUT
Description	Set the parameters of certificate expiry alarm.
Query	format: determine the format of request or response message.
Request	<i>JSON_CertificateRevocation</i>
Response	<i>JSON_ResponseStatus</i>

E.128 /ISAPI/Security/doubleVerification/users/<ID>?format=json

Operations about a specified double verification user.

Request URI Definition

Table C-179 GET /ISAPI/Security/doubleVerification/users/<ID>?format=json

Method	GET
Description	Get a double verification user.
Query	format: determine the format of request or response message.

	security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	<u>JSON_User</u>

Table C-180 PUT /ISAPI/Security/doubleVerification/users/<ID>?format=json

Method	PUT
Description	Set a double verification user.
Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	<u>JSON_User</u>
Response	<u>JSON_ResponseStatus</u>

Table C-181 DELETE /ISAPI/Security/doubleVerification/users/<ID>?format=json

Method	DELETE
Description	Delete a double verification user.
Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.

Request	None.
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the URI indicates the user ID.

E.129 /ISAPI/Security/doubleVerification/users/capabilities?format=json

Get the double verification configuration capability.

Request URI Definition

Table C-182 GET /ISAPI/Security/doubleVerification/users/capabilities?format=json

Method	GET
Description	Get the double verification configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_UserCap</u>

E.130 /ISAPI/Security/doubleVerification/users?format=json

Get all double verification users and add a double verification user.

Request URI Definition

Table C-183 GET /ISAPI/Security/doubleVerification/users?format=json

Method	GET
Description	Get all double verification users.
Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates

	that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	<u>JSON_UserList</u>

Table C-184 POST /ISAPI/Security/doubleVerification/users?format=json

Method	POST
Description	Add a double verification user.
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	<u>JSON_User</u>
Response	Succeeded: <u>JSON_ResponseStatus</u> and <u>JSON_id</u> Failed: <u>JSON_ResponseStatus</u>

E.131 /ISAPI/Security/doubleVerification/UsersPermission/<ID>?format=json

Get and set the permission of double verification user.

Request URI Definition

Table C-185 GET /ISAPI/Security/doubleVerification/UsersPermission/<ID>?format=json

Method	GET
Description	Get the configuration parameters of double verification user permission.
Query	format : determine the format of request or response message.

Request	None.
Response	Succeeded: <u>JSON_UserPermission</u> Failed: <u>JSON_ResponseStatus</u>

Table C-186 PUT /ISAPI/Security/doubleVerification/UsersPermission/<ID>?format=json

Method	PUT
Description	Set the double verification user permission.
Query	format: determine the format of request or response message.
Request	<u>JSON_UserPermission</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the URI indicates the user ID.

E.132 /ISAPI/Security/doubleVerification/UsersPermission/capabilities?format=json

Get the capability of permission configuration for double verification user.

Request URI Definition**Table C-187 GET /ISAPI/Security/doubleVerification/UsersPermission/capabilities?format=json**

Method	GET
Description	Get the capability of permission configuration for double verification user.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_UserPermissionCap</u> Failed: <u>JSON_ResponseStatus</u>

E.133 /ISAPI/Security/doubleVerification?format=json

Get the capability of double verification enabling configuration.

Request URI Definition

Table C-188 GET /ISAPI/Security/doubleVerification?format=json

Method	GET
Description	Get the capability of double verification enabling configuration.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_DoubleVerification</u>

Table C-189 PUT /ISAPI/Security/doubleVerification?format=json

Method	PUT
Description	Enable or disable the double verification.
Query	format: determine the format of request or response message.
Request	<u>JSON_DoubleVerification</u>
Response	<u>JSON_ResponseStatus</u>

E.134 /ISAPI/Security/email/parameter/capabilities?format=json

Get recovery email configuration capability (only available in LAN and for admin user).

Request URI Definition

Table C-190 GET /ISAPI/Security/email/parameter/capabilities?format=json

Method	GET
Description	Get recovery email configuration capability.
Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.

Request	None.
Response	<u>JSON_SecurityEmailCap</u>

E.135 /ISAPI/Security/email/parameter?format=json

Operations about recovery email configuration (only available for LAN and for admin user).

Request URI Definition

Table C-191 GET /ISAPI/Security/email/parameter?format=json

Method	GET
Description	Get recovery email parameters.
Query	<p>format: determine the format of request or response message.</p> <p>security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode.</p> <p>iv: the initialization vector, and it is required when security is 1 or 2.</p>
Request	None.
Response	<u>JSON_SecurityEmail</u>

Table C-192 PUT /ISAPI/Security/email/parameter?format=json

Method	PUT
Description	Set recovery email parameters.
Query	<p>format: determine the format of request or response message.</p> <p>security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode.</p> <p>iv: the initialization vector, and it is required when security is 1 or 2.</p>

Request	<u>JSON_SecurityEmail</u>
Response	<u>JSON_ResponseStatus</u>

E.136 /ISAPI/Security/email/qrCode?format=json

Get the QR code of the configured recovery email (only available for LAN and for admin user).

Request URI Definition

Table C-193 GET /ISAPI/Security/email/qrCode?format=json

Method	GET
Description	Get the QR code of the configured recovery email.
Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	<u>JSON_SecurityEmailQrCode</u>

E.137 /ISAPI/Security/emailCertification?format=json

Reset password by the verification code via recovery email (only available for LAN and for admin user).

Request URI Definition

Table C-194 PUT /ISAPI/Security/emailCertification?format=json

Method	PUT
Description	Reset password by the verification code via recovery email.
Query	format: determine the format of request or response message.

Request	<u>JSON_EmailCertification</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

For the encryption of passwords and security answers, first transform them by UTF8, and then transcode them by BASE64, finally, encrypt them by AES128CB.

E.138 /ISAPI/Security/extern/capabilities

Get capability of other security configuration.

Request URI Definition

Table C-195 GET /ISAPI/Security/extern/capabilities

Method	GET
Description	Get capability of other security configuration.
Query	None.
Request	None.
Response	<u>XML_externSecurityCap</u>

E.139 /ISAPI/Security/illegalLoginLock

Get or set locking parameters of illegal login.

Request URI Definition

Table C-196 GET /ISAPI/Security/illegalLoginLock

Method	GET
Description	Get locking parameters of illegal login.
Query	None.
Request	None.
Response	Succeeded: <u>XML_IllegalLoginLock</u> Failed: <u>XML_ResponseStatus</u>

Table C-197 PUT /ISAPI/Security/illegalLoginLock

Method	PUT
Description	Set locking parameters of illegal login.
Query	None.
Request	<u>XML_IllegalLoginLock</u>
Response	<u>XML_ResponseStatus</u>

E.140 /ISAPI/Security/questionConfiguration

Get or set device security questions.

Request URI Definition

Table C-198 GET /ISAPI/Security/questionConfiguration

Method	GET
Description	Get device security questions.
Query	None.
Request	None.
Response	Succeeded: <u>XML_SecurityQuestion</u> Failed: <u>XML_ResponseStatus</u>

Table C-199 PUT /ISAPI/Security/questionConfiguration

Method	PUT
Description	Set device security questions.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	<u>XML_SecurityQuestion</u>
Response	<u>XML_ResponseStatus</u>

E.141 /ISAPI/Security/RTSPCertificate

Get or set parameters of devices' RTSP authentication mode.

Request URI Definition

Table C-200 GET /ISAPI/Security/RTSPCertificate

Method	GET
Description	Get parameters of devices' RTSP authentication mode.
Query	None.
Request	None.
Response	Succeeded: <u>XML_RTSPCertificate</u> Failed: <u>XML_ResponseStatus</u>

Table C-201 PUT /ISAPI/Security/RTSPCertificate

Method	PUT
Description	Set parameters of devices' RTSP authentication mode.
Query	None.
Request	<u>XML_RTSPCertificate</u>
Response	<u>XML_ResponseStatus</u>

E.142 /ISAPI/Security/RTSPCertificate/capabilities

Get the capability of devices' RTSP authentication mode.

Request URI Definition

Table C-202 GET /ISAPI/Security/RTSPCertificate/capabilities

Method	GET
Description	Get the capability of devices' RTSP authentication mode.
Query	None.
Request	None.
Response	Succeeded: <u>XML_RTSPCertificateCap</u>

	Failed: <i>XMLResponseStatus</i>
--	----------------------------------

E.143 /ISAPI/Security/UserPermission

Operations about the user permission of the device.

Request URI Definition

Table C-203 GET /ISAPI/Security/UserPermission

Method	GET
Description	Get the user permission of the device.
Query	None.
Request	None.
Response	Succeeded: <i>XML_UserPermissionList</i> Failed: <i>XMLResponseStatus</i>

Table C-204 PUT /ISAPI/Security/UserPermission

Method	PUT
Description	Set the user permission of the device.
Query	None.
Request	<i>XML_UserPermissionList</i>
Response	<i>XMLResponseStatus</i>

E.144 /ISAPI/Security/UserPermission/adminCap

Get the capability of managing the user permission.

Request URI Definition

Table C-205 GET /ISAPI/Security/UserPermission/adminCap

Method	GET
Description	Get the capability of managing the user permission.
Query	None.

Request	None.
Response	Succeeded: <u>XML_UserPermissionCap</u> Failed: <u>XMLResponseStatus</u>

Remarks

Only permissions that can be configured by the admin user should be returned.

E.145 /ISAPI/Security/UserPermission/capabilities

Get the configuration capability of all users' permission.

Request URI Definition

Table C-206 GET /ISAPI/Security/UserPermission/capabilities

Method	GET
Description	Get the configuration capability of all users' permission.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Cap_UserPermissionList</u> Failed: <u>XMLResponseStatus</u>

E.146 /ISAPI/Security/UserPermission/<ID>

Operations about a specific user's permission.

Request URI Definition

Table C-207 GET /ISAPI/Security/UserPermission/<ID>

Method	GET
Description	Get a specific user's permission.
Query	None.
Request	None.
Response	Succeeded: <u>XML_UserPermission</u> Failed: <u>XMLResponseStatus</u>

Table C-208 PUT /ISAPI/Security/UserPermission/<ID>

Method	PUT
Description	Set a specific user's permission.
Query	None.
Request	<u>XML_UserPermission</u>
Response	<u>XMLResponseStatus</u>

E.147 /ISAPI/Security/users

Get, set, or delete information of all users; add a new user.

Request URI Definition

Table C-209 GET /ISAPI/Security/users

Method	GET
Description	Get information of all users.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	None
Response	Succeeded: <u>XML_UserList</u> Failed: <u>XMLResponseStatus</u>

Table C-210 PUT /ISAPI/Security/users

Method	PUT
Description	Set information of all users.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates

	that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	<u>XML_UserList</u>
Response	<u>XML_ResponseStatus</u>

Table C-211 DELETE /ISAPI/Security/users

Method	DELETE
Description	Delete information of all users.
Query	None
Request	None
Response	<u>XML_ResponseStatus</u>

Table C-212 POST /ISAPI/Security/users

Method	POST
Description	Add a user.
Query	security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	<u>XML_User</u>
Response	<u>XML_ResponseStatus</u>

Remarks

- A default account "admin" must be provided with the administrator permission and it cannot be deleted.
- Passwords can only be uploaded and cannot be displayed when getting the user list of the device.
- The user ID should be returned when adding users to the user list.

Example

Sample Code for Adding User

```
POST /ISAPI/Security/users HTTP/1.1
Host: 10.17.132.49
Content-Length: 335
Connection: Keep-Alive
Authorization: Digest username="admin",
realm="DS-2CD2F12FWD-IWS",
nonce="4e6a4d774d6a63304f544936593255335a474d334f54673d",
uri="/ISAPI/Security/users",
cnonce="178e14d4977c835db891b5392b5d0a67",
nc=00000015,
response="d92518fb00cd2772cf212f5326b251be",
qop="auth"
```

```
<?xml version="1.0" encoding="UTF-8"?>
<User>
<id>0</id>
<userName>test</userName>
<password>1qaz2wsx</password>
<bondIpList>
<bondIp>
<id>1</id>
<ipAddress>0.0.0.0</ipAddress>
<ipv6Address>::</ipv6Address>
</bondIp>
</bondIpList>
<macAddress></macAddress>
<userLevel>Operator</userLevel>
<attribute>
<inherent>false</inherent>
</attribute>
</User>
```

```
HTTP/1.1 200 OK
Date: Tue, 14 Mar 2017 20:24:31 GMT
Server: App-webs/
Connection: close
Content-Length: 288
Content-Type: application/xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus version="2.0" xmlns="http://www.isapi.com/ver20/XMLSchema">
<requestURL>/ISAPI/Security/users</requestURL>
<statusCode>1</statusCode>
<statusString>OK</statusString>
<id>2</id>
<subStatusCode>ok</subStatusCode>
</ResponseStatus>
```

Example

Sample Code for Setting User Permission

```
PUT /ISAPI/Security/UserPermission/2 HTTP/1.1
Host: 10.17.132.49
Content-Length: 891
Connection: Keep-Alive
Authorization: Digest username="admin",
realm="DS-2CD2F12FWD-IWS",
nonce="4e6a4d774d6a63304f544936593255335a474d334f54673d",
uri="/ISAPI/Security/UserPermission/2",
cnonce="178e14d4977c835db891b5392b5d0a67",
nc=00000016,
response="f69a20db00b36e9bb46c26c06e9bae80",
qop="auth"

<?xml version="1.0" encoding="utf-8"?>
<UserPermission>
  <id>2</id>
  <userID>2</userID>
  <userType>operator</userType>
  <remotePermission>
    <parameterConfig>false</parameterConfig>
    <logOrStateCheck>true</logOrStateCheck>
    <upgrade>false</upgrade>
    <voiceTalk>true</voiceTalk>
    <restartOrShutdown>false</restartOrShutdown>
    <alarmOutOrUpload>false</alarmOutOrUpload>
    <contorlLocalOut>false</contorlLocalOut>
    <transParentChannel>false</transParentChannel>
    <preview>true</preview>
    <record>true</record>
    <ptzControl>true</ptzControl>
    <playBack>true</playBack>
    <videoChannelPermissionList>
      <videoChannelPermission>
        <id>1</id>
        <preview>true</preview>
        <record>true</record>
        <playBack>true</playBack>
      </videoChannelPermission>
    </videoChannelPermissionList>
    <ptzChannelPermissionList>
      <ptzChannelPermission>
        <id>1</id>
        <ptzControl>true</ptzControl>
      </ptzChannelPermission>
    </ptzChannelPermissionList>
  </remotePermission>
</UserPermission>
```

```
HTTP/1.1 200 OK
Date: Tue, 14 Mar 2017 20:24:31 GMT
Server: App-webs/
Connection: close
```

Content-Length: 288
 Content-Type: application/xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus version="2.0" xmlns="http://www.isapi.com/ver20/XMLSchema">
    <requestURL>/ISAPI/Security/UserPermission/2</requestURL>
    <statusCode>1</statusCode>
    <statusString>OK</statusString>
    <subStatusCode>ok</subStatusCode>
</ResponseStatus>
```

E.148 /ISAPI/Security/users/<ID>

Get, set, or delete information of a specific user.

Request URI Definition

Table C-213 GET /ISAPI/Security/users/<ID>

Method	GET
Description	Get a specific user's configuration of the device.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	Succeeded: <u>XML_User</u> Failed: <u>XMLResponseStatus</u>

Table C-214 PUT /ISAPI/Security/users/<ID>

Method	PUT
Description	Get information of a specific user.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates

	that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	<u>XML_User</u>
Response	<u>XML_ResponseStatus</u>

Table C-215 DELETE /ISAPI/Security/users/<ID>

Method	DELETE
Description	Delete information of a specific user.
Query	security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2. loginPassword: encrypted administrator password.
Request	None
Response	<u>XML_ResponseStatus</u>

Remarks

- The <ID> in the request URI refer to user ID.
- The administrator account cannot be deleted.

E.149 /ISAPI/Security/users/<ID>/capabilities

Get the configuration capability of a specific user.

Request URI Definition

Table C-216 GET /ISAPI/Security/users/<ID>/capabilities

Method	GET
Description	Get the configuration capability of a specific user.
Query	security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is

	1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	Succeeded: <u>XML_Cap_User</u> Failed: <u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the user ID.

E.150 /ISAPI/Security/webCertificate

Get or set the certificate type parameters of web service.

Request URI Definition

Table C-217 GET /ISAPI/Security/webCertificate

Method	GET
Description	Get the certificate type parameters of web service.
Query	None.
Request	None.
Response	Succeeded: <u>XML_WebCertificate</u> Failed: <u>XML_ResponseStatus</u>

Table C-218 PUT /ISAPI/Security/webCertificate

Method	PUT
Description	Set the certificate type parameters of web service.
Query	None.
Request	<u>XML_WebCertificate</u>
Response	<u>XML_ResponseStatus</u>

E.151 /ISAPI/Streaming/channels/<ID>

Operations about the encoding configurations of a specific channel.

Request URI Definition

Table C-219 GET /ISAPI/Streaming/channels/<ID>

Method	GET
Description	Get the encoding parameters of a specific channel.
Query	None.
Request	None.
Response	Succeeded: <i>XML_StreamingChannel</i> Failed: <i>XML_ResponseStatus</i>

Table C-220 PUT /ISAPI/Streaming/channels/<ID>

Method	PUT
Description	Set the encoding parameters of a specific channel.
Query	None.
Request	<i>XML_StreamingChannel</i>
Response	<i>XML_ResponseStatus</i>

Table C-221 DELETE /ISAPI/Streaming/channels/<ID>

Method	DELETE
Description	Delete the encoding parameters of a specific channel.
Query	None
Request	None.
Response	<i>XML_ResponseStatus</i>

Remarks

The <ID> in the URI refers to the channel ID.

Example

Interaction Example of Getting Encoding Parameters of A Specific Channel

GET /ISAPI/Streaming/channels/444 HTTP/1.1

...

HTTP/1.1 200 OK

Content-Type: application/xml; charset="UTF-8"

Content-Length: ISAPI

```
<?xml version="1.0" encoding="UTF-8"?>
<StreamingChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>444</id>
  <channelName>Input 1 MPEG-4 ASP</channelName>
  <enabled>true</enabled>
  <Transport>
    <rtspPortNo>554</rtspPortNo>
    <maxPacketSize>1446</maxPacketSize>
    <ControlProtocolList>
      <ControlProtocol>
        <streamingTransport>RTSP</streamingTransport>
      </ControlProtocol>
      <ControlProtocol>
        <streamingTransport>HTTP</streamingTransport>
      </ControlProtocol>
    </ControlProtocolList>
  </Transport>
  <Video>
    <enabled>true</enabled>
    <videoInputChannelID>2</videoInputChannelID>
    <videoCodecType>MPEG4</videoCodecType>
    <videoScanType>progressive</videoScanType>
    <videoResolutionWidth>640</videoResolutionWidth>
    <videoResolutionHeight>480</videoResolutionHeight>
    <videoPositionX>0</videoPositionX>
    <videoPositionY>0</videoPositionY>
    <videoQualityControlType>CBR</videoQualityControlType>
    <constantBitRate>2000</constantBitRate>
    <maxFrameRate>2500</maxFrameRate>
    <keyFrameInterval>1000</keyFrameInterval>
    <rotationDegree>0</rotationDegree>
    <mirrorEnabled>false</mirrorEnabled>
    <snapShotImageType>JPEG</snapShotImageType>
  </Video>
  <Audio>
    <enabled>false</enabled>
    <audioInputChannelID>2</audioInputChannelID>
    <audioCompressionType>G.726</audioCompressionType>
    <audioBitRate>24</audioBitRate>
    <audioSamplingRate>8</audioSamplingRate>
  </Audio>
</StreamingChannel>
```

E.152 /ISAPI/Streaming/channels/<ID>/capabilities

Get encoding capability of a specific channel.

Request URI Definition

Table C-222 GET /ISAPI/Streaming/channels/<ID>/capabilities

Method	GET
Description	Get the encoding capability of a specific channel.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_StreamingChannel</i> Failed: <i>XML_ResponseStatus</i>

Remarks

The <ID> in the URI refers to the channel ID.

Example

Interaction Example of Getting Encoding Capability of A Specific Channel

```
GET /ISAPI/Streaming/channels/444/capabilities HTTP/1.1
```

...

HTTP/1.1 200 OK

Content-Type: application/xml; charset="UTF-8"

Content-Length: ISAPI

```
<?xml version="1.0" encoding="UTF-8"?>
<StreamingChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id opt="111,222,333,444">444</id>
  <channelName min="0" max="64">Input 1 MPEG-4 ASP</channelName>
  <enabled opt="true,false" def="true">true</enabled>
  <Transport>
    <rtspPortNo min="0" max="65535" def="554">554</rtspPortNo>
    <maxPacketSize min="0" max="1500">1446</maxPacketSize>
    <audioPacketLength min="0" max="5000"/>
    <audioInboundPacketLength min="0" max="5000"/>
    <audioInboundPortNo min="0" max="65535"/>
    <videoSourcePortNo min="0" max="65535"/>
    <audioSourcePortNo min="0" max="65535"/>
  <ControlProtocolList>
    <ControlProtocol>
      <streamingTransport opt="RTSP/RTP,HTTP">RTSP</streamingTransport>
    </ControlProtocol>
    <ControlProtocol>
      <streamingTransport opt="RTSP/RTP,HTTP">HTTP</streamingTransport>
    </ControlProtocol>
  </ControlProtocolList>
  <Unicast>
    <enabled opt="true,false" def="false"/>
    <rtpTransportType opt="RTP/UDP,RTP/TCP"/>
  
```

```

</Unicast>
<Multicast>
    <enabled opt="true,false" def="false"/>
    <userTriggerThreshold/>
    <videoDestPortNo min="0" max="65535"/>
    <audioDestPortNo min="0" max="65535"/>
    <destIPAddress min="8" max="16"/>
    <destIPv6Address min="15" max="39"/>
    <ttl min="0" max="127" def="1"/>
</Multicast>
<Security>
    <enabled opt="true,false" def="false"/>
</Security>
</Transport>
<Video>
    <enabled opt="true,false">true</enabled>
    <videoInputChannelID opt="1,2,3,4">2</videoInputChannelID>
    <videoCodecType opt="MJPEG,MPEG4">MPEG4</videoCodecType>
    <videoScanType opt="interlaced,progressive">progressive</videoScanType>
    <videoResolutionWidth min="0" max="640">640</videoResolutionWidth>
    <videoResolutionHeight min="0" max="480">480</videoResolutionHeight>
    <videoPositionX min="0" max="640">0</videoPositionX>
    <videoPositionY min="0" max="480">0</videoPositionY>
    <videoQualityControlType opt="CBR,VBR">CBR</videoQualityControlType>
    <constantBitRate min="50" max="4000" dynamic="true">2000</constantBitRate>
    <maxFrameRate opt="2500,1250,625,312,156,78, 830" dynamic="true">2500</maxFrameRate>
    <keyFrameInterval min="0", max="10000">1000</keyFrameInterval>
    <rotationDegree opt="0,90,180,270" def="0">0</rotationDegree>
    <mirrorEnabled opt="true,false" def="false">false</mirrorEnabled>
    <snapShotImageType opt="JPEG" def="JPEG">JPEG</snapShotImageType>
</Video>
<Audio>
    <enabled opt="true,false" def="false">false</enabled>
    <audioInputChannelID opt="1,2,3,4">2</audioInputChannelID>
    <audioCompressionType opt="G.726,G.711ulaw" def="G.726">G.726</audioCompressionType>
    <audioBitRate opt="16,24,32,40" def="32" dynamic="true">24</audioBitRate>
    <audioSamplingRate opt="8" dynamic="true">8</audioSamplingRate>
    <audioResolution opt="3,4,5,6" dynamic="true"/>
</Audio>
</StreamingChannel>

```

E.153 /ISAPI/Streaming/channels/<ID>/smartOverlap/capabilities?format=json

Get the capability of configuring the stream for displaying VCA rules of smart events.

Request URI Definition

Table C-223 GET /ISAPI/Streaming/channels/<ID>/smartOverlap/capabilities?format=json

Method	GET
Description	Get the capability of configuring the stream for displaying VCA rules of smart events.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_SmartOverlapCap</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.154 /ISAPI/Streaming/channels/<ID>/smartOverlap?format=json

Get or set the stream parameters for displaying VCA rules of smart events.

Request URI Definition

Table C-224 GET /ISAPI/Streaming/channels/<ID>/smartOverlap?format=json

Method	GET
Description	Get the stream parameters for displaying VCA rules of smart events.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_SmartOverlap</u> Failed: <u>JSON_ResponseStatus</u>

Table C-225 PUT /ISAPI/Streaming/channels/<ID>/smartOverlap?format=json

Method	PUT
Description	Set the stream parameters for displaying VCA rules of smart events.
Query	format: determine the format of request or response message.

Request	<u>JSON_SmartOverlap</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.155 /ISAPI/System/accessDevice/associatedChannel?format=json

Get the linked channel informaiton of added IoT device.

Request URI Definition

Table C-226 GET /ISAPI/System/accessDevice/associatedChannel?format=json

Method	GET
Description	Get the linked channel informaiton of added IoT device.
Query	format: determine the format of request or response message.
Request	<u>JSON_IOT_ChannelInfoList</u>
Response	<u>JSON_AssociatedChannelList</u>

E.156 /ISAPI/System/algorithmsVersion

Get the version information of algorithm library.

Request URI Definition

Table C-227 GET /ISAPI/System/algorithmsVersion

Method	GET
Description	Get the version information of algorithm library.
Query	None
Request	None
Response	Succeeded: <u>XML_AlgorithmsVersion</u> Failed: <u>XML_ResponseStatus</u>

E.157 /ISAPI/System/autoMaintenance/capabilities?format=json

Get the capability of automatic maintenance configuration.

Request URI Definition

Table C-228 GET /ISAPI/System/autoMaintenance/capabilities?format=json

Method	GET
Description	Get the capability of automatic maintenance configuration.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_AutoMaintenanceCap</u>

E.158 /ISAPI/System/autoMaintenance?format=json

Get automatic maintenance configuration parameters.

Request URI Definition

Table C-229 GET /ISAPI/System/autoMaintenance?format=json

Method	GET
Description	Get automatic maintenance configuration parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_AutoMaintenance</u>

Table C-230 PUT /ISAPI/System/autoMaintenance?format=json

Method	PUT
Description	Set automatic maintenance parameters.
Query	format: determine the format of request or response message.
Request	<u>JSON_AutoMaintenance</u>
Response	<u>JSON_ResponseStatus</u>

E.159 /ISAPI/System/autoSwitch/capabilities?format=json

Get configuration capability of auto power on or off.

Request URI Definition

Table C-231 GET /ISAPI/System/autoSwitch/capabilities?format=json

Method	GET
Description	Get configuration capability of auto power on or off.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_AutoSwitchCap</u> Failed: <u>JSON_ResponseStatus</u>

E.160 /ISAPI/System/autoSwitch?format=json

Get or set parameters of auto power on or off.

Request URI Definition

Table C-232 GET /ISAPI/System/autoSwitch?format=json

Method	GET
Description	Get parameters of auto power on or off.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_AutoSwitch</u> Failed: <u>JSON_ResponseStatus</u>

Table C-233 PUT /ISAPI/System/autoSwitch?format=json

Method	PUT
Description	Set parameters of auto power on or off.
Query	None.

Request	<u>JSON_AutoSwitch</u>
Response	<u>JSON_ResponseStatus</u>

E.161 /ISAPI/System/BVCorrect/channels/<ID>/capabilities?format=json

Get the configuration capability of camera correction.

Request URL Definition**Table C-234 GET /ISAPI/System/BVCorrect/channels/<ID>/capabilities?format=json**

Method	GET
Description	Get the configuration capability of camera correction.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_BVCorrectParamCap</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URL refers to the channel No.

E.162 /ISAPI/System/BVCorrect/channels/<ID>?format=json

Operations about the camera correction parameters.

Request URL Definition**Table C-235 GET /ISAPI/System/BVCorrect/channels/<ID>?format=json**

Method	GET
Description	Get the camera correction parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_BVCorrectParam</u> Failed: <u>JSON_ResponseStatus</u>

Table C-236 PUT /ISAPI/System/BVCorrect/channels/<ID>?format=json

Method	PUT
Description	Set the camera correction parameters.
Query	format: determine the format of request or response message.
Request	<u>JSON_BVCorrectParam</u>
Response	<u>JSONResponseStatus</u>

Remarks

The <ID> in the request URL refers to the channel No.

E.163 /ISAPI/System/capabilities

Get device capability.

Request URI Definition**Table C-237 GET /ISAPI/System/capabilities**

Method	GET
Description	Get device capability.
Query	None
Request	None.
Response	Succeeded: <u>XML_DeviceCap</u> Failed: <u>XML_ResponseStatus</u>

**E.164 /ISAPI/System/channels/<ID>/vibrationDetection/capabilities?
format=json**

Get the capability of vibration detection.

Request URI Definition**Table C-238 GET /ISAPI/System/channels/<ID>/vibrationDetection/capabilities?format=json**

Method	GET
Description	Get the capability of vibration detection.

Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_VibrationDetectionCap</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

The <ID> in the URI refers to the channel ID.

E.165 /ISAPI/System/channels/<ID>/vibrationDetection?format=json

Get or set the vibration detection parameters.

Request URI Definition

Table C-239 GET /ISAPI/System/channels/<ID>/vibrationDetection?format=json

Method	GET
Description	Get vibration detection parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_VibrationDetection</u> Failed: <u>JSON_ResponseStatus</u>

Table C-240 PUT /ISAPI/System/channels/<ID>/vibrationDetection?format=json

Method	PUT
Description	Set vibration detection parameters.
Query	format: determine the format of request or response message.
Request	<u>JSON_VibrationDetection</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the URI refers to the channel ID.

E.166 /ISAPI/System/deviceInfo

Operations about the device information.

Request URI Definition

Table C-241 GET /ISAPI/System/deviceInfo

Method	GET
Description	Get the device information.
Query	None
Request	None.
Response	Succeeded: <i>XML_DeviceInfo</i> Failed: <i>XML_ResponseStatus</i>

Table C-242 PUT /ISAPI/System/deviceInfo

Method	PUT
Description	Set the device information.
Query	None
Request	<i>XML_DeviceInfo</i>
Response	<i>XML_ResponseStatus</i>

E.167 /ISAPI/System/deviceInfo/capabilities

Get the device information configuration capability.

Request URI Definition

Table C-243 GET /ISAPI/System/deviceInfo/capabilities

Method	GET
Description	Get the device information configuration capability.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_DeviceInfo</i>

	Failed: <u>XMLResponseStatus</u>
--	--

E.168 /ISAPI/System/deviceInfo/characteristicCode?format=json

Get the device attribute code.

Request URI Definition

Table C-244 GET /ISAPI/System/deviceInfo/characteristicCode?format=json

Method	GET
Description	Get the device attribute code.
Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	Succeeded: <u>JSON_Characteristic</u> Failed: <u>JSONResponseStatus</u>

E.169 /ISAPI/System/DeviceLanguage

Operations about device language configuration.

Request URI Definition

Table C-245 GET /ISAPI/System/DeviceLanguage

Method	GET
Description	Get device language configuration parameters.
Query	None.

Request	None.
Response	Succeeded: <u>XML_DeviceLanguage</u> Failed: <u>XML_ResponseStatus</u>

Table C-246 PUT /ISAPI/System/DeviceLanguage

Method	PUT
Description	Set device language parameters.
Query	None.
Request	<u>XML_DeviceLanguage</u>
Response	<u>XML_ResponseStatus</u>

E.170 /ISAPI/System/DeviceLanguage/capabilities

Get the capability of device language configuration.

Request URI Definition

Table C-247 GET /ISAPI/System/DeviceLanguage/capabilities

Method	GET
Description	Get the capability of device language configuration.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Cap_DeviceLanguage</u> Failed: <u>XML_ResponseStatus</u>

E.171 /ISAPI/System/diagnosedData/parameter

Get or set the parameters of exporting device diagnose information.

Request URI Definition**Table C-248 GET /ISAPI/System/diagnosedData/parameter**

Method	GET
Description	Get the parameters of exporting device diagnose information.
Query	None.
Request	None.
Response	Succeeded: <i>XML_DiagnosedDataParameter</i> Failed: <i>XMLResponseStatus</i>

Table C-249 PUT /ISAPI/System/diagnosedData/parameter

Method	PUT
Description	Set the parameters of exporting device diagnose information.
Query	None.
Request	<i>XML_DiagnosedDataParameter</i>
Response	<i>XMLResponseStatus</i>

Remarks

The speed dome does not support getting parameters of exporting diagnose information.

E.172 /ISAPI/System/diagnosedData/parameter/capabilities

Get the capability of exporting device diagnose information.

Request URI Definition**Table C-250 GET /ISAPI/System/diagnosedData/parameter/capabilities**

Method	GET
Description	Get the capability of exporting device diagnose information.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_DiagnosedDataParameter</i> Failed: <i>XMLResponseStatus</i>

E.173 /ISAPI/System/diagnosedData/server?format=json

Get or set the diagnostic server parameters.

Request URI Definition**Table C-251 GET /ISAPI/System/diagnosedData/server?format=json**

Method	GET
Description	Get the diagnostic server configuration.
Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	Succeeded: <u>JSON_DiagnosedDataServerList</u> Failed: <u>JSON_ResponseStatus</u>

Table C-252 PUT /ISAPI/System/diagnosedData/server?format=json

Method	PUT
Description	Set the diagnostic server parameters.
Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	<u>JSON_DiagnosedDataServerList</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The following nodes in the message ***JSON_DiagnosedDataServerList*** should be encrypted: **password**, **userName**, **consoleCommand**, and **compressionKey**.

E.174 /ISAPI/System/diagnosedData/server/capabilities?format=json

Get the diagnostic server capability.

Request URI Definition

Table C-253 GET /ISAPI/System/diagnosedData/server/capabilities?format=json

Method	GET
Description	Get the diagnostic server capability.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_DiagnosedDataServerCap</i> Failed: <i>JSON_ResponseStatus</i>

E.175 /ISAPI/System/diagnosedData/server/test?format=json

Test the diagnostic server.

Request URI Definition

Table C-254 POST /ISAPI/System/diagnosedData/server/test?format=json

Method	POST
Description	Test the diagnostic server.
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.

Request	<i>JSON_TestDescription</i>
Response	<i>JSON_ResponseStatus</i>

Remarks

The two nodes **password** and **userName** in the message *JSON_DiagnosedDataServerList* should be encrypted.

E.176 /ISAPI/System/externalDevice/capabilities

Get peripherals configuration capability.

Request URI Definition

Table C-255 GET /ISAPI/System/externalDevice/capabilities

Method	GET
Description	Get peripherals configuration capability.
Query	None.
Request	None.
Response	<i>XML_Cap_ExternalDevice</i>

E.177 /ISAPI/System/factoryReset?mode=

Restore the device to default settings.

Request URI Definition

Table C-256 PUT /ISAPI/System/factoryReset?mode=

Method	PUT
Description	Restore the device to default settings.
Query	mode: The recovery mode, which contains "full", "basic", "part", and "other" mode. The default mode is "full", and all the device parameters will be restored to default settings; for "basic" mode, the device parameters, except network parameters, will be restored to default settings; for "other" mode, the device parameters, except network and user parameters, will be restored to default settings; for "part" mode, only a part of default settings will be restored.

Request	None.
Response	<u>XML_ResponseStatus</u>

Remarks

- For access control devices, this URI is only supported by facial recognition terminal, and the "full" and "basic" recovery modes are not supported by passthrough method.
- For facial recognition terminal, if the recovery mode is set to "part", i.e., /ISAPI/System/factoryReset?mode=part, the settings of card, face, event, fingerprint, schedule template, network, user name, and password will be reserved without being restored.

E.178 /ISAPI/System/GPSCalibration/channels/<ID>/capabilities?format=json

Get GPS calibration configuration capability.

Request URI Definition

Table C-257 GET /ISAPI/System/GPSCalibration/channels/<ID>/capabilities?format=json

Method	GET
Description	Get GPS calibration configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_GPSCalibrationCap</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the video channel ID.

E.179 /ISAPI/System/GPSCalibration/channels/<ID>/points/<PID>?format=json

Get, set or delete configuration parameters of a single GPS calibration point.

Request URI Definition

Table C-258 GET /ISAPI/System/GPSCalibration/channels/<ID>/points/<PID>?format=json

Method	GET
Description	Get the configuration parameters of a single GPS calibration point.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON GPSCalibration</i> Failed: <i>JSON ResponseStatus</i>

Table C-259 PUT /ISAPI/System/GPSCalibration/channels/<ID>/points/<PID>?format=json

Method	PUT
Description	Set the configuration parameters of a single GPS calibration point.
Query	format: determine the format of request or response message.
Request	<i>JSON GPSCalibration</i>
Response	<i>JSON ResponseStatus</i>

Table C-260 DELETE /ISAPI/System/GPSCalibration/channels/<ID>/points/<PID>?format=json

Method	DELETE
Description	Delete the configuration parameters of a single GPS calibration point.
Query	format: determine the format of request or response message.
Request	None.
Response	<i>JSON ResponseStatus</i>

Remarks

The first <ID> in the request URI refers to the video channel ID and the second <ID> refers to calibration point ID.

E.180 /ISAPI/System/GPSCalibration/channels/<ID>/points? format=json

Get, set or delete configuration parameters of multiple GPS calibration points in a batch.

Request URI Definition

Table C-261 GET /ISAPI/System/GPSCalibration/channels/<ID>/points?format=json

Method	GET
Description	Get configuration parameters of multiple GPS calibration points in a batch.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_GPSCalibrationList</u> Failed: <u>JSONResponseStatus</u>

Table C-262 PUT /ISAPI/System/GPSCalibration/channels/<ID>/points?format=json

Method	PUT
Description	Set configuration parameters of multiple GPS calibration points in a batch.
Query	format: determine the format of request or response message.
Request	<u>JSON_GPSCalibrationList</u>
Response	<u>JSONResponseStatus</u>

Table C-263 DELETE /ISAPI/System/GPSCalibration/channels/<ID>/points?format=json

Method	DELETE
Description	Delete configuration parameters of multiple GPS calibration points in a batch.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSONResponseStatus</u>

Remarks

The <ID> in the request URI refers to the video channel ID.

E.181 /ISAPI/System/GPSPTZCalibration/channels/<ID>/calibration?format=json

Enable GPS calibration.

Request URI Definition

Table C-264 PUT /ISAPI/System/GPSPTZCalibration/channels/<ID>/calibration?format=json

Method	PUT
Description	Enable GPS calibration.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSONResponseStatus</u>

Remarks

The <ID> in the request URI refers to the video channel ID.

E.182 /ISAPI/System/GPSVerification/channels/<ID>/display?format=json

Get or set the parameters of displaying GPS calibration verification result.

Request URI Definition

Table C-265 GET /ISAPI/System/GPSVerification/channels/<ID>/display?format=json

Method	GET
Description	Get the parameters of displaying GPS calibration verification result.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_VerificationDisplay</u> Failed: <u>JSONResponseStatus</u>

Table C-266 PUT /ISAPI/System/GPSVerification/channels/<ID>/display?format=json

Method	PUT
Description	Set the parameters of displaying GPS calibration verification result.
Query	format: determine the format of request or response message.
Request	<u>JSON_VerificationDisplay</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the URI refers to channel No.

E.183 /ISAPI/System/GPSVerification/channels/<ID>/points/<ID>?format=json

Get, set, or delete a specified verification point.

Request URI Definition**Table C-267 GET /ISAPI/System/GPSVerification/channels/<ID>/points/<ID>?format=json**

Method	GET
Description	Get the parameters of a specified verification point.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_VerificationPoint</u> Failed: <u>JSON_ResponseStatus</u>

Table C-268 PUT /ISAPI/System/GPSVerification/channels/<ID>/points/<ID>?format=json

Method	PUT
Description	Set the parameters of a specified verification point.
Query	format: determine the format of request or response message.
Request	<u>JSON_VerificationPoint</u>
Response	<u>JSON_ResponseStatus</u>

Table C-269 DELETE /ISAPI/System/GPSVerification/channels/<ID>/points/<ID>?format=json

Method	DELETE
Description	Delete a specified verification point.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSONResponseStatus</u>

Remarks

The first <ID> in the URI refers to channel No., and second <ID> refers to verification point No.

E.184 /ISAPI/System/GPSVerification/channels/<ID>/points/capabilities?format=json

Get the capability of verifying GPS calibration.

Request URI Definition**Table C-270 GET /ISAPI/System/GPSVerification/channels/<ID>/points/capabilities?format=json**

Method	GET
Description	Get the capability of verifying GPS calibration.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_VerificationPointList</u> Failed: <u>JSONResponseStatus</u>

Remarks

The <ID> in the URI refers to channel No.

E.185 /ISAPI/System/GPSVerification/channels/<ID>/points/resultInfo?format=json

Get the verification results of GPS calibration in a batch according to channel No.

Request URI Definition

Table C-271 GET /ISAPI/System/GPSVerification/channels/<ID>/points/resultInfo?format=json

Method	GET
Description	Get the verification results of GPS calibration in a batch according to channel No.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_GPSVerification_ResultInfoList</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

The <ID> in the URI refers to channel No.

E.186 /ISAPI/System/GPSVerification/channels/<ID>/points? format=json

Get, set, or delete multiple verification points in a batch, or add a verification point.

Request URI Definition

Table C-272 GET /ISAPI/System/GPSVerification/channels/<ID>/points?format=json

Method	GET
Description	Get the parameters of verification points in a batch according to channel No.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_VerificationPointList</u> Failed: <u>JSON_ResponseStatus</u>

Table C-273 PUT /ISAPI/System/GPSVerification/channels/<ID>/points?format=json

Method	PUT
Description	Set the parameters of verification points in a batch according to channel No.

Query	format: determine the format of request or response message.
Request	<i>JSON_VerificationPointList</i>
Response	<i>JSON_ResponseStatus</i>

Table C-274 DELETE /ISAPI/System/GPSVerification/channels/<ID>/points?format=json

Method	DELETE
Description	Delete all verification points in a batch according to channel No.
Query	format: determine the format of request or response message.
Request	None.
Response	<i>JSON_ResponseStatus</i>

Table C-275 POST /ISAPI/System/GPSVerification/channels/<ID>/points?format=json

Method	POST
Description	Add a verification point according to channel No.
Query	format: determine the format of request or response message.
Request	<i>JSON_VerificationPoint</i>
Response	Succeeded: <i>JSON_GPSVerification_ResultInfo</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URI refers to channel No.

E.187 /ISAPI/System/guideConfig/<guideEvent>/capabilities?format=json

Get the capability of quick setup instruction for specified event.

Request URI Definition

Table C-276 GET /ISAPI/System/guideConfig/<guideEvent>/capabilities?format=json

Method	GET
Description	Get the capability of quick setup instruction for specified event.
Query	format: determine the format of request or response message.

Request	None.
Response	Succeeded: <u>JSON_GuideConfigCap</u> Failed: <u>JSONResponseStatus</u>

Remarks

The <guideEvent> in the URI refers to the event, which supports quick setup, and now only "linkageCapture" (capture by linkage) is supported, i.e., the URI is: /ISAPI/System/guideConfig/linkageCapture/capabilities?format=json

E.188 /ISAPI/System/guideConfig/<guideEvent>?format=json

Get the parameters of quick setup instruction for specified event.

Request URI Definition

Table C-277 GET /ISAPI/System/guideConfig/<guideEvent>?format=json

Method	GET
Description	Get the parameters of quick setup instruction for specified event.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_GuideConfig</u> Failed: <u>JSONResponseStatus</u>

Remarks

The <guideEvent> in the URI refers to the event, which supports quick setup, and now only "linkageCapture" (capture by linkage) is supported, i.e., the URI is: /ISAPI/System/guideConfig/linkageCapture?format=json

E.189 /ISAPI/System/installationAngleCalibration/channels/<ID>/capabilities?format=json

Get the capability of installation angle calibration.

Request URI Definition

Table C-278 GET /ISAPI/System/installationAngleCalibration/channels/<ID>/capabilities?format=json

Method	GET
Description	Get the capability of installation angle calibration.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_Cap_CalibrationStatus</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.190 /ISAPI/System/installationAngleCalibration/channels/<ID>?format=json

Get the installation angle calibration status or calibrate the installation angle.

Request URI Definition

Table C-279 GET /ISAPI/System/installationAngleCalibration/channels/<ID>?format=json

Method	GET
Description	Get the installation angle calibration status.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_CalibrationStatus</u> Failed: <u>JSON_ResponseStatus</u>

Table C-280 PUT /ISAPI/System/installationAngleCalibration/channels/<ID>?format=json

Method	PUT
Description	Calibrate the installation angle.
Query	format: determine the format of request or response message.

Request	None.
Response	<u>JSONResponseStatus</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.191 /ISAPI/System/IOT/channelConfig?format=json

Import or export the list of added IoT devices.

Request URI Definition**Table C-281 GET /ISAPI/System/IOT/channelConfig?format=json**

Method	GET
Description	Export the list of added IoT devices.
Query	<p>format: determine the format of request or response message.</p> <p>security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode.</p> <p>iv: the initialization vector, and it is required when security is 1 or 2.</p>
Request	None.
Response	Succeeded: Opaque Data Failed: <u>JSONResponseStatus</u>

Table C-282 PUT /ISAPI/System/IOT/channelConfig?format=json

Method	PUT
Description	Import the list of added IoT devices.
Query	<p>format: determine the format of request or response message.</p> <p>security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates</p>

	that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	Opaque Data
Response	<u>JSON_ResponseStatus</u> <u>JSON_IOT_ErrorList</u>

Remarks

The Opaque Data should be encrypted.

E.192 /ISAPI/System/IOT/channels/<ID>/<EventType>/capabilities?format=json

Get the event configuration capability of IoT device.

Request URI Definition

Table C-283 GET /ISAPI/System/IOT/channels/<ID>/<EventType>/capabilities?format=json

Method	GET
Description	Get the event configuration capability of IoT device.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_XXCap</u> Failed: <u>JSON_ResponseStatus</u>

E.193 /ISAPI/System/IOT/channels/<ID>/<EventType>?format=json

Get or set the event/alarm parameters.

Request URI Definition

Table C-284 GET /ISAPI/System/IOT/channels/<ID>/<EventType>?format=json

Method	GET
Description	Get the event/alarm parameters.
Query	format : determine the format of request or response message.

Request	None.
Response	Succeeded: <u>JSON_XX</u> Failed: <u>JSON_ResponseStatus</u>

Table C-285 PUT /ISAPI/System/IOT/channels/<ID>/<EventType>?format=json

Method	PUT
Description	Set the event/alarm parameters.
Query	format: determine the format of request or response message.
Request	<u>JSON_XX</u>
Response	<u>JSON_ResponseStatus</u>

E.194 /ISAPI/System/IOT/channels/<ID>/alarmStatistics?format=json

Get the alarm statistics of added IoT device according to channel ID.

Request URI Definition

Table C-286 GET /ISAPI/System/IOT/channels/<ID>/alarmStatistics?format=json

Method	GET
Description	Get the alarm statistics of added IoT device according to channel ID.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_AlarmStatistics</u> Failed: <u>JSON_ResponseStatus</u>

E.195 /ISAPI/System/IOT/channels/<ID>/all?format=json

Delete all channels (video channel and IoT channel) of added IoT device

Request URI Definition**Table C-287 DELETE /ISAPI/System/IOT/channels/<ID>/all?format=json**

Method	DELETE
Description	Delete all channels (video channel and IoT channel) of added IoT device
Query	format: determine the format of request or response message.
Request	None.
Response	<i>JSONResponseStatus</i>

E.196 /ISAPI/System/IOT/channels/<ID>/basicParam?format=json

Set the basic parameters of added IoT device according to channel ID.

Request URI Definition**Table C-288 PUT /ISAPI/System/IOT/channels/<ID>/basicParam?format=json**

Method	PUT
Description	Set the basic parameters of added IoT device according to channel ID.
Query	format: determine the format of request or response message.
Request	<i>JSON_BasicParam</i>
Response	<i>JSONResponseStatus</i>

E.197 /ISAPI/System/IOT/channels/<ID>/OSD/capabilities?format=json

Get the OSD configuration capability of specified IoT device channel.

Request URI Definition**Table C-289 GET /ISAPI/System/IOT/channels/<ID>/OSD/capabilities?format=json**

Method	GET
Description	Get the OSD configuration capability of specified IoT device channel.
Query	format: determine the format of request or response message.

Request	None.
Response	Succeeded: <u>JSON OSDCap</u> Failed: <u>JSON ResponseStatus</u>

E.198 /ISAPI/System/IOT/channels/<ID>/OSD?format=json

Get or set the OSD parameters.

Request URI Definition

Table C-290 GET /ISAPI/System/IOT/channels/<ID>/OSD?format=json

Method	GET
Description	Get the OSD parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON OSD</u> Failed: <u>JSON ResponseStatus</u>

Table C-291 PUT /ISAPI/System/IOT/channels/<ID>/OSD?format=json

Method	PUT
Description	Set the OSD parameters.
Query	format: determine the format of request or response message.
Request	<u>JSON OSD</u>
Response	<u>JSON ResponseStatus</u>

Remarks

The <ID> in the URI is defied as the channel ID.

E.199 /ISAPI/System/IOT/channels/<ID>/status?format=json

Get the status of specified channel of added IoT device.

Request URI Definition

Table C-292 GET /ISAPI/System/IOT/channels/<ID>/status?format=json

Method	GET
Description	Get the status of specified channel of added IoT device.
Query	<p>format: determine the format of request or response message.</p> <p>security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode.</p> <p>iv: the initialization vector, and it is required when security is 1 or 2.</p>
Request	None.
Response	<u>JSON_IOTChannelStatus</u>

Remarks

The <ID> in the URI is defined as channel ID.

E.200 /ISAPI/System/IOT/channels/<ID>?format=json

Operations about specified added IoT device.

Request URI Definition

Table C-293 GET /ISAPI/System/IOT/channels/<ID>?format=json

Method	GET
Description	Get the information of added IoT device according to channel ID.
Query	<p>format: determine the format of request or response message.</p> <p>security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode.</p> <p>iv: the initialization vector, and it is required when security is 1 or 2.</p>

Request	None.
Response	Succeeded: <u>JSON_IOTChannel</u> Failed: <u>JSONResponseStatus</u>

Table C-294 PUT /ISAPI/System/IOT/channels/<ID>?format=json

Method	PUT
Description	Set the added IoT device according to channel ID.
Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	<u>JSON_IOTChannel</u>
Response	<u>JSONResponseStatus</u>

Table C-295 DELETE /ISAPI/System/IOT/channels/<ID>?format=json

Method	DELETE
Description	Delete the specified IoT device according to channel ID.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSONResponseStatus</u>

Remarks

The <ID> in the URI is defined as channel ID.

E.201 /ISAPI/System/IOT/channels/status?format=json

Get the status of added IoT device.

Request URI Definition

Table C-296 GET /ISAPI/System/IOT/channels/status?format=json

Method	GET
Description	Get the status of added IoT device.
Query	<p>format: determine the format of request or response message.</p> <p>security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode.</p> <p>iv: the initialization vector, and it is required when security is 1 or 2.</p>
Request	None.
Response	<p>Succeeded: <u>JSON_IOTChannelStatusList</u></p> <p>Failed: <u>JSON_ResponseStatus</u></p>

E.202 /ISAPI/System/IOT/channels/status? format=json&deviceInductiveType=

Get the status of added IoT device according to inductive type.

Request URI Definition

Table C-297 GET /ISAPI/System/IOT/channels/status?format=json&deviceInductiveType=

Method	GET
Description	Get the status of added IoT device according to inductive type.
Query	<p>format: determine the format of request or response message.</p> <p>security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode.</p> <p>iv: the initialization vector, and it is required when security is 1 or 2.</p>

	deviceInductiveType : device inductive type, its value can be obtained by URI: GET <u>/ISAPI/System/IOT/sourceSupport/capabilities?format=json</u> , now there are two available values: "inductiveType1" and "inductiveType2"
Request	None.
Response	<u>JSON_IOTChannelStatusList</u>

E.203 /ISAPI/System/IOT/channels?format=json

Operations about IoT devices.

Request URI Definition

Table C-298 GET /ISAPI/System/IOT/channels?format=json

Method	GET
Description	Get the information of added IoT devices.
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	<u>JSON_IOTChannelList</u>

Table C-299 POST /ISAPI/System/IOT/channels?format=json

Method	POST
Description	Add the IoT devices to NVR/DVR.
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates

	that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	<u>JSON_IOTChannel</u>
Response	Succeeded: <u>JSON_IOT_ChannelInfo</u> Failed: <u>JSON_ResponseStatus</u>

E.204 /ISAPI/System/IOT/channels?format=json&deviceInductiveType=

Get the information of added IoT devices according to device inductive type.

Request URI Definition

Table C-300 GET /ISAPI/System/IOT/channels?format=json&deviceInductiveType=

Method	GET
Description	Get the added IoT devices according to device inductive type.
Query	format: determine the format of request or response message. deviceInductiveType: device inductive type, its value can be obtained by URI: GET <u>/ISAPI/System/IOT/sourceSupport/capabilities?format=json</u> , now there are two available values: "inductiveType1" and "inductiveType2" security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	<u>JSON_IOTChannelList</u>

E.205 /ISAPI/System/IOT/linkageChannels?format=json

Get the linked channel of searched event.

Request URI Definition

Table C-301 POST /ISAPI/System/IOT/linkageChannels?format=json

Method	POST
Description	Get the linked channel of searched event.
Query	format : determine the format of request or response message.
Request	<u>JSON_LinkageChansCond</u>
Response	Succeeded: <u>JSON_Result</u> Failed: <u>JSON_ResponseStatus</u>

E.206 /ISAPI/System/IOT/search?format=json

Get the IoT devices which can be added.

Request URI Definition

Table C-302 GET /ISAPI/System/IOT/search?format=json

Method	GET
Description	Get the IoT devices which can be added.
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	<u>JSON_IOTSourceList</u>

E.207 /ISAPI/System/IOT/sourceCapabilities?format=json

Get the number of IoT device channels.

Request URI Definition

Table C-303 GET /ISAPI/System/IOT/sourceCapabilities?format=json

Method	GET
Description	Get the number of IoT device channels.
Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	<u>JSON_IOTSourceDescription</u>
Response	<u>JSON_SourceCapabilities</u>

E.208 /ISAPI/System/IOT/sourceSupport/capabilities?format=json

Get supported protocol type of IoT devices.

Request URI Definition

Table C-304 GET /ISAPI/System/IOT/sourceSupport/capabilities?format=json

Method	GET
Description	Get supported protocol type of IoT devices.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_IOTSourceSupport</u>

E.209 /ISAPI/System/laserOpticalAxis/capabilities?format=json

Get the configuration capability of laser optical axis.

Request URI Definition

Table C-305 GET /ISAPI/System/laserOpticalAxis/capabilities?format=json

Method	GET
Description	Get the configuration capability of laser optical axis.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_LaserOpticalAxisCap</i> Failed: <i>JSON_ResponseStatus</i>

E.210 /ISAPI/System/laserOpticalAxis/goto?format=json

Adjust the laser optical axis.

Request URI Definition

Table C-306 PUT /ISAPI/System/laserOpticalAxis/goto?format=json

Method	PUT
Description	Adjust the laser optical axis.
Query	format: determine the format of request or response message.
Request	<i>JSON_LaserOpticalAxis_direction</i>
Response	<i>JSON_ResponseStatus</i>

E.211 /ISAPI/System/laserOpticalAxis?format=json

Get or set laser optical axis parameters.

Request URI Definition

Table C-307 GET /ISAPI/System/laserOpticalAxis?format=json

Method	GET
Description	Get the laser optical axis parameters.
Query	format: determine the format of request or response message.

Request	None.
Response	Succeeded: <u>JSON_LaserOpticalAxis</u> Failed: <u>JSON_ResponseStatus</u>

Table C-308 PUT /ISAPI/System/laserOpticalAxis?format=json

Method	PUT
Description	Configure the laser optical axis.
Query	format: determine the format of request or response message.
Request	<u>JSON_LaserOpticalAxis</u>
Response	<u>JSON_ResponseStatus</u>

E.212 /ISAPI/System/logServer

Get or set log server parameters.

Request URI Definition

Table C-309 GET /ISAPI/System/logServer

Method	GET
Description	Get log server parameters.
Query	None.
Request	None.
Response	Succeeded: <u>XML_LogServer</u> Failed: <u>XML_ResponseStatus</u>

Table C-310 PUT /ISAPI/System/logServer

Method	PUT
Description	Set log server parameters.
Query	None.
Request	<u>XML_LogServer</u>
Response	<u>XML_ResponseStatus</u>

E.213 /ISAPI/System/logServer/capabilities

Get log server configuration capability.

Request URI Definition

Table C-311 GET /ISAPI/System/logServer/capabilities

Method	GET
Description	Get log server configuration capability.
Query	None.
Request	None.
Response	Succeeded: <i>XML_LogServerCap</i> Failed: <i>XML_ResponseStatus</i>

E.214 /ISAPI/System/Network/capabilities

Get the network capability.

Request URI Definition

Table C-312 GET /ISAPI/System/Network/capabilities

Method	GET
Description	Get the network capability.
Query	None.
Request	None.
Response	Succeeded: <i>XML_NetworkCap</i> Failed: <i>XML_ResponseStatus</i>

E.215 /ISAPI/System/Network/Ehome/<ISUPID>

Get or set the parameters of accessing a server via ISUP.

Request URI Definition

Table C-313 GET /ISAPI/System/Network/Ehome/<ISUPID>

Method	GET
Description	Get parameters of accessing a server via ISUP.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2. centerID : optional, string, center ID (for access control devices only).
Request	None.
Response	Succeeded: <u>XML_Ehome</u> Failed: <u>XMLResponseStatus</u>

Table C-314 PUT /ISAPI/System/Network/Ehome/<ISUPID>

Method	PUT
Description	Set parameters of accessing a server via ISUP.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2. centerID : optional, string, center ID (for access control devices only).
Request	<u>XML_Ehome</u>
Response	<u>XMLResponseStatus</u>

E.216 /ISAPI/System/Network/Ehome/capabilities

Get the configuration capability of accessing servers via ISUP.

Request URI Definition

Table C-315 GET /ISAPI/System/Network/Ehome/capabilities

Method	GET
Description	Get the configuration capability of accessing servers via ISUP.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_EHome</i> Failed: <i>XML_ResponseStatus</i>

E.217 /ISAPI/System/Network/EhomeList?centerID=

Get the parameters of accessing all servers via ISUP.

Request URI Definition

Table C-316 GET /ISAPI/System/Network/EhomeList?centerID=

Method	GET
Description	Get the parameters of accessing all servers via ISUP.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2. centerID : optional, string, center ID (for access control devices only).
Request	None
Response	Succeeded: <i>XML_EhomeList</i> Failed: <i>XML_ResponseStatus</i>

E.218 /ISAPI/System/Network/extension

Get and set multicast parameters.

Request URL Definition

Table C-317 GET /ISAPI/System/Network/extension

Method	GET
Description	Get multicast parameters.
Query	None.
Request	None.
Response	Succeeded: <i>XML_networkExtension</i> Failed: <i>XML_ResponseStatus</i>

Table C-318 PUT /ISAPI/System/Network/extension

Method	PUT
Description	Set multicast parameters.
Query	None.
Request	<i>XML_networkExtension</i>
Response	<i>XML_ResponseStatus</i>

E.219 /ISAPI/System/Network/EZVIZ

Operations about Hik-Connect access configurations.

Request URI Definition

Table C-319 GET /ISAPI/System/Network/EZVIZ

Method	GET
Description	Get Hik-Connect parameters.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.

Request	None.
Response	<u>XML_EZVIZ</u>

Table C-320 PUT /ISAPI/System/Network/EZVIZ

Method	PUT
Description	Set Hik-Connect parameters.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	<u>XML_EZVIZ</u>
Response	<u>XML_ResponseStatus</u>

E.220 /ISAPI/System/Network/EZVIZ/capabilities

Get the Guarding Vision access configuration capability.

Request URI Definition

Table C-321 GET /ISAPI/System/Network/EZVIZ/capabilities

Method	GET
Description	Get the Guarding Vision access configuration capability.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Cap_EZVIZ</u> Failed: <u>XML_ResponseStatus</u>

E.221 /ISAPI/System/Network/EZVIZ/QRCode

Get the Guarding Vision QR code of the device.

Request URI Definition

Table C-322 GET /ISAPI/System/Network/EZVIZ/QRCode

Method	GET
Description	Get the Guarding Vision QR code of the device.
Query	None.
Request	None.
Response	Succeeded: Binary data (QR code in binary format) Failed: <u>XMLResponseStatus</u>

E.222 /ISAPI/System/Network/EZVIZ/QRCode?format=json

Get the Guarding Vision QR code.

Request URI Definition

Table C-323 GET /ISAPI/System/Network/EZVIZ/QRCode?format=json

Method	GET
Description	Get the Guarding Vision QR code.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2. format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_EZVIZQRCode</u> Failed: <u>JSONResponseStatus</u>

Remarks

- You can use smart phone to scan the QR code to download Guarding Vision app or use Guarding Vision to scan the QR code to bind the device.
- If the node **isSupportEZVIZQRCode** is returned in the message [XML Cap_EZVIZ](#) (related URI: [/ISAPI/System/Network/EZVIZ/capabilities](#)) and the value is true, it indicates that the device supports this function.

E.223 /ISAPI/System/Network/EZVIZ/secretKey?format=json

Edit the verification code for Guarding Vision.

Request URI Definition

Table C-324 PUT /ISAPI/System/Network/EZVIZ/secretKey?format=json

Method	PUT
Description	Edit the verification code for Guarding Vision.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2. format : determine the format of request or response message.
Request	<u>JSON_EZVIZSecretKey</u>
Response	<u>JSON_ResponseStatus</u>

E.224 /ISAPI/System/Network/ftp/<ID>

Operations about the configuration of a specific FTP.

Request URI Definition

Table C-325 GET /ISAPI/System/Network/ftp/<ID>

Method	GET
Description	Get the parameters of a specific FTP.

Query	None.
Request	None.
Response	Succeeded: <u>XML_FTPNotification</u> Failed: <u>XML_ResponseStatus</u>

Table C-326 PUT /ISAPI/System/Network/ftp/<ID>

Method	PUT
Description	Set the parameters of a specific FTP.
Query	None.
Request	<u>XML_FTPNotification</u>
Response	<u>XML_ResponseStatus</u>

E.225 /ISAPI/System/Network/ftp/capabilities

Get the FTP capability.

Request URI Definition

Table C-327 GET /ISAPI/System/Network/ftp/capabilities

Method	GET
Description	Get the FTP capability.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Cap_FTPNotificationList</u> Failed: <u>XML_ResponseStatus</u>

E.226 /ISAPI/System/Network/Integrate

Get or set the access protocol parameters.

Request URI Definition

Table C-328 GET /ISAPI/System/Network/Integrate

Method	GET
Description	Get the access protocol parameters.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Integrate</i> Failed: <i>XMLResponseStatus</i>

Table C-329 PUT /ISAPI/System/Network/Integrate

Method	PUT
Description	Set the access protocol parameters.
Query	None.
Request	<i>XML_Integrate</i>
Response	<i>XMLResponseStatus</i>

E.227 /ISAPI/System/Network/Integrate/capabilities

Get the configuration capability of the access protocol.

Request URI Definition

Table C-330 GET /ISAPI/System/Network/Integrate/capabilities

Method	GET
Description	Get the configuration capability of the access protocol.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_Integrate</i> Failed: <i>XMLResponseStatus</i>

E.228 /ISAPI/System/Network/ipFilter

Operations about IP address filter configuration.

Request URI Definition

Table C-331 GET /ISAPI/System/Network/ipFilter

Method	GET
Description	Get the IP filter configuration parameters.
Query	None.
Request	None.
Response	Succeeded: <u>XML_IPFilter</u> Failed: <u>XML_ResponseStatus</u>

Table C-332 PUT /ISAPI/System/Network/ipFilter

Method	PUT
Description	Set the IP address filter.
Query	None.
Request	<u>XML_IPFilter</u>
Response	<u>XML_ResponseStatus</u>

E.229 /ISAPI/System/Network/ipFilter/capabilities

Get IP address filter configuration capability.

Request URI Definition

Table C-333 GET /ISAPI/System/Network/ipFilter/capabilities

Method	GET
Description	Get the IP address configuration capability.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Cap_IPFilter</u>

	Failed: <u>XML_ResponseStatus</u>
--	---

E.230 /ISAPI/System/Network/resourceStatistics?format=json

Get the information of network resource statistics.

Request URI Definition

Table C-334 GET /ISAPI/System/Network/resourceStatistics?format=json

Method	GET
Description	Get the information of network resource statistics.
Query	None.
Request	None.
Response	Succeeded: <u>JSON_resourceStatistics</u> Failed: <u>JSON_ResponseStatus</u>

E.231 /ISAPI/System/Network/SNMP

Get or set batch configuration parameters of SNMP (Simple Network Management Protocol).

Request URI Definition

Table C-335 GET /ISAPI/System/Network/SNMP

Method	GET
Description	Get batch configuration parameters of SNMP.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	Succeeded: <u>XML_SNMP</u>

	Failed: <i>XMLResponseStatus</i>
--	----------------------------------

Table C-336 PUT /ISAPI/System/Network/SNMP

Method	PUT
Description	Set batch configuration parameters of SNMP.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	<i>XML_SNPMP</i>
Response	<i>XML_ResponseStatus</i>

E.232 /ISAPI/System/Network/SNMP/advanced

Get or set advanced parameters of SNMP (Simple Network Management Protocol).

Request URI Definition

Table C-337 GET /ISAPI/System/Network/SNMP/advanced

Method	GET
Description	Get advanced parameters of SNMP.
Query	None.
Request	None.
Response	Succeeded: <i>XML_SNMPAdvanced</i> Failed: <i>XML_ResponseStatus</i>

Table C-338 PUT /ISAPI/System/Network/SNMP/advanced

Method	PUT
Description	Set advanced parameters of SNMP.
Query	None.

Request	<i><u>XML_SNMPAdvanced</u></i>
Response	<i><u>XML_ResponseStatus</u></i>

E.233 /ISAPI/System/Network/SNMP/advanced/trapReceivers

Get, set, delete parameters of all trap address, or add a new trap address of SNMP in version 3.

Request URI Definition

Table C-339 GET /ISAPI/System/Network/SNMP/advanced/trapReceivers

Method	GET
Description	Get all trap address parameters of SNMP in version 3.
Query	None.
Request	None.
Response	Succeeded: <i><u>XML_SNMPTrapReceiverList</u></i> Failed: <i><u>XML_ResponseStatus</u></i>

Table C-340 PUT /ISAPI/System/Network/SNMP/advanced/trapReceivers

Method	PUT
Description	Set parameters for all trap addresses of SNMP in version 3.
Query	None.
Request	<i><u>XML_SNMPTrapReceiverList</u></i>
Response	<i><u>XML_ResponseStatus</u></i>

Table C-341 DELETE /ISAPI/System/Network/SNMP/advanced/trapReceivers

Method	DELETE
Description	Delete all trap address parameters of SNMP in version 3.
Query	None.
Request	None.
Response	<i><u>XML_ResponseStatus</u></i>

Table C-342 POST /ISAPI/System/Network/SNMP/advanced/trapReceivers

Method	POST
Description	Add a new trap address of SNMP.
Query	None.
Request	<i>XML_SNMPTrapReceiver</i>
Response	<i>XMLResponseStatus</i>

E.234 /ISAPI/System/Network/SNMP/advanced/trapReceivers/<ID>

Get, set or delete parameters for a single trap address.

Request URI Definition**Table C-343 GET /ISAPI/System/Network/SNMP/advanced/trapReceivers/<ID>**

Method	GET
Description	Get trap address parameters of SNMP in version 3.
Query	None.
Request	None.
Response	Succeeded: <i>XML_SNMPTrapReceiver</i> Failed: <i>XMLResponseStatus</i>

Table C-344 PUT /ISAPI/System/Network/SNMP/advanced/trapReceivers/<ID>

Method	PUT
Description	Set parameters for all trap addresses of SNMP in version 3.
Query	None.
Request	<i>XML_SNMPTrapReceiver</i>
Response	<i>XMLResponseStatus</i>

Table C-345 DELETE /ISAPI/System/Network/SNMP/advanced/trapReceivers/<ID>

Method	DELETE
Description	Delete a trap address of SNMP in version 3.
Query	None.

Request	None.
Response	<u>XML_ResponseStatus</u>

E.235 /ISAPI/System/Network/socketIP

Get the socket IP of current connection.

Request URI Definition

Table C-346 GET /ISAPI/System/Network/socketIP

Method	GET
Description	Get the socket IP of current connection.
Query	None.
Request	None.
Response	<u>XML_SocketIP</u>

E.236 /ISAPI/System/Network/verificationCodeCheck

Verify the encryption key or verification code.

Request URI Definition

Table C-347 POST /ISAPI/System/Network/verificationCodeCheck

Method	POST
Description	Verify the encryption key or verification code.
Query	None.
Request	<u>XML_CheckInfo</u>
Response	<u>XML_ResponseStatus</u>

E.237 /ISAPI/System/Network/WirelessDial/Interfaces

Get or set parameters for all dial network cards.

Request URI Definition

Table C-348 GET /ISAPI/System/Network/WirelessDial/Interfaces

Method	GET
Description	Get parameters for all dial network cards.
Query	None.
Request	None.
Response	Succeeded: <i>XML_WirelessDialInterface</i> Failed: <i>XMLResponseStatus</i>

Table C-349 PUT /ISAPI/System/Network/WirelessDial/Interfaces

Method	PUT
Description	Set parameters for all dial network cards.
Query	None.
Request	<i>XML_WirelessDialInterface</i>
Response	<i>XMLResponseStatus</i>

E.238 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>

Get or set parameters for a single dial network card.

Request URI Definition

Table C-350 GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>

Method	GET
Description	Get parameters for a single dial network card.
Query	None.
Request	None.
Response	Succeeded: <i>XML_WirelessDialInterfaceList</i> Failed: <i>XMLResponseStatus</i>

Table C-351 PUT /ISAPI/System/Network/WirelessDial/Interfaces/<ID>

Method	PUT
Description	Set the wireless dial-up parameters.
Query	None.
Request	<u>XML_WirelessDialInterfaceList</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the network interface ID.

E.239 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial

Operations about the wireless dial-up configuration.

Request URI Definition**Table C-352 GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial**

Method	GET
Description	Get the configuration parameters of wireless dial-up.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	Succeeded: <u>XML_Dial</u> Failed: <u>XML_ResponseStatus</u>

Table C-353 PUT /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial

Method	PUT
Description	Set the wireless dial-up parameters.

Query	security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	<u>XML_Dial</u>
Response	<u>XML_ResponseStatus</u>

E.240 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial/capabilities

Get the wireless dial-up capability.

Request URI Definition

Table C-354 GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial/capabilities

Method	GET
Description	Get the wireless dial-up capability.
Query	security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	None.
Response	Succeeded: <u>XML_Cap_Dial</u> Failed: <u>XML_ResponseStatus</u>

E.241 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive/capabilities?format=json

Get the configuration capability of network keepalive.

Request URI Definition

Table C-355 GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive/capabilities?format=json

Method	GET
Description	Get the configuration capability of network keepalive.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_KeepAliveCap</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the network interface ID.

E.242 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive/test?format=json

Test network keepalive configuration.

Request URI Definition

Table C-356 POST /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive/test?format=json

Method	POST
Description	Test network keepalive configuration.
Query	format: determine the format of request or response message.
Request	<u>JSON_KeepAlive</u>
Response	<u>JSON_TestResult</u>

Remarks

The <ID> in the request URI refers to the network interface ID.

E.243 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive?format=json

Set network keepalive parameters.

Request URI Definition

Table C-357 PUT /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive?format=json

Method	PUT
Description	Set network keepalive parameters.
Query	format: determine the format of request or response message.
Request	<u>JSON_KeepAlive</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the network interface ID.

E.244 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/schedule

Get or set dial-up schedule parameters.

Request URI Definition

Table C-358 GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/schedule

Method	GET
Description	Get dial-up schedule parameters.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Schedule</u> Failed: <u>XML_ResponseStatus</u>

Table C-359 PUT /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/schedule

Method	PUT
Description	Set dial-up schedule parameters.

Query	None.
Request	<u>XML_Schedule</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the network interface ID.

E.245 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor/capabilities?format=json

Get traffic monitoring configuration capability.

Request URI Definition

Table C-360 GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor/capabilities?format=json

Method	GET
Description	Get traffic monitoring configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_TrafficMonitorCap</u> Failed: <u>JSON_ResponseStatus</u>

E.246 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor/status?format=json

Get traffic monitoring status.

Request URI Definition

Table C-361 GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor/status?format=json

Method	GET
Description	Get traffic monitoring status.

Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_TrafficMonitorStatus</u> Failed: <u>JSON_ResponseStatus</u>

E.247 /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor?format=json

Get traffic monitoring parameters.

Request URI Definition

Table C-362 GET /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor?format=json

Method	GET
Description	Get traffic monitoring parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_TrafficMonitor</u> Failed: <u>JSON_ResponseStatus</u>

Table C-363 PUT /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/trafficMonitor?format=json

Method	PUT
Description	Set traffic monitoring parameters.
Query	None.
Request	<u>JSON_TrafficMonitor</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the network interface ID.

E.248 /ISAPI/System/onlineUpgrade/deviceParameter?format=json

Get and set the device online upgrade parameters.

Request URI Definition

Table C-364 GET /ISAPI/System/onlineUpgrade/deviceParameter?format=json

Method	GET
Description	Get the device online upgrade parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_OnlineUpgradeParameter</u> Failed: <u>JSON_ResponseStatus</u>

Table C-365 PUT /ISAPI/System/onlineUpgrade/deviceParameter?format=json

Method	PUT
Description	Set the device online upgrade parameters.
Query	format: determine the format of request or response message.
Request	<u>JSON_OnlineUpgradeParameter</u>
Response	<u>JSON_ResponseStatus</u>

E.249 /ISAPI/System/onlineUpgrade/downloadPackage/pause? format=json

Pause the upgrade package download.

Request URI Definition

Table C-366 PUT /ISAPI/System/onlineUpgrade/downloadPackage/pause?format=json

Method	PUT
Description	Pause upgrade package download.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_ResponseStatus</u>

E.250 /ISAPI/System/onlineUpgrade/downloadPackage/resume?format=json

Resume the upgrade package download.

Request URI Definition

Table C-367 PUT /ISAPI/System/onlineUpgrade/downloadPackage/resume?format=json

Method	PUT
Description	Resume the upgrade package download.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_ResponseStatus</u>

E.251 /ISAPI/System/onlineUpgrade/downloadPackage/status?format=json

Get the upgrade package download progress.

Request URI Definition

Table C-368 GET /ISAPI/System/onlineUpgrade/downloadPackage/status?format=json

Method	GET
Description	Get the upgrade package download progress.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_DownloadPackageStatus</u> Failed: <u>JSON_ResponseStatus</u>

E.252 /ISAPI/System/onlineUpgrade/downloadPackage?format=json

Start or cancel downloading upgrade package to device.

Request URI Definition

Table C-369 PUT /ISAPI/System/onlineUpgrade/downloadPackage?format=json

Method	PUT
Description	Start to download upgrade package to device.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSONResponseStatus</u>

Table C-370 DELETE /ISAPI/System/onlineUpgrade/downloadPackage?format=json

Method	DELETE
Description	Cancel the upgrade package download.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSONResponseStatus</u>

E.253 /ISAPI/System/onlineUpgrade/status

Get the online upgrade progress of device.

Request URI Definition

Table C-371 GET /ISAPI/System/onlineUpgrade/status

Method	GET
Description	Get the online upgrade progress of device.
Query	None.
Request	None.
Response	<u>XML_OnlineUpgradeStatus</u>

E.254 /ISAPI/System/onlineUpgrade/upgradeWithoutDownload? format=json

Start device upgrade with existed upgrade package.

Request URI Definition

Table C-372 PUT /ISAPI/System/onlineUpgrade/upgradeWithoutDownload?format=json

Method	PUT
Description	Start upgrade with existed upgrade package.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSONResponseStatus</u>

E.255 /ISAPI/System/Serial/authentication/capabilities?format=json

Get capability of serial authentication service.

Request URI Definition

Table C-373 GET /ISAPI/System/Serial/authentication/capabilities?format=json

Method	GET
Description	Get capability of serial authentication service.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_AuthenticationCap</u> Failed: <u>JSONResponseStatus</u>

E.256 /ISAPI/System/Serial/authentication?format=json

Get or set parameters of serial authentication service.

Request URI Definition

Table C-374 GET /ISAPI/System/Serial/authentication?format=json

Method	GET
Description	Get parameters of serial authentication service.
Query	format: determine the format of request or response message.

Request	None.
Response	Succeeded: <u>JSON_Authentication</u> Failed: <u>JSON_ResponseStatus</u>

Table C-375 PUT /ISAPI/System/Serial/authentication?format=json

Method	PUT
Description	Set parameters of serial authentication service.
Query	format: determine the format of request or response message.
Request	<u>JSON_Authentication</u>
Response	<u>JSON_ResponseStatus</u>

E.257 /ISAPI/System/Serial/capabilities

Get the serial port capability of the device.

Request URI Definition

Table C-376 GET /ISAPI/System/Serial/capabilities

Method	GET
Description	Get the serial port capability of the device.
Query	None.
Request	None.
Response	Succeeded: <u>XML_SerialCap</u> Failed: <u>XML_ResponseStatus</u>

E.258 /ISAPI/System/Software/channels/<ID>

Operation about software service configuration.

Request URI Definition

Table C-377 GET /ISAPI/System/Software/channels/<ID>

Method	GET
Description	Get software service parameters.
Query	None.
Request	None.
Response	<u>XML_SoftwareService</u>

Table C-378 PUT /ISAPI/System/Software/channels/<ID>

Method	PUT
Description	Set software service parameters.
Query	None.
Request	<u>XML_SoftwareService</u>
Response	<u>XML_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.259 /ISAPI/System/Software/channels/<ID>/capabilities

Get the configuration capability of software service.

Request URI Definition

Table C-379 GET /ISAPI/System/Software/channels/<ID>/capabilities

Method	GET
Description	Get the configuration capability of software service.
Query	None.
Request	None.
Response	<u>XML_Cap_SoftwareService</u>

Remarks

The <ID> in the request URI refers to the channel No.

E.260 /ISAPI/System/SSDFileSystem/capacity?format=json

Search for SSD storage quotas.

Request URI Definition

Table C-380 GET /ISAPI/System/SSDFileSystem/capacity?format=json

Method	GET
Description	Search for SSD storage quotas.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_SSDCapacity</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

If the node **isSupportSSDFileSystemCapacity** is returned in the message *XML_DeviceCap* (related URI: [/ISAPI/System/capabilities](#)) and its value is true, it indicates that the device supports this function.

E.261 /ISAPI/System/SSDFileSystem/format/status?format=json

Search for the progress of SSD file system formatting.

Request URI Definition

Table C-381 GET /ISAPI/System/SSDFileSystem/format/status?format=json

Method	GET
Description	Search for the progress of SSD file system formatting.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_SSDFormatStatus</i> Failed: <i>JSON_ResponseStatus</i>

E.262 /ISAPI/System/SSDFileSystem/format?format=json

Format SSD file system.

Request URI Definition

Table C-382 PUT /ISAPI/System/SSDFileSystem/format?format=json

Method	PUT
Description	Format SSD file system.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_ResponseStatus</u>

Remarks

Formatting SSD will erase its data, including those in face picture library, and cancel the alarms linked to the library. The device will restart after formatting.

E.263 /ISAPI/System/SSDFileSystem/upgrade/status?format=json

Search for the upgrade progress of SSD file system.

Request URI Definition

Table C-383 GET /ISAPI/System/SSDFileSystem/upgrade/status?format=json

Method	GET
Description	Search for the upgrade progress of SSD file system.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_SSDUpgradeStatus</u> Failed: <u>JSON_ResponseStatus</u>

E.264 /ISAPI/System/SSDFileSystem/upgrade?format=json

Get the upgrade status of SSD file system or upgrade the SSD file system. The device will automatically reboot after upgrade.

Request URI Definition

Table C-384 GET /ISAPI/System/SSDFileSystem/upgrade?format=json

Method	GET
Description	Get the upgrade status of SSD file system.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_SSDUpgrade</i> Failed: <i>JSON_ResponseStatus</i>

Table C-385 PUT /ISAPI/System/SSDFileSystem/upgrade?format=json

Method	PUT
Description	Upgrade the SSD file system.
Query	format: determine the format of request or response message.
Request	None.
Response	<i>JSON_ResponseStatus</i>

E.265 /ISAPI/System/status

Get device status, e.g., CPU, memory, and so on.

Request URI Definition

Table C-386 GET /ISAPI/System/status

Method	GET
Description	Get device status, e.g., CPU, memory, and so on.
Query	None.
Request	None.
Response	Succeeded: <i>XML_DeviceStatus</i> Failed: <i>XML_ResponseStatus</i>

E.266 /ISAPI/System/time/capabilities

Get the time capability of the device.

Request URI Definition

Table C-387 GET /ISAPI/System/time/capabilities

Method	GET
Description	Get the time capability of the device.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Cap_Time</u> Failed: <u>XML_ResponseStatus</u>

E.267 /ISAPI/System/TwoWayAudio/channels

Get audio parameters of all two-way audio channels.

Request URI Definition

Table C-388 GET /ISAPI/System/TwoWayAudio/channels

Method	GET
Description	Get parameters of all two-way audio channels.
Query	None.
Request	None.
Response	Succeeded: <u>XML_TwoWayAudioChannelList</u> Failed: <u>XML_ResponseStatus</u>

Example

Sample Code for Getting Parameters of All Two-Way Audio Channels

```
GET /ISAPI/System/TwoWayAudio/channels HTTP/1.1
Host: 10.17.132.49
Connection: Keep-Alive
Authorization: Digest username="admin",
realm="DS-2CD2F12FWD-IWS",
nonce="4e4446464e6a6333516a63365a4445304f47526a4e32553d",
uri="/ISAPI/System/TwoWayAudio/channels",
```

```
cnonce="145ef1bc3ab456be09918d39b77e78ae",
nc=00000021,
response="d10bab23689c60e95241230fff3181b0",
qop="auth"
```

HTTP/1.1 200 OK
Date: Wed, 15 Mar 2017 09:25:29 GMT
Server: App-webs/
Connection: close
Content-Length: 467
Content-Type: application/xml

```
<?xml version="1.0" encoding="UTF-8"?>
<TwoWayAudioChannelList version="2.0" xmlns="http://www.isapi.com/ver20/XMLSchema">
<TwoWayAudioChannel version="2.0" xmlns="http://www.isapi.com/ver20/XMLSchema">
<id>1</id>
<enabled>false</enabled>
<audioCompressionType>G.711ulaw</audioCompressionType>
<audioInputType>MicIn</audioInputType>
<speakerVolume>50</speakerVolume>
<noisereduce>false</noisereduce>
</TwoWayAudioChannel>
</TwoWayAudioChannelList>
```

E.268 /ISAPI/System/TwoWayAudio/channels/<ID>

Operations about the configuration of a specific two-way audio channel.

Request URI Definition

Table C-389 GET /ISAPI/System/TwoWayAudio/channels/<ID>

Method	GET
Description	Get the parameters of a specific two-way audio channel.
Query	None.
Request	None.
Response	Succeeded: <u>XML_TwoWayAudioChannel</u> Failed: <u>XML_ResponseStatus</u>

Table C-390 PUT /ISAPI/System/TwoWayAudio/channels/<ID>

Method	PUT
Description	Set the parameters of a specific two-way audio channel.

Query	None.
Request	<u>XML_TwoWayAudioChannel</u>
Response	<u>XML_ResponseStatus</u>

E.269 /ISAPI/System/unitConfig/capabilities?format=json

Get the capability of unit unified configuration.

Request URI Definition

Table C-391 GET /ISAPI/System/unitConfig/capabilities?format=json

Method	GET
Description	Get the capability of unit unified configuration.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_unitConfigCap</u> Failed: <u>JSON_ResponseStatus</u>

E.270 /ISAPI/System/unitConfig?format=json

Get and set the unit unified configuration parameters.

Request URI Definition

Table C-392 GET /ISAPI/System/unitConfig?format=json

Method	GET
Description	Get the unit unified configuration parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_unitConfig</u> Failed: <u>JSON_ResponseStatus</u>

Table C-393 PUT /ISAPI/System/unitConfig?format=json

Method	PUT
Description	Set the unit unified configuration parameters.
Query	format: determine the format of request or response message.
Request	<u>JSON_unitConfig</u>
Response	<u>JSON_ResponseStatus</u>

E.271 /ISAPI/System/usb?format=json

Get or set the USB parameters.

Request URI Definition

Table C-394 GET /ISAPI/System/usb?format=json

Method	GET
Description	Get the USB parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_USB</u> Failed: <u>JSON_ResponseStatus</u>

Table C-395 PUT /ISAPI/System/usb?format=json

Method	PUT
Description	Set the USB parameters.
Query	format: determine the format of request or response message.
Request	<u>JSON_USB</u>
Response	<u>JSON_ResponseStatus</u>

E.272 /ISAPI/System/usb/capabilities?format=json

Get the USB parameters configuration capability.

Request URI Definition

Table C-396 GET /ISAPI/System/usb/capabilities?format=json

Method	GET
Description	Get the USB parameters configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_USBCap</i> Failed: <i>JSON_ResponseStatus</i>

E.273 /ISAPI/System/Video/capabilities

Get video channel capability.

Request URI Definition

Table C-397 GET /ISAPI/System/Video/capabilities

Method	GET
Description	Get video channel capability.
Query	None
Request	None
Response	Succeeded: <i>XML_VideoCap</i> Failed: <i>XML_ResponseStatus</i>

E.274 /ISAPI/System/Video/inputs/channels/<ID>/privacyMask/regions/<ID>

Get, set, or delete parameters of a privacy mask region.

Request URI Definition

Table C-398 GET /ISAPI/System/Video/inputs/channels/<ID>/privacyMask/regions/<ID>

Method	GET
Description	Get parameters of a privacy mask region.
Query	None.
Request	None.
Response	Succeeded: <i>XML_PrivacyMaskRegion</i> Failed: <i>XML_ResponseStatus</i>

Table C-399 PUT /ISAPI/System/Video/inputs/channels/<ID>/privacyMask/regions/<ID>

Method	PUT
Description	Set parameters of a privacy mask region.
Query	None.
Request	<i>XML_PrivacyMaskRegion</i>
Response	<i>XML_ResponseStatus</i>

Table C-400 DELETE /ISAPI/System/Video/inputs/channels/<ID>/privacyMask/regions/<ID>

Method	DELETE
Description	Delete parameters of a privacy mask region.
Query	None.
Request	None.
Response	<i>XML_ResponseStatus</i>

Remarks

The first <ID> in the request URI refers to the video input channel ID; and second <ID> is the privacy mask region ID.

E.275 /ISAPI/System/Video/inputs/channels/<ID>/VCAResource

Operation about configuration of intelligent resources switch.

Request URI Definition

Table C-401 GET /ISAPI/System/Video/inputs/channels/<ID>/VCAResource

Method	GET
Description	Get parameters of intelligent resources switch.
Query	None.
Request	None.
Response	Succeeded: <u>XML_VCAResource</u> Failed: <u>XML_ResponseStatus</u>

Table C-402 PUT /ISAPI/System/Video/inputs/channels/<ID>/VCAResource

Method	PUT
Description	Set parameters of intelligent resources switch.
Query	None.
Request	<u>XML_VCAResource</u>
Response	<u>XML_ResponseStatus</u>

Remarks

- The <ID> in the request URI refers to the video input channel ID.
- The intelligent resources switch supports three capture mode, i.e., target capture, face capture, and perimeter capture.
 - For target capture mode, the panoramic camera captures the human body and vehicle pictures, the speed dome tracks and capture the human body for human body or face recognition and comparison. If matched, the thumbnails and original pictures of human body and face will be uploaded; otherwise, only the thumbnail and original picture of face will be uploaded.
 - For face capture mode, the panoramic camera transmits the videos without any intelligent analysis, the speed dome captures face picture in up to 4 scenes.
 - For perimeter capture, the panoramic camera captures vehicle pictures and performs HMS +DGS detection, and it supports detecting and tracking the small target within 40 m; while the speed dome tracks and captures the detected target, and outputs thumbnails and original pictures of human body and face.

E.276 /ISAPI/System/Video/inputs/channels/<ID>/VCAResource/capabilities

Get capability of intelligent resources switch.

Request URI Definition

Table C-403 GET /ISAPI/System/Video/inputs/channels/<ID>/VCAResource/capabilities

Method	GET
Description	Get capability of intelligent resources switch.
Query	None.
Request	None.
Response	Succeeded: <u>XML_Cap_VCAResource</u> Failed: <u>XML_ResponseStatus</u>

E.277 /ISAPI/System/Video/inputs/channels/counting/collection/capabilities?format=json

Get the capability of people counting data replenishment.

Request URI Definition

Table C-404 GET /ISAPI/System/Video/inputs/channels/counting/collection/capabilities?
format=json

Method	GET
Description	Get the capability of people counting data replenishment.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_PeopleCounting_CollectionDescriptionCap</u>

E.278 /ISAPI/System/Video/inputs/channels/counting/collection? format=json

Perform people counting replenishment.

Request URI Definition

Table C-405 POST /ISAPI/System/Video/inputs/channels/counting/collection?format=json

Method	POST
Description	Perform people counting replenishment.
Query	format: determine the format of request or response message.
Request	<i><u>JSON_PeopleCounting_CollectionDescription</u></i>
Response	<i><u>JSON_PeopleCounting_CollectionResult</u></i>

E.279 /ISAPI/System/Video/inputs/channels/heatMap/collection/capabilities?format=json

Get the capability of heat map data replenishment.

Request URI Definition

Table C-406 GET /ISAPI/System/Video/inputs/channels/heatMap/collection/capabilities?format=json

Method	GET
Description	Get the capability of heat map data replenishment.
Query	format: determine the format of request or response message.
Request	None.
Response	<i><u>JSON_HeatMap_CollectionDescriptionCap</u></i>

E.280 /ISAPI/System/Video/inputs/channels/heatMap/collection?format=json

Perform heat map data replenishment.

Request URI Definition

Table C-407 POST /ISAPI/System/Video/inputs/channels/heatMap/collection?format=json

Method	POST
Description	Perform heat map data replenishment.
Query	format: determine the format of request or response message.
Request	<u>JSON HeatMap CollectionDescription</u>
Response	<u>JSON HeatMap CollectionResult</u>

E.281 /ISAPI/System/Video/inputs/OSDLanguag

Operations about OSD language configuration.

Request URI Definition

Table C-408 GET /ISAPI/System/Video/inputs/OSDLanguag

Method	GET
Description	Get OSD language parameters.
Query	None.
Request	None.
Response	<u>XML Language</u>

Table C-409 PUT /ISAPI/System/Video/inputs/OSDLanguag

Method	PUT
Description	Set OSD language parameters.
Query	None.
Request	<u>XML Language</u>
Response	<u>XML ResponseStatus</u>

E.282 /ISAPI/System/workingstatus?format=json

Get device working status.

Request URI Definition

Table C-410 GET /ISAPI/System/workingstatus?format=json

Method	GET
Description	Get device working status.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_WorkingStatus</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

Up to 20 bytes are allowed for a device's working status; up to 100 bytes are allowed for a channel's status, a HDD's status, and an IO's status; up to 30 bytes are allowed for a two-way audio channel's status.

E.283 /ISAPI/System/zeroBiasCalibration/channels/<ID>?format=json

Calibrate the accelerometer bias.

Request URI Definition

Table C-411 PUT /ISAPI/System/zeroBiasCalibration/channels/<ID>?format=json

Method	PUT
Description	Calibrate the accelerometer bias.
Query	format: determine the format of request or response message.
Request	None.
Response	<i>JSON_ResponseStatus</i>

Remarks

The <ID> in the request URI refers to the channel No.

E.284 /ISAPI/Thermal/capabilities

Get thermal capability.

Request URI Definition

Table C-412 GET /ISAPI/Thermal/capabilities

Method	GET
Description	Get the thermal capability.
Query	None.
Request	None.
Response	<u>XML_ThermalCap</u>

E.285 /ISAPI/Thermal/temperature/collection/capabilities?format=json

Get the capability of temperature data replenishment.

Request URI Definition

Table C-413 GET /ISAPI/Thermal/temperature/collection/capabilities?format=json

Method	GET
Description	Get the capability of temperature data replenishment.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_Temperature_CollectionDescriptionCap</u>

E.286 /ISAPI/Thermal/temperature/collection?format=json

Perform the temperature data replenishment.

Request URI Definition

Table C-414 GET /ISAPI/Thermal/temperature/collection?format=json

Method	POST
Description	Perform the temperature data replenishment.
Query	format: determine the format of request or response message.

Request	<i>JSON_CollectionDescription</i>
Response	<i>JSON_CollectionResult</i>

E.287 /ISAPI/Traffic/ContentMgmt/InputProxy/channels/<ID>/ocrScene/capabilities

Get the capability for configuring the parameters of a scene.

Request URI Definition

Table C-415 GET /ISAPI/Traffic/ContentMgmt/InputProxy/channels/<ID>/ocrScene/capabilities

Method	GET
Description	Get the capability for configuring the parameters of a scene.
Query	None.
Request	None.
Response	Succeeded: <i>XML_OcrConfigCap</i> Failed: <i>XML_ResponseStatus</i>

Remarks

The <ID> in the request URI refers to the channel ID.

Appendix F. Request and Response Messages

F.1 JSON_AlarmStatistics

AlarmStatistics message in JSON format

```
{
  "AlarmStatistics":{
    /*optional, alarm statistics of added IoT devices*/
    "totalNumber": "",
    /*required, the total number of alarms, integer*/
    "time": "",
    /*required, statistic time, integer, unit: second*/
  }
}
```

F.2 JSON_AssociatedChannelList

AssociatedChannelList message in JSON format

```
{
  "AssociatedChannelList": [
    /*optional, list of linked channels*/
    {
      "channel": "",
      /*required, channel No., integer*/
      "type": "",
      /*required, channel type: "video"-video channel, "IOT"-IoT channel, string*/
      "associatedVideoChannels": "",
      /*optional, linked video channel No., array, and the unit is integer*/
      "associatedIOTChannels": ""
    }
  ]
}
```

F.3 JSON_AudioAlarm

AudioAlarm message in JSON format

```
{
  "AudioAlarm":{
    "audioID": ,
    /*required, alarm sound types, 1-siren, 2-"Warning, this is a restricted area", 3-"Waring, this is a restricted area, please keep away", 4-"Warning, this is a no-parking zone", 5-"Warning, this is a no-parking zone, please keep away", 6-"Attention please. The area is under surveillance", 7-"Welcome, Please notice that the area is under surveillance", 8-"Welcome", 9-"Danger! Please keep away", 10-siren + "Danger, please keep away", 11-Audio Warning, 12-Beep Sound, 13-custom alarm sound 1, 14-custom alarm sound 2, 15-custom alarm sound 3*/
  }
}
```

```
"audioVolume": ,  
/*required, int, volume, which is between 1 and 100*/  
"alarmTimes": ,  
/*required, int, times of alarm, which is between 1 and 50*/  
"TimeRangeList": [{  
/*optional, alarm output schedule list*/  
"week": ,  
"TimeRange": [{  
"id": ,  
/*required, int, ID of time period for each day*/  
"beginTime": "",  
/*required, string, start time in ISO8601 format*/  
"endTime": ""  
/*required, string, end time in ISO8601 format*/  
}]  
}  
"audioClass": "",  
/*optional, string, read-only, sound type: "alertAudio"-alarm sound, "promptAudio"-prompt sound, "customAudio"-  
custom sound*/  
"alertAudioID": ,  
/*dependent, int, read-only, alarm sound ID, this node is valid only when audioClass is alertAudio, 1-"Siren",  
2-"Warning, this is a restricted area", 3-"Waring, this is a restricted area, please keep away", 4-"Warning, this is a no-  
parking zone", 5-"Warning, this is a no-parking zone, please keep away", 6-"Attention please. The area is under  
surveillance", 7-"Welcome, Please notice that the area is under surveillance", 8-"Welcome", 9-"Danger! Please keep  
away", 10-"(Siren)&Danger, please keep away, 11-  
Audio Warning"*/  
"customAudioID": ,  
/*dependent, int, read-only, custom sound ID, this node will be returned when audioClass is "customAudio". The  
alarm sound file can only be uploaded via Guarding Vision in specified format. If there is no value returned in this  
node, the custom sound (customAudio) will not be returned,either*/  
"alarmType": ""  
/*optional, string, alarm type: "behavior"-behavior analysis, "thermometry"-thermometry, "dynamicFire"-fire  
detection, "smokingMode"-smoke detection, "noMaskDetection"-no wearing mask detection, "AIOP"-AI Open  
Platform event, "PQA"-people amount exception (the sub type of people density detection). The alarmType should be  
linked to the audioID*/  
}  
}
```

Remarks

1. The user can configure the sound file via the node **audioID** only, and when the **audioID** is specified, its value will be automatically synchronized to the nodes of **audioClass**, **alertAudioID**, and **customAudioID**, which are read-only.
2. When the value of **audioID** is from 1 to 11, the corresponding value of **audioClass** is "alertAudio", and the value of **alertAudioID** is also from 1 to 11; when the value of **audioID** is 12, the corresponding value of **audioClass** is "promptAudio"; when the value of **audioID** is from 13 to 15, the corresponding value of **audioClass** is "customAudio", and the value of **customAudioID** is from 1 to 3.

F.4 JSON_AudioAlarmCap

JSON message about audible warning configuration capability

```
{
  "AudioAlarmCap": {
    "audioTypeListCap": [
      /*required, alarm sound types, 1-siren, 2-"Warning, this is a restricted area", 3-"Waring, this is a restricted area, please keep away", 4-"Warning, this is a no-parking zone", 5-"Warning, this is a no-parking zone, please keep away", 6-"Attention please. The area is under surveillance", 7-"Welcome, Please notice that the area is under surveillance", 8-"Welcome", 9-"Danger! Please keep away", 10-siren + "Danger, please keep away", 11-Audio Warning, 12-Beep Sound, 13-custom alarm sound 1, 14-custom alarm sound 2, 15-custom alarm sound 3, 16-AI custom audio file*/
      "audioID": 1,
      "audioDescription": "siren"
    ],
    {
      "audioID": 2,
      "audioDescription": "Warning, this is a restricted area"
    },
    {
      "audioID": 3,
      "audioDescription": "Waring, this is a restricted area, please keep away"
    }],
    "audioVolume": {
      /*required, int, volume, which is between 1 ad 100*/
      "@min": 1,
      "@max": 100,
      "@def": 100
    },
    "alarmTimes": {
      /*required, int, times of audible warning, which is between 1 and 50*/
      "@min": 1,
      "@max": 50,
      "@def": 5
    },
    "TimeRangeCap": {
      /*optional, audible warning schedule capability*/
      "week": {
        /*required, int, days of the week: 1-Monday, 2-Tuesday, 3-Wednesday, 4-Thursday, 5-Friday, 6-Saturday, 7-Sunday*/
        "@opt": "1, 2, 3, 4, 5, 6, 7"
      },
      "id": {
        /*required, int, ID of time period for each day*/
        "@maxSize": 8
      },
      "beginTime": {
        /*required, string, start time in ISO8601 format*/
        "@min": "00:00",
        "@max": "24:00"
      },
      "endTime": {
        /*required, string, end time in ISO8601 format*/
        "@min": "00:00",
        "@max": "24:00"
      }
    }
  }
}
```

```

/*required, string, end time in ISO8601 format*/
    "@min":"00:00",
    "@max":"24:00"
}
}

"audioClass": {

/*optional, string, read-only, sound type, "alertAudio"-alarm sound (get the supported alarm sound types from the node AlertAudioTypeListCap), "promptAudio"-prompt sound, "customAudio"-custom sound (configure it according to customAudioID); if this node is not returned, you should get the supported sound types from the node audioTypeListCap*/

    "@opt": "alertAudio, promptAudio, customAudio",
    "@def": "alertAudio"
},
"AlertAudioTypeListCap": [{

/*dependent, read-only, list of supported alarm sound types, this node will be returned when audioClass is "alertAudio": 1-"Siren", 2-"Warning, this is a restricted area", 3-"Waring, this is a restricted area, please keep away", 4-"Warning, this is a no-parking zone", 5-"Warning, this is a no-parking zone, please keep away", 6-"Attention please. The area is under surveillance", 7-"Welcome, Please notice that the area is under surveillance", 8-"Welcome", 9-"Danger! Please keep away", 10-(Siren)&Danger, please keep away, 11-Audio Warning*/
        "alertAudioID": 1,
        "alertAudioDescription": "Siren"
    },
    {
        "alertAudioID": 2,
        "alertAudioDescription": "Warning, this is a restricted area"
    },
    {
        "alertAudioID": 3,
        "alertAudioDescription": "Waring, this is a restricted area, please keep away"
    }
],
"customAudioID": {

/*dependent, int, read-only, custom sound ID, this node will be returned when audioClass is "customAudio". The alarm sound file can only be uploaded via Guarding Vision in specified format. If there is no value returned in this node, the custom sound (customAudio) will not be returned,either*/
        "@min": 1,
        "@max": 3,
        "@def": 1
    },
"alarmType":{

/*optional, string, alarm type: "behavior"-behavior analysis, "thermometry"-thermometry, "dynamicFire"-fire detection, "smokingMode"-smoke detection, "noMaskDetection"-no wearing mask detection, "AIOP"-AI Open Platform event, "PQA"-people amount exception (the sub type of people density detection)*/
        "@opt": "behavior,thermometry,dynamicFire,smokingMode,noMaskDetection,AIOP,PQA",
        "@def": "thermometry"
    },
"AlarmBindAudioList":[]

/*optional, alarm type and the list of audio related to alarm type*/
    "alarmType": "behavior",
/*optional, string, alarm type: "behavior"-behavior analysis, "thermometry"-thermometry, "dynamicFire"-fire detection, "smokingMode"-smoke detection, "noMaskDetection"-no wearing mask detection, "AIOP"-AI Open Platform event, "PQA"-people amount exception (the sub type of people density detection)*/
        "audioID":{

/*optional, string, the audible warning type ID corresponding to alarm type, behavior analysis corresponds to 1-12,
```

```
thermometry corresponds to 13, fire detection corresponds to 14, smoke detection corresponds to 15*/
    "@opt":"1,2,3,4,5,6,7,8,9,10,11,12"
}
}]
"isSupportCustomAudio": true,
/*optional, boolean, whether the device supports uploading, downloading, and editing custom alarm sound file*/
    "isSupportAudioTest": true,
/*optional, boolean, whether the device supports sound file test*/
    "customAudioName": {
/*optional, string, name of custom sound file, the maximum size is 24 bytes*/
        "@min": 1,
        "@max": 24
    }
    "isSupportCustomAudioFormatInfo":true,
/*optional, boolean, whether the device supports getting the custom audio file format*/
    "isSupportCustomAudioURL":true,
/*optional, boolean, whether the device supports importing the custom audio file in URL format; if this node is not
returned, the device dose not support the function*/
    "isSupportCustomAudioFiles":true
/*optional, boolean, whether the device supports importing the custom audio files in a batch; if this node is not
returned, the device dose not support the function*/
}
}
```

Remarks

1. The user can configure the sound file via the node **audioID** only, and when the **audioID** is specified, its value will be automatically synchronized to the nodes of **audioClass**, **alertAudioID**, and **customAudioID**, which are read-only.
2. When the value of **audioID** is from 1 to 11, the corresponding value of **audioClass** is "alertAudio", and the value of **alertAudioID** is also from 1 to 11; when the value of **audioID** is 12, the corresponding value of **audioClass** is "promptAudio"; when the value of **audioID** is from 13 to 15, the corresponding value of **audioClass** is "customAudio", and the value of **customAudioID** is from 1 to 3.

F.5 JSON_Authentication

JSON message about parameters of serial authentication service

```
{
    "Authentication":{
        "enabled":,
/*required, boolean, whether to enable serial authentication service*/
        "serviceClosurePeriod":,
/*dependent, integer, service life, unit: day; it is required when enabled is "false"*/
    }
}
```

F.6 JSON_AuthenticationCap

JSON message about serial authentication service

```
{  
  "AuthenticationCap":{  
    "enabled":{  
      /*required, boolean, whether to enable serial authentication service*/  
      "@opt": [  
        true,  
        false  
      ]  
    },  
    "serviceClosurePeriod":{  
      /*dependent, integer, service life, unit: day; it is required when enabled is "false"*/  
      "@min":1,  
      "@max":30,  
      "@def":7  
    }  
  }  
}
```

F.7 JSON_AutoEagleFocusing

AutoEagleFocusing message in JSON format

```
{  
  "AutoEagleFocusing": {  
    /*required, auto calibration of rapid focus*/  
    "EagleFocusingRegion": [{  
      /*optional, string, rapid focus region*/  
      "sid": "",  
      /*optional, string, scene ID*/  
      "spotNum": ,  
      /*optional, integer, the number of calibration points*/  
      "type": "",  
      /*optional, string, region type now only supports "line"*/  
      "Region": [{  
        /*optional, region list*/  
        "x": ,  
        "y": ,  
      }],  
      "AbsoluteHigh": {  
        /*optional, PTZ position*/  
        "elevation": ,  
        /*optional, xs:integer, tilt angle, range: [-900,2700]*/  
        "azimuth": ,  
        /*optional, xs:integer, azimuth, range: [0,3600]*/  
        "absoluteZoom": 33  
      }  
    }]  
  }  
}
```

```
/*optional, xs:integer, zoom, range: [1,1000]*/
    }
}
}
}
```

F.8 JSON_AutoEagleFocusingCap

AutoEagleFocusingCap message in JSON format

```
{
  "AutoEagleFocusingCap": {
    /*required, auto calibration capability of rapid focus*/
    "EagleFocusingRegion": {
      "sid": {
        /*optional, string, scene ID*/
        "@min": 1,
        "@max": 2
      },
      "spotNum": {
        /*optional, integer, the number of calibration points*/
        "@min": 1,
        "@max": 64
      },
      "type": {
        /*optional, string, region type now only supports "line"*/
        "@opt": "line"
      },
      "RegionCap": {
        "minSize": 3,
        /*required, integer, the minimum number of region edges*/
        "maxSize": 10,
        /*required, integer, the maximum number of region edges*/
        "x": {
          /*required, float, X-coordinate, value range: 0.000 to 1*/
          "@min": 0.000,
          "@max": 1.000,
          "#text": 0.120
        },
        "y": {
          /*required, float, Y-coordinate, value range: 0.000 to 1*/
          "@min": 0.000,
          "@max": 1.000,
          "#text": 0.120
        }
      },
      "AbsoluteHighCap": {
        "elevation": {
          /*optional, xs:integer, tilt angle, range: [-900,2700]*/
          "@min": -900,
          "@max": 2700
        }
      }
    }
  }
}
```

```
    "@max": 2700
},
"azimuth": {
/*optional, xs:integer, azimuth, range: [0,3600]*/
    "@min": 0,
    "@max": 3600
},
"absoluteZoom": {
/*optional, xs:integer, zoom, range: [1,1000]*/
    "@min": 1,
    "@max": 1000
}
}
```

F.9 JSON_AutoMaintenance

AutoMaintenance message in JSON format

```
{
"AutoMaintenance":{
    "enabled": true,
/*required, boolean, enable or not*/
    "dayOfWeek": "",
/*required, integer, day of the week, ranges from 1 to 7, 1-Monday, 2-Tuesday, and so on*/
    "rebootTime": "",
/*required, reboot time, corrects to minute, it is 0 by default, ISO8601 format, string*/
    }
}
```

F.10 JSON_AutoMaintenanceCap

AutoMaintenanceCap message in JSON format

```
{
"AutoMaintenanceCap":{
    "enabled": "true,false",
/*required, boolean, enable or not*/
    "dayOfWeek": {
/*required, integer, day of the week, ranges from 1 to 7, 1-Monday, 2-Tuesday, and so on*/
        "@min": 1,
        "@max": 7,
        "#text": 1
    },
}
}
```

F.11 JSON_AutoSwitch

AutoSwitch message in JSON format

```
{  
  "AutoSwitch":{  
    "enabled":true,  
    /*required, boolean, whether to enable power on or off*/  
    "openTime":"10:00:00",  
    /*required, string, power on time, which is accurate to minute with the time format of ISO8601*/  
    "closeTime":"10:00:00"  
  }  
}
```

F.12 JSON_AutoSwitchCap

AutoSwitchCap message in JSON format

```
{  
  "AutoSwitchCap":{  
    "enabled":"true,false"  
    /*required, boolean, whether to enable auto power on or off*/  
  }  
}
```

F.13 JSON_BasicParam

BasicParam message in JSON format

```
{  
  "BasicParam":{  
    /*required, basic parameters of added IoT device*/  
    "channelName": ""  
    /*required, channel name, string, the max. length is 32*/  
  }  
}
```

F.14 JSON_BVCorrectParam

BVCorrectParam message in JSON format

```
{  
  "BVCorrectParam":{  
    /*camera correction parameters*/  
    "lensType":""  
  }  
}
```

```
/*required, string type, focal length of the lens: "unknown", "8mm", "12mm", "16mm", "25mm", "35mm", "50mm",
"4mm", "6mm", "2.0mm", "2.8mm"*/
    "ReprojectMatrix"::[
/*required, reprojection matrix parameters, it is a 4*4 two-dimensional array (array[4][4])*/
        "index": ,
/*required, int type, the one-dimensional array subscript of the matrix parameter*/
        "Reproject":[{{
/*required, specific matrix parameters*/
            "index": ,
/*required, int type, the two-dimensional array subscript of the matrix parameter*/
            "value": 
/*required, float type, matrix parameters*/
        }]}
    ],
    "CameraCorrect":[{{
/*required, correction parameters of the left camera and the right camera*/
        "cameraType":"",
/*required, string type, camera type: "LCamera"-left camera, "RCamera"-right camera*/
        "CamInternalMatrix":[{{
/*required, matrix of camera internal parameters (including focal length and principle point coordinates), it is a 3*3
two-dimensional array (array[3][3])*/
            "index": ,
/*required, int type, the one-dimensional array subscript of the matrix parameter*/
            "CamInternal":[{{
/*required, specific matrix parameters*/
                "index": ,
/*required, int type, the two-dimensional array subscript of the matrix parameter*/
                "value": 
/*required, float type, matrix parameters*/
            }]}
        }],
        "DistCoeffs":[{{
/*required, lens distortion coefficient, it is a one-dimensional array with 8 elements (array[8])*/
            "index": ,
/*required, int type, the array subscript*/
            "value": 
/*required, float type, matrix parameters*/
        }]},
        "RotateMatrix":[{{
/*required, rotation matrix after correction, it is a 3*3 two-dimensional array (array[3][3])*/
            "index": ,
/*required, int type, the one-dimensional array subscript of the matrix parameter*/
            "Rotate":[{{
/*required, specific matrix parameters*/
                "index": ,
/*required, int type, the two-dimensional array subscript of the matrix parameter*/
                "value": 
/*required, float type, matrix parameters*/
            }]}
        }],
        "ProjectMatrix":[{{
/*required, projection matrix after correction, it is a 3*4 two-dimensional array (array[3][4])*/

```

```

    "index": ,
/*required, int type, the one-dimensional array subscript of the matrix parameter*/
    "Project":[{
/*required, specific matrix parameters*/
        "index": ,
/*required, int type, the two-dimensional array subscript of the matrix parameter*/
        "value": 
/*required, float type, matrix parameters*/
    }]
},
}],
"RotateMatrix":[{
/*required, rotation matrix of the left camera relative to the right camera after calibrating dual-lens camera, it is a 3*3
two-dimensional array (array[3][3]). RotateMatrix and TransMatrix are used to describe the position of the left
camera relative to the right camera*/
    "index": ,
/*required, int type, the one-dimensional array subscript of the matrix parameter*/
    "Rotate":[{
/*required, specific matrix parameters*/
        "index": ,
/*required, int type, the two-dimensional array subscript of the matrix parameter*/
        "value": 
/*required, float type, matrix parameters*/
    }]
},
}],
"TransMatrix":[{
/*required, translation matrix, it is a one-dimensional array with three elements (array[3]). RotateMatrix and
TransMatrix are used to describe the position of the left camera relative to the right camera*/
    "index": ,
/*required, int type, the one-dimensional array subscript of the matrix parameter*/
    "value": 
/*required, float type, matrix parameters*/
},
{
    "OrilImgResolution":{
/*optional, original picture resolution*/
        "orilImgResolutionWidth":{
/*required, int type, width of the original picture resolution*/
            "@opt": 
        },
        "orilImgResolutionHeight":{
/*required, int type, height of the original picture height*/
            "@opt": 
        }
    }
}
]
}

```

F.15 JSON_BVCorrectParamCap

BVCorrectParamCap capability message in JSON format

```
{
    "BVCorrectParamCap":{

        /*configuration capability of camera correction*/
        "lensType":{

            /*required, string type, focal length of the lens: "unknown", "8mm", "12mm", "16mm", "25mm", "35mm", "50mm",
            "4mm", "6mm", "2.0mm", "2.8mm"*/
            "@opt":["unknown", "8mm", "12mm", "16mm", "25mm", "35mm", "50mm", "4mm", "6mm", "2.0mm", "2.8mm"]
        },
        "ReprojectMatrixCap":{

            /*required, configuration capability of reprojection matrix, it is a 4*4 two-dimensional array (array[4][4])*/
            "index":{

                /*required, int type, range of the one-dimensional array subscript of the matrix parameter*/
                "@min":0,
                "@max":3
            },
            "ReprojectCap":{

                /*required, specific matrix configuration capability*/
                "index":{

                    /*required, int type, range of the two-dimensional array subscript of the matrix parameter*/
                    "@min":0,
                    "@max":3
                },
                "value":{

                    /*required, float type, matrix parameters*/
                    "@min":-4000.000000,
                    "@max":8000.000000
                }
            }
        },
        "CameraCorrectCap":{

            /*required, configuration capability of the left camera and the right camera correction*/
            "cameraType":{

                /*required, string type, camera type: "LCamera"-left camera, "RCamera"-right camera*/
                "@opt":["LCamera", "RCamera"]
            },
            "CamInternalMatrixCap":{

                /*required, matrix of camera internal parameters (including focal length and principle point coordinates), it is a 3*3
                two-dimensional array (array[3][3])*/
                "index":{

                    /*required, int type, range of the one-dimensional array subscript of the matrix parameter*/
                    "@min":0,
                    "@max":2
                },
                "CamInternalCap":{

                    /*required, specific matrix configuration capability*/
                    "index":{

                        /*required, int type, range of the two-dimensional array subscript of the matrix parameter*/
                        "@min":0,
                        "@max":2
                    },
                    "value":{

                        /*required, float type, matrix parameters*/
                    }
                }
            }
        }
    }
}
```

```
    "@min": -4000.000000,
    "@max": 8000.000000
  }
}
},
"DistCoeffsCap": {
/*required, lens distortion coefficient, it is a one-dimensional array with 8 elements (array[8])*/
  "index": {
/*required, int type, range of the array subscript*/
    "@min": 0,
    "@max": 7
  },
  "value": {
/*required, float type, matrix parameters*/
    "@min": -4000.000000,
    "@max": 8000.000000
  }
},
"RotateMatrixCap": {
/*required, rotation matrix after correction, it is a 3*3 two-dimensional array (array[3][3])*/
  "index": {
/*required, int type, range of the one-dimensional array subscript of the matrix parameter*/
    "@min": 0,
    "@max": 2
  },
  "RotateCap": {
/*required, specific matrix configuration capability*/
    "index": {
/*required, int type, range of the two-dimensional array subscript of the matrix parameter*/
      "@min": 0,
      "@max": 2
    },
    "value": {
/*required, float type, matrix parameters*/
      "@min": -4000.000000,
      "@max": 8000.000000
    }
  }
},
"ProjectMatrixCap": {
/*required, projection matrix after correction, it is a 3*4 two-dimensional array (array[3][4])*/
  "index": {
/*required, int type, range of the one-dimensional array subscript of the matrix parameter*/
    "@min": 0,
    "@max": 2
  },
  "ProjectCap": {
/*required, specific matrix configuration capability*/
    "index": {
/*required, int type, range of the two-dimensional array subscript of the matrix parameter*/
      "@min": 0,
      "@max": 3
    }
  }
}
```

```

    },
    "value":{

/*required, float type, matrix parameters*/
    "@min":-4000.000000,
    "@max":8000.000000
}
},
},
"OrilImgResolutionCap":{

/*optional, original picture resolution capability*/
    "orilImgResolutionWidth":{

/*required, int type, width of the original picture resolution*/
        "@opt":"1920"
    },
    "orilImgResolutionHeight":{

/*required, int type, height of the original picture height*/
        "@opt":"1080"
    }
},
"RotateMatrixCap":{

/*required, rotation matrix of the left camera relative to the right camera after calibrating dual-lens camera, it is a 3*3 two-dimensional array (array[3][3]). RotateMatrix and TransMatrix are used to describe the position of the left camera relative to the right camera*/
    "index":{

/*required, int type, range of the one-dimensional array subscript of the matrix parameter*/
        "@min":0,
        "@max":2
    },
    "RotateCap":{

/*required, specific matrix configuration capability*/
        "index":{

/*required, int type, range of the two-dimensional array subscript of the matrix parameter*/
            "@min":0,
            "@max":2
        },
        "value":{

/*required, float type, matrix parameters*/
            "@min":-4000.000000,
            "@max":8000.000000
        }
    }
},
"TransMatrixCap":{

/*required, translation matrix, it is a one-dimensional array with three elements (array[3]). RotateMatrix and TransMatrix are used to describe the position of the left camera relative to the right camera*/
    "index":{

/*required, int type, range of the one-dimensional array subscript of the matrix parameter*/
        "@min":0,
        "@max":2
    },
    "value":{

```

```
/*required, float type, matrix parameters*/
    "@min": -4000.000000,
    "@max": 8000.000000
}
}
}
}
```

F.16 JSON_CalibrationStatus

CalibrationStatus message in JSON format

```
{
  "CalibrationStatus": {
    /*required, string type, installation angle calibration status: "Calibrated"-calibrated, "Uncalibrated"-uncalibrated*/
  }
}
```

F.17 JSON_Cap_CalibrationStatus

CalibrationStatus message in JSON format

```
{
  "CalibrationStatus": {
    /*required, string type, installation angle calibration status: "Calibrated"-calibrated, "Uncalibrated"-uncalibrated*/
    "@opt": "Calibrated,Uncalibrated"
  }
}
```

F.18 JSON_Cap_CaputreDescription

JSON message about capability of capturing picture in URL format

```
{
  "CaputreDescription": {
    "imageType": {
      /*required, string, picture type: "JEPG"*/
      "@opt": ["JPEG"]
    },
    "URLType": {
      /*required, string, URL type: "cloudURL"*/
      "@opt": ["cloudURL"]
    }
  }
}
```

F.19 JSON_Cap_ClearingSpaceConfig

JSON message about configuration capability of HDD clearing

```
{
  "ClearingSpaceConfig": {
    /*when the free space of HDD reaches the configured value of insufficientSpace, and if the value of clearingOriginalData is true, the configured HDD space size will be cleared*/
    "insufficientSpace": {
      /*required, int, HDD free space, which is used as the threshold to start clearing*/
      "@min":0,
      /*minimum value of the range*/
      "@max":100
      /*maximum value of the range*/
    },
    "clearingOriginalData": {
      /*required, boolean, whether to enable clearing the original data*/
      "opt":"true,false"
    },
    "clearingSpace": {
      /*required, int, space size to clear*/
      "@min":0,
      /*minimum value of the range*/
      "@max":100
      /*maximum value of the range*/
    }
  }
}
```

F.20 JSON_Cap_IntelliManagement

JSON message about capability of intelligent management

```
{
  "requestURL":"",
  "statusCode": ,
  "statusString":"",
  "subStatusCode":"",
  "errorCode": ,
  "errorMsg":"",
  /*see the description of this node and the above nodes in the message of JSON_ResponseStatus*/
  "isSupportTask":true,
  /*optional, boolean type, whether it supports task: "true"-yes, "false"-no*/
  "isSupportCtrlCenter":true,
  /*optional, boolean type, whether it supports control center: "true"-yes, "false"-no*/
  "isSupportIntelligentSearch":true,
  /*optional, boolean type, whether it supports intelligent search: "true"-yes, "false"-no*/
  "isSupportIntelligentSearchAsync":true,
  /*optional, boolean, whether it supports asynchronous intelligent search, related URI: /ISAPI/SDT/Management/
```

```
IntelligentSearch/async/capabilities?format=json*/
  "isSupportExecuteControl":true,
  /*optional, boolean type, whether it supports arming: "true"-yes, "false"-no*/
  "isSupportDataStatistics":true,
  /*optional, boolean type, whether it supports statistics: "true"-yes, "false"-no*/
  "isSupportIntelligentStorage":true,
  /*optional, boolean type, whether it supports intelligent storage: "true"-yes, "false"-no*/
  "isSupportServer":true,
  /*optional, boolean type, whether it supports service configuration: "true"-yes, "false"-no*/
  "isSupportAlgorithmModel":true,
  /*optional, boolean type, whether it supports algorithm model configuration: "true"-yes, "false"-no*/
  "isSupportCluster":true,
  /*optional, boolean type, whether it supports cluster: "true"-yes, "false"-no*/
  "isSupportSearchResultDisplay":true,
  /*optional, boolean type, whether it supports display configuration: "true"-yes, "false"-no*/
  "isSupportSearchResultDisplayCfg":true,
  /*optional, boolean, whether it supports configuring global parameters for displaying search results, related URI: /ISAPI/SDT/Management/SearchResultDisplayCfg/capabilities?format=json*/
  "isSupportTranscodeAnalysisUnit":true,
  /*optional, boolean type, whether it supports transcoding analysis unit: "true"-yes, "false"-no*/
  "isSupportFactoryReset":true,
  /*optional, boolean type, whether it supports restoring to factory settings: "true"-yes, "false"-no*/
  "isSupportUPS":true,
  /*optional, boolean type, whether it supports UPS: "true"-yes, "false"-no*/
  "isSupportCity":true,
  /*optional, boolean type, whether it supports city: "true"-yes, "false"-no*/
  "isSupportClusterEnableStatus":true,
  /*optional, boolean type, whether it supports enabling cluster status: "true"-yes, "false"-no*/
  "isSupportSensitivity":true,
  /*optional, boolean type, whether it supports sensitivity configuration: "true"-yes, "false"-no*/
  "isSupportConfidence":true,
  /*optional, boolean type, whether it supports confidence configuration: "true"-yes, "false"-no*/
  "isSupportEventSearch":true,
  /*optional, boolean, whether it supports event search: "true"-yes, "false"-no*/
  "isSupportAudioFileUpload":true,
  /*optional, boolean type, whether it supports importing audio file: "true"-yes, "false"-no*/
  "isSupportIntelligentMode":true,
  /*optional, boolean type, whether it supports intelligent mode configuration: "true"-yes, "false"-no*/
  "saveCapturedDataMaxDays":90,
  /*optional, int*, the captured pictures can be saved in local HDD up to 90 days by default, when the number of days reaches the limit, the records will be overwritten by day.*/
  "isSupportAlgorithmTraining": true,
  /*optional, boolean, whether it supports algorithm training, "true"-yes, "false"-no*/
  "isSupportDataResync":true,
  /*optional, boolean, whether it supports data resynchronization: "true"-yes, "false"-no*/
  "isSupportCertificateImport":true,
  /*optional, boolean, whether it supports importing software license certificate: "true"-yes, "false"-no*/
  "isSupportFingerprint":true,
  /*optional, boolean, whether it supports exporting hardware fingerprint: "true"-yes, "false"-no*/
  "isSupportAuthorization":true,
  /*optional, boolean, whether it supports authorization configuration of the algorithm center: "true"-yes, "false"-no*/
  "isSupportLicensingDongle":true,
```

```
/*optional, boolean, whether it supports dongle: "true"-yes, "false"-no*/
    "isSupportAlgorithmUpdate":true,
/*optional, boolean, whether it supports updating algorithm: "true"-yes, "false"-no*/
    "isSupportModellingMode":true,
/*optional, boolean, whether it supports modeling mode configuration*/
    "isSupportListLoading":true,
/*optional, boolean, whether it supports list library loading configuration*/
    "isSupportTasksStatistics": true,
/*optional, boolean, whether it supports analysis task statistics*/
    "isSupportTargetsStatistics": true,
/*optional, boolean, whether it supports analysis target statistics*/
    "isSupportAuthAddress":true,
/*optional, boolean, whether it supports authorizing address*/
    "isSupportCertificateStatus": true,
/*optional, boolean, whether it supports getting status of software authorization certificate*/
    "certificateImportProductID": 555,
/*optional, int, product ID of software authorization, corresponding URI: /ISAPI/SDT/Management/certificateImport/
<productID>?format=json and /ISAPI/SDT/Management/fingerprint/<productID>?format=json*/
    "algorithmAuthProductID": 556,
/*optional, int, product ID of algorithm authorization*/
    "isSupportAnalysisResult":true,
/*optional, boolean, whether it supports searching for and counting the number of behavior analysis events: "true"-yes, "false"-no*/
    "isShowCenterComputing":true,
/*optional, boolean, whether to display the configuration interface of algorithm resources: "true"-yes, "false"-no. The configuration interface of algorithm resources will be displayed only when this field is "true" in the single smart mode*/
    "isSupportVehicleRegion": true,
/*optional, boolean, whether it supports vehicle region configuration*/
    "isSupportStorageParam":true,
/*optional, boolean, whether it supports storage parameters configuration*/
    "isSupportFingerprintByName": true,
/*optional, boolean, whether it supports exporting hardware fingerprint*/
    "isSupportAnalysisResultPerimeter": true,
/*optional, boolean, whether it supports perimeter protection (including line crossing detection, intrusion detection, region entrance detection, and region exiting detection); if this node is not returned, it indicates that the events are not classified as perimeter protection and behavior analysis*/
    "isSupportGuardTask":true,
/*optional, boolean, whether it supports camera arming tasks. This node is read-only*/
    "isSupportIntelligentUpload1400": ,
/*optional, boolean, whether it supports adding the service of the image and video library, related URI: /ISAPI/SDT/Management/intelligentUpload1400?format=json*/
    "activeIdType":"",
/*optional, string, value type of activeId: "activeId" (activation ID), "serialNo" (serial No.). The default value is "activeId"*/
    "isSupportTrajectory": ,
/*optional, boolean, whether it supports person pattern function, related URI: /ISAPI/SDT/Management/trajectory/capabilities?format=json*/
    "isSupportMap": ,
/*optional, boolean, whether it supports e-map function, related URI: /ISAPI/SDT/Management/map/capabilities?format=json*/
    "isSupportAlgTrainResourceInfo": ,
```

```
/*optional, boolean, whether it supports getting the source information of the algorithm training platform, related
URI: /ISAPI/SDT/algorithmTraining/ResourceInfo?format=json*/
"isSupportAlgTrainAuthInfo": ,
/*optional, boolean, whether it supports getting the authorization information of the algorithm training platform,
related URI: /ISAPI/SDT/algorithmTraining/SoftLock/AuthInfo?format=json*/
"isSupportAlgTrainNodeList": ,
/*optional, boolean, whether it supports getting the node information of the algorithm training platform, related
URI: /ISAPI/SDT/algorithmTraining/NodeList?format=json*/
"isSupportHumanSelectStrategy": ,
/*optional, boolean, whether it supports configuring the strategy of selecting the human body analysis target*/
"isSupportEventSearchAsync": ,
/*optional, boolean, whether it supports searching for events asynchronously, related URI: /ISAPI/SDT/Management/
EventSearchAsync/capabilities?format=json*/
"isSupportMPVideoStructuredTask": ,
/*optional, boolean, whether it supports creating the video structuralization task of the camera, related URI: /
ISAPI/SDT/Management/Task/Video/monitorPort/capabilities?format=json*/
"isSupportCurrentDataStatistics": ,
/*optional, boolean, whether it supports real-time data statistics, related URI: /ISAPI/SDT/Management/dataStatistics/
current/capabilities?format=json*/
"isSupportTraining": ,
/*optional, boolean, whether it supports submitting the training task, related URI: /ISAPI/SDT/Management/Task/
Training?format=json*/
"isSupportApplySceneFilter": ,
/*optional, boolean, whether it supports filtering application scene parameters for displaying some specific event
types only, related URI: /ISAPI/SDT/Management/applySceneFilter?format=json*/
}
```

See Also

[JSON_ResponseStatus](#)

F.21 JSON_Cap_RemarkList

JSON message about capability of adding remarks for videos by video ID

```
{
  "RemarkList": {
    "@size": 100,
    /*maximum number of remarks can be added for once, the default value is 100*/
    "trackId": ,
    /*whether it supports file No.*/
    "remarks": ,
    /*whether it supports adding remarks for file*/
    "recordName": {
      /*optional, string, file name of video, audio, or picture*/
      "@min":1,
      "@max":100
    },
    "recorderCode": {
      /*optional, string, ID of device that collects data, which should be encrypted*/
      "@min":1,
```

```
    "@max":100
},
"policeCode":{
/*optional, string, ID of police that collects the data, which should be encrypted*/
    "@min":1,
    "@max":100
}
}
}
```

F.22 JSON_Cap_SSD

JSON message about SSD management capability

```
{
"SSD":{
    "isSupportSSDUpgrade":true
/*optional, boolean, whether the device supports SSD firmware upgrade (related URI: /ISAPI/ContentMgmt/storage/ssd/<ID>/upgrade?format=json)*/
}
}
```

F.23 JSON_Cap_TimeSearchCond

Capability message about searching for recording start and end time in JSON format

```
{
"TimeSearchCond": {
    "@size": 1024,
/*the maximum number of search conditions that can be set*/
    "channelID": {
/*required, int, channel No.*/
        "@min": 1,
        "@max": 256
/*value range of the channel No.*/
    },
    "streamType": {
/*optional, string, stream type, "main,sub,adaptive"*/
        "@opt": [
            "main",
            "sub",
            "adaptive"
        ]
    }
},
"TimeSearchResult": {
    "@size": 1024,
/*the maximum number of results that can be returned*/
    "channelID": {

```

```

/*required, int, channel No.*/
    "@min": 1,
    "@max": 256
/*value range of the channel No.*/
},
"streamType": {
/*optional, string, stream type, "main,sub,adaptive"*/
    "@opt": [
        "main",
        "sub",
        "adaptive"
    ]
},
"hasRecord": {
/*required, boolean, whether the video exists, startTime and endTime are valid only when the value of hasRecord is "true"*/
    "@opt": [
        true,
        false
    ]
},
"startTime": {
/*optional, string, recording start time, e.g., 2010-01-01T00:00:00+08:00, this field is required when the value of hasRecord is "true"*/
    "@min": 0,
    "@max": 64
},
"endTime": {
/*optional, string, recording end time, e.g., 2010-01-01T00:00:00+08:00, this field is required when the value of hasRecord is "true"*/
    "@min": 0,
    "@max": 64
}
}
}

```

F.24 JSON_CaputreDescription

JSON message about parameters of capturing picture in URL format

```

{
    "CaputreDescription": {
        "imageType": "JPEG",
        /*required, string, picture type: "JPEG"*/
        "URLType": "cloudURL",
        /*required, string, URL type: "cloudURL"-cloud storage or third party storage, "localURL"-device local storage*/
        /*When the message is transmitted via ISUP protocol, the "URLType" should be "cloudURL"*/
    }
}

```

F.25 JSON_CaputreResult

JSON message about picture capture result

```
{  
    "CaputreResult":{  
        "URLType":"cloudURL",  
        /*required, string, URL type: "cloudURL"-cloud storage or third party storage, "localURL"-device local storage*/  
        "imageURL":""  
        /*required, string, picture URL*/  
    }  
}
```

F.26 JSON_CertificateRevocation

CertificateRevocation message in JSON format

```
{  
    "CertificateRevocation":{  
        "enabled": ,  
        /*required, boolean, whether to enable certificate expiry alarm*/  
        "expireAlarmTime": ,  
        /*required, integer, number of days to prompt in advance before the certificate is expired*/  
        "detecteTime": "",  
        /*required, time, detection time of certificate expiry alarm*/  
        "intervalTime":  
        /*required, integer, alarm interval, unit: day*/  
    }  
}
```

F.27 JSON_CertificateRevocationCap

CertificateRevocationCap capability message in JSON format

```
{  
    "CertificateRevocationCap":{  
        "enabled":{  
            /*required, boolean, whether to enable certificate expiry alarm*/  
            "@opt": [true, false]  
        },  
        "expireAlarmTime":{  
            /*required, integer, number of days to prompt in advance before the certificate is expired*/  
            "@min": 1,  
            "@max": 30,  
            "@def": 7  
        },  
        "detecteTime":{  
    }
```

```

/*required, time, detection time of certificate expiry alarm*/
    "@min":"00:00",
    "@max":"24:00",
    "@def":"10:00"
},
"intervalTime":{

/*required, integer, alarm interval, unit: day*/
    "@min":1,
    "@max":30,
    "@def":1
}
}
}

```

F.28 JSON_ChangedStatus

ChangedStatus message in JSON format

```

{
  ChangedStatus: {
    "seq": ,
    /*32-byte integer type, alarm No., for alarm acknowledge*/
    "devStatus": ,
    /*integer type, value of device status: 0-normal, 1-CPU usage, higher than 85%, 2-hardware error (e.g., serial port exception)*/
    "devCapHash": "";
    /*global Hash value of device capability, which is generated by MD5, and encrypted by Base64; for NVR, the Hash value of network camera should be the calculation source*
     "zeroChanEnable": ,
    /*boolean type, whether the channel-zero is enabled: true-yes, false-no*/
     "cycleRecordEnable": true,
    /*boolean type, whether to enable recording, true-yes, false-no*/
     "chanChanged": [.....]
    /*integer type, channel status changes, e.g., delete, edit, add, the Hash value of this channel changed*/
     "hdChanged": [.....]
    /*integer type, HDD status changes, e.g., delete, edit, add*/
     "ChanStatus": [
      "chanNo": ,
      /*integer type, channel No., read-only/
       "enable": 1,
      /*integer type, it is valid for analog channel only, 0-disable, 1-enable*/
       "online": ,
      /*required, integer type, online status: 0-offline, 1-online*/
       "recordStatus": ,
      /*integer type, recording status: 0-recording, 1-recording exception (HDD exception), 2-recording exception (network camera offline), 3-recording exception (other reason)*/
       "signal": ,
      /*required, integer type, signal status: 0-normal, 1-signal loss*/
       "arming": ,
      /*integer type, network camera arming status: 0-armed, 1-arming failed*/
    ]
  }
}

```

```

}],  

    "HDStatus": [{  

        "hdNo": ,  

        /*required, integer type, HDD No., which starts from 1*/  

        "status": ,  

        /*required, integer type, HDD status: 0-activate, 1-sleep, 2-exception, 3-sleepy HDD error, 4-unformated, 5-  

        disconnected (for network HDD), 6-formatting*/  

    }],  

    "CapChanged": [{  

        "name": "",  

        /*required, root node of changed capability set*/  

        "chan": ,  

        /*channel No., it is required when transmitting channel capability; it is optional when transmitting device capability*/  

        "url": "/ISAPI/Thermal/channels/1/fireDetection/capabilities",  

        /*capability set URL, it is required for capability set based on ISAPI protocol*/  

    }],  

    {  

        "name": "",  

        /*required, capability set name, which is usually the root node*/  

        "chan": 1,  

        /*channel No., it is required when transmitting channel capability; it is optional when transmitting device capability*/  

        "abilityType": 0x700,  

        /*capability type, it is required if the capability is not based on ISAPI protocol*/  

    }]  

}

```

F.29 JSON_Characteristic

JSON message about device attribute code

```
{
    "Characteristic": {
        /*required, attribute code*/
        "code": "+/"
        /*required, string, device attribute code, value range: [1,128]; following characters in the quotes are supported:  

        "ABCDEFIGHJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/*/
    }
}
```

F.30 JSON_ClearingSpaceConfig

JSON message about HDD clearing parameters

```
{
    "ClearingSpaceConfig": {
        /*when the free space of HDD reaches the configured value of insufficientSpace, and if the value of  

        clearingOriginalData is true, the configured HDD space size will be cleared*/
        "insufficientSpace": ,
    }
}
```

```
/*required, int, HDD free space, which is used as the threshold to start clearing*/
    "clearingOriginalData": ,
/*required, boolean, whether to enable clearing the original data*/
    "clearingSpace": 
/*required, int, space size to clear*/
}
}
```

F.31 JSON_CloudUserManage

JSON message about the cloud user information

```
{
  "CloudUserManage": {
    /*required, object, cloud user information*/
    "userName": "test",
    /*required, string, user name*/
    "userNickName": "test",
    /*optional, string, user's nickname, if the user's nickname is configured, the nickname will replace the user name when the message is uploaded via EZVIZ cloud*/
    "userNameSessionAuthInfo": "test",
    /*required, string, session authentication information of the user name, generation method: SHA256(user +salt2+SHA256(user+salt+password)), this node is not required when the value of userOperateType is 2*/
    "phoneNum": "test",
    /*optional, string, phone number, which should be encrypted and can replace the user name for login*/
    "phoneNumSessionAuthInfo": "test",
    /*optional, string, session authentication information of the phone number, generation method: SHA256(user +salt2+SHA256(user+salt+password))*/
    "emailAddress": "test",
    /*optional, string, email address*/
    "emailAddressSessionAuthInfo": "test",
    /*optional, string, session authentication information of the email address, generation method: SHA256(user +salt2+SHA256(user+salt+password))*/
    "duressPassword": "test",
    /*optional, string, duress password, which should be encrypted*/
    "keypadPassword": "test",
    /*optional, string, keypad password, which should be encrypted*/
    "singleKeypadEnable": true,
    /*optional, boolean, whether to enable using the keypad password only once: true (the keypad password can only be used once)*/
    "loginPassword": "test",
    /*optional, string, login password (admin's password) which should be encrypted, this node is required when security exists in the URL*/
    "userOperateType": 1,
    /*optional, int, user operation type: 1 (network user), 2 (keypad user), 3 (network user+keypad user). When the value of this node is 1 or not configured, only password is valid, and for the request method POST, the password is required; when the value of this node is 2, only keypadPassword is valid, and for the request method POST, keypadPassword is required; when the value of this node is 3, both password and keypadPassword are valid, and for the request method POST, password and keypadPassword are required*/
    "password": "test",
  }
}
```

```
/*optional, string, password, which should be encrypted*/
  "userLevel": "test",
/*optional, string, user type*/
  "delRelatedAccountInfoEnabled": true,
/*optional, boolean, whether to delete all the existing installer and operator users in the device when adding the admin user. If either delRelatedOperatorInfoEnabled or delRelatedInstallerInfoEnabled is configured, this node will not be parsed*/
  "delRelatedOperatorInfoEnabled": true,
/*optional, boolean, whether to delete all the existing operator users in the device when adding the admin user*/
  "delRelatedInstallerInfoEnabled": true,
/*optional, boolean, whether to delete all the existing installer users in the device when adding the admin user*/
  "salt": "test",
/*optional, string, salt value 1 downloaded from the cloud*/
  "salt2": "test",
/*optional, string, salt value 2 downloaded from the cloud*/
  "CardList": [
/*optional, array, list of the cards owned by the user*/
    "Card": {
/*optional, object, card information*/
      "id": 1,
/*optional, int, card ID*/
      "name": "test"
    },
/*optional, string, card name*/
    }],
/*optional, array, keyfob list*/
  "remoteCtrlList": [
/*optional, object, keyfob*/
    "remoteCtrl": {
      "id": 1,
/*optional, int, ID*/
      "name": "test"
    },
/*optional, string, name*/
    }],
/*optional, string, EZVIZ user ID which consists of letters and digits, the maximum length is 64 bytes*/
  "userNo": 501,
/*optional, int, displayed No. corresponding to the user ID*/
  "adminType": "Cloud",
/*optional, string, admin type, "Cloud" (user account created in the cloud), "LAN" (user account created on LAN)*/
  "installerType": "Cloud"
/*optional, string, installer type, "Cloud" (user account created in the cloud), "LAN" (user account created on LAN)*/
  }
}
```

F.32 JSON_CloudUserManageCap

JSON message about the capability of managing cloud users

```
{  
    "CloudUserManage":{  
        /*required, object, cloud user information*/  
        "userName":{  
            /*required, string, user name*/  
            "@min":1,  
            "@max":2  
        },  
        "userNickName":{  
            /*optional, string, user's nickname, if the user's nickname is configured, the nickname will replace the user name  
            when the message is uploaded via EZVIZ cloud*/  
            "@min":1,  
            "@max":2  
        },  
        "userNickNameSessionAuthInfo":{  
            /*required, string, session authentication information of the user name: SHA256(user+salt2+SHA256(user+salt  
            +password)), this node is not required when the value of userOperateType is 2*/  
            "@min":1,  
            "@max":2  
        },  
        "phoneNum":{  
            /*optional, string, phone number, which should be encrypted and can replace the user name for login*/  
            "@min":1,  
            "@max":2  
        },  
        "phoneNumSessionAuthInfo":{  
            /*optional, string, session authentication information of the phone number: SHA256(user+salt2+SHA256(user+salt  
            +password))*/  
            "@min":1,  
            "@max":2  
        },  
        "emailAddress":{  
            /*optional, string, email address*/  
            "@min":1,  
            "@max":2  
        },  
        "emailAddressSessionAuthInfo":{  
            /*optional, string, session authentication information of the email address: SHA256(user+salt2+SHA256(user+salt  
            +password))*/  
            "@min":1,  
            "@max":2  
        },  
        "duressPassword":{  
            /*optional, string, duress password, which should be encrypted*/  
            "@min":1,  
            "@max":2  
        },  
        "keypadPassword":{  
            /*optional, string, keypad password, which should be encrypted*/  
            "@min":1,  
            "@max":2  
        },  
    }  
}
```

```
"singleKeypadEnable":{  
/*optional, boolean, whether to enable using the keypad password only once: true (the keypad password can only be  
used once)*/  
    "@opt": [true, false]  
},  
"isNotSupportSingleUserCard":true,  
/*optional, boolean, whether card cannot be linked to users who use the password only once: true (linking card is not  
supported), false or this node is not returned (linking card is supported)*/  
"isNotSupportSingleUserRemoteCtrl":true,  
/*optional, boolean, whether keyfob cannot be linked to users who use the password only once: true (linking keyfob is  
not supported), false or this node is not returned (linking keyfob is supported)*/  
"isNotSupportSingleUserDuressPassword":true,  
/*optional, boolean, whether configuring duress password is not supported for users who use the password only  
once: true (configuring duress password is not supported), false or this node is not returned (configuring duress  
password is supported)*/  
"loginPassword":{  
/*optional, string, login password (admin's password) which should be encrypted, this node is required when security  
exists in the URL*/  
    "@min":1,  
    "@max":2  
},  
"userOperateType":{  
/*optional, int, user operation type: 1 (network user), 2 (keypad user), 3 (network user+keypad user). When the value  
of this node is 1 or not configured, only password is valid, and for the request method POST, the password is required;  
when the value of this node is 2, only keypadPassword is valid, and for the request method POST, keypadPassword is  
required; when the value of this node is 3, both password and keypadPassword are valid, and for the request method  
POST, password and keypadPassword are required*/  
    "@min":1,  
    "@max":2  
},  
"password":{  
/*optional, string, password, which should be encrypted*/  
    "@min":1,  
    "@max":2  
},  
"userLevel":{  
/*optional, string, user type*/  
    "@opt": ["Administrator", "Operator", "Viewer", "installer", "manufacturer", "single", "installerAdmin",  
"installEmployee"]  
},  
"delRelatedAccountInfoEnabled":{  
/*optional, boolean, whether to delete all the existing installer and operator users in the device when adding the  
admin user. If either delRelatedOperatorInfoEnabled or delRelatedInstallerInfoEnabled is configured, this node will  
not be parsed*/  
    "@opt": ["true", "false"]  
},  
"delRelatedOperatorInfoEnabled":{  
/*optional, boolean, whether to delete all the existing operator users in the device when adding the admin user*/  
    "@opt": [true, false]  
},  
"delRelatedInstallerInfoEnabled":{  
/*optional, boolean, whether to delete all the existing installer users in the device when adding the admin user*/  
}
```

```
    "@opt": [true, false]
},
"CardList": {
/*optional, array, list of the cards owned by the user*/
    "Card": {
/*optional, object, card information*/
        "id": {
/*optional, int, card ID*/
            "@min": 1,
            "@max": 2
        },
        "name": {
/*optional, string, card name*/
            "@min": 1,
            "@max": 2
        }
    },
    "@size": 1
},
"remoteCtrlList": {
/*optional, array, keyfob list*/
    "remoteCtrl": {
/*optional, object, keyfob*/
        "id": {
/*optional, int, ID*/
            "@min": 1,
            "@max": 2
        },
        "name": {
/*optional, string, name*/
            "@min": 1,
            "@max": 2
        }
    },
    "@size": 1
},
"salt": {
/*optional, string, salt value 1 downloaded from the cloud*/
    "@min": 1,
    "@max": 2
},
"salt2": {
/*optional, string, salt value 2 downloaded from the cloud*/
    "@min": 1,
    "@max": 2
},
"userID": {
/*optional, string, EZVIZ user ID which consists of letters and digits, the maximum length is 64 bytes*/
    "@min": 1,
    "@max": 64
},
"userNo": {
```

```
/*optional, int, displayed No. corresponding to the user ID*/
  "@min": 501,
/*optional, int, the minimum value*/
  "@max": 10000
/*optional, int, the maximum value*/
},
"adminType": {
/*optional, object, admin type, "Cloud" (user account created in the cloud), "LAN" (user account created on LAN)*/
  "@opt": ["Cloud", "LAN"]
/*optional, array of string*/
},
"installerType": {
/*optional, object, installer type, "Cloud" (user account created in the cloud), "LAN" (user account created on LAN)*/
  "@opt": ["Cloud", "LAN"]
/*optional, array of string*/
}
}
```

F.33 JSON_CloudUserManageList

JSON message about information of multiple cloud users

```
{
  "List": [
/*required, array, list of cloud users*/
    "CloudUserManage": {
/*required, object, cloud user information, see details in JSON_CloudUserManage*/
    }
  ]
}
```

See Also

[JSON_CloudUserManage](#)

F.34 JSON_CustomAudioID

JSON message about custom alarm sound ID

```
{
  "customAudioID":
/*required, string, custom alarm sound ID, the value can be 1 to 3*/
}
```

F.35 JSON_CustomAudioTypeInfoList

JSON message about encoding format of custom audio file

```
{  
    "CustomAudioTypeInfoList": [  
        /*encoding format list of custom audio file*/  
        "CustomAudioTypeInfo": {  
            /*encoding format of custom audio file*/  
            "channelID": 1,  
            /*required, channel ID*/  
            "audioType": ["OPUS", "G726"],  
            /*required, string, audio encoding format type: OPUS, G726; now supports analog channels only*/  
            "bitRate": 64,  
            /*required, int, bit rate, unit: kbps*/  
            "samplingRate": 48,  
            /*optional, int, sampling rate, unit: k*/  
            "samplingBits": 10,  
            /*optional, int, sampling bits, unit: bits*/  
            "AudioChannelNum": 1,  
            /*optional, int, the number of audio channels*/  
            "AudioFileMaxCapacity": 10  
            /*optional, int, the maximum size of audio file, unit: MB*/  
        }  
    ]  
}
```

F.36 JSON_DataReconstruction

JSON message about disk data reconstruction parameters

```
{  
    "DataReconstruction": {  
        "dataType": "record",  
        /*required, string, reconstruction data type: "all", "record"*/  
        "diskID": [1,2],  
        /*optional, array of integers, disk ID for single disk mode*/  
        "RAID": [1,2]  
        /*optional, array of integers, RAID ID for RAID mode*/  
    }  
}
```

F.37 JSON_DiagnosedDataServerCap

JSON message about diagnostic server capability

```
{
    "DiagnosedDataServerCap": {
        /*required, diagnostic server capability*/
        "id": {
            "@min": 1,
            "@max": 1
        },
        "protocol": {
            /*required, string, transmission protocol: "FTP,SFTP"*/
            "@opt": ["FTP", "SFTP"]
        },
        "enabled": {
            /*required, boolean, whether to enable*/
            "@opt": [true, false]
        },
        "addressingFormatType": {
            /*optional, string, address type: "ipaddress" (IP address), "hostname" (host name)*/
            "@opt": ["ipaddress", "hostname"]
        },
        "hostName": {
            /*dependent, string, host name, the maximum length is 64 bytes; it is valid when the value of addressingFormatType is "hostname"*/
            "@min": 1,
            "@max": 64
        },
        "ipVersion": {
            /*dependent, string, IP version: "ipv4", "ipv6"; it is valid when the value of addressingFormatType is "ipaddress"*/
            "@opt": ["ipv4", "ipv6"]
        },
        "portNo": {
            /*optional, int, port No., value range: [1,65535]*/
            "@min": 1,
            "@max": 65535
        },
        "userName": {
            /*required, string, user name, the maximum length is 32 bytes*/
            "@min": 0,
            "@max": 32
        },
        "password": {
            /*string, write-only, password, the maximum length is 16 bytes*/
            "@min": 0,
            "@max": 16
        },
        "isSupportStartTime": true,
        /*required, boolean, whether the device supports start time, i.e., uploads the diagnosis information and running logs of device from the start time; the time is in ISO 8601 format*/
        "isSupportEndTime": true,
        /*required, boolean, whether the device supports end time., i.e., no longer uploads the diagnosis information and running logs of device after end time; the time is in ISO 8601 format*/
        "isSupportTest": true,
        /*required, boolean, whether the device supports server test (related URI: /ISAPI/System/diagnosedData/server/test?*/
    }
}
```

```
format=json)*/  
    "isSupportUpload":true,  
/*required, boolean, whether the device supports uploading diagnosis information (such as running logs, 4G network, MCU, and SOC information)*/  
    "consoleCommand":{  
/*required, string, console command, the length is between 1 to 512 bytes*/  
        "@min":1,  
        "@max":512  
    },  
    "compressionKey":{  
/*optional, string, the decompression password of uploaded diagnosis information in compression file; the length is between 8 to 16 bytes; digits and letters are supported*/  
        "@min":8,  
        "@max":16  
    }  
}
```

F.38 JSON_DiagnosedDataServerList

JSON message about diagnostic server configuration parameters.

```
{  
    "DiagnosedDataServerList":[{  
/*required, diagnostic server parameters*/  
        "id":1,  
/*required, string, index*/  
        "protocol":"FTP",  
/*required, string, transmission protocol: "FTP,SFTP"*/  
        "enabled":true,  
/*required, boolean, whether to enable*/  
        "addressingFormatType":"ipaddress",  
/*optional, string, address type: "ipaddress" (IP address), "hostname" (host name)*/  
        "hostName":"www.baidu.com",  
/*dependent, string, host name, the maximum length is 64 bytes; it is valid when the value of addressingFormatType is "hostname"*/  
        "ipVersion":"ipv4",  
/*dependent, string, IP version: "ipv4", "ipv6"; it is valid when the value of addressingFormatType is "ipaddress"*/  
        "ipV4Address":"10.17.132.254",  
/*dependent, string, IPv4 address; it is valid when the value of ipVersion is "ipv4", and addressingFormatType is "ipaddress"*/  
        "ipv6Address":"fe80::884a:67cb:9b67:b3a0%15",  
/*dependent, string, IPv6 address; it is valid when the value of ipVersion is "ipv6", and addressingFormatType is "ipaddress"*/  
        "portNo":20,  
/*optional, int, port No., value range: [1,65535]*/  
        "userName":"admin",  
/*required, string, user name, the maximum length is 32 bytes; this node should be encrypted*/  
        "password":"admin",  
/*string, write-only, password, the maximum length is 16 bytes; this node should be encrypted*/  
    }]
```

```
"DiagnosedDataUpload":{  
    "consoleCommand":"qwertyuiopasdfghj",  
    /*required, string, write-only, console command, the length is between 1 to 512 bytes; this node should be  
    encrypted*/  
    "compressionKey":"qwertyuiopasdfghj",  
    /*required, string, the decompression password of uploaded diagnosis information in compression file; the length is  
    between 8 to 16 bytes; digits and letters are supported; this node should be encrypted*/  
    "startTime":"2017-06-10T13:30:00+08:00",  
    /*dependent, datetime, start time in ISO 8601 format, e.g., "2017-06-10T13:30:00+08:00"; device uploads the  
    diagnosis information and running logs of device from the start time*/  
    "endTime":"2017-06-10T13:30:00+08:00"  
    /*dependent, datetime, time in ISO 8601 format, e.g., "2017-06-10T13:30:00+08:00"; device no longer uploads the  
    diagnosis information and running logs of device after end time*/  
}  
}  
}
```

F.39 JSON_DoubleVerification

DoubleVerification message in JSON format

```
{  
    "DoubleVerification":{  
        "enable": ""  
        /*optional, whether enables double verification, boolean*/  
    }  
}
```

F.40 JSON_DownloadPackageStatus

JSON message about upgrade package upload status

```
{  
    "DownloadPackageStatus":{  
        /*required, upgrade package download status*/  
        "status": "notDownload,downloading,downloadFailed,pause,finish,incorrectPackage,hdOperationFailed",  
        /*required, string, download status: "notDownload"-not download, "downloading"-downloading, "downloadFailed"-  
        download failed, "pause"-paused, "finish"-completed, "incorrectPackage"-incorrect upgrade package format,  
        "hdOperationFailed"-HD operation failed*/  
        "total": "",  
        /*optional, float, upgrade package total size, unit: MB, corrects to one decimal place*/  
        "remain": "",  
        /*optional, float, remaining space, unit: MB, corrects to one decimal place*/  
        "speed": "",  
        /*optional, float, download speed, unit: KB/s, corrects to two decimal places*/  
        "remainTime": "",  
        /*optional, int, estimated remaining time, unit: s*/  
        "progress": 0,  
        /*required, int, progress, ranges from 0 to 100*/  
    }  
}
```

```
}
```

F.41 JSON_EmailCertification

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

Enter the syntax information of your reference here (optional).

Enter the actual information in this section (optional).

Example

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

F.42 JSON_EncryptFormat

EncryptFormat message in JSON format

```
{
  "EncryptFormat":{
    "password":"",
    /*required, string, password, it will be encrypted according to the encryption vector*/
    "encryptFormatType":""
    /*optional, string, HDD formatting type: "FAT32", "EXT4"*/
  }
}
```

F.43 JSON_EncryptVerify

EncryptVerify message in JSON format

```
{
  "EncryptVerify":{
    "password":"",
    /*required, string, password, it will be encrypted according to the encryption vector*/
  }
}
```

F.44 JSON_EPTZMode

EPTZMode message in JSON format

```
{
  "EPTZMode":{
    "mode":"",
    /*required, modes supported by e-PTZ: "cruise"-patrol (PTZ supports preset, patrol, zoom, and eight directions),
```

```
"autoTrack"-automatic tracking*/
"AutoTrack":{
/*dependency, auto-tracking mode configuration*/
"sensitivity": ,
/*optional, integer, sensitivity*/
"detectionTarget":["human", "vehicle"],
/*detection target*/
"ReginonList":[{
"id": ,
/*required, rule ID*/
"Region":>[
/*required, rule area, the number of edges is between 3 and 10*/
"x": ,
/*required, float, X-coordinate, range: [0.000,1]*/
"y": ,
/*required, float, Y-coordinate, range: [0.000,1]*/
}]
}],
"FilterSize":{
/*optional, size filter of rectangle*/
"MaxTargetRect":{
"height": ,
/*required, float, height, range: [0.000,1.000], the value is accurate to three decimal places*/
"width": ,
/*required, float, width, range: [0.000,1.000], the value is accurate to three decimal places*/
"x": ,
/*required, float, X-coordinate, range: [0.000,1.000], the value is accurate to three decimal places. The upper-left corner is the origin*/
"y": ,
/*required, float, Y-coordinate, range: [0.000,1.000], the value is accurate to three decimal places. The upper-left corner is the origin*/
},
"MinTargetRect":{
"height": ,
/*required, float, height, range: [0.000,1.000], the value is accurate to three decimal places*/
"width": ,
/*required, float, width, range: [0.000,1.000], the value is accurate to three decimal places*/
"x": ,
/*required, float, X-coordinate, range: [0.000,1.000], the value is accurate to three decimal places. The upper-left corner is the origin*/
"y": ,
/*required, float, Y-coordinate, range: [0.000,1.000], the value is accurate to three decimal places. The upper-left corner is the origin*/
}
}
}
```

F.45 JSON_EPTZModeCap

EPTZModeCap capability message in JSON format

```
{  
    "EPTZModeCap":{  
        "mode":{  
            /*required, modes supported by e-PTZ: "cruise"-patrol (PTZ supports preset, patrol, zoom, and eight directions),  
            "autoTrack"-automatic tracking*/  
            "@opt":["cruise", "autoTrack"]  
        },  
        "AutoTrackCap":{  
            /*optional, this node is returned when mode contains "autoTrack"*/  
            "sensitivity":{  
                /*optional, integer, sensitivity*/  
                "@min":0,  
                "@max":100  
            },  
            "detectionTarget":{  
                /*detection target*/  
                "@opt":["human", "vehicle", "all"]  
            },  
            "ReginonList":{  
                "id":{  
                    /*required, rule ID*/  
                    "@min":1,  
                    "@max":4  
                },  
                "Region":{  
                    /*required, rule area, the number of edges is between 3 and 10*/  
                    "number":{  
                        /*required, supported number of edges*/  
                        "@min":3,  
                        "@max":10  
                    },  
                    "x":{  
                        /*required, float, X-coordinate, range: [0.000,1]*/  
                        "@min":0,  
                        "@max":1  
                    },  
                    "y":{  
                        /*required, float, Y-coordinate, range: [0.000,1]*/  
                        "@min":0,  
                        "@max":1  
                    }  
                },  
                "FilterSize":{  
                    /*optional, size filter of rectangle*/  
                    "MaxTargetRect":{  
                        "height":{  
                            /*optional, height of rectangle*/  
                            "@min":0,  
                            "@max":100  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

```
/*required, float, height, range: [0.000,1.000], the value is accurate to three decimal places*/
    "@min":0,
    "@max":1
},
"width":{
/*required, float, width, range: [0.000,1.000], the value is accurate to three decimal places*/
    "@min":0,
    "@max":1
},
"x":{
/*required, float, X-coordinate, range: [0.000,1.000], the value is accurate to three decimal places. The upper-left
corner is the origin*/
    "@min":0,
    "@max":1
},
"y":{
/*required, float, Y-coordinate, range: [0.000,1.000], the value is accurate to three decimal places. The upper-left
corner is the origin*/
    "@min":0,
    "@max":1
}
},
"MinTargetRect":{
    "height":{

/*required, float, height, range: [0.000,1.000], the value is accurate to three decimal places*/
    "@min":0,
    "@max":1
},
"width":{

/*required, float, width, range: [0.000,1.000], the value is accurate to three decimal places*/
    "@min":0,
    "@max":1
},
"x":{

/*required, float, X-coordinate, range: [0.000,1.000], the value is accurate to three decimal places. The upper-left
corner is the origin*/
    "@min":0,
    "@max":1
},
"y":{

/*required, float, Y-coordinate, range: [0.000,1.000], the value is accurate to three decimal places. The upper-left
corner is the origin*/
    "@min":0,
    "@max":1
}
}
}
}
```

F.46 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.47 JSON_EventNotificationAlert_CertificateExpiryAlarmMsg

The certificate expiry alarm details are uploaded in JSON format of EventNotificationAlert message, here shows an example.

```
Content-Type: multipart/form-data; boundary=MIME_boundary  

--MIME_boundary  

Content-Type: application/json  

Content-Length:  

{
```

```
"ipAddress":172.6.64.7,  
"ipv6Address":"",
"portNo":80,  
"protocol":"HTTP",
"macAddress":"01:17:24:45:D9:F4",
"channelID":1,  
"dateTime":"2018-11-29T15:32:55+08:00",
"activePostCount":1,  
"eventType":"certificateRevocation",
"eventState":"active",
"eventDescription":"Certificate Revocation Detection",
/*refer to the message JSON_EventNotificationAlert_Alarm/EventInfo for description details of the above nodes*/
"channelName":"ABC",
/*required, channel name (camera name)*/
"deviceID":"test0123",
/*optional, device ID, it is also the PUID and it must be returned when passing through ISAPI event by ISUP*/
"Result":[{
    "customID":"",
/*required, string, custom certificate ID*/
    "certificateType":"CA",
/*required, string, certificate type: "CA"-CA certificate, "client/server"-client or server certificate*/
    "type":["securityLog"],
/*optional, function type: "HTTPS", "WebSockets", "SDK_OVER_TLS", "SRTP", "securityLog", "ieee802.1x"*/
    "status":"expiring",
/*optional, certificate status: "expiring"-expiring, "expired"-expired, "exceptional"-exceptional*/
    "startDate":"2010-04-17",
/*required, date, start date of the certificate expiry date, it is accurate to day*/
    "endDate":"2010-04-17"
/*required, date, end date of the certificate expiry date, it is accurate to day*/
}]
}
--MIME_boundary--
```

See Also

[JSON_EventNotificationAlert_Alarm/EventInfo](#)

F.48 JSON_EventNotificationAlert_HDDBadSectorEventMsg

The event details of the HDD bad sector detection are uploaded in JSON format of EventNotificationAlert message, here shows an example.

```
{
    "ipAddress":"",
/*required, string type, IPv4 address of the alarm device, the maximum length is 32 bytes*/
    "ipv6Address":"",
/*optional, string type, IPv6 address of the alarm device, the maximum length is 128 bytes*/
    "portNo": ,
/*optional, integer32 type, port No. of the alarm device*/
    "protocolType":"",
/*optional, string type, protocol type: "HTTP", "HTTPS", the maximum length is 32 bytes*/
```

```
"macAddress":"",
/*optional, string type, MAC address, the maximum length is 32 bytes*/
"channelID": ,
/*optional, integer32 type, device channel No. that triggers alarms*/
"dateTime":"",
/*required, string type, alarm triggering time in ISO8601 time format, e.g., 2018-03-13T19:42:27+08:00, the maximum length is 32 bytes*/
"activePostCount": ,
/*required, integer32 type, times that the same alarm has been uploaded*/
"eventType":"",
/*required, triggered event type: "hdBadBlock"-HDD bad sector detection event*/
"eventState":"",
/*required, string type, event triggering status: "active"-triggered, "inactive"-not triggered (heartbeat data), the maximum length is 32 bytes*/
"eventDescription":"",
/*required, string type, event description: "HDD bad sector detection event", the maximum length is 128 bytes*/
"deviceID":"",
/*optional, device ID, which is the PUID and should be returned for ISUP alarms, e.g., "test0123"*/
"HDBadBlock":{
  "diskNo": ,
  /*required, integer32 type, HDD No.*/
}
}
```

F.49 JSON_EventNotificationAlert_HDDHighTemperatureEventMsg

The event details of the HDD high temperature detection are uploaded in JSON format of EventNotificationAlert message, here shows an example.

```
{
  "ipAddress":"",
/*required, string type, IPv4 address of the alarm device, the maximum length is 32 bytes*/
  "ipv6Address":"",
/*optional, string type, IPv6 address of the alarm device, the maximum length is 128 bytes*/
  "portNo": ,
/*optional, integer32 type, port No. of the alarm device*/
  "protocolType":"",
/*optional, string type, protocol type: "HTTP", "HTTPS", the maximum length is 32 bytes*/
  "macAddress":"",
/*optional, string type, MAC address, the maximum length is 32 bytes*/
  "channelID": ,
/*optional, integer32 type, device channel No. that triggers alarms*/
  "dateTime":"",
/*required, string type, alarm triggering time in ISO8601 time format, e.g., 2018-03-13T19:42:27+08:00, the maximum length is 32 bytes*/
  "activePostCount": ,
/*required, integer32 type, times that the same alarm has been uploaded*/
  "eventType":"",
/*required, triggered event type: "highHDTemperature"-HDD high temperature detection event*/
  "eventState": ,
```

```
/*required, string type, event triggering status: "active"-triggered, "inactive"-not triggered (heartbeat data), the maximum length is 32 bytes*/
"eventDescription":"",
/*required, string type, event description: "HDD high temperature detection event", the maximum length is 128 bytes*/
"deviceID":"",
/*optional, device ID, which is the PUID and should be returned for ISUP alarms, e.g., "test0123"*/
"HighHDTemperature":{
  "diskNo":
/*required, integer32 type, HDD No.*/
}
}
```

F.50 JSON_EventNotificationAlert_HDDImpactEventMsg

The event details of the HDD impact detection are uploaded in JSON format of EventNotificationAlert message, here shows an example.

```
{
  "ipAddress":"",
/*required, string type, IPv4 address of the alarm device, the maximum length is 32 bytes*/
  "ipv6Address":"",
/*optional, string type, IPv6 address of the alarm device, the maximum length is 128 bytes*/
  "portNo": ,
/*optional, integer32 type, port No. of the alarm device*/
  "protocolType":"",
/*optional, string type, protocol type: "HTTP", "HTTPS", the maximum length is 32 bytes*/
  "macAddress":"",
/*optional, string type, MAC address, the maximum length is 32 bytes*/
  "channelID": ,
/*optional, integer32 type, device channel No. that triggers alarms*/
  "dateTime":"",
/*required, string type, alarm triggering time in ISO8601 time format, e.g., 2018-03-13T19:42:27+08:00, the maximum length is 32 bytes*/
  "activePostCount": ,
/*required, integer32 type, times that the same alarm has been uploaded*/
  "eventType":"",
/*required, triggered event type: "hdImpact"-HDD impact detection event*/
  "eventState":"",
/*required, string type, event triggering status: "active"-triggered, "inactive"-not triggered (heartbeat data), the maximum length is 32 bytes*/
  "eventDescription":"",
/*required, string type, event description: "HDD impact detection event", the maximum length is 128 bytes*/
  "deviceID":"",
/*optional, device ID, which is the PUID and should be returned for ISUP alarms, e.g., "test0123"*/
  "HDImpact":{
    "diskNo":
/*required, integer32 type, HDD No.*/
  }
}
```

F.51 JSON_EventNotificationAlert_HDDLowTemperatureEventMsg

The event details of the HDD low temperature detection are uploaded in JSON format of EventNotificationAlert message, here shows an example.

```
{
  "ipAddress":"",
  /*required, string type, IPv4 address of the alarm device, the maximum length is 32 bytes*/
  "ipv6Address":"",
  /*optional, string type, IPv6 address of the alarm device, the maximum length is 128 bytes*/
  "portNo": ,
  /*optional, integer32 type, port No. of the alarm device*/
  "protocolType":"",
  /*optional, string type, protocol type: "HTTP", "HTTPS", the maximum length is 32 bytes*/
  "macAddress":"",
  /*optional, string type, MAC address, the maximum length is 32 bytes*/
  "channelID": ,
  /*optional, integer32 type, device channel No. that triggers alarms*/
  "dateTime":"",
  /*required, string type, alarm triggering time in ISO8601 time format, e.g., 2018-03-13T19:42:27+08:00, the maximum length is 32 bytes*/
  "activePostCount": ,
  /*required, integer32 type, times that the same alarm has been uploaded*/
  "eventType":"",
  /*required, triggered event type: "lowHDTemperature"-HDD low temperature detection event*/
  "eventState":"",
  /*required, string type, event triggering status: "active"-triggered, "inactive"-not triggered (heartbeat data), the maximum length is 32 bytes*/
  "eventDescription":"",
  /*required, string type, event description: "HDD low temperature detection event", the maximum length is 128 bytes*/
  "deviceID":"",
  /*optional, device ID, which is the PUID and should be returned for ISUP alarms, e.g., "test0123"*/
  "LowHDTemperature":{
    "diskNo":
    /*required, integer32 type, HDD No.*/
  }
}
```

F.52 JSON_EventNotificationAlert_HDDSevereFaultEventMsg

The event details of the HDD severe fault detection are uploaded in JSON format of EventNotificationAlert message, here shows an example.

```
{
  "ipAddress":"",
  /*required, string type, IPv4 address of the alarm device, the maximum length is 32 bytes*/
  "ipv6Address":"",
  /*optional, string type, IPv6 address of the alarm device, the maximum length is 128 bytes*/
  "portNo": ,
  /*optional, integer32 type, port No. of the alarm device*/
  "protocolType":"",
  /*optional, string type, protocol type: "HTTP", "HTTPS", the maximum length is 32 bytes*/
  "macAddress":"",
  /*optional, string type, MAC address, the maximum length is 32 bytes*/
  "channelID": ,
  /*optional, integer32 type, device channel No. that triggers alarms*/
  "dateTime":"",
  /*required, string type, alarm triggering time in ISO8601 time format, e.g., 2018-03-13T19:42:27+08:00, the maximum length is 32 bytes*/
  "activePostCount": ,
  /*required, integer32 type, times that the same alarm has been uploaded*/
  "eventType":"",
  /*required, triggered event type: "lowHDTemperature"-HDD low temperature detection event*/
  "eventState":"",
  /*required, string type, event triggering status: "active"-triggered, "inactive"-not triggered (heartbeat data), the maximum length is 32 bytes*/
  "eventDescription":"",
  /*required, string type, event description: "HDD low temperature detection event", the maximum length is 128 bytes*/
  "deviceID":"",
  /*optional, device ID, which is the PUID and should be returned for ISUP alarms, e.g., "test0123"*/
  "LowHDTemperature":{}
```

```
/*optional, string type, IPv6 address of the alarm device, the maximum length is 128 bytes*/
"portNo": ,
/*optional, integer32 type, port No. of the alarm device*/
"protocolType":"",
/*optional, string type, protocol type: "HTTP", "HTTPS", the maximum length is 32 bytes*/
"macAddress":"",
/*optional, string type, MAC address, the maximum length is 32 bytes*/
"channelID": ,
/*optional, integer32 type, device channel No. that triggers alarms*/
"dateTime":"",
/*required, string type, alarm triggering time in ISO8601 time format, e.g., 2018-03-13T19:42:27+08:00, the maximum length is 32 bytes*/
"activePostCount": ,
/*required, integer32 type, times that the same alarm has been uploaded*/
"eventType":"",
/*required, triggered event type: "severeHDFailure"-HDD severe fault detection event*/
"eventState":"",
/*required, string type, event triggering status: "active"-triggered, "inactive"-not triggered (heartbeat data), the maximum length is 32 bytes*/
"eventDescription":"",
/*required, string type, event description: "HDD severe fault detection event", the maximum length is 128 bytes*/
"deviceID":"",
/*optional, device ID, which is the PUID and should be returned for ISUP alarms, e.g., "test0123"*/
"SevereHDFailure":{
  "diskNo":
/*required, integer32 type, HDD No.*/
}
}
```

F.53 JSON_EventNotificationAlert_NetworkTrafficPrealarmMsg

JSON message about details of network traffic pre-alarm.

```
{
  "ipAddress": "",
/*required, string, IPv4 address of security control device, the maximum size is 32 bytes*/
  "ipv6Address": "",
/*optional, string, IPv6 address of security control device, the maximum size is 128 bytes*/
  "portNo": ,
/*optional, integer32, port No. of security control device*/
  "protocolType": "",
/*optional, string, protocol types: "HTTP,HTTPS,Ehome", the maximum size is 32 bytes*/
  "macAddress": "",
/*optional, string, MAC address, the maximum size is 32 bytes*/
  "channelID": ,
/*optional, integer32, device channel No. of the device that triggers alarm*/
  "dateTime": "",
/*required, string, time of alarm trigger with the format of ISO8601, the maximum size is 32 bytes*/
  "activePostCount": ,
/*required, integer32, number of times that an alarm has been uploaded*/
```

```
"eventType": "",  
/*required, string, event type, here it should be set to "dataPrealarm"; the maximum size is 128 bytes*/  
"eventState": "",  
/*required, event status: "active"-triggered alarm, "inactive"-not triggered alarm, the maximum size is 32 bytes*/  
"eventDescription": "",  
/*required, string, event description*/  
"deviceID": "",  
/*optional, PUID, e.g., test0123*/  
"DataPrealarm":{  
/*req, traffic pre-alarm event*/  
"interfaceType": "",  
/*required, string, network card type: "4g,wiredAndWireless"*/  
"interfaceID":,  
/*required, integer, network card ID*/  
"packageType": "",  
/*required, string, data plan type: "day"-daily plan, "month"-monthly plan, "year"-annual plan*/  
"trafficValue": ,  
/*required, float, data value in the plan, which should be accurate to three decimal places, unit: MB*/  
"usedTrafficValue":  
/*required, float, data usage, which should be accurate to three decimal places, unit: MB*/  
}  
}
```

F.54 JSON_EventNotificationAlert_vibrationDetection

Message about alarm details of vibration detection in JSON format

```
{  
"ipAddress": "",  
/*required, string, IPv4 address of alarm device, the maximum size is 32 bytes*/  
"ipv6Address": "",  
/*optional, string, IPv6 address of alarm device, the maximum size is 128 bytes*/  
"portNo": ,  
/*optional, integer32, port No. of alarm device*/  
"protocol": "",  
/*optional, string, protocol type: "HTTP, HTTPS, EHome", it should be the same as the protocolType in /ISAPI/Event/notification/httpHosts/<ID>, and the maximum size is 32 bytes. This node should be set to "HTTP" when the JSON message is transmitted by SDK; it should be set to "EHome" when the JSON message is transmitted by ISUP; it should be set to "HTTPS" when the transmission protocol is HTTPS*/  
"macAddress": "",  
/*optional, string, MAC address, the maximum size is 32 bytes*/  
"channelID": ,  
/*optional, integer32, channel ID of the device that triggers alarm*/  
"dateTime": "",  
/*required, string, alarm triggering time in ISO8601 format, the maximum size is 32 bytes*/  
"activePostCount": ,  
/*required, integer32, times of the same alarm being uploaded*/  
"eventType": "",  
/*required, string, event type that triggers alarm, here it should be set to "vibrationDetection" (vibration detection), the maximum size is 128 bytes*/
```

```
"eventState": "",  
/*required, string, event triggering status: "active"-triggered, "inactive"-not triggered (heartbeat data), the maximum size is 32 bytes*/  
"eventDescription": "",  
/*required, string, event description (e.g., vibration detection), the maximum size is 128 bytes*/  
"channelName": " ",  
/*optional, string, channel name (camera name), which should be the same as the channelName in /ISAPI/Streaming/channels/<ID>*/  
"deviceID": " ",  
/*optional, device ID which is also the PUID, which should be returned in ISUP alarm and should be the same as the deviceID in /ISAPI/System/Network/Ehome*/  
"VibrationDetection": {  
    "sensitivity":  
        /*optional, int, sensitivity, value range: [0-100]*/  
    }  
}
```

F.55 JSON_EventSearchCap

EventSearchCap message in JSON format

```
{  
    "startTime": "2004-05-03T17:30:08Z",  
    /*required, start time, ISO8601_time, string*/  
    "endTime": "2004-05-03T17:30:08Z",  
    /*required, end time, ISO8601_time, string*/  
    "resultMaxNum": 100,  
    /*required, supported maximum number of searching, int*/  
    "channelIDLen": {  
        /*optional, channel ID length*/  
        "@min": ,  
        "@max":  
    },  
    "eventType": {  
        /*event type: studentStoodUp-student stands up, accessController-access controller, videoIntercom-video intercom, OPTEX-OPTEX security control panel, Luminite-Luminite security control panel, GJD-GJD security control panel, cameraDetector-detector, securityControlPanel-security control panel, multiple selections are separated by comma, string*/  
        "@opt": "studentStoodUp, accessController, videoIntercom, OPTEX, Luminite, GJD, cameraDetector, securityControlPanel",  
        "#text": "studentStoodUp"  
    },  
    "type": {  
        /*optional, channel type: video-video channel, IOT-IOT channel, if this node does not exist, it indicates video channel, string*/  
        "@opt": "video,IOT",  
        "#text": "video"  
    },  
    "AccessController": {  
        /*optional*/  
    }  
}
```

```
"eventType":{  
/*optional, event type, if this node does not exist, it indicates searching all supported events, string*/  
    "@opt": "",  
    "#text": ""  
},  
"nameLen":{  
/*optional, name length*/  
    "@min": ,  
    "@max":  
},  
"cardNoLen":{  
/*optional, card No. length*/  
    "@min": ,  
    "@max":  
},  
"VideoIntercom":{  
/*optional*/  
    "eventType":{  
/*optional, optional, event type, if this node does not exist, it indicates searching all supported events, string*/  
        "@opt": "",  
        "#text": ""  
},  
    "nameLen":{  
/*optional, name length*/  
        "@min": ,  
        "@max":  
},  
    "cardNoLen":{  
/*optional, card No. length*/  
        "@min": ,  
        "@max":  
},  
    "OPTEX":{  
/*optional*/  
        "eventType":{  
/*optional, optional, event type, if this node does not exist, it indicates searching all supported events, string*/  
            "@opt": "",  
            "#text": ""  
},  
        "Luminite":{  
/*optional*/  
            "eventType":{  
/*optional, optional, event type, if this node does not exist, it indicates searching all supported events, string*/  
                "@opt": "",  
                "#text": ""  
},  
        },  
        "GJD":{  
/*optional*/  
    
```

```

"eventType":{
/*optional, optional, event type, if this node does not exist, it indicates searching all supported events, string*/
    "@opt": "",
    "#text": ""
},
},
"CameraDetector":{
/*optional*/
    "eventType":{
/*optional, optional, event type, if this node does not exist, it indicates searching all supported events, string*/
        "@opt": "",
        "#text": ""
    },
},
"SecurityControlPanel":{
/*optional, security control panel event*/
    "eventType":{
/*optional, optional, event type, if this node does not exist, it indicates searching all supported events, string*/
        "@opt": ""
    },
    "zoneNo":{
/*required, zone No., integer*/
        "size": 1,
/*required, the maximum number of items, integer*/
        "@min": 1,
        "@max": 16,
        "#text": 1
    },
}
}

```

F.56 JSON_EventSearchCond

EventSearchCond message in JSON format

```

{
    "searchID": "",
/*required, string, unique record search ID.*/
    "searchResultPosition": "",
/*required, initial position of search result in the list, integer, 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*/
    "maxResults": "",
/*required, the max. number of results for current search, int*/
    "startTime": "",
/*required, start time, ISO8601_time, string*/
    "endTime": "",
/*required, end time, ISO8601_time, string*/
    "choiceChannel": [
/*optional*/
        "channelID": ""
    ]
}

```

```
/*optional, channel ID,string*/
},
"eventType": "",  
/*required, event type: studentStoodUp-student stands up, accessController-access controller, videoIntercom-video  
intercom, OPTEX-OPTEX security control panel, Luminite-Luminite security control panel, GJD-GJD security control  
panel, cameraDetector-detector, securityControlPanel-security control panel, multiple selections are separated by  
comma, string*/
"type": "video,IOT",
/*optional, channel type: video-video channel, IOT-IOT channel, if this node does not exist, it indicates video channel,  
string*/
"AccessController":{
"eventType": "",  
/*optional, IoT event type, if this node does not exist, it indicates searching all supported events, string*/
"name": "",  
/*optional, name*/
"cardNo": "",  
/*optional, card No.*/
},
"VideoIntercom":{
"eventType": "",  
/*optional, IoT event type, if this node does not exist, it indicates searching all supported events, string*/
"name": "",  
/*optional, name*/
"cardNo": "",  
/*optional, card No.*/
},
"OPTEX":{
"eventType": "",  
/*optional, IoT event type, if this node does not exist, it indicates searching all supported events, string*/
},
"Luminite":{
"eventType": "",  
/*optional, IoT event type, if this node does not exist, it indicates searching all supported events, string*/
},
"GJD":{
"eventType": "",  
/*optional, IoT event type, if this node does not exist, it indicates searching all supported events, string*/
},
"CameraDetector":{
"eventType": "",  
/*optional, IoT event type, if this node does not exist, it indicates searching all supported events, string*/
},
"SecurityControlPanel":{
"eventType": "",  
/*optional, IoT event type, if this node does not exist, it indicates searching all supported events, string*/
"zoneNo": [1,2]
/*optional, zone No., it is valid when eventType values "zone", array, the sub type is integer*/
}
```

F.57 JSON_EventSearchResult

EventSearchResult message in JSON format

```
{  
    "responseStatusStrg": "",  
    /*optional, searching status: OK- Searching ended, NO MATCHES-No matched data found, MORE-Search again for  
    more results, string, the max. length is 32, {dep if errcode == 1 && errMsg == ok}*/  
    "numOfMatches": "",  
    /*optional, returned number of results for current search, integer32, {dep if errcode == 1 && errMsg == ok}*/  
    "totalMatches": "",  
    /*optional, total number of matched results, integer32, {dep if errcode == 1 && errMsg == ok}*/  
    "eventInfo": [  
        {"captureTime": "",  
        /*required, capture time, ISO8601_time, string*/  
        "channelID": "",  
        /*optional, camera ID, string*/  
        "channelName": "",  
        /*optional, camera name*/  
        "picUrl": "",  
        /*optional, picture URL, string*/  
        "subPicUrl": "",  
        /*optional, thumbnail URL, string */  
        "eventType": "",  
        /*required, event type: studentStoodU-student stands up, videoIntercom-video intercom, string*/  
        "studentStoodUp": {  
            /*optional, student stand up event information*/  
            "studentsStoodUp_number": ""  
            /* optional, xs:integer, the number of students, who are standing up*/  
        },  
        "AccessController": {  
            "eventType": "",  
            /*required, event type, string*/  
            "name": "",  
            /*optional, name*/  
            "cardNo": "",  
            /*optional, card No.*/  
            "cardType": "",  
            /*optional, card type: invalid-invalid, ordinary-normal card, disabled-card for the disabled,blocklist-card in blocklist,  
            patrol-patrol card, super-super card, guest-visitor card, remove-removed card*/  
            "videoChannellInfo": [  
                /*optional, video channel information*/  
                "videoChannel": "",  
                /*required, video channel No., integer*/  
                "picUrl": "",  
                /*optional, picture URL, string*/  
            ]  
        },  
        "VideoIntercom": {  
            "eventType": "",  
            /*required, event type, string*/  
        }  
    ]  
}
```

```
    "name": "",  
/*optional, name*/  
    "cardNo": "",  
/*optional, card No.*/  
    "cardType": "invalid,ordinary,disabled,blocklist,patrol,super,guest,remove",  
/*optional, card type: invalid-invalid, ordinary-normal card, disabled-card for the disabled,blocklist-card in blocklist,  
patrol-patrol card, super-super card, guest-visitor card, remove-removed card*/  
    "videoChannellInfo": [{  
/*optional, video channel information*/  
        "videoChannel": "",  
/*required, video channel No., integer*/  
        "picUrl": "",  
/*optional, picture URL, string*/  
    }]  
},  
"OPTEX":{  
    "eventType": "",  
/*required, event type, string*/  
    "videoChannellInfo": [{  
/*optional, video channel information*/  
        "videoChannel": "",  
/*required, video channel No., integer*/  
        "picUrl": "",  
/*optional, picture URL, string*/  
    }]  
},  
"Luminite":{  
    "eventType": "",  
/*required, event type, string*/  
    "videoChannellInfo": [{  
/*optional, video channel information*/  
        "videoChannel": "",  
/*required, video channel No., integer*/  
        "picUrl": "",  
/*optional, picture URL, string*/  
    }]  
},  
"GJD":{  
    "eventType": "",  
/*required, event type, string*/  
    "videoChannellInfo": [{  
/*optional, video channel information*/  
        "videoChannel": "",  
/*required, video channel No., integer*/  
        "picUrl": "",  
/*optional, picture URL, string*/  
    }]  
},  
"CameraDetector":{  
    "eventType": "",  
/*required, event type, string*/  
    "videoChannellInfo": [{
```

```
/*optional, video channel information*/
    "videoChannel": "",
/*required, video channel No., integer*/
    "picUrl": "",
/*optional, picture URL, string*/
}
},
"SecurityControlPanel":{
    "eventType": "",
/*required, event type, string*/
    "videoChannellInfo": [
/*optional, video channel information*/
        "videoChannel": "",
/*required, video channel No., integer*/
        "picUrl": "",
/*optional, picture URL, string*/
    ]
    "zoneNo": "",
/*optional, zone No., it is valid when eventType values "zone", integer*/
    "code":1103,
/*optional, CID event No., it is valid when eventType values "zone", integer*/
    "zoneName": ""
/*optional, zone name, string*/
}
}
}
```

F.58 JSON_EZVIZSecretKey

EZVIZSecretKey message in JSON format

```
{
    "EZVIZSecretKey":{
/*required, verificaiton code for Guarding Vision, string, sensitive information should be encrypted, the string length ranges from 0 to 64. E.g., 34020000002000000001*/
        "secretKey": ""
    }
}
```

F.59 JSON_EZVIZQRCode

JSON message about Guarding Vision QR code

```
{
    "EZVIZQRCode":{
        "QRCode": ""
/*required, string, Guarding Vision QR code string, the maximum length is 1024 bytes*/
}
```

```
}
```

F.60 JSON_FocusStatus

JSON message about focus status

```
{
  "focusStatus":"success",
  /*required, string, focus status: "success", "failed", "working" (focusing)*/
  "motorMovementType":"manual"
  /*optional, string, motor movement type: "manual", "auto" (automatic), "armingLocation" (arming location)*/
}
```

F.61 JSON_GetCustomAudio

JSON message about the custom alarm audio file

```
{
  "CustomAudioInfo": {
    "CustomAudioList": [
      /*required, the file list of custom alarm sound*/
      {
        "customAudioID": ,
        /*required, int, custom alarm sound ID, the value can be 1 to 3*/
        "customAudioName": "",
        /*required, string, name of the custom alarm sound, the maximum size is 24 bytes*/
        "AudioType": "AI",
        /*optional, string, audio file type: AI-AI audio file*/
        "fileSize": 1014,
        /*optional, int, file size, unit: KB*/
        "isDefaultFile": true
        /*optional, boolean, whether it is a default audio file of the device*/
      }
    ]
  }
}
```

F.62 JSON_GetUserInfoByType

JSON message about conditions of getting cloud user information

```
{
  "GetUserInfoByType": {
    /*required, object, search condition*/
    "mode": "email",
    /*optional, enum, condition type: "email", "phoneNo", "userName"*/
    "UserNameMode": {

```

```
/*optional, object, user name mode*/
    "userName": "test"
/*optional, string, user name*/
},
"EmailMode": {
/*optional, object, email mode*/
    "Email": "test",
/*optional, string, email address*/
    "userType": "test"
/*optional, string, user type*/
},
"PhoneNoMode": {
/*optional, object, phone number mode*/
    "phoneNo": "test"
/*optional, string, phone number*/
}
}
```

F.63 JSON_GPSCalibration

Message about configuration parameters of a single GPS calibration point in JSON format

```
{
    "GPSCalibration": {
        "id": ,
/*optional, int, calibration point ID, which is used for parsing in upper layer and cannot be configured*/
        "calibrated": ,
/*optional, boolean, calibration status*/
        "longitude": ,
/*required, float, longitude, accurate to 6 decimal places*/
        "latitude": ,
/*required, float, latitude, accurate to 6 decimal places*/
        "x": ,
/*required, float, x-coordinate, accurate to 6 decimal places*/
        "y": ,
/*required, float, y-coordinate, accurate to 6 decimal places*/
    }
}
```

F.64 JSON_GPSCalibrationCap

Message about GPS calibration capability in JSON format

```
{
    "GPSCalibrationCap": {
/*required, GPS calibration configuration capability*/
        "enabled": "true,false",
/*required, boolean, whether the channel supports GPS calibration configuration*/
    }
}
```

```

    "id": {
        /*required, int, calibration No.*/
        "@min":1,
        "@max":64,
        "@text":1
    },
    "Longitude":{
        /*required, float, longitude, accurate to 6 decimal places*/
        "@min":-180.000000,
        "@max":180.000000,
        "@text":56.000000
    },
    "Latitude":{
        /*required, float, latitude, accurate to 6 decimal places*/
        "@min":-90.000000,
        "@max":90.000000,
        "@text":32.000000
    },
    "x":{
        /*required, float, x-coordinate, accurate to 6 decimal places*/
        "@min":0.000000,
        "@max":1.000000,
        "@text":1.000000
    },
    "y":{
        /*required, float, y-coordinate, accurate to 6 decimal places*/
        "@min":0.000000,
        "@max":1.000000,
        "@text":1.000000
    },
    "showEnabled": "true,false",
    /*optional, boolean, whether to enable calibration stream overlay, if it is set to "true", calibration identification will be displayed on the live view screen*/
    "calibratated": "true,false",
    /*optional, boolean, calibration status*/
    "isSupportCalibrationStatus":true
    /*optional, boolean, whether it supports getting all the calibration status (related URI: /ISAPI/System/GPSCalibration/channels/<ID>/points/status?format=json)*/
}
}

```

F.65 JSON_GPSCalibrationList

Message about configuration parameters of multiple GPS calibration points in JSON format

```
{
    "GPSCalibrationList":[{
        "id": ,
        /*required, int, calibration point ID*/
        "calibratated": ,

```

```
/*optional, boolean, calibration status*/
    "longitude":,
/*required, float, longitude, accurate to 6 decimal places*/
    "latitude":,
/*required, float, latitude, accurate to 6 decimal places*/
    "x":,
/*required, float, x-coordinate, accurate to 6 decimal places*/
    "y":,
/*required, float, y-coordinate, accurate to 6 decimal places*/
}
}
```

F.66 JSON_GPSCalibrationStatus

Message about GPS calibration status in JSON format

```
{
  "GPSCalibrationStatus": {
/*required, GPS calibration status*/
    "status": ""
/*required, string, GPS calibration status, calib-calibrated already,noCalib-Not been calibrated yet*/
  }
}
```

F.67 JSON_GPSInfo

JSON message about GPS information

```
{
  "GPSInfo": {
/*required, actual GPS information*/
    "longitude":,
/*required, float, longitude, range: [-180.000000,180.000000], corrects to 6 decimal places*/
    "latitude":,
/*required, float latitude, range: [-90.000000,90.000000], corrects to 6 decimal places*/
  }
}
```

F.68 JSON_GPSVerification_ResultInfo

JSON message about calibration verification result information

```
{
  "ResultInfo": {
/*required, calibration verification result information*/
    "id": ,
/*required, int, verification point ID; PanoVu series camera supports up to 16 points, and speed dome supports up to
```

```
64 points*/
    "longitude": ,
/*required, float, longitude, corrects to 6 decimal places, range:[-180.000000,180.000000]*/
    "latitude":
/*required, float, latitude, corrects to 6 decimal places, range: [-90.000000,90.000000]*/
}
}
```

F.69 JSON_GPSVerification_ResultInfoList

JSON message about calibration verification results

```
{
  "ResultInfoList":[{
/*required, verification results information*/
    "id": ,
/*required, int, verification point ID; PanoVu series camera supports up to 16 points, and speed dome supports up to
64 points*/
    "longitude": ,
/*required, float, longitude which is calculated based on the pan and tilt, corrects to 6 decimal places, range:
[-180.000000,180.000000]*/
    "latitude":
/*required, float, latitude which is calculated based on the pan and tilt, corrects to 6 decimal places, range:
[-90.000000,90.000000]*/
  }]
}
```

F.70 JSON_GyroVerify

JSON message about gyroscope calibration parameters

```
{
  "GyroVerify": {
/*required, gyroscope calibration*/
    "enabled":
/*required, whether to enable gyroscope calibration: true-yes, false-no*/
  }
}
```

F.71 JSON_GyroVerifyCap

JSON message about gyroscope calibration capability.

```
{
  "GyroVerifyCap":{
    "enabled":{
/*required, whether to enable gyroscope calibration*/
  }
}
```

```
    "@opt": [true,false],  
    "@def":false  
}  
}  
}
```

F.72 JSON_GuideConfig

GuideConfig message in JSON format

```
{  
  "GuideConfig": {  
    "GuideStep": [{  
      /*required, configuration steps of quick setup instruction*/  
      "stepIndex": ,  
      /*required, integer, the maximum supported index*/  
      "StepFunction": {  
        /*required, string, instruction functions*/  
        "channelID": ,  
        /*optional, integer, channel number*/  
        "type": ""  
      },  
      /*required, "masterSlaveTracking" (linkage calibration, corresponding URL: /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCameraCalibrating/capabilities), "mixedTargetDetection" (multi-target type detection, corresponding URL : /ISAPI/Intelligent/channels/<ID>/mixedTargetDetection/capabilities?format=json), eagleFocusing (rapid focus, corresponding URL: /ISAPI/PTZCtrl/channels/<ID>/EagleFocusing/capabilities)*/  
      "status": "unfinished"  
    },  
    /*required, string, instruction status, whether the single configuration has completed*/  
    "guideStatus": "unfinished"  
  },  
  /*required, string, instruction overall status, whether all configurations have completed*/  
}
```

F.73 JSON_GuideConfigCap

GuideConfigCap message in JSON format

```
{  
  "GuideConfigCap": {  
    "GuideStep": {  
      /*required, configuration steps of quick setup instruction*/  
      "stepIndexNum": ,  
      /*required, integer, the maximum supported index*/  
      "StepFunction": {  
        /*required, string, instruction functions*/  
        "channels": ,  
        /*optional, integer, channel number*/  
        "type": {
```

```
    "@opt": ["masterSlaveTracking", "mixedTargetDetection", "eagleFocusing"]
/*required, "masterSlaveTracking" (linkage calibration, corresponding URL: /ISAPI/MasterSlaveTracking/channels/<ID>/slaveCameraCalibrating/capabilities), "mixedTargetDetection" (multi-target type detection, corresponding URL : /ISAPI/Intelligent/channels/<ID>/mixedTargetDetection/capabilities?format=json), eagleFocusing (rapid focus, corresponding URL: /ISAPI/PTZCtrl/channels/<ID>/EagleFocusing/capabilities)*/
},
"guideStatus": {
/*required, string, instruction status, whether the instruction configuration has completed*/
    "@opt": ["finished", "unfinished"]
}
},
"MasterSlaveTrackingCap": {
/*optional, dep:StepFunction, linkage calibration capability*/
    "enabled": {
/*required, boolean, whether to enable, the default value is "true"*/
        "@opt": "true,false",
        "@def": "true"
    },
    "CalibratingMode": {
/*required, string, calibration mode*/
        "@opt": "auto",
        "@def": "auto"
    }
},
"MixedTargetDetectionCap": {
/*optional, dep:StepFunction, capability of multi-target type detection*/
    "enabled": {
/*required, boolean, whether to enable, the default value is "true"*/
        "@opt": "true,false",
        "@def": "true"
    },
    "RuleInfoCap": {
/*required, rule information*/
        "maxSize": 1,
/*required, integer, number of supported rules*/
        "ruleID": {
/*required, integer, rule ID*/
            "@min": 1,
            "@max": 1
        },
        "RegionCap": {
/*required, rule region capability*/
            "minSize": 3,
/*required, integer, the minimum region edges*/
            "maxSize": 10,
/*required, integer, the maximum region edges*/
            "x": {
/*required, float, X-coordinate, range: from 0.000 to 1*/
                "@min": 0.000,
                "@max": 1.000,
                "#text": 0.120
            },
        }
    }
}
```

```
"y": {
/*required, float, Y-coordinate, range: from 0.000 to 1*/
    "@min": 0.000,
    "@max": 1.000,
    "#text": 0.120
}
}
},
"EagleFocusingCap": {
/*optional, dep:StepFunction, rapid focus capability*/
    "focusMode": {
/*required, string, focus mode*/
        "@opt": "auto",
        "@def": "auto"
    },
    "controlEnabled": {
/*required, boolean, whether to enable rapid focus, corresponding URL: /ISAPI/PTZCtrl/channels/<ID>/EagleFocusing/control/capabilities*/
        "@opt": "true,false",
        "@def": "true"
    }
}
}
```

F.74 JSON_HddFormatList

HddFormatList message in JSON format

```
{
    "HddFormatList": [
        {
/*required, HDD list to be formatted*/
            "HddFormat": {
/*required, initialization parameter of a HDD*/
                "id": ,
/*required, string type, ID of HDDs that specified to be formatted*/
                "formatType": ""
/*optional, string type, formatting type: FAT32, EXT4; this node is only available for SD card; if this node does not exist, the formatting type is the default type "FAT32"*/
            }
        }
    ]
}
```

F.75 JSON_HeatMap_CollectionDescription

CollectionDescription message in JSON format

```
{
    "CollectionDescription": {
        "searchID": "",
        /*required, string, unique search ID, the content of searchID remains unchanged when search condition remains unchanged.*/
        "searchResultPosition": "",
        /*required, initial position of search result list, integer 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*/
        "maxResults": "",
        /*required, number of matched records per search, integer type*/
        "timeSpanList": [
            /*required, time interval list*/
            "startTime": "",
            /*required, start time, ISO8601_time, string*/
            "endTime": "",
            /*required, end time, ISO8601_time, string*/
        ],
        "channels": "",
        /*optional, sub type, array, integer32, related channel No., array. If the channel information is not specified, it indicates searching all channels*/
        "statisticsType": "PDC",
        /*required, string, statistic type: OLD-heat map, PDC-people counting, DURATION-people dwell time, INTERSECTION-people counting of intersection*/
    }
}
```

F.76 JSON_HeatMap_CollectionDescriptionCap

CollectionDescriptionCap message in JSON format

```
{
    "CollectionDescriptionCap": {
        "timeRange": "",
        /*required, supported time range for replenishment, unit: day, integer*/
        "timeSpanMaxNum": "",
        /*required, the maximum number of time buckets supported for one time searching, integer*/
        "resultMaxNum": "",
        /*required, the maximum number of items can be searched, integer*/
        "channelMaxNum": "",
        /*required, the maximum number of channels supported for one time search, integer*/
        "statisticsType": {
            /*required, string, statistics type: OLD-heat map, PDC-people counting statistics, DURATION-people staying time duration, INTERSECTION-people counting of intersection*/
            "@opt": "OLD,PDC,DURATION,INTERSECTION"
        }
    }
}
```

F.77 JSON_HeatMap_CollectionResult

CollectionResult message in JSON format

```
{  
  "CollectionResult":{  
    "responseStatusStrg": "",  
    /*required, searching status: OK- Searching ended, NO MATCHES-No matched data found, MORE-Search again for  
    more results, string, the max. length is 32, {dep if errcode == 1 && errMsg == ok}*/  
    "numOfMatches": "",  
    /*required, returned number of results for current search, integer32, {dep if errcode == 1 && errMsg == ok}*/  
    "totalMatches": "",  
    /*required, total number of matched results, integer32, {dep if errcode == 1 && errMsg == ok}*/  
    "targets": [{  
      /*optional, people gathering result*/  
      "startTime": "",  
      /*required, start time*/  
      "endTime": "",  
      /*required, end time*/  
      "channel": "",  
      /*required, integer, channel No.*/  
      "statisticsType": "",  
      /*required, string, statistic type: OLD-heat map, PDC-people counting, DURATION-people staying time, INTERSECTION-  
      people counting of intersection*/  
      "HeatMapView": [{  
        "heatmapDataType": "",  
        /*optional, string, heat map data type, it is valid only when statisticsType values "PDC". "PDC_stayNum"-the number  
        of people dwell in the image, "PDC_leaveNum"-the number of exiting people in the image*/  
        "maxHeatMapView": ,  
        /*required, integer, the max. heat value, 4-byte*/  
        "minHeatMapView": ,  
        /*required, integer, the min. heat value, 4-byte*/  
        "timeHeatMapView": ,  
        /*required, integer, the average heat value, 4-byte*/  
        "lineValue": ,  
        /*required, integer, row value, 4-byte*/  
        "columnValue": ,  
        /*required, integer, column value, 4-byte*/  
        "curNumber": ,  
        /*optional, integer, the number of current people, it is valid only when statisticsType values "PDC"*/  
        "leaveNumber": ,  
        /*optional, integer, the number of exiting people, it is valid only when statisticsType values "PDC"*/  
        "totalTime": ,  
        /*optional, integer, total dwell time, it is valid only when statisticsType values "DURATION"*/  
      }],  
      "heatmapURL": "",  
      /*optional, string, URL of pixel data for heat map image, it is valid when statisticsType values "OLD", "DURATION" or  
      "INTERSECTION"*/  
      "PDC_stayURL": "",  
      /*optional, string, URL of pixel data for dwell people in heat map image, it is valid when statisticsType values "PDC"*/  
      "PDC_leaveURL": ""},
```

```
/*optional, string, URL of pixel data for exiting people, it is valid when statisticsType values "PDC"*/
    "arrayUnitType": ""
/*optional, string, matrix unit data type (each pixel data type of matrix information): "byte"-a byte, "short"-2 bytes,
"int"-four bytes*/
}
}
}
```

F.78 JSON_id

id message in JSON format

```
{
  "id": "",
/*optional, integer, user ID, it returns when double verification user is added*/
}
```

F.79 JSON_ImportCustomAudio

JSON message about information of custom alarm audio file to be imported

Import Audio File in URL format

```
{
  "CustomAudioInfo": {
/*required, related information about custom alarm audio file*/
    "customAudioName": ""
/*required, string, custom alarm audio file name, the maximum size is 24 bytes*/
    "channels": [1,2],
/*optional, integer, ID of the channel, which the audio file should be sent to; the voice coding format of each channel
should be the same; if this node is empty or not configured, the audio file will be sent to all channels*/
    "AudioType": "AI",
/*optional, string, audio file type: AI-AI audio file*/
    "isCover": false
/*optional, boolean, whether to cover the audio file with same name: true-yes, false-no (default)*/
  }
}
```

Import Audio File in Binary Format

Content-Type: multipart/form-data; boundary=-----7e13971310878
-----7e13971310878

Content-Disposition: form-data; name="CustomAudioInfo";
Content-Type: text/json

```
{
  "CustomAudioInfo": {
/*required, related information about custom alarm audio file*/
```

```
"customAudioName": "",  
/*required, string, custom alarm audio file name, the maximum size is 24 bytes*/  
 "AudioType":"AI",  
/*optional, string, audio file type: AI-AI audio file*/  
 "isCover":false  
/*optional, boolean, whether to cover the audio file with same name: true-yes, false-no (default)*/  
 }  
}  
-----7e13971310878  
Content-Disposition: form-data; name="audioData";  
Content-Type: audio/wav  
  
Opaque Data, binary audio file  
-----7e13971310878--
```

F.80 JSON_IntelligentSearchCap

JSON message about intelligent search capability

```
{  
 "requestURL":"",
/*optional, string, request URL*/
 "statusCode":1,
/*required, int, status code*/
 "statusString":"",
/*required, string, status description*/
 "subStatusCode":"",
/*required, string, sub status code*/
 "errorCode":1,
/*optional, int, error code, this node is required when statusCode is not 1 and it corresponds to subStatusCode*/
 "errorMsg":"ok",
/*optional, string, error details, this node is required, when statusCode is not 1 and it can be error details about a specific parameter*/
 "startTime":"2004-05-03T17:30:08+08:00",
/*required, string, start time in ISO8601 time format. This node only indicates that the device supports this field*/
 "endTime":"2004-05-03T17:30:08+08:00",
/*required, string, end time in ISO8601 time format. This node only indicates that the device supports this field*/
 "resultMaxNum":100,
/*required, int, supported maximum number of results that can be searched*/
 "channelIDLen":{
/*optional, camera ID length*/
 "min":1,
 "max":32
 },
 "channelNameLen":{
/*optional, camera name length*/
 "min":1,
 "max":32
 },
 "streamType": "realtime,historyvideo,localvideo",
```

```
/*optional, string, streaming type, "realtime,historyvideo,localvideo"*/
"targetRect":{
/*required, target frame*/
"height":{
  "min":1,
  "max":100
},
"width":{
  "min":1,
  "max":100
},
"x":{
  "min":1,
  "max":100
},
"y":{
  "min":1,
  "max":100
}
},
"picUrlLen":{
/*required, URL length of the large picture*/
"min":1,
"max":128
},
"subPicUrlLen": {
/*required, length of the thumbnail URL*/
"min": 1,
"max": 1
},
"targetID": {
/*optional, string, linkage ID between face and human body*/
"min": 1,
"max": 1
},
"targetType":"vehicle",
/*string, search object type*/
"targetSize": "",
/*optional, string, target size*/
"direction": "",
/*optional, string, target direction*/
"speed": "",
/*optional, string, target speed*/
"ageGroup":"",
/*optional, string, age*/
"gender":"",
/*optional, string, gender, string, gender*/
"glass":"",
/*optional, string, wear glasses or not*/
"bag":"",
/*optional, string, whether with bag or not*/
"hat": "",
```

```
/*optional, string, whether with hat or not*/
"mask":"",
/*optional, string, whether with mask or not*/
"jacket": "",
/*optional, string, jacket type*/
"trousersType":"",
/*optional, string, trousers type*/
"hairStyle":"",
/*optional, string, hair style*/
"jacketColor":"",
/*optional, string, jacket color*/
"trousersColor":"",
/*optional, string, trousers coloe*/
"ride":"",
/*optional, string, ride a bike or not*/
"things":"",
/*optional, string, whether with handbag or not*/
"cyclingType":"",
/*optional, string, vehicle type*/
"cyclingPersonNumber":"",
/*optional, string, number of passengers*/
"isSupportLicense":true,
/*required, boolean type, whether it supports license plate number*/
"plateType":"unknown,92TypeCivil,arm,upDownMilitay,92TypeArm,leftRightMilitay,
02TypePersonalized,yellowTwoLine,
04NewMilitay,embassy,oneLineArm,twoLineArm,yellow1225FarmVehicle,green1325FarmVehicle,yellow1325FarmVehi
cle,motorola,coach,tempTravl,trailer,consulate,hongKongMacao,tempEntry,civilAviation,newEnergy",
/*optional, string, license plate type: license plate type: "unknown", "92TypeCivil"-92-style civil vehicle, "arm"-police
vehicle, "upDownMilitay"-military vehicle (top-bottom type), "92TypeArm"-92-style police vehicle, "leftRightMilitay"-military
vehicle (left-right type), "02TypePersonalized"-02-style customized vehicle, "yellowTwoLine"-yellow two-line rear
license plate, "04NewMilitay"-04-style new military vehicle, "embassy"-embassy vehicle, "oneLineArm"-new armed
police vehicle (one-line), "twoLineArm"-new armed police vehicle (two-line), "yellow1225FarmVehicle"-yellow agricultural
vehicle with 1225 structure, "green1325FarmVehicle"-green agricultural vehicle with 1325 structure,
"yellow1325FarmVehicle"-yellow agricultural vehicle with 1325 structure, "motorola"-motorcycle, "coach"-driver-training
vehicle, "tempTravl"-vehicle with temporary license plate, "trailer"-trailer, "consulate"-consular vehicle,
"hongKongMacao"-vehicle entering and leaving Hong Kong/Macao, "tempEntry"-temporary entry vehicle,
"civilAviation"-civil aviation license plate, "newEnergy"-new energy license plate. Multiple types should be separated
by commas*/
"vehicleColor":"white,silver,gray,black,red,deepBlue,blue,yellow,green,brown,pink,purple,deepGray,cyan,orange,unkn
own",
/*optional, string, vehicle color: "unknown", "white", "silver"-silvery, "gray", "black", "red", "deepBlue"-dark blue,
"blue", "yellow", "green", "brown", "pink", "purple", "deepGray"-dark gray, "cyan", "orange". Multiple colors should
be separated by commas*/
"plateColor":"white,yellow,blue,black,green,civilAviationBlack,civilAviationGreen,other",
/*optional, string, license plate color: "white", "yellow", "blue", "black", "green", "civilAviationBlack"-civil aviation
black, "civilAviationGreen"-civil aviation green, "0xff" or "other"-other color, "unknown"-unrecognized. Multiple colors
should be separated by commas*/
"vehicleType":"largeBus,truck,vehicle,van,buggy,pedestrian,twoWheelVehicle,threeWheelVehicle,SUVMPV,mediumBus
,motorVehicle,nonmotorVehicle,smallCar,miniCar,pickupTruck,unknown",
/*optional, string, vehicle type: "largeBus"-large-sized bus, "truck"-truck, "vehicle"-salon car, "van"-minivan, "buggy"-
```

```
light truck, "pedestrian", "twoWheelVehicle"-two wheeler, "threeWheelVehicle"-tricycle, "SUVMPV"-SUV/MPV,
"mediumBus"-middle-sized bus, "motorVehicle"-motor vehicle (it will be transformed to "vehicle" (salon car) in the
platform), "nonmotorVehicle"-non-motor vehicle (it will be transformed to "threeWheelVehicle" (tricycle) in the
platform), "smallCar"-small sedan (it will be transformed to "vehicle" (salon car) in the platform), "miniCar"-mini
sedan (it will be transformed to "vehicle" (salon car) in the platform), "pickupTruck"-pick-up truck, "unknown".
Multiple types should be separated by commas*/
"isSupportVehicleLogo":true,
/*optional, boolean type, whether it supports vehicle parent brand*/
"isSupportVehicleSubLogo":true,
/*optional, boolean type, whether it supports vehicle sub brand*/
"isSupportVehicleModel":true,
/*optional, boolean type, whether it supports vehicle model year*/
"pilotSafebelt":"",
/*optional, string type, whether the driver buckled up*/
"pilotSunvisor":"",
/*optional, string type, whether the driver's sun visor is pulled down*/
"vicePilotSafebelt":"",
/*optional, string type, whether the front passenger buckled up*/
"vicePilotSunvisor":"",
/*optional, string type, whether the front passenger's sun visor is pulled down*/
"uphone":"",
/*optional, string type, whether the person is making a call*/
"dangmark": "",
/*optional, string, whether it is dangerous goods vehicle*/
"envprosign": "",
/*optional, string, whether to display yellow label vehicle*/
"muckTruck": "",
/*optional, string, whether it is a dump truck*/
"vehicleState": "",
/*optional, string, license plate status*/
"pdvs": "",
/*optional, string, whether there are people sticking out of the sunroof*/
"vehicleHead": "",
/*optional, string, license plate recognition direction*/
"pendant": "",
/*optional, string, whether there is window hangings detected*/
"temporaryLicense": "",
/*optional, string, whether it is a temporary license plate*/
"tricycleCanopy": "",
/*optional, string, whether there is a hood on the tricycle*/
"decoration": "",
/*optional, string, whether there is decoration detected*/
"tissueBox": "",
/*optional, string, whether there is tissue box detected*/
"card": "",
/*optional, string, whether it is with cards*/
"cardType": "",
/*optional, string, card type*/
"copilot": "",
/*optional, string, whether there is a person in the co-driver room*/
"frontChild": "",
/*optional, string, whether the co-driver is with baby in arm*/
```

```
"sunroof": "",  
/*optional, string, whether it is with a sunroof*/  
"luggageRack": "",  
/*optional, string, whether it is with a luggage rack*/  
"vehicleSprayPainted": "",  
/*optional, string, whether the vehicle is painted*/  
"spareTire": "",  
/*optional, string, whether it is with a spare tire*/  
"coverPlate": "",  
/*optional, string, whether the heavy truck is covered with back cover*/  
"label": "",  
/*optional, string, whether there is inspection label detected*/  
"plateRecogRegionIDNum":  
/*optional, number of IDs of the license plate recognition region*/ {  
    "min": 1,  
    "max": 1  
},  
"discuss": {  
/*optional, discussion*/  
    "confidence": 99.99990,  
/*optional, float, confidence*/  
    "value": ""  
/*optional, discussion type, "unknown", "no", "yes"*/  
},  
"channelMaxNum":1,  
/*required, integer type, maximum number of channels that support simultaneous search*/  
"supportSearchByUTC":true  
/*optional, boolean type, whether it supports searching by UTC*/  
"linkFaceBodyId": {  
    "@min": 0,  
    "@max": 64  
},  
/*optional, linkage ID between human face and body, the maximum length is 64 bytes*/  
"sortType": {  
/*optional, string, sorting type: similarity-by similarity(by default), time-by time, the maximum length is 32 bytes*/  
    "@opt":["similarity", "time"]  
},  
"plateCategory": {  
/*optional, additional information of license plate*/  
    "@min": 0,  
    "@max": 0  
},  
"taskID":[{  
/*optional, string, video analysis task ID, which is the taskID returned by APIs such as /ISAPI/SDT/Management/Task/  
Video/monitorPort?format=json, /ISAPI/SDT/Management/Task/Video/monitorPort/batch?format=json, etc. */  
    "@size":  
}],  
"minTaskCreateTime": "",  
/*optional, string, the creation time of the task with the minimum task ID in the taskIDs. The time is in ISO 8601 time  
format. If this node is returned, it only indicates that the device supports this node*/  
"maxTaskFinishTime": "",  
/*optional, string, the end time of the task with the maximum task ID in the taskIDs. The time is in ISO 8601 time
```

```
format. If this node is returned, it only indicates that the device supports this node*/
/*The node minTaskCreateTime and maxTaskFinishTime are used to speed up searching. The task creation time
(taskCreateTime) and end time (taskFinishTime) can be obtained by calling the URI: /ISAPI/SDT/Management/Task/
Video/monitorPort/search?format=json*/
"fuzzyMatch":{
/*optional, boolean, whether it supports fuzzy match of license plates. If this node is returned, it indicates that fuzzy
matching of license plates is supported*/
    "@opt":  
}  
}
```

F.81 JSON_IntelligentSearchCondition

JSON message about intelligent search conditions

```
{
    "searchID":"",
/*required, string, search ID. The content of searchID remains unchanged if the search conditions remain unchanged.
This node is used to check the same search. When the device performance is limited, you can search asynchronously
by applying conditions with the same searchID several times and getting the search progress*/
    "searchResultPosition": ,
/*required, int, 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, int, maximum number of search results obtained this time*/
    "startTime":"",
/*required, string, start time in ISO8601 time format*/
    "endTime":"",
/*required, string, end time in ISO8601 time format*/
    "choiceChannel":[{
/*optional*/
        "channelID":"",
/*optional, string, camera No.*/
        "plateRecogRegionID":
/*optional, int, license plate recognition area No. If this node is not configured, it refers to all areas*/
        }],
        "targetType":"",
/*optional, string, search object type. If this node is not returned, it indicates no limit*/
        "targetSize": "",
/*optional, string, target size*/
        "direction": "",
/*optional, string, target direction*/
        "speed": "",
/*optional, string, target speed*/
        "sortType": "",
/*optional, string, sorting type: "similarity"-by similarity(by default), "time"-by time, the maximum length is 32 bytes*/
        "humanInfo": {
/*dependent, this node is valid only when the value of targetType is human*/
            "ageGroup":"",
/*optional, string, age*/
        }
    }
}
```

```
"gender":"",
/*optional, string, gender, string, gender*/
"glass":"",
/*optional, string, wear glasses or not*/
"bag":"",
/*optional, string, whether with bag or not*/
"hat":"",
/*optional, string, whether with hat or not*/
"mask":"",
/*optional, string, whether with mask or not*/
"jacketType":"",
/*optional, string, jacket type*/
"trousersType":"",
/*optional, string, trousers type*/
"hairStyle":"",
/*optional, string, hair style*/
"jacketColor":"",
/*optional, string, jacket color, "white,black"*/
"trousersColor":"",
/*optional, string, trousers color, "white,black"*/
"ride":"",
/*optional, string, ride a bike or not*/
"things":"",
/*optional, string, whether with handbag or not*/
"cyclingType":"",
/*optional, string, vehicle type*/
"cyclingPersonNumber":"",
/*optional, string, number of passengers*/
"hID":"",
/*required, string, person ID, the maximum length is 32 bytes*/
"name":"",
/*required, string, name of the person, the maximum length is 32 bytes*/
"linkFaceBodyId":""
/*optioal, linkage ID between human face and body, the maximum length is 64 bytes*/
},
"vehicleInfo":{
/*this node is valid when targetType is "vehicle"*/
"license":"",
/*optional, string, license plate number*/
"plateCategory": "",
/*optional, additional information of license plate*/
"plateType":"",
/*optional, string, license plate type: license plate type: "unknown", "92TypeCivil"-92-style civil vehicle, "arm"-police vehicle, "upDownMilitay"-military vehicle (top-bottom type), "92TypeArm"-92-style police vehicle, "leftRightMilitay"-military vehicle (left-right type), "02TypePersonalized"-02-style customized vehicle, "yellowTwoLine"-yellow two-line rear license plate, "04NewMilitay"-04-style new military vehicle, "embassy"-embassy vehicle, "oneLineArm"-new armed police vehicle (one-line), "twoLineArm"-new armed police vehicle (two-line), "yellow1225FarmVehicle"-yellow agricultural vehicle with 1225 structure, "green1325FarmVehicle"-green agricultural vehicle with 1325 structure, "yellow1325FarmVehicle"-yellow agricultural vehicle with 1325 structure, "motorola"-motorcycle, "coach"-driver-training vehicle, "tempTravl"-vehicle with temporary license plate, "trailer"-trailer, "consulate"-consular vehicle, "hongKongMacao"-vehicle entering and leaving Hong Kong/Macao, "tempEntry"-temporary entry vehicle, "civilAviation"-civil aviation license plate, "newEnergy"-new energy license plate. If this node is not returned, it
```

```
indicates no limit*/
"vehicleColor": "",  
/*optional, string, vehicle color: "unknown", "white", "silver"-silvery, "gray", "black", "red", "deepBlue"-dark blue,
"blue", "yellow", "green", "brown", "pink", "purple", "deepGray"-dark gray, "cyan", "orange". Multiple colors should
be separated by commas. If this node is not returned, it indicates no limit*/
"vehicleType": "",  
/*optional, string, vehicle type: "largeBus"-large-sized bus, "truck"-truck, "vehicle"-salon car, "van"-minivan, "buggy"-  
light truck, "pedestrian", "twoWheelVehicle"-two wheeler, "threeWheelVehicle"-tricycle, "SUVMPV"-SUV/MPV,  
"mediumBus"-middle-sized bus, "unknown". If this node is not returned, it indicates no limit*/
"vehicleLogo": ,  
/*optional, int, vehicle parent brand. If this node is not returned, it indicates no limit*/
"vehicleSubLogo": ,  
/*optional, int, vehicle sub brand. If this node is not returned, it indicates no limit*/
"vehicleModel": ,  
/*optional, int, vehicle model year. If this node is not returned, it indicates no limit*/
"plateColor": "",  
/*optional, string, license plate color: "white", "yellow", "blue", "black", "green", "civilAviationBlack"-civil aviation  
black, "civilAviationGreen"-civil aviation green, "0xff" or "other"-other color. If this node is not returned, it indicates no  
limit*/
"pilotSafebelt": "",  
/*optional, string, whether the driver buckled up. If this node is not returned, it indicates no limit*/
"pilotSunvisor": "",  
/*optional, string, whether the driver's sun visor is pulled down. If this node is not returned, it indicates no limit*/
"vicePilotSafebelt": "",  
/*optional, string, whether the front passenger buckled up*/
"vicePilotSunvisor": "",  
/*optional, string, whether the front passenger's sun visor is pulled down*/
"uphone": "",  
/*optional, string, whether the person is making a call*/
"dangmark": "",  
/*optional, string, whether it is dangerous goods vehicle*/
"envprosign": "",  
/*optional, string, whether it is a yellow-label vehicle*/
"vehicleState": "",  
/*optional, string, license plate status*/
"pdvs": "",  
/*optional, string, whether there are people sticking out of the sunroof*/
"vehicleHead": "",  
/*optional, string, license plate recognition direction*/
"pendant": "",  
/*optional, string, whether there is window hangings detected*/
"temporaryLicense": "",  
/*optional, string, whether it is a temporary license plate*/
"tricycleCanopy": "",  
/*optional, string, whether there is a hood on the tricycle*/
"decoration": "",  
/*optional, string, whether there is decoration detected*/
"tissueBox": "",  
/*optional, string, whether there is tissue box detected*/
"card": "",  
/*optional, string, whether it is with cards*/
"cardType": "",
```

```
/*optional, string, card type*/
  "copilot": "",

/*optional, string, whether there is a person in the co-driver room*/
  "frontChild": "",

/*optional, string, whether the co-driver is with baby in arm*/
  "muckTruck": "",

/*optional, string, whether it is a dump truck*/
  "sunroof": "",

/*optional, string, whether it is with a sunroof*/
  "luggageRack": "",

/*optional, string, whether it is with a luggage rack*/
  "vehicleSprayPainted": "",

/*optional, string, whether the vehicle is painted*/
  "spareTire": "",

/*optional, string, whether it is with a spare tire*/
  "coverPlate": "",

/*optional, string, whether the heavy truck is covered with back cover*/
  "label": "",

/*optional, string, whether there is inspection label detected*/
  "fuzzyMatch":true

/*optional, boolean, whether it supports fuzzy match of license plates: true, false (default). If fuzzy matching is disabled, you need to input the complete license plate number to search. Fuzzy matching is time-consuming*/
  },

"twoWheelVehicle": {

/*dependent, this node is valid only when the value of targetType is "twoWheelVehicle"*/
  "ageGroup": "",

/*optional, string, whether to display age group*/
  "gender": "",

/*optiona, string, whether to display gender*/
  "glass": "",

/*optional, string, whether to display glasses*/
  "bag": "",

/*optional, string, whether to display back bag*/
  "hat": "",

/*optional, string, whether to display hat*/
  "mask": "",

/*optional, string, whether to display mask*/
  "jacketType": "",

/*optional, string, whether to display jacket type*/
  "hairStyle": "",

/*optional, string, whether to display hairstyle*/
  "jacketColor": "",

/*optional, string, whether to display jacket color*/
  "cyclingType": "",

/*optional, string, whether to display cycling type*/
  "cyclingPersonNumber": "",

/*optional, string, whether to display number of people cycling*/
  },

"threeWheelVehicle": {

/*dependent, this node is valid only when the value of targetType is "threeWheelVehicle"*/
  "ageGroup": "",

/*optional, string, whether to display age group*/
```

```

    "gender": "",
    /*optional, string, whether to display gender*/
    "glass": "",
    /*optional, string, whether to display glasses*/
    "bag": "",
    /*optional, string, whether to display back bag*/
    "hat": "",
    /*optional, string, whether to display hat*/
    "mask": "",
    /*optional, string, whether to display mask*/
    "jacketType": "",
    /*optional, string, whether to display jacket type*/
    "hairStyle": "",
    /*optional, string, whether to display hairstyle*/
    "jacketColor": "",
    /*optional, string, whether to display jacket color*/
    "cyclingPersonNumber": ""
    /*optional, string, whether to display number of people cycling*/
},
"taskIDs":["","",""],
/*optional, string, task ID group which can contain multiple task IDs*/
"minTaskCreateTime": "2004-05-03T17:30:08+08:00",
/*optional, string, the creation time of the task with the minimum task ID in the taskIDs. The time is in ISO 8601 time format*/
"maxTaskFinishTime ": ""
/*optional, string, the end time of the task with the maximum task ID in the taskIDs. The time is in ISO 8601 time format*/
/*The node minTaskCreateTime and maxTaskFinishTime are used to speed up searching. The task creation time (taskCreateTime) and end time (taskFinishTime) can be obtained by calling the URI: /ISAPI/SDT/Management/Task/Video/monitorPort/search?format=json*/
}

```

F.82 JSON_IntelligentSearchResult

JSON message about intelligent search results

```
{
    "requestURL":"",
    /*required, string type, request URL*/
    "statusCode": ,
    /*required, int type, status code*/
    "statusString":"",
    /*required, string type,status description*/
    "subStatusCode":"",
    /*required, string type, sub status code*/
    "errorCode": ,
    /*optional, int type, error code, this node is required when statusCode is not 1 and it corresponds to subStatusCode*/
    "errorMsg":"",
    /*optional, string type, error details, this node is required, when statusCode is not 1 and it can be error details about a specific parameter*/
}
```

```
"responseStatusStrg": "",  
/*required, string type, searching status: "OK"-searching completed, "NO MATCH"-no matched results, "MORE"-  
searching for more results. The maximum length is 32 bytes. This node is valid when errorCode is 1 and errorMsg is  
"ok"*/  
"numOfMatches": ,  
/*required, integer32, number of results returned this time. This node is valid when errorCode is 1 and errorMsg is  
"ok"*/  
"totalMatches": ,  
/*required, integer32, total number of matched results. This node is valid when errorCode is 1 and errorMsg is "ok"*/  
"progress": ,  
/*optional, integer32 type, VCA search progress*/  
"targetInfo": [{  
/*this node is valid when progress is 100*/  
"captureTime": "",  
/*required, string type, capture time in ISO8601 time format*/  
"isSummerTime": ,  
/*optional, boolean, whether the time returned currently is in DST system*/  
"channelID": "",  
/*optional, string type, camera No.*/  
"channelName": "",  
/*optioanl, string type, camera name*/  
"streamType": "",  
/*optional, string type, streaming type: "realtime,historyvideo,localvideo"*/  
"targetRect": {  
/*required, target frame*/  
"height": ,  
"width": ,  
"x": ,  
"y":  
},  
"picUrl": "",  
/*optional, string type, large picture URL*/  
"subPicUrl": "",  
/*optional, string, thumbnail URL*/  
"targetType": "",  
/*optional, string type, search object type*/  
"targetSize": "",  
/*optional, string, target size*/  
"direction": "",  
/*optional, string, target direction*/  
"speed": "",  
/*optional, string, target speed*/  
"modelKey": "",  
/*optional, string, mark of human body which is used to raise up the efficiency of searching for human body by  
picture*/  
"AbsoluteHigh": {  
/* optional, PTZ position*/  
"elevation": ,  
/* optional, int, tilting parameter, the value is between -900 and 2700*/  
"azimuth": ,  
/* optional, int, panning parameter, the value is between 0 and 3600*/  
"absoluteZoom":
```

```
/* optional, int, zooming parameter, the value is between 1 and 1000*/
},
"humanInfo":{

/*dependent, this node is valid only when the value of targetType is human*/
    "ageGroup":"",
/*optional, string, age*/
    "gender":"",
/*optional, string, gender, string, gender*/
    "glass":"",
/*optional, string, wear glasses or not*/
    "bag":"",
/*optional, string, whether with bag or not*/
    "hat":"",
/*optional, string, whether with hat or not*/
    "mask":"",
/*optional, string, whether with mask or not*/
    "jacketType":"",
/*optional, string, jacket type*/
    "trousersType":"",
/*optional, string, trousers type*/
    "hairStyle":"",
/*optional, string, hair style*/
    "jacketColor":"",
/*optional, string, jacket color*/
    "trousersColor":"",
/*optional, string, trousers coloe*/
    "ride":"",
/*optional, string, ride a bike or not*/
    "things":"",
/*optional, string, whether with handbag or not*/
    "cyclingType":"",
/*optional, string, vehicle type*/
    "cyclingPersonNumber":"",
/*optional, string, number of passengers*/
    "posture":{

/*optional, posture*/
        "confidence":,
/*optional, float, confidence*/
        "value":""
    },
/*optional, string, posture type*/
    },
    "listening":{

/*optional, listening*/
        "confidence":,
/*optional, float, listening*/
        "value":""
    },
/*optional, string, listening type*/
    },
    "reading":{

/*optional, reading*/
        "confidence":,
/*optional, float, confidence*/
        }
}
```

```
        "value":""  
/*optional, string, reading type*/  
    },  
    "writing":{  
/*optional, writing*/  
    "confidence":,  
/*optional, float, confidence*/  
    "value":""  
/*optional, string, writing type*/  
    },  
    "raiseHand":{  
/*optional, raise hand*/  
    "confidence":,  
/*optional, float, confidence*/  
    "value":""  
/*optional, string, hand raising type*/  
    },  
    "lyingOnTable":{  
/*optional, lying on the table*/  
    "confidence":,  
/*optional, float, confidence*/  
    "value":""  
/*optional, string, types of lying on the table*/  
    },  
    "playPhone":{  
/*optional, playing phone*/  
    "confidence":,  
/*optional, float, confidence*/  
    "value":""  
/*optional, string , types of playing phone*/  
    },  
    "orientation":{  
/*optional, orientation*/  
    "confidence":,  
/*optional, float, confidence*/  
    "value":""  
/*optional, string, types of orientation*/  
    },  
    "discuss":{  
/*optional, discussion*/  
    "confidence":,  
/*optional, float, confidence*/  
    "value":""  
/*optional, string, types of discussion: unknown, no discussion, have discussion*/  
    },  
    "studentScene": "",  
/*optional, string, school scene: "takeLesson", "answer", "discuss"*/  
    "studentNum": ,  
/*optional, int, number of students*/  
    "linkFaceBodyId":""  
/*optional, linkage ID between human body and face, the maximum length is 64 bytes*/  
    },
```

```
"vehicleInfo":{  
    /*this node is valid when targetType is "vehicle"*/  
    "license": "",  
    /*optional, string type, license plate number*/  
    "plateCategory": "",  
    /*optional, additional information of license plate*/  
    "plateType": "",  
    /*optional, string type, license plate type: license plate type: "unknown", "92TypeCivil"-92-style civil vehicle, "arm"-police vehicle, "upDownMilitay"-military vehicle (top-bottom type), "92TypeArm"-92-style police vehicle, "leftRightMilitay"-military vehicle (left-right type), "02TypePersonalized"-02-style customized vehicle, "yellowTwoLine"-yellow two-line rear license plate, "04NewMilitay"-04-style new military vehicle, "embassy"-embassy vehicle, "oneLineArm"-new armed police vehicle (one-line), "twoLineArm"-new armed police vehicle (two-line), "yellow1225FarmVehicle"-yellow agricultural vehicle with 1225 structure, "green1325FarmVehicle"-green agricultural vehicle with 1325 structure, "yellow1325FarmVehicle"-yellow agricultural vehicle with 1325 structure, "motorola"-motorcycle, "coach"-driver-training vehicle, "tempTravl"-vehicle with temporary license plate, "trailer"-trailer, "consulate"-consular vehicle, "hongKongMacao"-vehicle entering and leaving Hong Kong/Macao, "tempEntry"-temporary entry vehicle, "civilAviation"-civil aviation license plate, "newEnergy"-new energy license plate*/  
    "vehicleColor": "",  
    /*optional, string type, vehicle color: "unknown", "white", "silver"-silvery, "gray", "black", "red", "deepBlue"-dark blue, "blue", "yellow", "green", "brown", "pink", "purple", "deepGray"-dark gray, "cyan", "orange". Multiple colors should be separated by commas*/  
    "vehicleType": "",  
    /*optional, string type, vehicle type: "largeBus"-large-sized bus, "truck"-truck, "vehicle"-salon car, "van"-minivan, "buggy"-light truck, "pedestrian", "twoWheelVehicle"-two wheeler, "threeWheelVehicle"-tricycle, "SUVMPV"-SUV/MPV, "mediumBus"-middle-sized bus, "unknown"*/  
    "vehicleLogo": ,  
    /*optional, int type, vehicle parent brand*/  
    "vehicleSubLogo": ,  
    /*optional, int type, vehicle sub brand*/  
    "vehicleModel": ,  
    /*optional, int type, vehicle model year*/  
    "plateColor": "",  
    /*optional, string type, license plate color: "white", "yellow", "blue", "black", "green", "civilAviationBlack"-civil aviation black, "civilAviationGreen"-civil aviation green, "0xff" or "other"-other color.*/  
    "pilotSafebelt": "",  
    /*optional, string type, whether the driver buckled up*/  
    "pilotSunvisor": "",  
    /*optional, string type, whether the driver's sun visor is pulled down*/  
    "vicePilotSafebelt": "",  
    /*optional, string type, whether the front passenger buckled up*/  
    "vicePilotSunvisor": "",  
    /*optional, string type, whether the front passenger's sun visor is pulled down*/  
    "uphone": "",  
    /*optional, string type, whether the person is making a call*/  
    "dangmark": "",  
    /*optional, string, whether it is dangerous goods vehicle*/  
    "envprosign": "",  
    /*optioal, string, whether it is a yellow-label vehicle*/  
    "vehicleState": "",  
    /*optional, string, license plate status*/  
    "pdvs": "",  
    /*optional, string, whether there are people sticking out of the sunroof*/
```

```
"vehicleHead": "",  
/*optional, string, license plate recognition direction*/  
    "pendant": "",  
/*optional, string, whether there is window hangings detected*/  
    "temporaryLicense": "",  
/*optional, string, whether it is a temporary license plate*/  
    "tricycleCanopy": "",  
/*optional, string, whether there is a hood on the tricycle*/  
    "decoration": "",  
/*optional, string, whether there is decoration detected*/  
    "tissueBox": "",  
/*optional, string, whether there is tissue box detected*/  
    "card": "",  
/*optional, string, whether it is with cards*/  
    "cardNum": ,  
/*optional, int, number of cards*/  
    "cardType": [{  
/*optional, array, card type*/  
        "value": ""  
/*optional, string, value of different card types*/  
    }],  
    "copilot": "",  
/*optional, string, whether there is a person in the co-driver room*/  
    "frontChild": "",  
/*optional, string, whether the co-driver is with baby in arm*/  
    "muckTruck": "",  
/*optional, string, whether it is a dump truck*/  
    "sunroof": "",  
/*optional, string, whether it is with a sunroof*/  
    "luggageRack": "",  
/*optional, string, whether it is with a luggage rack*/  
    "vehicleSprayPainted": "",  
/*optional, string, whether the vehicle is painted*/  
    "spareTire": "",  
/*optional, string type, whether it is with a spare tire*/  
    "coverPlate": "",  
/*optional, string, whether the heavy truck is covered with back cover*/  
    "label": "",  
/*optional, string, whether there is inspection label detected*/  
    "labelNum": ,  
/*optional, int, number of labels*/  
    "labelPosition": [{  
/*optional, label position*/  
        "height": ,  
/*required, float, height*/  
        "width": ,  
/*required, float, width*/  
        "x": ,  
/*required, float, X-coordinate*/  
        "y": ,  
/*required, float, Y-coordinate*/  
    }],
```

```
"region": "",  
/*optional, string type, region that the license plate belongs to: "EU"-Europe, "ER"-Russian region, "EUandCIS"-Europe  
and Russia, "ME"-the Middle East, "All"-all regions, "other"*/  
"country": ""  
/*optional, string type, country/area that the license plate belongs to: "EU"-Europe, "ER"-Russian region, "EUandCIS"-  
Europe and Russia, "ME"-the Middle East, "All"-all regions, "other"*/  
,  
"twoWheelVehicle": {  
/*dependent, this node is valid only when the value of targetType is "twoWheelVehicle"*/  
"ageGroup": "",  
/*optional, string, whether to display age group*/  
"gender": "",  
/*optiona, string, whether to display gender*/  
"glass": "",  
/*optional, string, whether to display glasses*/  
"bag": "",  
/*optional, string, whether to display back bag*/  
"hat": "",  
/*optional, string, whether to display hat*/  
"mask": "",  
/*optional, string, whether to display mask*/  
"jacketType": "",  
/*optional, string, whether to display jacket type*/  
"hairStyle": "",  
/*optional, string, whether to display hairstyle*/  
"jacketColor": "",  
/*optional, string, whether to display jacket color*/  
"cyclingType": "",  
/*optional, string, whether to display cycling type*/  
"cyclingPersonNumber": ""  
/*optional, string, whether to display number of people cycling*/  
,  
"threeWheelVehicle": {  
/*dependent, this node is valid only when the value of targetType is "threeWheelVehicle"*/  
"ageGroup": "",  
/*optional, string, whether to display age group*/  
"gender": "",  
/*optional, string, whether to display gender*/  
"glass": "",  
/*optional, string, whether to display glasses*/  
"bag": "",  
/*optional, string, whether to display back bag*/  
"hat": "",  
/*optional, string, whether to display hat*/  
"mask": "",  
/*optional, string, whether to display mask*/  
"jacketType": "",  
/*optional, string, whether to display jacket type*/  
"hairStyle": "",  
/*optional, string, whether to display hairstyle*/  
"jacketColor": "",  
/*optional, string, whether to display jacket color*/
```

```
"cyclingType": "",  
/*optional, string, whether to display cycling type*/  
"cyclingPersonNumber": ""  
/*optional, string, whether to display number of people cycling*/  
,  
"humanID": "",  
/*optional, string, face picture library ID*/  
"humanURL": "",  
/*optional, string, face picture URL*/  
"name": "",  
/*optional, string, name of the student*/  
"ClassRegions": [{  
/*optional, coordinates of the people in the classroom*/  
"x": ,  
"y":  
},  
"taskID": ""  
/*optional, string, task ID*/  
}  
}
```

F.83 JSON_IOT_ChannelInfo

ChannelInfo message in JSON format

```
{  
"ChannelInfo":{  
"channel": "",  
/*required, channel No., it should be returned when the channel is added, integer*/  
}  
}
```

F.84 JSON_IOT_ChannelInfoList

ChannelInfoList message in JSON format

```
{  
"ChannelInfoList": [{  
/*optional, channel information, if this node does not exist, it indicates all channels, array*/  
"channel": "",  
/*required, channel No., integer*/  
"type": ""  
/*required, channel type: "video"-video channel, "IOT"-IoT channel, string*/  
}]  
}
```

F.85 JSON_IOT_ErrorList

ErrorList message in JSON format

```
{
  "ErrorList": [
    /*dep, detailed error information, it is valid when subStatusCode values "badParameters"*/
    {"errorRowNo": "",  

     /*required, error row No., integer*/  

     "errorType": "",  

     /*required, error type, string, channelNoInvalid-invalid channel No., channelNoConflict-conflicted channel No.,  

      channel IP/Domain invalid-invalid channel IP or domain name, channel IP/Domain conflict-conflicted channel IP or  

      domain name, "channel IP conflict with local IP"-channel IP is conflicted with local IP, protocolError-incorrect protocol,  

      adminPortError-incorrect admin port, channelError-incorrect channel No., UserNameInvalid-invalid user name,  

      passwordInvalid-invalid password, transProtocolError-incorrect transmission protocol, inductiveTypeInvalid-invalid  

      inductive type*/
    }]
}
```

F.86 JSON_IOTChannel

IOTChannel message in JSON format

```
{
  "IOTChannel": {
    /*optional, added IoT devices information*/
    "channel": "",  

    /*optional, channel No., if this node does not exist, it indicates any free channel, integer*/
    "IOTProtocolType": "",  

    /*required, IOT protocol type: "HIKVISION"-Hikvision device, "OPTEX"-OPTEX security control panel, "Luminite"-  

     Luminite security control panel, "GJD"-GJD security control panel, "SIA-CID"-SIA-CID protocol, string, the max. length is  

     32*/
    "addressingFormatType": "",  

    /*optional, address type, "ipaddress"-ip address, "hostname"-host name, string, the max. length is 32*/
    "hostName": "",  

    /*optional, host name, it is required when addressingFormatType values "hostname", string, the max. length is 64*/
    "ipAddress": "",  

    /*optional, device IPV4 address, it is required when addressingFormatType values "ipaddress", string, the max. length  

     is 32*/
    "ipv6Address": "",  

    /*optional, device IPV6 address, it is required when addressingFormatType values "ipaddress", string, the max. length  

     is 128*/
    "protocol": "",  

    /*optional, protocol type: "tcp"-TCP protocol, "ud"-UDP protocol, "multicast"-Multicast protocol, string, the max.  

     length is 32*/
    "portNo": "",  

    /*optional, device port No., integer*/
    "userName": "",  

    /*optional, user name, string, the sensitive information is encrypted, the max. length is 32*/
  }
}
```

```
"password": "",  
/*read-only, password, string, the sensitive information is encrypted, the max. length is 16. It will be applied only  
when configuring, and it is not returned when getting information*/  
"deviceChannel": "",  
/*required, added IoT channel No., integer*/  
"videoChannels": [1, 2],  
/*optional, added video channel No., including the cameras and IoT devices, array, and the array unit is integer*/  
"addAll": "",  
/*optional, whether adds all channels (including video and IoT channel) of device, boolean*/  
"inductiveDeviceType": "",  
/*optional, inductive device type, "inductiveType1"-inductive type 1, "inductiveType2"-inductive type 2. When getting  
added channel information according inductive type, if this node does not exist, all added device channels will be  
returned, string*/  
"zoneNum": ""  
/*optional, the number of added zones, it is valid when IOTProtocolType values "SIA-CID", integer*/  
}  
}
```

Remarks



For different IoT devices, the inputted parameters in IOTChannel message are different.

Example

IOTChannel Message of IoT Devices via Private Protocol

```
"IOTChannel":{  
    "IOTProtocolType": "HIKVISION",  
    "addressingFormatType": "ipaddress",  
    "ipAddress": "10.11.12.13",  
    "protocol": "tcp",  
    "portNo": 80,  
    "userName": "admin",  
    "password": "12345",  
    "deviceChannel": 1,  
    "videoChannels": [1, 2]  
}
```

Example

IOTChannel Message of IoT Devices via SIA-CID Protocol

```
"IOTChannel":{  
    "IOTProtocolType": "HIKVISION",  
    "addressingFormatType": "ipaddress",  
    "ipAddress": "10.11.12.13",  
    "protocol": "tcp",  
    "portNo": 80,  
    "userName": "admin",  
    "password": "12345",  
    "deviceChannel": 1,  
    "videoChannels": [1, 2],
```

```
"zoneNum": 1  
}
```

Example

IOTChannel Message of OPTEX IoT Devices

```
"IOTChannel":{  
    "IOTProtocolType": "OPTEX",  
    "addressingFormatType": "ipaddress",  
    "ipAddress": "10.11.12.13",  
    "protocol": "tcp",  
    "deviceChannel": 1  
}
```

Example

IOTChannel Message of Luminite IoT Devices

```
"IOTChannel":{  
    "IOTProtocolType": "Luminite",  
    "addressingFormatType": "ipaddress",  
    "ipAddress": "10.11.12.13",  
    "protocol": "tcp",  
    "portNo": 80,  
    "password": "12345",  
    "deviceChannel": 1  
}
```

Example

IOTChannel Message of GJD IoT Devices

```
"IOTChannel":{  
    "IOTProtocolType": "GJD",  
    "addressingFormatType": "ipaddress",  
    "ipAddress": "10.11.12.13",  
    "deviceChannel": 1  
}
```

F.87 JSON_IOTChannelEventCap

IOTChannelEventCap message in JSON format

```
{  
    "IOTChannelEventCap":{  
        /*required, event capability*/  
        "AccessController":{  
            /*optional, access controller event*/  
            "eventType":{  
                /*required, event type, array, the sub type is string*/  
                "@opt": ["authenticated", "openingDoor"]  
            },  
        },  
    },
```

```
"VideoIntercom":{  
/*optional, video intercom event*/  
  "eventType":{  
/*required, event type, string*/  
    "@opt": ["tampering", "duressAlarm"]  
  },  
},  
"GJD":{  
/*optional, GJD security control panel event*/  
  "eventType":{  
/*required, event type, string*/  
    "@opt": ["pirDetection", "tampering"]  
  },  
},  
"Luminite":{  
/*optional, Luminite security control panel*/  
  "eventType":{  
/*required, event type, string*/  
    "@opt": ["pirDetection", "pirTamper"]  
  },  
},  
"OPTEX":{  
/*optional, OPTEX security control panel*/  
  "eventType":{  
/*required, event type, string*/  
    "@opt": ["pcFn", "pcFr"]  
  },  
},  
"cameraDetector":{  
/*optional, detector evnet*/  
  "eventType":{  
/*required, event type, string*/  
    "@opt": ["wirelessTemperature", "wirelessPIR"]  
  },  
},  
"SecurityControlPanel":{  
/*optional, security control panel*/  
  "eventType":{  
/*required, event type, string*/  
    "@opt": ["zone", "host"]  
  },  
  "zoneNum":{  
/*required, the number of zones, integer*/  
    "@min": 1,  
    "@max": 16,  
    "#text": 1  
  },  
}  
}
```

F.88 JSON_IOTChannelList

IOTChannelList message in JSON format

```
{  
    "IOTChannelList": [  
        "IOTChannel"  
        /*optional, added IoT devices information, see details in  
         * JSON\_IOTChannel  
        */  
    ]  
}
```

See Also

[JSON_IOTChannel](#)

F.89 JSON_IOTChannelStatus

IOTChannelStatus message in JSON format

```
{  
    "IOTChannelStatus":{  
        /*optional, added IoT device status*/  
        "IOTChannel":{  
            /*optional, added IoT device information, see details in  
             * JSON\_IOTChannel  
             */  
        },  
        "channelName": "",  
        /*required, channel name, string, type, the max. length is 32*/  
        "onlineStatus": "",  
        /*required, online stauts, boolean*/  
        "channelDetectResult": "connecting,networkUnreachable,accessAbnormal,normal,guardFailed",  
        /*optional, channel detection status: "connecting"-connecting, "networkUnreachable"-unreachable network,  
        "accessAbnormal"-abnormal access, "normal"-normal, "guardFailed"-arming failed, string*/  
        "SecurityStatus":{  
            /*optional, security status*/  
            "passwordStatus": ""  
            /*optional, password status: notActivated-device is inactivated, risk-risky password, weak-weak password, medium-  
            medium password, strong-strong password, invalid-invalid status, string*/  
        },  
        "DeviceInfo":{  
            /*optional, device information*/  
            "deviceName": "",  
            /*optional, device name, string, the max. length is 32*/  
            "deviceType": "",  
            /*required, device type: "accessController"-access controller, "videoIntercom"-video intercom, "OPTEX"-OPTEX  
            security control panel, Luminite-Luminite security control panel, GJD-GJD security control panel, securityControlPanel-
```

```
security control panel, string*/  
    "model": "",  
/*optional, device model, string, the max. length is 128*/  
    "firmwareVersion": "",  
/*optional, firmware version, string, the max. length is 128*/  
    "firmwareReleasedDate": ""  
/*optional, firmware compiled date, string, the max. length is 128*/  
    }  
}  
}
```

F.90 JSON_IOTChannelStatusList

IOTChannelStatusList message in JSON format

```
{  
    "IOTChannelStatusList": [  
        {  
            "IOTChannelStatus":  
/*optional, added IoT device status, see details in JSON_IOTChannelStatus*/  
        }  
    ]  
}
```

See Also

[JSON_IOTChannelStatus](#)

F.91 JSON_IOTSourceDescription

IOTSourceDescription message in JSON format

```
{  
    "IOTSourceDescription":{  
/*optional, protocol information*/  
        "IOTProtocolType": "",  
/*required, IOT protocol type: "HIKVISION"-Hikvision device, "OPTEX"-OPTEX security control panel, "Luminite"-  
Luminite security control panel, "GJD"-GJD security control panel, "SIA-CID"-SIA-CID protocol, string, the max. length is  
32*/  
        "addressingFormatType": "",  
/*optional, address type, "ipaddress"-ip address, "hostname"-host name, string, the max. length is 32*/  
        "hostName": "",  
/*optional, host name, it is required when addressingFormatType values "hostname", string, the max. length is 64*/  
        "ipAddress": "",  
/*optional, device IPV4 address, it is required when addressingFormatType values "ipaddress", string, the max. length  
is 32*/  
        "ipv6Address": "",  
/*optional, device IPV6 address, it is required when addressingFormatType values "ipaddress", string, the max. length  
is 128*/  
        "protocol": "",  
/*optional, protocol type: "tcp"-TCP protocol, "ud"-UDP protocol, "multicast"-Multicast protocol, string, the max.
```

```
length is 32*/
    "portNo": "",
/*optional, device port No., integer*/
    "userName": "",
/*optional, user name, string, the sensitive information is encrypted, the max. length is 32*/
    "password": ""
/*optional, password, string, the sensitive information is encrypted, the max. length is 16*/
}
}
```

F.92 JSON_IOTSourceList

IOTSourceList message in JSON format

```
{
    "IOTSourceList": [
/*optional, source informaiton, array*/
        "IOTProtocolType": "",
/*required, IOT protocol type: "HIKVISION"-Hikvision device, "OPTEX"-OPTEX security control panel, "Luminite"-Luminite security control panel, "GJD"-GJD security control panel, "SIA-CID"-SIA-CID protocol, string, the max. length is 32*/
        "addressingFormatType": "",
/*optional, address type, "ipaddress"-ip address, "hostname"-host name, string, the max. length is 32*/
        "hostName": "",
/*optional, host name, it is required when addressingFormatType values "hostname", string, the max. length is 64*/
        "ipAddress": "",
/*optional, device IPV4 address, it is required when addressingFormatType values "ipaddress", string, the max. length is 32*/
        "ipv6Address": "",
/*optional, device IPV6 address, it is required when addressingFormatType values "ipaddress", string, the max. length is 128*/
        "portNo": "",
/*optional, device port No., integer*/
        "userName": "",
/*optional, user name, string, sensitive information will be encrypted, the max. length is 32*/
        "password": "",
/*optional, password, string, sensitive information will be encrypted, the max. length is 16*/
        "activeStatus": "",
/*required, active or not, boolean*/
        "deviceType": "",
/*optional, string, devive type: accessController-access controller, videoIntercom-video intercom, OPTEX-OPTEX security control panel, Luminite-Luminite security control panel, GJD-GJD security control panel, securityControlPanel-security control panel*/
        "macAddress": "",
/*optional, device MAC address, string, the max. length is 48*/
        "serialNumber": "",
/*optional, device serial No., string, the max. length is 48*/
        "firmwareVersion": "",
/*optional, firmware version (including compiled date), string, the max. length is 128*/
    ]
}
```

```
    }]
}
```

F.93 JSON_IOTSourceSupport

IOTSourceSupport message in JSON format

```
{
  "IOTSourceSupport":{
    /*required, supported source information*/
    "sourceNum": "",

    /*required, the number of protocols, integer*/
    "IOTSourceDescriptions": [
      /*optional, protocol informaiton, array*/
      "IOTProtocolType": "",

      /*required, IOT protocol type: "HIKVISION"-Hikvision device, "OPTEX"-OPTEX security control panel, "Luminite"-Luminite security control panel, "GJD"-GJD security control panel, "SIA-CID"-SIA-CID protocol, string, the max. length is 32*/
      "addressingFormatType":{

        /*optional, address type, "ipaddress"-ip address, "hostname"-host name, string, the max. length is 32*/
        "@opt": "ipaddress,hostname",
        "#text": "ipaddress"
      },
      "protocol":{

        /*optional, protocol type, "tcp, udp, muticast", string, the max. length is 32*/
        "@opt": "tcp,udp,micast",
        "#text": "tcp"
      },
      "portNo":{

        /*optional, device port No., integer*/
        "@min": 0,
        "@max": 65535,
        "#text": 80
      },
      "userName":{

        /*optional, user name, string, the max. length is 32*/
        "@min": 0,
        "@max": 32,
        "#text": "admin"
      },
      "password":{

        /*optional, password, string, the max. length is 16*/
        "@min": 0,
        "@max": 16,
        "#text": "12345"
      },
      "maxVideoChannelNum": ""
    }

    /*optional, the max. number of supported video channel, integer*/
    "zoneNum":{

      /*optional, number of connected zones, integer, it is valid when the IOTProtocolType values "SIA-CID"*/
    }
  }
}
```

```
    "@min": 1,
    "@max": 256,
    "#text": 1
},
}],
"DeviceInductiveType":{
/*optional, device inductive type*/
    "inductiveType1": ["accessController", "videoIntercom"],
/*optional, array, inductive type 1: accessController-access controller, videoIntercom-video intercom, OPTEX-OPTEX security control panel, Luminite-Luminite security control panel, GJD-GJD security control panel, securityControlPanel-security control panel, the sub type is string*/
    "inductiveType2": ["OPTEX", "Luminite", "GJD", "securityControlPanel"]
/*optional, array, inductive type 2: accessController-access controller, videoIntercom-video intercom, OPTEX-OPTEX security control panel, Luminite-Luminite security control panel, GJD-GJD security control panel, securityControlPanel-security control panel, the sub type is string*/
}
}
```

F.94 JSON_KeepAlive

KeepAlive message in JSON format

```
{
    "KeepAlive":{
        "enabled":"",
/*required, boolean, whether to enable network keepalive*/
        "interval":"",
/*required, integer, heartbeat interval, unit: second*/
        "addressingFormatType":"",
/*required, string, address types: "ipaddress"-IP address, "hostname"-host name, the max. length is 32*/
        "hostName":"",
/*optional, string, host name, which is required when addressingFormatType is "hostname", the max. length is 64*/
        "ipAddress":"",
/*optional, string, IPv4 address, which is required when addressingFormatType is "ipaddress", the max. length is 32*/
        "ipv6Address": "",
/*optional, string, IPv6 address, which is valid when addressingFormatType is "ipaddress", the max. length is 128*/
    }
}
```

F.95 JSON_KeepAliveCap

KeepAliveCap message in JSON format

```
{
    "KeepAliveCap":{
        "enabled": "true,false",
    }
}
```

```
/*required, boolean, whether to enable network keepalive*/
  "interval":{
    "@min": 60,
    "@max": 300,
    "@def": 300
  },
/*required, integer, heart beat interval which is 300s by default, unit: second*/
  "addressingFormatType":{
    "@opt": ["ipaddress", "hostname"],
  },
/*required, string, address types: "ipaddress"-IP address, "hostname"-host name, the max. length is 32*/
  "supportServerTest": true,
/*optional, boolean, whether to support server test*/
}
}
```

F.96 JSON_LaserOpticalAxis

JSON message about laser optical axis parameters.

```
{
  "LaserOpticalAxis": {
    "adjustmentEnabled": ,
/*required, boolean, whether it enables adjusting laser optical axis*/
    "sensitivity": ,
/*required, int, sensitivity*/
  }
}
```

F.97 JSON_LaserOpticalAxis_direction

JSON message about direction of adjusting laser optical axis

```
{
  "direction": """
/*required, string, direction of adjusting laser optical axis: "up"-up, "down"-down, "left"-left, "right"-right*/
}
```

F.98 JSON_LaserOpticalAxisCap

JSON message about capability of configuring laser optical axis

```
{
  "LaserOpticalAxisCap": {
    "adjustmentEnabled": {
/*required, boolean, whether it enables adjusting laser optical axis*/
      "@opt": [true, false]
    }
  }
}
```

```

},
"sensitivity": {
/*required, int, sensitivity*/
    "@min": 1,
    "@max": 100
}
"direction": {
/*required, string, direction of adjusting laser optical axis: "up"-up, "down"-down, "left"-left, "right"-right*/
    "@opt":["up","down","left","right"],
    "#text":"up"
}
}
}

```

F.99 JSON_LinkageChansCond

LinkageChansCond message in JSON format

```

{
  "LinkageChansCond": {
    "eventType": "",
    /*required, string, for access controller or video intercom, the EventType == AcsEvent_major_minor, for security
    control panel, the EventType == AlarmHost_zoneNo*/
    "monitorId": "",
    /*dep, string, camera ID, strlen=[32,64]. For access controller or video intercom, the format is: device serial
    No._DOOR_door No., for security control panel, the format is: device serial No._ALARM_security control panel No.*/
    "unlockType": ""
    /optional, unlock type, string, password-unlock by password, hijacking-hijacking unlock, card-unlock by swiping card,
    householder-unlock by householder, centerplatform-unlock by center platform, bluetooth-bluetooth unlock, qrcode-
    unlock by QR code, face-unlock by face, fingerprint-unlock by fingerprint*/
  }
}

```

F.100 JSON_LinkedTracking

JSON message about advanced parameters of linked tracking capture

```

{
  "LinkedTracking": {
    /*required, advanced parameters of linked tracking capture*/
    "trackingTakeoverEnabled": ,
    /*optional, boolean, whether to enable tracking takeover, it is disabled by default; the function is supported only
    when you select VCA resource as Smart Event*/
    "linkedTrackingCaptureEnabled": ,
    /*optional, boolean, whether to enable linked tracking capture, it is disabled by default; the function is supported only
    when you select VCA resource as Smart Event*/
    "captureThreshold": ,
    /*optional, int, capture threshold, it refers for the quality of face to trigger capture and alarm, range: [0,100], default
    value: 20*/
  }
}

```

```

    "Exposure":{
    /*optional, face exposure*/
        "exposureEnabled": ,
    /*optional, boolean, whether to enable face exposure, it is disabled by default*/
        "brightRef": ,
    /*optional, int, reference brightness of a face in the face exposure mode, range: [0,100]*/
        "expDurationTime": ,
    /*optional, int, minimum duration, it is the extra time the device keeps the face exposure lever after the face disappears from the scene; range: [0,60], unit: minute, default value: 5m*/
        },
    "RemoveDuplicate":{
    /*optional, remove duplicates of captured face pictures*/
        "enabled": ,
    /*optional, whether to enable removing duplicates of captured face pictures*/
        "threshold": ,
    /*optional, int, threshold of removing duplicates, when the picture similarity is larger than the value, the picture is duplicated*/
        "faceScore": ,
    /*optional, int, the face grading threshold for removing duplicates, only when the face grading is larger than the configured threshold, the face picture will be compared for removing duplicates, in case that low quality face pictures influence the comparison effect*/
        "updateTime": ,
    /*optional, int, face modal data updating time interval (the time interval between the modal data being added into the library and being deleted)*/
        }
    }
}

```

F.101 JSON_LinkedTrackingCap

JSON message about configuration capability of advanced parameters of linked tracking capture

```

{
    "LinkedTrackingCap":{
    /*required, advanced parameters capability of linked tracking capture*/
        "trackingTakeoverEnabled":{
    /*optional, boolean, whether to enable tracking takeover, it is disabled by default; the function is supported only when you select VCA resource as Smart Event*/
            "@opt": [true, false],
            "@def": false
        },
        "linkedTrackingCaptureEnabled":{
    /*optional, boolean, whether to enable linked tracking capture, it is disabled by default; the function is supported only when you select VCA resource as Smart Event*/
            "@opt": [true, false],
            "@def":false
        },
        "captureThreshold":{
    /*optional, int, capture threshold, it refers for the quality of face to trigger capture and alarm, range: [0,100], default value: 20*/
            }
    }
}

```

```
"@min":0,
"@max":100,
"@def":20
},
"ExposureCap":{
/*optional, face exposure capability*/
    "exposureEnabled":{
        "@opt": [true, false],
        "@def":false
    },
    "brightRef":{
        /*optional, int, reference brightness of a face in the face exposure mode, range: [0,100]*/
        "@min":0,
        "@max":100,
        "@def":50
    },
    "expDurationTime":{
        /*optional, int, minimum duration, it is the extra time the device keeps the face exposure lever after the face disappears from the scene; range: [0,60], unit: minute, default value: 5m*/
        "@min":1,
        "@max":60,
        "@def":5
    }
},
"RemoveDuplicateCap":{
/*optional, capability of removing duplicates of captured face pictures*/
    "enabled":{
        /*optional, whether to enable removing duplicates of captured face pictures*/
        "@opt": [true, false],
        "@def":false
    },
    "threshold":{
        /*optional, int, threshold of removing duplicates, when the picture similarity is larger than the value, the picture is duplicated*/
        "@min":70,
        "@max":100,
        "@def":88
    },
    "faceScore":{
        /*optional, int, the face grading threshold for removing duplicates, only when the face grading is larger than the configured threshold, the face picture will be compared for removing duplicates, in case that low quality face pictures influence the comparison effect*/
        "@min":20,
        "@max":60,
        "@def":20
    },
    "updateTime":{
        /*optional, int, face modal data updating time interval (the time interval between the modal data being added into the library and being deleted)*/
        "@min":0,
        "@max":300,
```

```
    "@def":30
    }
}
}
}
```

Remarks

Tracking takeover and linked tracking capture are two mutually exclusive modes for panorama tracking, they cannot be enabled at the same time.

F.102 JSON_logProTestResult

JSON message about preverification results of logging HDD.

```
{
  "requestURL":"",
  /*optional, string type, request URL*/
  "statusCode": ,
  /*required, integer type, status code*/
  "statusString":"",
  /*required, string type, status description*/
  "subStatusCode":"",
  /*required, string type, sub status code*/
  "errorCode": ,
  /*optional, integer type, error code, which corresponds to subStatusCode, this field is required when statusCode is not 1. The returned value is the transformed decimal number*/
  "errorMsg":"",
  /*optional, string type, error details, this field is required when statusCode is not 1*/
  "logProTestResult":{
    /*preverification results*/
    "enough": ,
    /*required, boolean, whether there is enough storage space in the HDD: "true"-enough, "false"-not enough*/
    "CoverDate":{
      /*dependent, it is returned when the field enough is "false"*/
      "beginTime":"",
      /*required, string, start time of HDD coverage, which should in ISO 8601 format,the maximum size is 32 bytes*/
      "endTime":"",
      /*required, string, end time of HDD coverage, which should in ISO 8601 format,the maximum size is 32 bytes*/
      "coverSpace":
        /**required, integer, size of the covered HDD, unit: GB*/
        }
    }
  }
```

F.103 JSON_MatchList

JSON Message about log search results

```
{
    "requestURL": "",
    /*optional, string, request URL*/
    "statusCode": ,
    /*required, int, status code*/
    "statusString": "",
    /*required, string, status description*/
    "subStatusCode": "",
    /*required, string, sub status code*/
    "errorCode": ,
    /*optional, int, error code, which corresponds to the subStatusCode; this node is required when the value of statusCode is not 1*/
    "errorMsg": "",
    /*optional, string, error description, which corresponds to errorCode*/
    "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, the maximum length is 32 bytes*/
    "responseStatusStrg": "",
    /*required, string, search status: "OK"-search completed, "FAILED"-search failed, "MORE"-more data waiting to be searched, "PARM ERROR"-incorrect parameters, "INVALID TIME"-invalid time, "TIMEOUT"-timeout, "NO MATCHES"-no matched data, the maximum length is 32 bytes*/
    "numOfMatches": ,
    /*required, integer32, number of matched data records*/
    "matchList": [
        /*optional, array, list of matched data records*/
        "searchMatchItem": {
            /*optional, single matched record*/
            "logDescriptor": {
                /*required, log description*/
                "metaID": "",
                /*required, string, log type string, log type includes "Alarm"-alarm log, "Exception"-exception log, "Operation"-operation log, and "Information"-additional information; e.g., "log.std-cgi.com/Alarm", the maximum length is 64 bytes*/
                "startTime": "",
                /*required, string, log time, ISO8601 time format, the maximum length is 32 bytes*/
                "localId": "",
                /*optional, string, local channel, different logs are corresponding to different local channel types, the maximum length is 32 bytes*/
                "paraType": "",
                /*optional, string, function modules that generate the logs: "videoout, image, encode, network, alarm, exception, decoder, rs232, preview, security, datatime, frametype"; the maximum length is 32 bytes*/
                "userName": "",
                /*optional, string, user name that performs current search task, the maximum length is 32 bytes*/
                "ipAddress": "",
                /*optional, string, device IP address, the maximum length is 32 bytes*/
                "logInfo": ""
            }
        }
    ]
}
```

```
checked, 12-network status checked*/
    "remarks": ""
/*optional, operation remarks*/
}
}
}]
}
```

F.104 JSON_MoveAutoTracking

JSON message about status of moving object tracking

```
{
  "MoveAutoTracking": {
    "enabled": true
  }
}
```

F.105 JSON_OnlineUpgradeParameter

OnlineUpgradeParameter message in JSON format

```
{
  "OnlineUpgradeParameter": {
    /*required, online upgrade parameters*/
    "autoDownloadPackage": false,
    /*required, whether enables automatic download of upgrade package, boolean, it values "false" by default*/
    "timingUpgrade": false,
    /*optional, whether enable scheduled upgrade, boolean, it values "false" by default*/
    "upgradeTime": "2023-01-01T00:00:00Z",
    /*optional, upgrade time, corrects to minute, and second is 0 by default, ISO8601 format, string type, it is valid when timingUpgrade values "true"*/
  }
}
```

F.106 JSON OSD

OSD message in JSON format

```
{
  "OSD": {
    /*required, OSD information*/
    "overlayItem": "deviceName-deviceName, cardNo-card number, eventName-event name, name-name, identityCard-ID card, privacyInfo-private information"
  }
}
```

```

    "characterEncoding":"",
/*required, character encoding, string*/
    "fontSize": "",
/*required, string, font size: big-large, medium-medium, small-small*/
    "fontColor": "",
/*required, font color, hexBinary*/
    "overlayType": "",
/*required, overlay type: "flip, scroll", string*/
    "displayTime": "",
/*required, display time, unit; second, integer*/
    "PrivacyInfo": [
/*optional, private information list, it is required when the value of overlayItem contains "privacyInfo", array*/
        "content": "",
/*required, private information, string, the max. length is 16*/
        ],
    "normalizedScreenSize":{
/*required, read-only, normalized coordinates*/
        "normalizedScreenWidth": "",
/*required, normalized width, integer*/
        "normalizedScreenHeight": ""
/*required, normalized height, integer*/
        },
    "Rect":{
/*required, OSD position*/
        "height": "",
/*required, height, float*/
        "width": "",
/*required, width, float*/
        "x": "",
/*required, X-coordinate, float*/
        "y": ""
/*required, Y-coordinate, float*/
        }
    }
}

```

F.107 JSON_OSDCap

OSDCap message in JSON format

```

{
    "OSDCap":{
/*required, OSD information*/
        "overlayItem":{
/*required, string, overlaid item, supports multiple items, each item is separated by comma; deviceName-device
name, cardNo-card number, eventName-event name, name-name, identityCard-ID card, privacyInfo-private
information*/
            "@opt": "deviceName,cardNo,eventName,name,identityCard,privacyInfo",
            "#text": "deviceName"
        }
    }
}

```

```
"characterEncoding":{  
/*required, character encoding, string*/  
    "@opt": "GB2312,Latin-1",  
    "#text": "GB2312"  
},  
"fontSize":{  
/*required, string, font size: big-large, medium-medium, small-small*/  
    "@opt": "big,medium,small",  
    "#text": "big"  
},  
"fontColor":{  
/*required, font color, hexBinary*/  
    "@opt": "FFFFFF",  
},  
"overlayType":{  
/*required, overlay type: "flip, scroll", string*/  
    "@opt": "flip,scroll",  
    "#text": "flip"  
},  
"displayTime":{  
/*required, display time, unit: second, integer*/  
    "@min": 1,  
    "@max": 120,  
    "#text": 15  
},  
"PrivacyInfo": [{  
/*optional, private information list, it is required when the value of overlayItem contains "privacyInfo", array*/  
    "size": 3,  
/*required, the max. number of items, integer*/  
    "content":{  
/*required, private information, string, the max. length is 16*/  
        "@min": 1,  
        "@max": 16,  
        "#text": "12345"  
    },  
}],  
"Rect":{  
/*required, OSD position*/  
    "height":{  
/*required, height, float*/  
        "@min": 0.000,  
        "@max": 1.000,  
        "#text": 0.311  
    },  
    "width":{  
/*required, width, float*/  
        "@min": 0.000,  
        "@max": 1.000,  
        "#text": 0.311  
    },  
    "x":{  
/*required, X-coordinate, float*/  
    }  
}
```

```
    "@min": 0.000,
    "@max": 1.000,
    "#text": 0.311
},
"y":{
/*required, Y-coordinate, float*/
    "@min": 0.000,
    "@max": 1.000,
    "#text": 0.311
},
}
}
```

F.108 JSON_PeopleCounting_CollectionDescriptionCap

CollectionDescriptionCap message in JSON format.

```
{
"CollectionDescriptionCap":{
    "timeRange": "",
/*required, supported time range for people counting replenishment, unit: day, integer*/
    "timeSpanMaxNum": "",
/*required, the maximum number of time buckets supported for one time search integer*/
    "resultMaxNum": "",
/*required, the maximum number of items can be searched, integer*/
    "channelMaxNum": ""
/*required, the maximum number of channels supported for one time search, integer*/
}
}
```

F.109 JSON_PeopleCounting_CollectionDescription

CollectionDescription message in JSON format

```
{
"CollectionDescription":{
    "searchID": "",
/*required, string, unique search ID, the content of searchID remains unchanged when search condition remains unchanged.*/
    "searchResultPosition": "",
/*required, initial position of search result list, integer 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*/
    "maxResults": "",
/*required, number of matched records per search, integer type*/
    "timeSpanList": [
/*required, time interval list*/
        "startTime": "",
/*required, start time, ISO8601_time, string*/
    ]
}
}
```

```
        "endTime": "",  
        /*required, end time, ISO8601_time, string*/  
    }],  
    "channels": "",  
    /*optional, sub type, array, integer32, related channel No., array. If the channel information is not specified, it  
    indicates searching all channels*/  
}
```

F.110 JSON_PeopleCounting_CollectionResult

CollectionResult message in JSON format

```
{  
    "CollectionResult":{  
        "responseStatusStrg": "",  
        /*required, searching status: OK- Searching ended, NO MATCHES-No matched data found, MORE-Search again for  
        more results, string, the max. length is 32, {dep if errcode == 1 && errMsg == ok}*/  
        "numOfMatches": "",  
        /*required, returned number of results for current search, integer32, {dep if errcode == 1 && errMsg == ok}*/  
        "totalMatches": "",  
        /*required, total number of matched results, integer32, {dep if errcode == 1 && errMsg == ok}*/  
        "targets": [],  
        /*optional, people gathering result*/  
        "startTime": "",  
        /*required, start time of people gathering*/  
        "endTime": "",  
        /*required, end time of people gathering*/  
        "channel": "",  
        /*required, integer, channel No.*/  
        "enter": "",  
        /*required, integer, the number of people entered*/  
        "exit": "",  
        /*required, integer, the number of people exited*/  
        "pass": "",  
        /*required, integer, the number of people passed by*/  
    }  
}
```

F.111 JSON_Reconstruction_ProgressLists

JSON message about disk data reconstruction progress

```
{  
    "ProgressLists": [  
        {"diskID": 1,  
        /*required, int, disk ID*/  
        "progress": 100,
```

```
/*required, int, disk data reconstruction progress*/
    "status": 1
/*required, int, status: 0-reconstructing, 1-device exception (applying for memory failed/opening files failed/deleting files failed), 2-abnormal disconnection, 3-reconstructed*/
}
}
```

F.112 JSON_RemarkList

JSON message about remark information list of videos

```
{
  "RemarkList": [
    {
      "trackId": "",
      /*required, file No.*/
      "remarks": "",
      /*required, string, file remarks, which should be encrypted*/
      "recordName":"",
      /*optional, string, file name of video, audio, or picture*/
      "recorderCode":"",
      /*optional, string, ID of device that collects data, which should be encrypted*/
      "policeCode":""
    }
  ]
}
```

F.113 JSON_resourceStatistics

resourceStatistics message in JSON format

```
{
  "inputBandwidth": ,
  /*required, input bandwidth, unit: kbps, float type, corrects to one decimal*/
  "outputBandwidth": ,
  /*required, output bandwidth, unit: kbps, float type, corrects to one decimal*/
  "StreamingInformation": [
    /*optional, stream information, array*/
    "moduleType": "",
    /*required, module type: "remoteStreaming"-remote streaming, "remotePlayBack"-remote playback, string type, the maximum length is 32 bits*/
    "ipAddress": "",
    /*required, device IPv4 address, string type, the maximum length is 32 bits*/
    "ipv6Address": "",
    /*optional, device IPv6 address, string type, the maximum length is 128 bits*/
    "bandwidth":
    /*required, bandwidth, unit: kbps, float type, corrects to one decimal*/
  ],
  "inputPictureBandwidth": ,
  /*optional, picture input bandwidth (including all interfaces of inputting binary picture data), unit: kbps, float type,
}
```

```
corrects to one decimal*/
"outputPictureBandwidth": ,
/*optional, picture output bandwidth (including all interfaces of returning binary picture data), unit: kbps, float type,
corrects to one decimal*/
}
```

F.114 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.115 JSON_Result

Result message in JSON format

```
{
  "Result":{
    "channels": [1, 2],
    /*optional, integer, linked video channel No.*/
    "IoTChannels": [1, 2]
    /*optional, integer, linked IoT channel NO.*/
  }
}
```

F.116 JSON_SearchCondition

JSON message about condition information of log search

```
{  
    "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, the maximum length is 32 bytes*/  
    "timeSpanList": {  
        /*required, list of time periods*/  
        "timeSpan": {  
            /*required, time periods*/  
            "startTime": "",  
            /*required, string, start time of search, ISO8601 time format, the maximum length is 32 bytes*/  
            "endTime": ""  
            /*required, string, end time of search, ISO8601 time format, the maximum length is 32 bytes*/  
        }  
    },  
    "metaID": "",  
    /*required, string, log type string, log type includes "Alarm"-alarm log, "Exception"-exception log, "Operation"-operation log, and "Information"-additional information; e.g., "log.std-cgi.com/Alarm", the maximum length is 64 bytes*/  
    "searchResultPostion": ,  
    /*required, integer32, the start position of search result in result list*/  
    "maxResults": ,  
    /*required, integer32, the maximum number of records supported in this search*/  
    "wildcard": "",  
    /*optional, string, wildcard, which is used for fuzzy search, the maximum length is 48 bytes*/  
    "operationType": ,  
    /*optional, operation type, it is valid only when metaID is set to "Operation": 0-all, 1-software started, 2-software stopped, 3-configuration page entered, 4-parameters edited, 5-file search, 6-device connected, 7-information collection started, 8-information collection ended, 9-devices disconnected, 10-password changed, 11-disk space checked, 12-network status checked*/  
}
```

F.117 JSON_SecurityEmail

SecurityEmail message in JSON format

```
{  
    "SecurityEmail":{  
        "SecurityInformation": [{}  
        /*required, recovery email information*/  
        "emailAddress": "",  
        /*required, string type, email address, the sensitive information should be encrypted*/  
    },
```

```
 },  
 }
```

F.118 JSON_SecurityEmailCap

SecurityEmailCap message in JSON format

```
{  
  "SecurityEmailCap":  
  /*required, recovery email configuration capability*/  
  {  
    "emailAddressNum": ,  
    /*required, number of recovery emails*/  
    "emailAddress":  
    /*required, string type, recovery email information*/  
    {  
      "@min": 1,  
      "@max": 128,  
    },  
  }  
}
```

F.119 JSON_SecurityEmailQrCode

SecurityEmailQrCode message in JSON format

```
{  
  "SecurityEmailQrCode":  
  /*optional, QR code of recovery email*/  
  {  
    "SecurityInformation": [{  
      /*required, security information, the sensitive information should be encrypted*/  
      "emailAddress": "",  
      /*required, string type, email address, the sensitive information should be encrypted*/  
      "qrCode": "",  
      /*required, string type, QR code, the maximum size is 1024 bytes, the sensitive information should be encrypted*/  
      {}  
      "serverEmailAddress": "",  
      /*optional, string type, manufacturer service email address, the sensitive information should be encrypted*/  
      "isDisclaimerDisplay":  
      /*optional, boolean type, whether to display disclaimer*/  
      {}  
    }  
  }
```

F.120 JSON_SmartOverlap

SmartOverlap message in JSON format

```
{  
  "SmartOverlap":{  
    "enabled": ,  
    /*required, boolean, whether to enable the stream*/  
    "streamType": ,  
    /*required, integer, stream type: 1-main stream, 2-sub-stream, 3-third stream*/  
    "bkgImageOverlapEnabled":  
    /*required, boolean, whether to overlay rule frame and target frame on the background picture*/  
  }  
}
```

F.121 JSON_SmartOverlapCap

SmartOverlapCap capability message in JSON format

```
{  
  "SmartOverlapCap":{  
    "enabled":{  
      /*required, boolean, whether to enable the stream*/  
      "@opt": [ true, false]  
    },  
    "streamType":{  
      /*required, integer, stream type: 0-disable, 1-main stream, 2-sub-stream, 3-third stream*/  
      "@opt": [2]  
    },  
    "bkgImageOverlapEnabled":{  
      /*required, boolean, whether to overlay rule frame and target frame on the background picture*/  
      "@opt": [true, false]  
    },  
    "supportEventType":{  
      /*required, string, supported event type: "fieldDetection"-intrusion, "attendedBaggage"-object removal,  
      "unattendedBaggage"-unattended baggage, "regionExiting"-region exiting, "regionEntrance"-region entrance,  
      "lineDetection"-line crossing*/  
      "@opt": ["fieldDetection", "attendedBaggage", "unattendedBaggage", "regionExiting", "regionEntrance",  
      "lineDetection"]  
    }  
  }  
}
```

F.122 JSON_SourceCapabilities

SourceCapabilities message in JSON format

```
{  
    "SourceCapabilities":{  
        /*required, the supported source capabilities*/  
        "IOTChannelNum": "",  
        /*required, the number of IoT channels,integer*/  
        "videoChannelNum": "",  
        /*optional, the number of video channels, integer*/  
        "accessChannels": [{  
            /*optional, array, added channel No.*/  
            "channel": "",  
            /*required, channel No., integer*/  
            "type": "",  
            /*required, channel type: "video"-video channel, "IOT"-IoT channel, string*/  
        }]  
    }  
}
```

F.123 JSON_SSDCapacity

JSON message about SSD storage quotas

```
{  
    "SSDCapacity":{  
        "total":70.5,  
        /*required, float, SSD total capacity, unit: GB*/  
        "systemReserved":10.5,  
        /*required, float, reserved capacity, unit: GB*/  
        "free":10.5,  
        /*required, float, remained capacity, unit: GB*/  
        "faceLib":10.5,  
        /*optional, float, face picture library capacity, unit: GB*/  
        "strangerLib":10.5,  
        /*optional, float, strangers library capacity, unit: GB*/  
        "faceFrequencyLib":10.5,  
        /*optional, float, capacity of people frequency statistics library, unit: GB*/  
        "alarmCache":10.5,  
        /*optional, float, alarm cache capacity, unit: GB*/  
        "faceIndex":10.5,  
        /*optional, float, face picture index capacity, unit: GB*/  
    }  
}
```

F.124 JSON_SSDFormatStatus

JSON message about progress of SSD file system formatting

```
{  
    "SSDFormatStatus": {
```

```
"formatting": "false",
/*required, read-only, boolean, whether the device is formatting*/
"percent":100
/*required, read-only, int, device formatting progress, range: [0,100]*/
}
}
```

F.125 JSON_SSDUpgrade

JSON message about upgrade status of SSD file system

```
{
"SSDUpgrade": {
"upgrading": "false"
/*required, read-only, boolean, whether the device has been upgraded*/
}
}
```

F.126 JSON_SSDUpgradeStatus

JSON message about progress of SSD file system upgrade

```
{
"SSDUpgradeStatus": {
"upgrading": "false",
/*required, read-only, boolean, whether the SSD file system is upgrading*/
"percent":100
/*required, read-only, int, device upgrade progress, range: [0,100]*/
}
}
```

F.127 JSON_SSD_UpgradeStatus

JSON message about SSD firmware upgrade progress

```
{
"UpgradeStatus": {
"upgrading": false,
/*required, read-only, boolean, whether the SSD firmware is upgrading*/
"percent":100
/*required, read-only, int, SSD firmware upgrade progress, range: [0,100]; this node is valid only when the value of
upgrading is true*/
}
}
```

F.128 JSON_SyncStatus

JSON message about HDD data sync status

```
{  
  "SyncStatus":{  
    "synching":true,  
    /*required, boolean, whether the HDD is in sync*/  
    "percent":90  
    /*optional, int, sync progress, range: [0,100]%*/  
  }  
}
```

F.129 JSON_CollectionDescription

CollectionDescription message in JSON format

```
{  
  "CollectionDescription":{  
    "searchID": "",  
    /*required, string, unique search ID, the content of searchID remains unchanged when search condition remains unchanged.*/  
    "searchResultPosition": "",  
    /*required, initial position of search result list, integer 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*/  
    "maxResults": "",  
    /*required, number of matched records per search, integer type*/  
    "timeSpanList": [  
      /*required, time interval list*/  
      "startTime": "",  
      /*required, start time, ISO8601_time, string*/  
      "endTime": "",  
      /*required, end time, ISO8601_time, string*/  
    ],  
    "channels": "",  
    /*optional, sub type, array, integer32, related channel No., array. If the channel information is not specified, it indicates searching all channels*/  
    "alarmLevel": "TMA",  
    /*required, string, alarm level: TMA-thermometry alarm, TMPA-thermometry pre-alarm*/  
  }  
}
```

F.130 JSON_Temperature_CollectionDescriptionCap

CollectionDescriptionCap message in JSON format

```
{
    "CollectionDescriptionCap": {
        "timeRange": 15,
        /*required, supported time range for replenishment, unit: day, integer */
        "timeSpanMaxNum": 1,
        /*required, the maximum number of time buckets supported for one time search, integer*/
        "resultMaxNum": 100,
        /*required, the maximum number of items can be searched, integer*/
        "channelMaxNum": 1,
        /*required, the maximum number of channels supported for one time search, integer*/
        "alarmLevel": {
            "@opt": "TMA,TMPA"
        }
    }
}
```

F.131 JSON_CollectionResult

CollectionResult message in JSON format

```
{
    "CollectionResult": {
        "responseStatusStrg": "",
        /*required, searching status: OK- Searching ended, NO MATCHES-No matched data found, MORE-Search again for more results, string, the max. length is 32, {dep if errcode == 1 && errMsg == ok}*/
        "numOfMatches": "",
        /*required, returned number of results for current search, integer32, {dep if errcode == 1 && errMsg == ok}*/
        "totalMatches": "",
        /*required, total number of matched results, integer32, {dep if errcode == 1 && errMsg == ok}*/
        "targets": [
            /*optional, statistics result*/
            {
                "time": "2017-08-25T11:34:59+08:00",
                /*required, statistic time*/
                "channel": "",
                /*required, integer, channel No.*/
                "Region": [
                    /*required, float, X-coordinate, ranges from 0.000 to 1*/
                    "x": "",
                    /*required, float, Y-coordinate, ranges from 0.000 to 1*/
                    "y": ""
                ],
                "thermometryUnit": "",
                /*required, string, temperature unit: celsius, fahrenheit, kelvin*/
                "ruleTemperature": ,
                /*required, float, themometry rule*/
                "currTemperature": ,
                /*required, float, current temperature*/
                "ruleCalibType": "",
                /*required, string, rule calibration type: point,line,region*/
            }
        ]
    }
}
```

```

    "ruleType": "",  

/*required, string, rule: "highest temp is higher than"-the maximum temperature is higher than, "lowest temp is  

higher than"-the minimum temperature is higher than, "average temp is higher than"-the average temperature is  

higher than, "temp diff is higher than"-temperature difference is higher than, "highest temp is lower than"-the  

maximum temperature is lower than, "lowest temp is lower than"-the minimum temperature is lower than, "average  

temp is lower than"-the average temperature is lower than, "temp diff is lower than"-temperature difference is lower  

than*/  

    "MaximumTemperaturePoint":{  

/*optional, the maximum temperature point*/  

    "x": "",  

/*required, float, X-coordinate, ranges from 0.000 to 1*/  

    "y": ""  

/*required, float, Y-coordinate, ranges from 0.000 to 1*/  

},  

    "AbsoluteHigh":{  

/*optional, absolute height*/  

    "elevation": "",  

/*required, float, elevation*/  

    "azimuth": "",  

/*required, float, azimuth*/  

    "absoluteZoom": ""  

/*required, float, absolute zoom*/  

},  

    "ruleID": "",  

/*required, integer, rule ID*/  

    "presetNo": "",  

/*required, integer, preset ID*/  

    "visibleLightURL": "",  

/*optional, string, visible light picture URL*/  

    "thermalURL": "",  

/*optional, string, thermal picture URL*/  

    "thermalInfoURL": "",  

/*optional, string, thermal additional information URL*/  

}
}
}

```

F.132 JSON_TestDescription

JSON message about diagnostic server test parameters

```

    "TestDescription": {  

/*required, diagnostic server test parameters*/  

    "protocol": "FTP",  

/*required, string, transmission protocol: "FTP,SFTP"*/  

    "addressingFormatType": "ipaddress",  

/*optional, string, address type: "ipaddress" (IP address), "hostname" (host name)*/  

    "hostName": "www.baidu.com",  

/*dependent, string, host name, the maximum length is 64 bytes; it is valid when the value of addressingFormatType

```

```
is "hostname"*/
  "ipVersion": "ipv4",
/*dependent, string, IP version: "ipv4", "ipv6"; it is valid when the value of addressingFormatType is "ipaddress"*/
  "ipV4Address": "10.17.132.254",
/*dependent, string, IPv4 address; it is valid when the value of ipVersion is "ipv4", and addressingFormatType is "ipaddress"*/
  "ipV6Address": "fe80::884a:67cb:9b67:b3a0%15",
/*dependent, string, IPv6 address; it is valid when the value of ipVersion is "ipv6", and addressingFormatType is "ipaddress"*/
  "portNo": 20,
/*optional, int, port No., value range: [1,65535]*/
  "userName": "admin",
/*string, write-only, user name, the maximum length is 32 bytes*/
  "password": "admin"
/*string, write-only, password, the maximum length is 16 bytes*/
}
```

F.133 JSON_TestResult

TestResult message in JSON format

```
{
  "TestResult":{
    "status":"ok",
/*required, string, server status: "ok"-normal, "connectServerFail"-server connection failed
  }
}
```

F.134 JSON_TimeSearchCond

Message about recording time search condition in JSON format

```
{
  "TimeSearchCond": {
    "channelID": ,
/*required, int, channel No.*/
    "streamType": ""
/*required, string, stream type, "main,sub,adaptive"*/
  }
}
```

F.135 JSON_TimeSearchResult

Message about recording time search result in JSON format

```
{
  "TimeSearchResult": {
    "channelID": ,
    /*required, int, channel No.*/
    "streamType": "",
    /*required, string, stream type, "main,sub,adaptive"*/
    "hasRecord": ,
    /*required, boolean, whether video exists, startTime and endTime are valid only when the value of hasRecord is "true"*/
    "startTime": "",
    /*optional, string, recording start time, e.g., 2010-01-01T00:00:00+08:00, this field is required when the value of hasRecord is "true"*/
    "endTime": "",
    /*optional, string, recording end time, e.g., 2010-01-01T00:00:00+08:00, this field is required when the value of hasRecord is "true"*/
  }
}
```

F.136 JSON_TrafficMonitor

TrafficMonitor message in JSON format

```
{
  "TrafficMonitor": {
    "enabled": ,
    /*required, boolean, whether to enable traffic monitoring*/
    "packageType": "",
    /*required, string, data plan types: "day"-daily plan, "month"-monthly plan, "year"-annual plan */
    "trafficValue": ,
    /*required, float, data value, which should be accurate to three decimal places, unit:MB */
    "overrunOperation": "",
    /*optional, string, operations when the data exceeds: "brokenNetworkAndAlert"-turn off the cellular network and alert users, "alert"-notify users */
    "alertType": [
      "beep",
      "email",
      "SMS"
    ],
    /*required, string, notification methods: "beep"-audio warning, "email"-send email, "SMS"-send message, "center"-upload to center*/
    "alertValue": ,
    /*required, integer, alarm value(%)*/
    "startDay": ,
    /*optional, integer, the start day of the monthly plan, which is valid only when packageType is "month"*/
  }
}
```

F.137 JSON_TrafficMonitorCap

TrafficMonitorCap message in JSON format

```
{  
  "TrafficMonitorCap":{  
    "enabled":"" ,  
    /*required, boolean, whether to enable traffic monitoring*/  
    "packageType":{  
      "@opt": [  
        "day",  
        "month",  
        "year"  
      ]  
    },  
    /*required, string, data plan types: "day"-daily plan, "month"-monthly plan, "year"-annual plan*/  
    "overrunOperation":{  
      "@opt": [  
        "brokenNetworkAndAlert",  
        "alert"  
      ]  
    },  
    /*optional, string, operations after the data exceeds: "brokenNetworkAndAlert"-turn off the cellular network and alert  
    users, "alert"-notify users */  
    "alertType":{  
      "@opt": [  
        "beep",  
        "email",  
        "SMS",  
        "center"  
      ]  
    },  
    /*required, string, notification methods: "beep"-audio warning, "email"-send email, "SMS"-send message, "center"-  
    upload to center*/  
    "alertValue":{  
      "@min": ,  
      "@max":  
    },  
    /*required, integer, alarm value(%)*/  
    "supportTrafficMonitorStatus":  
    /*optional, boolean, whether it supports getting the current network data usage*/  
  }  
}
```

F.138 JSON_TrafficMonitorStatus

TrafficMonitorStatus message in JSON format

```
{
    "TrafficMonitorStatus": {
        "remainTrafficValue": ,
        /*required, float, remaining data, which should be accurate to three decimal places, unit:MB*/
        "trafficValue": ,
        /*required, float, data value, which should be accurate to three decimal places, unit:MB*/
        "todayUsedTrafficValue": ,
        /*required, float, today's data usage, which should be accurate to three decimal places, unit:MB*/
        "todayUsedUpstreamTrafficValue": ,
        /*optional, float, today's upstream data usage, which should be accurate to three decimal places, unit:MB*/
        "todayUsedDownstreamTrafficValue": ,
        /*optional, float, today's downstream data usage, which should be accurate to three decimal places, unit:MB*/
        "monthUsedTrafficValue": ,
        /*optional, float, data usage of this month, which should be accurate to three decimal places, unit:MB*/
        "monthUsedUpstreamTrafficValue": ,
        /*optional, float, upstream data usage of this month, which should be accurate to three decimal places, unit:MB*/
        "monthUsedDownstreamTrafficValue": ,
        /*optional, float, downstream data usage of this month, which should be accurate to three decimal places, unit:MB*/
        "yearUsedTrafficValue": ,
        /*optional, float, data usage of this year, which should be accurate to three decimal places, unit:MB*/
        "yearUsedUpstreamTrafficValue": ,
        /*optional, float, upstream data usage of this year, which should be accurate to three decimal places, unit:MB*/
        "yearUsedDownstreamTrafficValue": ,
        /*optional, float, downstream data usage of this year, which should be accurate to three decimal places, unit:MB*/
        "enabled": ,
        /*optional, boolean, whether traffic monitoring is enabled*/
        "packageType": "",
        /*optional, string, data plan types: "day"-daily plan, "month"-monthly plan, "year"-annual plan*/
        "alertValue": ,
        /*optional, integer, alarm value(%)*/
    }
}
```

F.139 JSON_unitConfig

unitConfig message in JSON format

```
{
    "enabled": "",
    /*required, enable unit unifying configuration or not, boolean type*/
    "temperatureRange": "",
    /*optional, string type, temperature unit: degreeCentigrade-Centigrade (°C), degreeFahrenheit-Fahrenheit (°F) , degreeKelvin-Kelvin (K), the default value is "degreeCentigrade"*/
    "distanceUnit": ""
    /*optional, string type, distance unit: centimeter, meter, feet, the default value is "centimeter"*/
}
```

Remarks

- When the **enabled** values "false", it indicates the unit unifying function is disabled, the unit can be configured in thermometry basic settings, and the configured unit has no influence on the unit in system settings.
- When the **enabled** values "false", the thermometry unit configuration in applied URL remains unchanged and takes effect; while when the **enabled** values "true" and the thermometry unit is different with the system unit, the thermometry unit configuration in applied URL takes no effect and the error code will be returned.

F.140 JSON_unitConfigCap

unitConfigCap message in JSON format

```
{  
    "enabled":"true,false",  
    /*required, enable unit unified configuration or not, boolean type*/  
    "temperatureRange":{  
        /*optional, string type, temperature unit: degreeCentigrade-Centigrade (°C), degreeFahrenheit-Fahrenheit (°F) ,  
        degreeKelvin-Kelvin (K), the default value is "degreeCentigrade"*/  
        "@opt":"degreeCentigrade,degreeFahrenheit,degreeKelvin"  
    },  
    "distanceUnit":{  
        /*optional, string type, distance unit: centimeter, meter, feet, the default value is "centimeter"*/  
        "@opt":"centimeter,meter,feet"  
    }  
}
```

F.141 JSON_USB

JSON message about USB parameters

```
{  
    "USB": {  
        "usbMode": ""  
        /*optional, string, USB mode: "UVC", "NCM"*/  
    }  
}
```

F.142 JSON_USBCap

JSON message about USB parameters configuration capability

```
{  
    "USBCap": {
```

```
"usbMode": {  
    /*optional, string, USB mode: "UVC", "NCM"*/  
    "@opt":["UVC","NCM"],  
    "#text":"UVC"  
}  
}  
}
```

F.143 JSON_User

User message in JSON format

```
{  
    "User":{  
        "userName": "",  
        /*required, string, double verification user name*/  
        "password": ""  
        /*required, string, double verification password*/  
    }  
}
```

F.144 JSON_UserCap

UserCap message in JSON format

```
{  
    "UserCap":{  
        "userMaxNumber": "",  
        /*required, the maximum number of supported double verification users*/  
        "userNameLen":{  
            /*required, the length of double verification user name*/  
            "@min": "",  
            "@max": "",  
            },  
        "passwordLen":{  
            /*required, the length of double verification user password*/  
            "@min": ,  
            "@max": ,  
            }  
    }  
}
```

F.145 JSON_UserList

UserList message in JSON format

```
{  
  "UserList": [{  
    "id":,  
    /*required, integer, user ID*/  
    "userName":""  
    /*required, string, double verification user name*/  
  }]  
}
```

F.146 JSON_UserPermission

UserPermission message in JSON format

```
{  
  "UserPermission":{  
    "localPlayback":{  
      /*optional, local playback permission*/  
      "enable": "",  
      /*required, enables permission or not, boolean*/  
      "channelPermission": [  
        /*optional, channel permission*/  
        "channelID":,  
        /*required, channel ID, integer*/  
        "enable": ""  
        /*required, enables permission or not, boolean*/  
      ]  
    },  
    "localBackup":{  
      /*optional, local backup permission*/  
      "enable": "",  
      /*required, enables permission or not, boolean*/  
      "channelPermission": [  
        /*optional, channel permission*/  
        "channelID":,  
        /*required, channel ID, integer*/  
        "enable": ""  
        /*required, enables permission or not, boolean*/  
      ]  
    },  
    "remotePlayback":{  
      /*optional, remote playback and download permission*/  
      "enable": "",  
      /*required, enables permission or not, boolean*/  
      "channelPermission": [  
        /*optional, channel permission*/  
        "channelID":,  
        /*required, channel ID, integer*/  
        "enable": ""  
        /*required, enables permission or not, boolean*/  
      ]  
    }  
  }  
}
```

```
    }
}
}
```

F.147 JSON_UserPermissionCap

UserPermissionCap message in JSON format

```
{
  "UserPermissionCap": {
    "localPlayback": {
      /*required, local playback permission*/
      "channelID": {
        /*required, the channel which supports configuring local playback permission*/
        "@min": "",
        "@max": ,
      },
      "enable": "true,false"
    /*required, enables permission or not, boolean*/
    },
    "localBackup": {
      /*required, local backup permission*/
      "channelID": {
        /*required, the channel which supports configuring local backup permission*/
        "@min": "",
        "@max": ,
      },
      "enable": "true,false"
    /*required, enables permission or not, boolean*/
    },
    "remotePlayback": {
      /*required, remote playback permission*/
      "channelID": {
        /*required, the channel which supports configuring remote playback permission*/
        "@min": "",
        "@max": ,
        "#text": ""
      },
      "enable": "true,false"
    /*required, enables permission or not, boolean*/
    }
  }
}
```

F.148 JSON_VerificationDisplay

JSON message about parameters of calibration verification result display

```
{  
    "VerificationDisplay": {  
        /*required, display calibration verification result*/  
        "enabled":  
            /*required, boolean, whether to display the calculated point position and verification point position on video: true-yes*/  
    }  
}
```

F.149 JSON_VerificationPoint

JSON message about calibration verification point information

```
{  
    "VerificationPoint": [  
        /*required, calibration verification point*/  
        "pan": ,  
        /*optional, float, pan, range: [0.00,360.00]*/  
        "tilt": ,  
        /*optional, float, tilt, range: [-90.00,90.00]*/  
        "zoom": ,  
        /*optional, float, zoom, range: [0.00,90.00]*/  
        "longitude": ,  
        /*required, float, longitude which is calculated based on the pan and tilt, corrects to 6 decimal places, range: [-180.000000,180.000000]*/  
        "latitude": ,  
        /*required, float, latitude which is calculated based on the pan and tilt, corrects to 6 decimal places, range: [-90.000000,90.000000]*/  
        "x": ,  
        /*optional, float, x-coordinate, [0.000,1.000]*/  
        "y": ,  
        /*optional, float, y-coordinate, [0.000,1.000]*/  
    ]  
}
```

F.150 JSON_VerificationPointList

JSON message about verification points information

```
{  
    "VerificationPointList": [  
        /*required, calibration verification points*/  
        "id": ,  
        /*required, int, verification point ID; PanoVu series camera supports up to 16 points, and speed dome supports up to 64 points*/  
        "pan": ,  
        /*optional, float, pan, range: [0.00,360.00]*/  
        "tilt": ,  
        /*optional, float, tilt, range: [-90.00,90.00]*/  
        "zoom": ,  
        /*optional, float, zoom, range: [0.00,90.00]*/  
        "longitude": ,  
        /*required, float, longitude which is calculated based on the pan and tilt, corrects to 6 decimal places, range: [-180.000000,180.000000]*/  
        "latitude": ,  
        /*required, float, latitude which is calculated based on the pan and tilt, corrects to 6 decimal places, range: [-90.000000,90.000000]*/  
        "x": ,  
        /*optional, float, x-coordinate, [0.000,1.000]*/  
        "y": ,  
        /*optional, float, y-coordinate, [0.000,1.000]*/  
    ]  
}
```

```
/*optional, float, tilt, range: [-90.00,90.00]*/
  "zoom": ,
/*optional, float, zoom, range: [0.00,90.00]*/
  "longitude": ,
/*required, float, longitude, corrects to 6 decimal places, range:[-180.000000,180.000000]*/
  "latitude": ,
/*required, float, latitude, corrects to 6 decimal places, range: [-90.000000,90.000000]*/
  "x": ,
/*optional, float, x-coordinate, [0.000,1.000]*/
  "y": ,
/*optional, float, y-coordinate, [0.000,1.000]*/
}
}
```

F.151 JSON_VerificationPointsCap

JSON message about capability of verify calibration

```
{
  "VerificationPointsCap":{
/*required, calibration verification capability*/
    "id":{
/*required, int, verification point ID; PanoVu series camera supports up to 16 points, and speed dome supports up to 64 points*/
      "@min":1,
      "@max":64
    },
    "pan":{
/*optional, float, pan*/
      "@min":0,
      "@max":360
    },
    "tilt":{
/*optional, float, tilt*/
      "@min":-90,
      "@max":90
    },
    "zoom":{
/*optional, float, zoom*/
      "@min":0,
      "@max":90
    },
    "longitude":{
/*required, float, longitude, corrects to 6 decimal places*/
      "@min":-180,
      "@max":180
    },
    "latitude":{
/*required, float, latitude, corrects to 6 decimal places*/
      "@min":-90,
```

```
    "@max":90
},
"x":{
/*optional, float, x-coordinate*/
    "@min":0,
    "@max":1
},
"y":{
/*optional, float, y-coordinate*/
    "@min":0,
    "@max":1
},
"isSupportDisplay":
/*optional, boolean, whether device supports display the verification result (related URI: /ISAPI/System/
GPSVerification/channels/<ID>/display?format=json)*/
}
```

F.152 JSON_VibrationDetection

Vibration detection message in JSON format

```
{
  "VibrationDetection": {
/*vibration detection*/
    "enabled": ,
/*required, boolean, whether to enable vibration detection, "true"-enable, "false"-disable*/
    "sensitivity":
/*required, int, sensitivity, value range: [0-100], by default: 50*/
  }
}
```

F.153 JSON_VibrationDetectionCap

Message about vibration detection capability in JSON format

```
{
  "VibrationDetectionCap": {
/*vibration detection capability*/
    "enabled": {
/*required, boolean, whether to enable vibration detection,"true"-enable, "false"-disable*/
      "@opt": [true, false]
    },
    "sensitivity": {
/*required, int, sensitivity, value range: [0-100]*/
      "@min": 0,
      "@max": 100,
      "@def": 50
    }
  }
}
```

```
}
```

F.154 JSON_WhiteLightAlarm

WhiteLightAlarm message in JSON format

```
{
  "WhiteLightAlarm": {
    "durationTime": ,
    /*required, int, flashing duration of supplement light, which is between 1 and 60 s*/
    "frequency": """",
    /*required, string, flashing frequency of supplement light: "high"-flashing 0.5s per second, "medium"-flashing 0.8s per
    1.6s, "low"-flashing 1s per 2s*/
    "brightness": ,
    /*optional, int, supplement light brightness, which is between 1 and 100*/
    "TimeRangeList": [
      /*optional, alarm output schedule list*/
      "week": ,
      /*required, int, days of the week: 1-Monday, 2-Tuesday, 3-Wednesday, 4-Thursday, 5-Friday, 6-Saturday, 7-Sunday*/
      "TimeRange": [
        "id": ,
        /*required, int, ID of time period for each day*/
        "beginTime": """",
        /*required, string, start time in ISO8601 format*/
        "endTime": """
        /*required, string, end time in ISO8601 format*/
        ]
      ]
    ]
  }
}
```

F.155 JSON_WhiteLightAlarmCap

WhiteLightAlarmCap message in JSON format

```
{
  "WhiteLightAlarmCap": {
    "durationTime": {
      "@min": 1,
      "@max": 60,
      "@def": 15
    },
    "frequency": {
      /*required, string, flashing frequency of supplement light: "high"-flashing 0.5s per second, "medium"-flashing 0.8s per
      1.6s, "low"-flashing 1s per 2s*/
      "@opt": "high,medium,low,normallyOn",
      "@def": "high"
    }
  }
}
```

```

},
"brightness":{
/*optional, int, supplement light brightness, which is between 1 and 100*/
    "@min": 1,
    "@max": 100,
    "@def": 50
},
"TimeRangeCap":{
/*optional, alarm output schedule capability*/
    "week":{
/*required, int, days of the week: 1-Monday, 2-Tuesday, 3-Wednesday, 4-Thursday, 5-Friday,6-Saturday, 7-Sunday*/
        "@opt":"1,2,3,4,5,6,7"
    },
    "id":{
/*required, int, ID of time period for each day*/
        "@maxSize": 8
    },
    "beginTime":{
/*required, string, start time in ISO8601 format*/
        "@min":"00:00",
        "@max":"24:00"
    },
    "endTime":{
/*required, string, end time in ISO8601 format*/
        "@min":"00:00",
        "@max":"24:00"
    }
},
"whiteLightMode":{
/*optional, string, supplement light mode: "keepOn"-always on, "flashing"-flashing*/
    "@opt":"keepOn,flashing",
    "@def":"keepOn"
}
}
}

```

F.156 JSON_WorkingStatus

WorkingStatus message in JSON format

```

{
    "WorkingStatus":{
        "devStatus": ,
        /*required, integer type, value of device status: 0-normal, 1-CPU usage, higher than 85%, 2-hardware error (e.g., serial
port exception)*/
        "ChanStatus"::[
            "chanNo": ,
            /*required, integer type, channel No., which starts from 1*/
            "enable": ,
            /*integer type, it is valid for analog channel only, 0-disable, 1-enable*/
        ]
    }
}

```

```

    "online": ,
/*required, integer type, online status: 0-offline, 1-online*/
    "record": ,
/*required, whether the device is recording, 0-no, 1-yes*/
    "recordStatus": ,
/*integer type, recording status: 0-recording, 1-recording exception (HDD exception), 2-recording exception (network camera offline), 3-recording exception (other reason)*/
    "signal": ,
/*required, integer type, signal status: 0-normal, 1-signal loss*/
    "linkNum": ,
/*required, integer type, number of software clients connected to this channel*/
    "bitRate": ,
/*required, integer type, channel bit rate, unit: Kpbs*/
},
"HDStatus":[{
    "hdNo": ,
/*required, integer type, HDD No., which starts from 1*/
    "status": ,
/*required, integer type, HDD status: 0-activate, 1-sleep, 2-exception, 3-sleepy HDD error, 4-unformatted, 5-disconnected (for network HDD), 6-formatting*/
    "volume": ,
/*required, integer type, HDD capacity, unit: MB*/
    "freeSpace": ,
/*required, integer type, free space, unit: MD*/
},
"IOStatus":{
    "IOInTrig": [...,...,...,...],
/*integer type, triggered alarm input No., two places and below: analog alarm input No., two places and above: digital alarm input No.; and the low 2-bit is the alarm input No., the 3-bit or above is the digital channel No., e.g., 3201-alarm input No.1 of digital channel No.32*/
    "IOOutTrig": [...,...,...,...]
/*integer type, triggered alarm output No., two places and below: analog alarm output No., two places and above: digital alarm output No.; and the lower 2 bits are the alarm output No., the 3-bit or above is the digital channel No., e.g., 3201-alarm output No.1 of digital channel No.32*/
}
}
}

```

F.157 JSON_XX

XX message in JSON format

```
{
    "XX":{
/*required, event/alarm configuration information*/
        "enabled": "",
/*required, enable or not, boolean*/
        "zoneName": ""
/*optional, zone name, it is valid only when event type is zone, string*/
    }
}
```

```
}
```

Remarks

The XX in the parameter "**XX**" corresponds to detailed event type. E.g., if the event type is `humanRecognition`, then the returned node is "HumanRecognition".

F.158 JSON_XXCap

XXCap message in JSON format

```
{
  "XXCap": {
    /*required, event configuration information*/
    "enabled": "true,false",
    /*required, enable or not, boolean*/
    "zoneName": {
      /*optional, zone name, it is valid only when event type is zone, string*/
      "@min": 1,
      "@max": 64,
    },
  }
}
```

Remarks

The XX in the parameter "**XXCap**" corresponds to detailed event type. E.g., if the event type is `humanRecognition`, then the returned node is "HumanRecognitionCap".

F.159 XML_AccessProtocolAbility

AccessProtocolAbility message in XML format

```
<AccessProtocolAbility version="2.0">
  <channelNO><!--req, channel No.--></channelNO>
  <EzvizParam>
    <enable opt="true,false"/><!--req, whether to enable EZVIZ video7 protocol: 0-no, 1-yes-->
    <deviceStatus attri="readonly" opt="offline,online"/><!--Device register status-->
    <allowRedirect opt ="0,1,2"/><!--opt, whether supports address redirection, if not support, the node will not be returned-->
    <domainLen min="" max="" /><!--opt, server domain length, if not support, the node will not be returned-->
    <netMode opt="auto,wiredNetworkPriority, wiredNetwork,wirelessNetwork"/>
    <!--opt, network mode: auto, wired network is preferred, wired network (default), wireless network-->
    <VerificationCode><!--opt, whether supports configuring verification code, if not support, the node will not be returned-->
    <verificationCodeType opt="normal,empty"><!--opt, xs: string--></verificationCodeType>
    <supportDeclarationURL>
      <!--opt, xs: string, service declaration hyperlink-->
```

```
</supportDeclarationURL>
<supportPrivacyPolicyURL>
  <!--opt, xs: string, privacy policy hyperlink-->
</supportPrivacyPolicyURL>
<verificationCodeModify>
  <!--opt, whether the verification code is edited, true-yes, false-no, if not support, the node will not be returned-->
</verificationCodeModify>
</VerificationCode>
<EZVIZSecretKey>
  <!--opt, whether supports capability of editing verification code for Guarding Vision-->
<offlineStatus>
  <!--ro,dep,xs:string; it is valid when registerStatus values "false", device offline status, opt="secretKeyInvalid"-->
  invalid verification code-->
</offlineStatus>
<secretKey min="0" max="64"><!--opt, xs:string, verification code for Guarding Vision--></secretKey>
</EZVIZSecretKey>
<operateCode min="" max=""><!--opt, operation code, this node is used for linking device. If linking device is not supported, this node will not be returned-->
</EzvizParam>
<SupportTransPrivateProtocol><!--opt, support private protocol passthrough via Guarding Vision-->
<NotSupportLongConfig>
  <!--opt, the persistent connection is supported by Guarding Vision when communicating between Guarding Vision and device via SDK-->
    <preview><!--opt, whether supports live view, "true"-yes--></preview>
    <voiceCom><!--opt, whether supports two-way audio, "true"-yes--></voiceCom>
    <playbackByTime><!--opt, whether supports playback by time, "true"-yes--></playbackByTime>
    <playbackByFile><!--opt, whether supports playback by file name, "true"-yes--></playbackByFile>
    <playReverseByTime><!--opt, whether supports reversely playback by time, "true"-yes--></playReverseByTime>
    <playReverseByFile><!--opt, wehtehr supports reversely playback by file name, "true"-yes--></playReverseByFile>
    <playbackByTimeAndId><!--opt, whether supports playback by ID and time, "true"-yes--></playbackByTimeAndId>
    <playbackByTimeLatitudeLongitude>
      <!--opt, whether supports playback by time and latitude/longitude, "true"-yes-->
    </playbackByTimeLatitudeLongitude>
    <playbackByTimePCNVR><!--opt, whether the PCNVR supports playback by time PCNVR, "true"-yes--></playbackByTimePCNVR>
      <arm><!--opt, whether supports arming channel, "true"-yes--></arm>
      <format><!--opt, whether supports remotely formatting HDD, "true"-yes--></format>
      <log><!--opt, whether supports device log, "true"-yes--></log>
      <matrixLog><!--opt, whether supports MVC log, "true"-yes--></matrixLog>
      <smartLog><!--opt, whether supports smart device log, "true"-yes--></smartLog>
      <alarmHostLog><!--opt, whether supports security control panel log, "true"-yes--></alarmHostLog>
      <uploadConfigFile><!--opt, whether supports importing configuration file, "true"-yes--></uploadConfigFile>
      <downloadConfigFile><!--opt, whether supports exporting configuration file, "true"-yes--></downloadConfigFile>
      <exportIPCCConfigFile>
        <!--opt, whether supports exporting configuration file of network camera, "true"-yes-->
      </exportIPCCConfigFile>
      <importIPCCConfigFile>
        <!--opt, whether supports importing configuration file of network camera, "true"-yes-->
      </importIPCCConfigFile>
      <transparencySerial><!--opt, whether supports transparent channel, "true"-yes--></transparencySerial>
      <alarmHostTransparencySerial>
        <!--opt, whether supports transparent channel of security control panel, "true"-yes-->
```

```
</alarmHostTransparencySerial>
<backupByFile><!--opt, whether supports video backup by file name, "true"-yes--></backupByFile>
<backupByTime><!--opt, whether supports video backup by time, "true"-yes--></backupByTime>
<backupPicture><!--opt, whether supports picture backup, "true"-yes--></backupPicture>
<backupLog><!--opt, whether supports log backup, "true"-yes--></backupLog>
<upgrade><!--opt, whether supports upgrade, "true"-yes--></upgrade>
<upgradeRaidAdapter><!--opt, whether supports upgrading adapter, "true"-yes--></upgradeRaidAdapter>
<upgradeIPChannel><!--opt, whether supports upgrading digital channel, "true"-yes--></upgradeIPChannel>
<upgradeVCALib><!--opt, whether supports upgrading intelligent library, "true"-yes--></upgradeVCALib>
<upgradeOptical><!--opt, whether supports upgrading optical terminal, "true"-yes--></upgradeOptical>
<inquireFaceDBRecord><!--opt, whether supports searching for face data records in face database, "true"-yes--></inquireFaceDBRecord>
    <searchFaceDB><!--opt, whether supports searching in face database, "true"-yes--></searchFaceDB>
    <inquireSnapDBRecord><!--opt, whether supports searching pictures in captured picture database, "true"-yes--></inquireSnapDBRecord>
        <searchSnapDB><!--opt, whether supports searching in captured picture database, "true"-yes--></searchSnapDB>
        <faceMatchAlarm><!--opt, whether supports face comparison in blocklist and allowlist alarm, "true"-yes--></faceMatchAlarm>
            <findDatabase><!--opt, whether supports searching in database, "true"-yes--></findDatabase>
            <findBlackList><!--opt, whether supports searching for blocklist, "true"-yes--></findBlackList>
            <findSnapPicture><!--opt, whether supports searching for captured picture, "true"-yes--></findSnapPicture>
            <findAdvanceSnapPicture><!--opt, whether supports dual-VCA, "true"-yes--></findAdvanceSnapPicture>
            <uploadAudio><!--opt, whether supports uploading audio file, "true"-yes--></uploadAudio>
            <downloadAudio><!--opt, whether supports downloading audio, "true"-yes--></downloadAudio>
            <passiveDecode><!--opt, whether supports passive decoding--></passiveDecode>
            <passiveTransCode><!--opt, whether supports passive transcoding--></passiveTransCode>
            <pictureView><!--opt, whether supports picture preview--></pictureView>
            <dvcsOperateDevice><!--opt, whether supports configuring distributed sub device--></dvcsOperateDevice>
            <uploadPicture><!--opt, whether supports uploading background picture--></uploadPicture>
            <t1Test><!--opt, whether supports T1 test--></t1Test>
            <uploadCert><!--opt, whether supports uploading certificate--></uploadCert>
            <downloadCert><!--opt, whether supports downloading certificate--></downloadCert>
            <downloadBaselineScenePicture><!--opt, whether supports downloading scene picture--></downloadBaselineScenePicture>
                <downloadVQDAlarmPicture><!--opt, whether supports downloading VQD alarm picture--></downloadVQDAlarmPicture>
                    <uploadInquestFile><!--opt, whether supports uploading inquest file--></uploadInquestFile>
                    <resumeInquestEvent><!--opt, whether supports restoring inquest event--></resumeInquestEvent>
                    <getNetworkFlow><!--opt, whether supports getting network traffic flow--></getNetworkFlow>
                    <emailTest><!--opt, whether supports email service test--></emailTest>
                    <alolibDebugInfo><!--opt, whether supports outputting algorithm library debug information--></alolibDebugInfo>
                        <eventSearch><!--opt, whether supports event search--></eventSearch>
                        <smartSearch><!--opt, whether supports VCA search--></smartSearch>
                        <findFile><!--opt, whether supports file search--></findFile>
                        <findIPSANDirectory><!--opt, whether supports searching in IPSAN file directory--></findIPSANDirectory>
                        <findPicture><!--opt, whether supports picture search--></findPicture>
                        <findInquestFile><!--opt, whether supports inquest file search--></findInquestFile>
                        <findRecordLabel><!--opt, whether supports searching videos by tag--></findRecordLabel>
                        <findPCNVRFile><!--opt, whether supports searching PCNVR file--></findPCNVRFile>
                        <setAccessCameraInfo><!--opt, whether supports setting camera information--></setAccessCameraInfo>
                        <securityPullRaidDisk><!--opt, whether supports removing disk in security--></securityPullRaidDisk>
```

```

<scanRaid><!--opt, whether supports scanning RAID--></scanRaid>
<GPSData><!--opt, whether supports getting GPS data--></GPSData>
<oneKeyConfig><!--opt, whether supports one-touch configuration--></oneKeyConfig>
<streamInfoList><!--opt, whether supports getting added stream ID--></streamInfoList>
<findNASDirectory><!--opt, whether supports searching in NAS directory--></findNASDirectory>
<ITSTransparencyChannel><!--opt, whether supports ITS transparent channel--></ITSTransparencyChannel>
<getInputVolume><!--opt, whether supports getting input volume--></getInputVolume>
<vehicleCheck><!--opt, whether supports checking vehicle blocklist--></vehicleCheck>
<fisheyeConfig><!--opt, whether supports fisheye function configuration by persistent connection--></fisheyeConfig>
<getVehicleList><!--opt, whether supports getting all vehicle allowlist or blocklist information--></getVehicleList>
<setVehicleList><!--opt, whether supports setting all vehicle allowlist/blocklist--></setVehicleList>
<NASServerTest><!--opt, whether supports NAS test--></NASServerTest>
<NTPServerTest><!--opt, whether supports NTP server test--></NTPServerTest>
<IPServerTest><!--opt, whether supports IP conflicting test--></IPServerTest>
<FTPServerTest><!--opt, whether supports FTP server test--></FTPServerTest>
<emailServerTest><!--opt, whether supports email server test--></emailServerTest>
<expand><!--opt, whether supports HDD remote expansion--></expand>
<raidFastConfig><!--opt, whether supports RAID one-touch configuration--></raidFastConfig>
<VQDDiagnoseInfo><!--opt, whether supports getting VQD information--></VQDDiagnoseInfo>
<ITSExternalDevice><!--opt, whether supports getting ITS external device information--></ITSExternalDevice>
<FCInfo><!--opt, whether supports getting FC information--></FCInfo>
<NPlusOneDeviceInfo><!--opt, whether supports getting device information in N+1 mode--></NPlusOneDeviceInfo>
<matrixDispRoute><!--opt, whether supports displaying the video saving path--></matrixDispRoute>
<Ipv6List><!--opt, whether supports IPv6 address--></Ipv6List>
<zoneInfo><!--opt, whether supports getting all zones' information--></zoneInfo>
<alarmoutInfo><!--opt, whether supports getting all triggers' information--></alarmoutInfo>
<variableInfo><!--opt, whether supports getting variable element information--></variableInfo>
<searchAlarmhostExternalDevice><!--opt, whether supports searching peripherals--></searchAlarmhostExternalDevice>
<registerAlarmhostExternalDevice><!--opt, whether supports registering peripherals--></registerAlarmhostExternalDevice>
<historyValue><!--opt, whether supports getting history data--></historyValue>
<ETCChannelState><!--opt, whether supports getting ECT channel status--></ETCChannelState>
<VQDmonitorPlan><!--opt, whether supports getting management schedule--></VQDmonitorPlan>
<VQDmonitorID><!--opt, whether supports getting camera information--></VQDmonitorID>
<CDWriteStatus><!--opt, whether supports getting CD burning status--></CDWriteStatus>
<pdcResult><!--opt, whether supports people counting--></pdcResult>
<ptzLockInfo><!--opt, whether supports PTZ locking--></ptzLockInfo>
<gopInfo><!--opt, whether supports GOP--></gopInfo>
<heatMapResult><!--opt, whether supports heat map--></heatMapResult>
<lockInfoList><!--opt, whether supports getting locked user information--></lockInfoList>
</NotSupportLongConfig>
</SupportTransPrivateProtocol>
</AccessProtocolAbility>

```

F.160 XML_AdminAccessProtocol

AdminAccessProtocol message in XML format

```
<AdminAccessProtocol version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:string, ID--></id>
  <enabled><!--optional, xs:boolean, "true,false"--></enabled>
  <protocol><!--required, xs:string, protocol name:
    "HTTP,HTTPS,RTSP,DEV_MANAGE,IOT,WebSocket,WebSocketS,SDK_OVER_TLS,TLS1_1Enable,TLS1_2Enable,SRTP,MATRIX_GATEWAY,Bonjour,104Proto,SRTP"--></protocol>
  <portNo><!--required, xs:integer--></portNo>
  <redirectToHttps>
    <!--opt, xs: boolean, whether to automatically go to HTTPS when connecting to HTTP port and HTTPS is enabled-->
  </redirectToHttps>
  <streamOverTls>
    <!--opt, xs: boolean, whether to enable TLS link encryption when the streaming mode is SDK_OVER_TLS, true-yes, false-no-->
  </streamOverTls>
  <TLS1_1Enable><!--dep, xs:boolean--></TLS1_1Enable>
  <TLS1_2Enable><!--dep, xs:boolean--></TLS1_2Enable>
</AdminAccessProtocol>
```

Remarks

"TLS1_1Enable" and "TLS1_2Enable" are not available for port configuration.

F.161 XML_AdminAccessProtocolList

AdminAccessProtocolList message in XML format

```
<AdminAccessProtocolList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <AdminAccessProtocol/><!--see details in the message of XML_AdminAccessProtocol-->
</AdminAccessProtocolList>
```

See Also

[XML_AdminAccessProtocol](#)

F.162 XML_AlgorithmsVersion

Message about version information of algorithm library in XML format

```
<AlgorithmsVersion version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <AlgorithmsVersionInfoList><!--optional-->
    <AlgorithmsVersionInfo>
      <majorVersion><!--required, xs:integer, major version No.--></majorVersion>
      <minorVersion><!--required, xs:integer, minor version No.--></minorVersion>
      <revisionNumber><!--required, xs:integer, revision No.--></revisionNumber>
      <buildNumber><!--required, xs:integer, build No.--></buildNumber>
      <versionYear><!--required, xs:integer, date: year--></versionYear>
      <versionMonth><!--required, xs:integer, date: month--></versionMonth>
      <versionDay><!--required, xs:integer, date: day--></versionDay>
      <name>
```

```
<!--required, xs:string, algorithm library name, "faceProperties"-face attributes, "behaviorAnalysis"-behavior analysis, "humanAttribute"-human body attributes, "faceSnap"-face capture, "faceRecognition"-face recognition, "faceContrast"-face picture comparison, "peopleCounting"-people counting, "faceScore"-face picture score, "personQueueDetection"-queue management, "depthMap"-depth map, "safetyHelmetDetection"-hard hat detection, "streetBehavior"-street behavior, "workBehavior"-work behavior, "cityManagement"-city management, "fireEscapeDetection"-fire engine access detection, "takingElevatorDetection"-elevator detection-->
<name>
</AlgorithmsVersionInfo>
</AlgorithmsVersionInfoList>
</AlgorithmsVersion>
```

Example

AlgorithmsVersion Message Example

```
<AlgorithmsVersion version="2.0" xmlns="http://www.isapi.com/ver20/XMLSchema">
<AlgorithmsVersionInfoList>
<AlgorithmsVersionInfo>
<majorVersion>1</majorVersion>
<minorVersion>0</minorVersion>
<revisionNumber>1</revisionNumber>
<versionYear>17</versionYear>
<versionMonth>12</versionMonth>
<versionDay>12</versionDay>
<name>faceSnap</name>
</AlgorithmsVersionInfo>
<AlgorithmsVersionInfo>
<majorVersion>1</majorVersion>
<minorVersion>0</minorVersion>
<revisionNumber>0</revisionNumber>
<versionYear>17</versionYear>
<versionMonth>10</versionMonth>
<versionDay>31</versionDay>
<name>faceScore</name>
</AlgorithmsVersionInfo>
<AlgorithmsVersionInfo>
<majorVersion>3</majorVersion>
<minorVersion>4</minorVersion>
<revisionNumber>1</revisionNumber>
<versionYear>17</versionYear>
<versionMonth>12</versionMonth>
<versionDay>25</versionDay>
<name>faceContrast</name>
</AlgorithmsVersionInfo>
</AlgorithmsVersionInfoList>
</AlgorithmsVersion>
```

F.163 XML_BasicCapability

BasicCapability message in XML format

```
<?xml version="1.0" encoding="utf-8"?>
<!--req, software and hardware capabilities-->
<BasicCapability version="2.0">
  <HardwareCapability><!--req, hardware capability-->
    <HardwareVersion>
      <!--req, the version of front panel: higher 16-bit means the major version, and lower 16-bit means the minor version-->
    </HardwareVersion>
    <AlarmInPortNum><!--req, number of alarm inputs--></AlarmInPortNum>
    <AlarmOutPortNum><!--req, number of alarm outputs--></AlarmOutPortNum>
    <RS232Num><!--req, number of RS232 ports--></RS232Num>
    <RS485Num><!--req, number of RS485 ports--></RS485Num>
    <NetworkPortNum><!--req, number of network interfaces--></NetworkPortNum>
    <USBNum><!--req, number of USB ports--></USBNum>
    <FlashSize><!--req, the flash size (unit:MB)--></FlashSize>
    <RamSize><!--req, RAM size (unit:M)--></RamSize>
    <USBVersion><!--req, USB version--></USBVersion>
    <SDNum><!--req, number of SD cards--></SDNum>
    <HardDiskNum>
      <!--req, number of HDDs (including SATA, eSATA and NAS), number of available HDDs-->
    </HardDiskNum>
    <SATANum><!--req, number of SATAs--></SATANum>
    <eSATANum><!--req, number of eSATAs--></eSATANum>
    <miniSASNum><!--req, number of miniSASs--></miniSASNum>
    <VideoInNum><!--req, number of video inputs--></VideoInNum>
    <AudioInNum><!--req, number of audio inputs--></AudioInNum>
    <VideoOutNum><!--req, number of video outputs --></VideoOutNum>
    <AudioOutNum><!--req, number of audio outputs--></AudioOutNum>
    <AudioTalkNum><!--req, number of two-way audio channels--></AudioTalkNum>
    <SDSupport><!--req, whether support SD card: 1-support, the node does not exist-not support --></SDSupport>
    <WiFiSupport><!--req, whether support WiFi: 1-support, the node does not exist-not support --></WiFiSupport>
    <POESupport><!--req, whether support POE: 1-support, the node does not exist-not support --></POESupport>
    <IRSupport><!--req, whether support IR: 1- support, the node does not exist-not support --></IRSupport>
    <VideoOutSupport>
      <!-- req, whether support local video output: 1- support, the node does not exist-not support -->
    </VideoOutSupport>
    <ResetSupport>
      <!-- req, whether support restoring factory settings: 1- support, the node does not exist-not support -->
    </ResetSupport>
    <AnalogChannelNum>
      <!-- req, number of analog channels, which is same with the value of byChanNum returned when logging in-->
    </AnalogChannelNum>
    <IPChannelNum><!-- req, number of IP channels --></IPChannelNum>
    <MultiNetworkCard>
      <!-- req, whether support multiple NICs: 1-support, the node does not exist-not support -->
    </MultiNetworkCard>
    <BondingSupport><!--req, whether support bonding: 1- support, the node does not exist-not support --></BondingSupport>
    <VGANumber><!--req, number of VGA ports --></VGANumber>
    <HDMIINumber><!--req, number of HDMI® ports --></HDMIINumber>
    <CVBSNumber><!--req, number of CVBS ports --></CVBSNumber>
    <AuxoutNumber><!--req, number of auxiliary outputs--></AuxoutNumber>
```

```
<RAIDType><!--req, RAID type: 0-hardware RAID, 1-software RAID--></RAIDType>
<RS485>
  <localRs485No min="" max="" /><!--req, local RS485 number-->
  <expandRS485No min="" max="" /><!--req, extended RS485 No.-->
  <fullDuplexRS485No min="" max="" /><!--req, full duplex RS485 No., relative to local RS485 No.-->
  <semiDuplexRS485No min="" max="" /><!--req, half duplex RS485 No., relative to local RS485 No.-->
  <RS485SlotNo min="" max="" /><!--req, RS485 slot No.-->
</RS485>
<ZoneNum>
  <!--alarm host uses the node-->
  <localZoneNo min="1" max="64"/>
  <!--req, local alarm input interface (local arming region) No.-->
  <extendZoneNo min="1" max="64"/>
  <!--req, extensible alarm input interface (extensible arming region) No.-->
</ZoneNum>
<sirenNo min="" max="" />
<!--req, host siren No.-->
<electroLockNo min="" max="" /><!--req, electric lock No.-->
<!--req, Electric Lock No. -->
<mobileGateNo min="" max="" />
<!--req, Mobile Gate No.-->
<TriggerNum><!--req, this node is for security control panel only-->
  <localTriggerNo min="1" max="64"/>
  <!--req, local alarm output No. (local trigger)-->
  <extendTriggerNo min="1" max="64"/>
  <!--req, extended alarm output No. (extended trigger)-->
<Sensor>
  <!--req, power supply monitoring alarm host uses the node-->
  <totalSensorNum&>
    <!--required, xs:int, number of analog sensors-->
  </totalSensorNum>
  <localSensorNo min="" max="" />
  <!--req, local analog sensor No.-->
  <expandSensorNo min="" max="" />
  <!--req, extensible analog sensor No.-->
</Sensor>
<rs232_rs485No min="" max="" />
<!--req, RS232/ number of RS485 serial ports-->
<MirrorChanNum>1</MirrorChanNum>
<!--req, number of mirror channels-->
<DeviceSubBoardInfo>
  <!--req, device sub board information, for use of integrated display controller-->
  <BackBoardType opt = "4U,8U,13U"/>
  <!--req, supported back board type-->
  <SubBoardType opt= "DVI-IInput,YPbPrInput,BNCInput,SDIInput,DVI DualLinkInput,OrdinaryDecodeInput,DVI-IOOutput,SDIOOutput,EnhanceDecodeInput"/>
  <!--req, supported sub board type-->
</DeviceSubBoardInfo>
</HardwareCapability>
<!-- req, software capability -->
<SoftwareCapability>
  <!-- req, whether support the new mapping type of hard disk number: 1- support, the node does not exist-not
```

```
support -->
<NewHdNo>1</NewHdNo>
<!-- req, the max number of network disks supported by the device(the node does not exist-not support), including
NAS and IPSAN -->
<MaxNetworkHDNum>8</MaxNetworkHDNum>
<!-- req, whether support NAS: 1- support, the node does not exist-not support -->
<NasSupport>1</NasSupport>
<!-- req, number of NAS -->
<NasNumber>8</NasNumber>
<!-- req, the max number of text overlay strings (special for analog channel) -->
<NetDiskIdentification>
<!-- req, support network disk access authentication-->
<NASIdentification>
<!--req, support NAS access authentication-->
<NFSMountType>true</NFSMountType>
<!--req, support NAS supporting NFS access-->
<CIFSMountType>
<!--req, support NAS supporting CIFS access-->
<usernameLen min = "" max= ""/>
<!--req, max. and min. value of name length on NAS authentication-->
<passwordLen min = "" max= ""/>
<!--req, max. and min. value of password length on NAS authentication-->
</CIFSMountType>
</NASIdentification>
</NetDiskIdentification>
<NasMountPara>
<addressType opt="IP, Domain"/>
<!--req, address type of mounted HDD 0-IP address; 1-domain name-->
<domainAddressLen min = "" max= ""/>
<!--req, max. and min. length of mounted HDD domain name-->
</NasMountPara>
<NetDiskDomain>
<domainSupport>true</domainSupport>
<!--req, whether mounted HDD supports domain name: true- yes; false- no-->
<domainAddressLen min = "" max= ""/>
<!--req, max. and min. length of mounted HDD domain name-->
</NetDiskDomain>

<ShowStringNumber>8</ShowStringNumber>
<!--req, max. quantity of string overlays (special for analog channel) -->
<MotionDetectAlarmSupport>1</MotionDetectAlarmSupport>
<!-- req, whether to support motion detection: 1- Yes, not displayed if not support -->
<VILostAlarmSupport>1</VILostAlarmSupport>
<!-- req, whether to support video loss detection: 1-Yes, not displayed if not support (special for analog channel) -->
<HideAlarmSupport>1</HideAlarmSupport>
<!-- req, whether to support tamper detection: 1- Yes, not displayed if not support (special for analog channel) -->
<ShelterSupport>1</ShelterSupport>
<!-- req, whether to support privacy mask: 1- Yes, not displayed if not support (special for analog channel) -->
<RtspSupport>1</RtspSupport>
<!-- req, whether to support rtsp protocol: 1- Yes, not displayed if not support -->
<RtpoverRtspSupport>
<!--whether support RTP over RTSP streaming mode: 1-yes, if not supported, this node will not be returned-->
```

```
</RtpoverRtspSupport>
<RtpoverHttpsSupport>
  <!--whether supports RTP over HTTPS streaming mode, 1-yes, if not supported, this node will not be returned-->
</RtpoverHttpsSupport>
<NtpSupport>1</NtpSupport>
  <!-- req, whether to support NTP timing: 1-Yes, not displayed if not support -->
<EptzSupport>1</EptzSupport>
  <!-- req, whether to support E-PTZ:1- Yes, not displayed if not support -->
<PtzSupport>1</PtzSupport>
  <!-- req, whether to support PTZ control: 1- Yes, not displayed if not support -->
<DDNSSupport>1</DDNSSupport>
  <!-- req, whether to support DDNS: 1- Yes, not displayed if not support. Compatible network camera cannot be deleted -->
<DDNSHostType>0,1,2,3,4</DDNSHostType>
  <!-- req, DDNS server type: 0- IP Server, 1-Dyndns,2-PeanutHull, 3-NO-IP, 4- hkDDNS -->
<DDNSStatus>1,2,3,4,5,6,7,8,9,10,11,12,13,14,15</DDNSStatus>
  <!--req, supported DDNS status type: 1- Failed to connect to address server, 2- Failed to parse address message, 3- Failed to connect to heartbeat server, 4- Failed to parse heartbeat server message, 5- Failed to connect to domain name server, 6- Failed to parse domain name server message, 7- Domain name registered, 8- Failed to register domain name, 9- Heartbeat sent, 10- DNS server is not configured, 11- Domain name is occupied, 12- Service exception, 13- Authentication required, 14- Invalid domain name, 15- Disabled-->
<SNMPSupport>1</SNMPSupport>
  <!-- req, SNMP version, 1-v1, 2-v2, 3-v3 -->
<SNMPVersion>1,2,3</SNMPVersion>
  <!-- req, whether support UPNP: 1- support, the node does not exist-not support -->
<UPNPSupport>1</UPNPSupport>
  <!-- req, whether support iSCSI: 1- support, the node does not exist-not support -->
<iSCSISupport>1</iSCSISupport>
  <!-- req, the max number of IP SAN supported by the device -->
<iSCSINum>1</iSCSINum>
  <!-- req, whether support ipv6: 1- support, the node does not exist-not support -->
<Ipv6Support>0</Ipv6Support>
  <!-- req, whether support intelligent function: 1- support, the node does not exist-not support -->
<VCASupport>1</VCASupport>
  <!-- req, capability set, this node will be returned when video stream is not supported -->
<VideoStreamnotSupport>true</VideoStreamnotSupport >
  <!-- req, whether support composite stream(video&audio): 1- support, the node does not exist-not support(special for analog channel) -->
<MultipleStreamSupport>1</MultipleStreamSupport>
  <!-- req, whether support sub stream: 1- support, the node does not exist-not support(special for analog channel) -->
>
<SubStreamSupport>1</SubStreamSupport>
  <!-- req, whether support EMAIL: 1- support, the node does not exist-not support -->
<EmailSupport>1</EmailSupport>
  <!-- req, SADP version: 0-V1.0, 1-V3.0 -->
<SADPVersion>0,1</SADPVersion>
<ZeroChanNumber>1</ZeroChanNumber>
  <!-- req, whether support backup: 1- support, the node does not exist-not support -->
<BackupSupport>1</BackupSupport>
  <!-- req, whether support searching record files by event: 1- support, the node does not exist-not support -->
<FindFileByEventSupport>1</FindFileByEventSupport>
  <!-- req, whether support smart searching: 1- support, the node does not exist-not support -->
```

```
<SmartSearchSupport>1</SmartSearchSupport>
<!-- req, whether support ATM configuration: 1- support, the node does not exist-not support-->
<ATMSupport>1</ATMSupport>
<!-- req, whether support to lock or unlock the files: 1- support, the node does not exist-not support -->
<FileLockSupport>1</FileLockSupport>
<!-- req, whether support dynamic limit for bit rate of main/sub stream: 1- support, the node does not exist-not support(could not configure the capability) -->
<BitrateLimitSupport>1</BitrateLimitSupport>
<!-- req, the max number of login supported by the device -->
<MaxLoginNum>0</MaxLoginNum>
<!-- req, the max number of live view supported by the device -->
<MaxPreviewNum>0</MaxPreviewNum>
<!-- req, the max number of playback supported by the device -->
<MaxPlayBackNum>0</MaxPlayBackNum>
<!-- req, the max number of connections supported by one channel -->
<MaxChanLinkNum>6</MaxChanLinkNum>
<ShutDownSupport>1</ShutDownSupport>
<!-- req, whether support to freeze preset: 1- support, the node does not exist-not support -->
<FrameFreezeSupport>1</FrameFreezeSupport>
<!-- req, whether support hard disk group configuration: 1- support, the node does not exist-not support -->
<HDgroupConfig>1</HDgroupConfig>
<!--req, whether supports auto saving mode, 1-Yes, if not supports, this node will not be displayed.-->
<AutoStorageMode>1</AutoStorageMode>
<!-- req, whether support RS232 configuration: 1- support, the node does not exist-not support -->
<RS232Config>1</RS232Config>
<!-- req, whether support PPPoE configuration: 1- support, the node does not exist-not support -->
<PPPoEConfig>1</PPPoEConfig>
<!-- req, whether support redundant recording: 1- support, the node does not exist-not support -->
<RedundancyRecord>1</RedundancyRecord>
<!-- req, whether support scale configuration of main or aux video output: 1- support, the node does not exist-not support -->
<VideoOutScaleConfig>1</VideoOutScaleConfig>
<!-- req, whether support guest user: 1- support, the node does not exist-not support -->
<GuestUser>1</GuestUser>
<!-- req, whether support FTP to upload pictures: 1- support, the node does not exist-not support -->
<UploadFTP>1</UploadFTP>
<!-- req, whether it does not support disk quota: 1- not support, no this node if support -->
<NotSupportDiskQuota>1</NotSupportDiskQuota>
<!-- req, 2012-12-8 support ratio-based disk quota allocation, 1- support, not displayed if not supported-->
<QuotaRatio>1</QuotaRatio>
<!-- req, 2014-4-11 support time-based disk quota allocation, 1- support both recording and picture quota, 2-support recording quota but not picture quota, not displayed if not supported-->
<QuotaByTime>1</QuotaByTime>
<QuotaWithoutChannel><!--for camera with multiple channels, channel No. is not required when performing disk quota--></QuotaWithoutChannel>
<LocalVout>
  <mainVoutType opt="mainCVBS,HDMI,VGA,auto"/>
  <!--req, the output type which is the local main port, nonhomologous device: 0-Auto,1-main CVBS, 2-HDMI®, 3-VGA; homologous device: 0-Auto, 1-main CVBS, 2-HDMI®/VGA-->
</LocalVout>
<SingleStorageMaxCap>16</SingleStorageMaxCap><!--req,the maximum capacity of single storage, unit: T-->
<IPCPlug>1</IPCPlug><!--req, whether support plug and play of IPC: 1- support, the node does not exist-not
```

```
support-->
<DrawFrameRecord>1</DrawFrameRecord><!--req, whether support frame extract recording: 1- support, the node does not exist-not support-->
<DeleteInvalidDisk>1</DeleteInvalidDisk><!--req, whether support to delete invalid disks: 1- support, the node does not exist-not support-->
<MountOrUnmountDisk>1</MountOrUnmountDisk><!--req, whether support to mount or unmount disks: 1- support, the node does not exist-not support-->
<MaxDvcsSubDevNumNum><!--req,the maximum device number which distributed device supported, the device which is not supported doesn't appear--></MaxDvcsSubDevNumNum>
<NotSupportInputOutputConfigFile>1</NotSupportInputOutputConfigFile><!--req,input/output the configurate file is not supported, 1-not support-->
<NotSupportLogSearch>1</NotSupportLogSearch><!--req,not support log search, if it is supported, it doesn't appear, 1-not support--><DateUpLoadAndDownLoad><!--data upload and download-->
<audioType opt="wave"/><!--audio type supported-->
<uploadAndDownload opt = "upload,download"/><!--support upload and download-->
<maxNum opt= "8"/><!--type:max audio number which is supported by device-->
</DateUpLoadAndDownLoad>
<DevModuleServerCfg><!--req,server configuration ability-->
<!--req,the configuration of telnet is only permitted by "admin" -->
<telnetServer opt="disable,enable"/><!--req 0-disable 1-enable-->
<irLampServer opt="disable,enable"/><!--req 0-disable 1-enable-->
<abfServer opt="enable,disable,"/>
<!--req 0-enable, 1-disable -->
<LEDStatus opt="disable,enable"/>
<!--req 0-disable, 1-enable-->
<autoDefog opt="disable,enable"/>
<!--req, auto defog control: 0- enable, 1- disable-->
<sshServer opt="disable,enable"/>
<!--req, SSH settings: 0- enable, 1- disable-->
<webAuthentication opt="disable,enable"/>
<!--req, WEB authentication: 0- disable, 1- enable-->
<supplementLight opt="disable,enable"/>
<!--req, illuminator control: 0- enable, 1- disable-->
<deicing opt="disable,enable"/>
<!--opt, deicing: 0- disable, 1-enable-->
<isSupportFireLaserLight><!--opt,xs:boolean, when true, it is thermal imaging fire source detection laser supplement light--></isSupportFireLaserLight>
<visibleMovementPower opt="disable,enable"/><!--opt 0-disable, 1-enable-->
<thermalMovementPower opt="disable,enable"/><!--opt 0-disable, 1-enable-->
<ptzPower opt="disable,enable"/><!--opt 0-disable, 1-enable-->
<powerSavingControl opt="sleepMode,lowConsumptionMode"/><!--opt xs:string "Low-power consumption: sleepMode-Sleepy mode, lowConsumptionMode-Low-power mode-->
<captureWithSupplimentLightEnabled opt="true,false"><!--opt, xs:boolean, enable snapshot supplement light--></captureWithSupplimentLightEnabled>
</DevModuleServerCfg>
<SearchLogAbilitySupport>1</SearchLogAbilitySupport><!--req,if it supports GB/T28181 protocol, 1-support, if not support,it does not appear-->
<AlarmTriggerRecordAbilitySupport>1</AlarmTriggerRecordAbilitySupport><!--req,if it supports log search 1-support, if not support,it does not appear-->
<CameraParaDynamicAbilitySupport>1</CameraParaDynamicAbilitySupport><!--req,if it supports camera parameters dymatic ability, 1-support, if not support,it does not appear-->
<IOAbilitySupport>1</IOAbilitySupport>
```

```
<AccessProtocolAbility>1</AccessProtocolAbility>
<!--req, protocol access capability, 1- support, not displayed if not supported-->
<CameraMountAbility>1</CameraMountAbility>
<!--req, camera mount capability, 1- support, not displayed if not supported-->
<VehicleRecogAbility>1</VehicleRecogAbility>
<!--req, vehicle secondary detection capability, 1- support, not displayed if not supported-->
<VcaChanAbility>1</VcaChanAbility>
<!--req, VCA smart channel capability, 1- support, not displayed if not supported-->
<Language><!--req Type of language-->
<supportType opt="0-noSupport,1-chinese,2-english"/>
<!--req 1-chinese,2-English-->
</Language>
<LongLinkConfigurationFile>1</LongLinkConfigurationFile>
<!--req, whether to support long connection import and export configuration file, 1- yes, not displayed if not supported-->
<IpViewDev>1</IpViewDev>
<!--req, support IP video intercom host capability set, corresponding to IpViewDevAbility-->
<TransDevice>
<transChannelNum min = "" max = "" />
<!--req, max. and min. value of transcode channel-->
</TransDevice>
<MultiNetworkCardMode>0,1</MultiNetworkCardMode>
<!--req, supported multi-NIC operating mode, 0- normal, 1- LAN & WAN isolation, not displayed if not supported-->
<DeviceWorkMode>1,2,3</DeviceWorkMode>
<!--req, device-supported operating mode, 1- channel mode, 2- stream ID mode, not displayed if not supported-->
<AllBackupLog>
<enabled>true</enabled>
<!--req, whether to support exporting all logs-->
</AllBackupLog>
<VoiceTalkAsAudioIn>
<enabled>true</enabled>
<!--req, whether to support two-way audio as audio input-->
<supportVoiceChan opt="1,2"/>
<!--opt, No. of supported two-way audio channel-->
</VoiceTalkAsAudioIn>
<Matrix>1</Matrix>
<!--req , support matrix capability set MATRIX_ABILITY, not displayed if not supported-->
<VCADetection opt="true,false"/>
<!--req, whether to support VCA detection-->
<PDC opt="true,false"/><!--req, whether supports people counting function-->
<TrialHostAbility>1</TrialHostAbility>
<!--req, support interrogation host capability, 1- support, not displayed if not supported-->
<BinocularSupport>1</BinocularSupport>
<!--req, support stereo cameras, 1- support, not displayed if not supported-->
<supportChoosePlaybackDrawframe>true</supportChoosePlaybackDrawframe>
<!--opt, whether supports extracting frames for playback, if not supported, no return-->
<isNotSupportSummerTime>true</isNotSupportSummerTime>
<!--opt, return this capability if DST is not supported-->
<BrokenNetHttpSupport opt="postMPR,PDC,heatMapDetection"/>
<!--opt,xs:string, support ANR, vehicle detection under MPR mode, people counting, heat map detection-->
<isSupportSyncIPCPassword>true</isSupportSyncIPCPassword><!--opt, whether to support IP camera password sync, not returned if not supported-->
```

```
<isSupportTransferIPC>true</isSupportTransferIPC><!--opt, whether to support transparent transmission of IP  
camera protocol function, not returned if not supported-->  
<supportPreviewHRUDP>true</supportPreviewHRUDP><!--opt, whether to support HRUDP live view streaming  
mode, not returned if not supported-->  
    <isSupportTimeCorrect>true</isSupportTimeCorrect>  
        <!--opt,Whether to support the timing operation of NET_DVR_SET_TIMECORRECT-->  
<HRUDP><!--opt-->  
    <LinkList>  
        <!--req-->  
        <Link>  
            <!--,req-->  
            <previewLink>1</previewLink>  
            <!--,req-->  
            <recommendResolution opt="39-1920*1080"></recommendResolution>  
            <!--,req-->  
        </Link>  
    </LinkList>  
</HRUDP>  
<!--opt, SD card unlocked time, if the time is uploaded, it refers to support, if not supported, it will not be  
displayed.-->  
<SDCardUnlockTime>3</SDCardUnlockTime>  
<isSupportOnLineUser>true</isSupportOnLineUser>  
<!--opt, device supports getting online user-->  
<NeedReboot>  
    <!-- req, auto restart after importing configuration file, 1- yes, 2- with prompt, not displayed if not supported-->  
    <ImportConfigurationFileReboot>1</ImportConfigurationFileReboot>  
    <!-- req, auto restart after ESATA application is modified, 1- restart, not displayed if not supported-->  
    <EsataUseageChange>1</EsataUseageChange>  
    <!-- req, auto restart after alarm input type is changed, 1- restart, 2- with prompt, not displayed if not supported-->  
    <AlarmInTypeChange>1</AlarmInTypeChange>  
    <!-- req, auto restart after modification in analog channel, 1- restart, not displayed if not supported -->  
    <AnalogChanEnableChange>1</AnalogChanEnableChange>  
    <!-- req, auto restart after restoring default settings, 1- restart, 2- with prompt, not displayed if not supported-->  
    <RestoreConfig>1</RestoreConfig>  
    <!-- req, auto restart after transmission mode of RS232 serial port is modified, 1- restart, not displayed if not  
supported-->  
    <RS232workModeChange>1</RS232workModeChange>  
    <!-- req, auto restart after network transmission port is modified, 1- restart, not displayed if not supported-->  
    <NetPortChange>1</NetPortChange>  
    <!-- req, auto restart after RTSP port is modified, 1- restart, not displayed if not supported-->  
    <RtspPortChange>1</RtspPortChange>  
    <!-- req, auto restart after DHCP status is modified, 1- restart, not displayed if not supported-->  
    <DhcpEnableChange>1</DhcpEnableChange>  
    <!-- req, auto restart after HTTP port is modified, 1- restart, not displayed if not supported -->  
    <HttpPortChange>1</HttpPortChange>  
    <!-- req, auto restart after PPPoE parameters are modified, 1- restart, not displayed if not supported -->  
    <PPPoEChange>1</PPPoEChange>  
    <!-- req, auto restart after multicast address is modified in network parameters, 1- restart, not displayed if not  
supported -->  
    <NetMultiCastIPChange>1</NetMultiCastIPChange>  
    <!-- req, auto restart after HDD parameters are modified, 1- restart, not displayed if not supported -->  
    <HardDiskParamChange>1</HardDiskParamChange>
```

```
<!-- req, auto restart after recording schedule time is modified, 1- restart, not displayed if not supported -->
<RecordTimeChange>1</RecordTimeChange>
<!-- req, auto restart after video compression type is modified, 1- restart, not displayed if not supported -->
<VideoEncodeTypeChange>1</VideoEncodeTypeChange>
<!-- req, auto restart after audio compression type is modified, 1- restart, not displayed if not supported -->
<AudioEncodeTypeChange>1</AudioEncodeTypeChange>
<!-- req, auto restart after video standard is modified, 1- restart, not displayed if not supported -->
<StandardTypeChange>1</StandardTypeChange>
<!-- req, auto restart after defog status is modified, 1- restart, not displayed if not supported -->
<DehazeEnableChange>1</DehazeEnableChange>
<!-- req, auto restart after line encoding status is modified, 1- restart, not displayed if not supported -->
<LineCodingEnableChange>1</LineCodingEnableChange>
<!-- req, auto restart after IP camera parameters local output switch status is modified, 1- restart, not displayed if
not supported -->
<LocalOutputEnableChange>1</LocalOutputEnableChange>
<!-- req, auto restart after main port is switched, 1- restart, not displayed if not supported -->
<LocalMainVoutTypeChange>1</LocalMainVoutTypeChange>
<!-- req, auto restart after NAS authentication parameters are modified, 1- restart, not displayed if not supported -->
>
<NASIdentificationChange>1</NASIdentificationChange>
<DeviceLanguageChange>
    <!--whether the reboot is required after editing device language, 1-required, this node does not exist-not
required-->
</DeviceLanguageChange>
<CardReaderFPAlgorithmUpgradeReboot>
    <!--opt, whether the reboot is required after upgrading the fingerprint algorithm program of the fingerprint
module: 1-required, this node does not exist-not required-->
</CardReaderFPAlgorithmUpgradeReboot>
<DeviceUpgradeReboot>
    <!--opt, whether the device will automatically reboot after upgrading: 1-yes (the device will automatically reboot
no matter whether the upgrading succeeded or not, this node does not exist-no)-->
</DeviceUpgradeReboot>
<!--whether the device restarts after switching the VoIP protocol of the video intercom device: 1-restart (the user
needs to click "Confirm" and the application will apply the command to restart the device), 2-restart automatically
(the upper-layer application applies the command to restart the device directly). If there is no need to restart the
device, this field will not be displayed-->
<IntercomProtocolTypeChange>1</IntercomProtocolTypeChange>
<!-- req, auto restart after operating mode is modified, 1- restart, not displayed if not supported -->
<DevWorkModeChange>1</DevWorkModeChange>
<!-- req, auto restart after SIP local port is modified, 1- restart, not displayed if not supported -->
<LocalPortChange>1</LocalPortChange>
<!-- req, auto restart after SIP server register cycle is modified, 1- restart, not displayed if not supported -->
<LoginCycleChange>1</LoginCycleChange>
<!-- req, auto restart after RTP port is modified, 1- restart, not displayed if not supported -->
<RtpPortChange>1</RtpPortChange>
<!-- req, auto restart after audio compression priority level is modified, 1- restart, not displayed if not supported -->
<AudioEncodePriorityChange>1</AudioEncodePriorityChange>
<!-- req, auto restart after preview delay time is modified, 1- restart, not displayed if not supported -->
<PreviewDelayTimeChange>1</PreviewDelayTimeChange>
<!-- req, auto restart after NIC type is modified, 0- no -->
<NetworkCardTypeChange>0</NetworkCardTypeChange>
<!--req, auto restart after device cloud storage mode is modified, 1- restart, not displayed if not supported-->
```

```
<CloudStorageModeChangeReboot>1</CloudStorageModeChangeReboot>
<VcaDecModeChange>1</VcaDecModeChange>
<!--req, auto restart after enabling smart decoding mode, 1- restart, not displayed if not supported-->
<CompleteRestoreReboot>1</CompleteRestoreReboot>
<!--req, auto restart after restoring to factory defaults, 1- restart, not displayed if not supported-->
<SwitchVehicleDetection>1</SwitchVehicleDetection>
<!--req, auto restart after vehicle detection mode is modified, 1- restart, not displayed if not supported-->
<SwitchHVTVehicleDetection>1</SwitchHVTVehicleDetection>
<!--req, auto restart after mixed vehicle detection mode is switched, 1- restart, not displayed if not supported-->
<SmartCodec>1</SmartCodec>
<!--req, auto restart after high performance compression detection mode is switched, 1- restart, not displayed if
not supported-->
<NUCLEUSToOther>1</NUCLEUSToOther>
<!--opt, auto restart after switching from nucleus to other protocol, 1- restart, not displayed if not supported-->
<SystemSwitchReboot>1</SystemSwitchReboot>
<!--opt, whether to reboot when switching video intercom system, 1-reboot, this node will not be displayed if
reboot is not required-->
<TerminalModeReboot>1</TerminalModeReboot>
<!--opt, whether to reboot when the face recognition terminal switching the terminal mode, 1-reboot, this node
will not be displayed if reboot is not required-->
<ThirdStream>1</ThirdStream>
<!--opt, whether to restart after third stream is enabled, 1- yes, not displayed if not supported-->
<ExtensionModuleUpgradeReboot>1</ExtensionModuleUpgradeReboot>
<!--opt, whether to reboot after upgrading extended module: 1-Reboot. If there is no need to reboot, this node
will not be displayed-->
<WorkModeReboot>
  <!--opt, whether the reboot is required when switching working mode of intelligent identity detection terminal,1-
yes, if reboot is not required, this node will not return-->
</WorkModeReboot>
<MultiFaceRecognizeChange><!--opt, whether the reboot is required when switching multi-face recognition,1-yes, if
reboot is not required, this node will not return--></MultiFaceRecognizeChange>
<FaceRecognizeModeChange><!--whether reboot is required after switching facial recognition mode: 1-yes, 2-
device reboots automatically, this node is not returned if reboot is not required--></FaceRecognizeModeChange>
<ChannelControllerUpgradeReboot>1</ChannelControllerUpgradeReboot>
<!--opt, whether to reboot after upgrading lane controller: 1- Reboot. If there is no need to reboot, this node will
not be displayed-->
<UbootUpgradeReboot><!--opt, whether to reboot after upgrading uboot: 1-reboot. If there is no need to reboot,
this node will be displayed--></UbootUpgradeReboot>
</NeedReboot>
<SupportMainBoardCameraPTZ>true</SupportMainBoardCameraPTZ><!--opt, PTZ that supports camera control of
master controller-->
<Support3DPTZ>true</Support3DPTZ>
<!--opt, whether to support 3D64 speed PTZ control-->
<SupportUnifiedIPCProtocol>true</SupportUnifiedIPCProtocol>
<!--opt, whether to support unified IPC protocol, using NET_DVR_IPC_ENUM_UNIFY type-->
<SupportRemoveStorage>true</SupportRemoveStorage>
<!--opt, remove device storage function (including video, playback, picture and log) xs:boolean -->
<CloudSupport>1</CloudSupport>
<!--opt, whether to support third-party cloud: 1- yes, not displayed if not supported-->
<isSupportBatchUploadPic>true</isSupportBatchUploadPic>
<!--opt, whether to support batch upload picture (BATCH_UPLOAD_PICTURE_FILE), not displayed if not supported-->
>
```

```
<isSupportDownloadVehicleInfo>true</isSupportDownloadVehicleInfo>
<!--opt, whether to support vehicle information search (NET_DVR_GET_VEHICLE_INFORMATION), not displayed if not supported-->
<isSupportConfirmMechanism>true</isSupportConfirmMechanism>
<!--Whether supports confirming arming, if not supports, no return.-->
<isSupportDNS>
  <!--opt, whether to support enabling manual DNS settings, 1- yes, not displayed if not supported--></isSupportDNS>
    <HRUDP>
      <!--opt,HRUDP (reliable transmission) capability-->
      <LinkList>
        <!--req,HRUDP connection list-->
        <Link>
          <!--req,HRUDP connection-->
          <previewLink>1</previewLink>
          <!--req,HRUDP number of preview connections-->
          <recommendResolution opt="39-1920*1080"></recommendResolution>
          <!--req,recommended resolution-->
        </Link>
      </LinkList>
    </HRUDP>
    <isSupportCountBinocular>true</isSupportCountBinocular>
    <!--opt,"supported means the device is dual-lens people counting camera, not displayed if not supported"-->
    <isSupportUploadCountBinocular>true</isSupportUploadCountBinocular>
    <!--opt,"If supports, the dual-lens people counting camera list will be uploaded, otherwise, no return."-->
    <supportChoosePlaybackDrawframe>true</supportChoosePlaybackDrawframe>
    <!--opt, whether supports extracting frames for playback, if not supported, no return-->
    <isSupportMinSAS>true</isSupportMinSAS>
    <isSupportMinSAS>true</isSupportMinSAS>
    <!--req, whether to support miniSAS HDD-->
    <isNotSupportSummerTime>true</isNotSupportSummerTime>
    <!--opt,"return this capability when DST is not supported"-->
    <isSupportPanoramicStitch>true</isSupportPanoramicStitch>
    <!--opt, whether to support uploading panoramic view configurationfile function, not returned if not supported-->
    <isSupportGetFigure/>
    <!--opt,xs:boolean, whether supports getting video thumbnails-->
    <isSupportGetThumbnails>
      <!--opt, whether supports getting video thumbnails by time segment-->
      <isSupportStreamID opt="true,false"/>
      <!--opt, whether supports stream ID-->
      <resultDataType opt="0,1"/><!--opt, search type: 0-I frame, 1-picture data-->
      <StartTime>
        <year min="" max="" />
        <month min="" max="" />
        <day min="" max="" />
        <hour min="" max="" />
        <minute min="" max="" />
        <second min="" max="" />
      </StartTime>
      <StopTime>
        <!--opt, search end time-->
        <year min="" max="" />
```

```
<month min="" max="" />
<day min="" max="" />
<hour min="" max="" />
<minute min="" max="" />
<second min="" max="" />
</StopTime>
<fileIntervalTime min="0" max="24*60*60"/>
<!--opt, video duration, unit: s, integer-->
</isSupportGetThumbnails>
<isSupportPrisonPanorama>true</isSupportPrisonPanorama>
<!--opt, "If supports, it means the device is PanoVo camera (cell); otherwise, no return."-->
<isSupportSystemLogFileExport>true</isSupportSystemLogFileExport>
<!--opt, Whether supports exporting system logs.If not supports, no return. -->
<isSupportNewFaceResult>true</isSupportNewFaceResult>
<!--opt, Whether supports representing age (age + age error) by new method. If not supports, no return-->
<isSupportSmoothDragging>
<!--opt, whether supports smooth drag during playback, xs:boolean,"true"-->
</isSupportSmoothDragging>
<PlaybackDrawframeCap><!--opt, supports extracting frames for playback-->
<drawType opt="0,1,2,3,4,5"/><!--opt, Frame extracting mode: 0-Only transfer I frame, 1-Drop 1/2 P frames (only supporte by SVC stream), 2-Drop 3/4 P frames (only supporte by SVC stream), 3-Transfer 1/2 I frames (only transfer I frames, and transfer one of each 2 I frames), 4-Treansfer 1/4 I frames (only transfer I frames, and transfer one of each 4 I frames), 5-Transfer 1/8 I frames (only transfer I frames, and transfer one of each 8 I frames)-->
</PlaybackDrawframeCap>
<isSupportIPCTiming>
<!--opt, xs:boolean,"true,false", whether supports the byEnableTiming field in access protocol configuration of network camera-->
</isSupportIPCTiming>
<isSupportEncryption>
<!--opt, whether supports stream encryption function, xs:boolean,"true"-->
</isSupportEncryption>
<isSupportMultiChannelSearch>
<!--opt, whether supports people counting of multiple channels, it is valid only when the value of PDC is "true"-->
</isSupportMultiChannelSearch>
<isSupportEncryption><!--opt, whether supports stream encryption, xs: boolean, "true"--></isSupportEncryption>
<isSupportClientProxyWEB>
<!--opt, xs:boolean, whether to support the client agent to pass through the remote web configuration: "true"-yes-->
</isSupportClientProxyWEB>
<WEBLocation>
<!--opt, string, web page location: "local"-local device, "remote". If this node is not returned, the web page location will be the local device by default-->
</WEBLocation>
<deviceId>
<!--opt, string, device ID. If WEBLocation is "remote", the unique device ID should be returned to mark the unique web package corresponding to the device. The deviceId is managed by web configuration package, and deviceId returned by different models of one device is the same-->
</deviceId>
</SoftwareCapability>
</BasicCapability>
```

F.164 XML_CAMERAPARA

CAMERAPARA message in XML format

```
<xml version="1.0" encoding="utf-8"?>
<!--req, camera parameter capability set description -->
<CAMERAPARA version="2.0">
  <ChannelList>
    <ChannelEntry>
      <ChannelNumber>1</ChannelNumber>
      <IPStartChanNoDefault><!--optional, xs:integer, start digital channel No. The device's digital channel No. will start from the returned No.--></IPStartChanNoDefault>
      <PowerLineFrequencyMode><!-- req, format -->
        <isNotSupportDigitalChanCfg opt="true,false"/><!--optional, whether setting digital channels is not supported: true-yes (not supported), false-no (supported). If this field is not returned, it indicates that setting digital channels is supported-->
        <Range>0,1</Range><!-- req, 0-50HZ; 1-60HZ -->
        <Default>0</Default><!-- req, default value -->
      </PowerLineFrequencyMode>
      <CaptureMode>
        <!--req, correspond to byCaptureMode in NET_DVR_CAMERAPARAMCFG_EX-->
        <!--req, the device supports captureModePWithIndex and captureModeNWithIndex, when returning captureModeP and captureModeN, the client resolves the capability set, first with captureModePWithIndex and captureModeNWithIndex, or captureModeP and captureModeN if the former nodes are not supported-->
        <!--req, 0-close, 1-640*480@25fps, 2-640*480@30fps, 3-704*576@25fps, 4-704*480@30fps, 5-1280*720@25fps, 6-1280*720@30fps, 7-1280*720@50fps, 8-1280*720@60fps, 9-1280*960@15fps, 10-1280*960@25fps, 11-1280*960@30fps, 12-1280*1024@25fps, 13-1280*1024@30fps, 14-1600*900@15fps, 15-1600*1200@15fps, 16-1920*1080@15fps, 17-1920*1080@25fps, 18-1920*1080@30fps, 19-1920*1080@50fps, 20-1920*1080@60fps, 21-2048*1536@15fps, 22-2048*1536@20fps, 23-2048*1536@24fps, 24-2048*1536@25fps, 25-2048*1536@30fps, 26-2560*2048@25fps, 27-2560*2048@30fps, 28-2560*1920@7.5fps, 29-3072*2048@25fps, 30-3072*2048@30fps, 31-2048*1536@12.5, 32-2560*1920@6.25, 33-1600*1200@25, 34-1600*1200@30, 35-1600*1200@12.5, 36-1600*900@12.5, 37-1600@900@15, 38-800*600@25, 39-800*600@30fps, 136-640*960@25fps, 137-640*960@24fps, 142-2992*2192@25fps, 143-2992*2192@30fps, 158-384*288@8.3fps, 159-640*512@8.3fps, 160-160*120@8.3fps, 161-1024*768@8.3fps, 162-640*480@8.3fps-->
        <captureModeP opt="close,640*480@25fps,640*480@30fps,704*576@25fps,704*480@30fps,1280*720@25fps, 1280*720@30fps,1280*720@50fps,1280*720@60fps,1280*960@15fps,1280*960@25fps, 1280*960@30fps,1280*1024@25fps,1280*1024@30fps,1600*900@15fps,1600*1200@15fps, 1920*1080@15fps,1920*1080@25fps,1920*1080@30fps,1920*1080@50fps,1920*1080@60fps, 2048*1536@15fps,2048*1536@20fps,2048*1536@24fps, 2048*1536@25fps,2048*1536@30fps, 2560*2048@25fps,2560*2048@30fps,2560*1920@7.5fps,3072*2048@25fps, 3072*2048@30fps, 2048*1536@12.5fps,2560*1920@6.25fps,1600*1200@25fps,1600*1200@30fps, 1600*1200@12.5fps, 1600*900@12.5fps,1600@900@15fps,800*600@25fps,800*600@30fps,640*960@25fps, 640*960@24fps"/>
        <!--req, The value of captureMode in P standard-->
        <captureModeN opt="close,640*480@25fps,640*480@30fps,704*576@25fps,704*480@30fps, 1280*720@25fps, 1280*720@30fps,1280*720@50fps,1280*720@60fps,1280*960@15fps,1280*960@25fps, 1280*960@30fps,1280*1024@25fps,1280*1024@30fps,1600*900@15fps,1600*1200@15fps, 1920*1080@15fps,1920*1080@25fps,1920*1080@30fps,1920*1080@50fps,1920*1080@60fps, 2048*1536@15fps,2048*1536@20fps,2048*1536@24fps, 2048*1536@25fps,2048*1536@30fps, 2560*2048@25fps,2560*2048@30fps,2560*1920@7.5fps,3072*2048@25fps, 3072*2048@30fps, 2048*1536@12.5fps,2560*1920@6.25fps,1600*1200@25fps,1600*1200@30fps, 1600*1200@12.5fps, 1600*900@12.5fps,1600@900@15fps,800*600@25fps,800*600@30fps,640*960@25fps, 640*960@24fps"/>
```

```

<!--req, The value of captureMode in N standard-->
<captureModePWithIndex opt="0-close, 1-640*480@25fps,2-640*480@30fps,3-704*576@25fps,
4-704*480@30fps,5-1280*720@25fps, 6-1280*720@30fps,7-1280*720@50fps,8-1280*720@60fps,
9-1280*960@15fps,10-1280*960@25fps, 11-1280*960@30fps,12-1280*1024@25fps,13-1280*1024@30fps,
14-1600*900@15fps,15-1600*1200@15fps, 16-1920*1080@15fps,17-1920*1080@25fps,18-1920*1080@30fps,
19-1920*1080@50fps,20-1920*1080@60fps, 21-2048*1536@15fps,22-2048*1536@20fps,23-2048*1536@24fps,
24-2048*1536@25fps,25-2048*1536@30fps, 26-2560*2048@25fps,27-2560*2048@30fps,28-2560*1920@7.5fps,
29-3072*2048@25fps,30-3072*2048@30fps, 31-2048*1536@12.5fps,32-2560*1920@6.25fps,33-1600*1200@25fps,
34-1600*1200@30fps,35-1600*1200@12.5fps, 36-1600*900@12.5fps,37-1280*960@12.5fps,38-800*600@25fps,
39-800*600@30fps,40-4000*3000@12.5fps, 41-4000*3000@15fps,42-4096*2160@20fps,43-3840*2160@20fps,
44-960*576@25fps,45-960*480@30fps, 46-752*582@25fps,47-768*494@30fps,48-2560*1440@25fps,
49-2560*1440@30fps,50-720P@100fps, 51-720P@120fps,52-2048*1536@50fps,53-2048*1536@60fps,
54-3840*2160@25fps,55-3840*2160@30fps, 56-4096*2160@25fps,57-4096*2160@30fps,58-1280*1024@50fps,
59-1280*1024@60fps,60-3072*2048@50fps, 61-3072*2048@60fps,62-3072*1728@25fps,63-3072*1728@30fps,
64-3072*1728@50fps,65-3072*1728@60fps, 66-336*256@50fps,67-336*256@60fps,68-384*288@50fps,
69-384*288@60fps,70-640*512@50fps, 71-640*512@60fps,72-2592*1944@25fps,73-2592*1944@30fps,
74-2688*1536@25fps,75-2688*1536@30fps, 76-2592*1944@20fps,77-2592*1944@15fps,78-2688*1520@20fps,
79-2688*1520@15fps,80-2688*1520@25fps, 81-2688*1520@30fps,82-2720*2048@25fps,83-2720*2048@30fps,84-
336*256@25fps,85-384*288@25fps, 86-640*512@25fps,87-1280*960@50fps,88-1280*960@60fps,
89-1280*960@100fps,90-1280*960@120fps, 91-4000*3000@20fps,141-2688*1520@12.5fps"/>

<!--req, captureMode value with index in P standard-->
<captureModeNWithIndex opt="0-close,1-640*480@25fps,2-640*480@30fps,3-704*576@25fps,
4-704*480@30fps,5-1280*720@25fps, 6-1280*720@30fps,7-1280*720@50fps,8-1280*720@60fps,
9-1280*960@15fps,10-1280*960@25fps, 11-1280*960@30fps,12-1280*1024@25fps,13-1280*1024@30fps,
14-1600*900@15fps,15-1600*1200@15fps, 16-1920*1080@15fps,17-1920*1080@25fps,18-1920*1080@30fps,
19-1920*1080@50fps,20-1920*1080@60fps, 21-2048*1536@15fps,22-2048*1536@20fps,23-2048*1536@24fps,
24-2048*1536@25fps,25-2048*1536@30fps, 26-2560*2048@25fps,27-2560*2048@30fps,28-2560*1920@7.5fps,
29-3072*2048@25fps,30-3072*2048@30fps, 31-2048*1536@12.5fps,32-2560*1920@6.25fps,33-1600*1200@25fps,
34-1600*1200@30fps,35-1600*1200@12.5fps, 36-1600*900@12.5fps,37-1280*960@12.5fps,38-800*600@25fps,
39-800*600@30fps,40-4000*3000@12.5fps, 41-4000*3000@15fps,42-4096*2160@20fps,43-3840*2160@20fps,
44-960*576@25fps,45-960*480@30fps, 46-752*582@25fps,47-768*494@30fps,48-2560*1440@25fps,
49-2560*1440@30fps,50-720P@100fps, 51-720P@120fps,52-2048*1536@50fps,53-2048*1536@60fps,
54-3840*2160@25fps,55-3840*2160@30fps, 56-4096*2160@25fps,57-4096*2160@30fps,58-1280*1024@50fps,
59-1280*1024@60fps,60-3072*2048@50fps, 61-3072*2048@60fps,62-3072*1728@25fps,63-3072*1728@30fps,
64-3072*1728@50fps,65-3072*1728@60fps, 66-336*256@50fps,67-336*256@60fps,68-384*288@50fps,
69-384*288@60fps,70-640*512@50fps, 71-640*512@60fps,72-2592*1944@25fps,73-2592*1944@30fps,
74-2688*1536@25fps,75-2688*1536@30fps, 76-2592*1944@20fps,77-2592*1944@15fps,78-2688*1520@20fps,
79-2688*1520@15fps,80-2688*1520@25fps, 81-2688*1520@30fps,82-2720*2048@25fps,83-2720*2048@30fps,84-
336*256@25fps,85-384*288@25fps, 86-640*512@25fps,87-1280*960@50fps,88-1280*960@60fps,
89-1280*960@100fps,90-1280*960@120fps, 91-4000*3000@20fps,141-2688*1520@12.5fps"/>

<!--req, captureMode value with index in N standard-->

<!--req, to enable 3D noise reduction, SMD, rotation or WDR in 1080p50/1080p60 mode, the prompt will show
"please set capture mode with normal frame rate"-->
<CaptureModeIndex19>
<!--1920*1080@50fps-->
<!--req, mutually exclusive capability, digital noise reduction, line crossing detection, rotation, WDR access
capability-->
<mutexAbility opt="digitalNoiseReduction,traversingVirtualPlane,fieldDetection,corridorMode,WDR"/>
</CaptureModeIndex19>

<CaptureModeIndex20>

```

```
<!--req, 1920*1080@60fps-->
<!--req, mutually exclusive capability, digital noise reduction, line crossing detection, rotation, WDR access
capability-->
<mutexAbility opt="digitalNoiseReduction,traversingVirtualPlane,fieldDetection,corridorMode,WDR"/>
</CaptureModelIndex20>

<!-req, to enable rotation or WDR in 720p50/720p60 mode, the prompt will show "please set capture mode with
normal frame rate"-->
<CaptureModelIndex7>
<!--1280*720@50fps-->
<!--req mutually exclusive capability rotation WDR access capability-->
<mutexAbility opt="corridorMode,WDR"/>
</CaptureModelIndex7>

<CaptureModelIndex8>
<!--1280*720@60fps-->
<!--req mutually exclusive capability rotation WDR access capability-->
<mutexAbility opt="corridorMode,WDR"/>
</CaptureModelIndex8>

<CaptureModelIndex52>
<!--2048*1536@50fps-->
<!--req mutually exclusive capability rotation WDR access capability-->
<mutexAbility opt="WDR"/>
</CaptureModelIndex52>

<CaptureModelIndex53>
<!--2048*1536@60fps -->
<!--req mutually exclusive capability rotation WDR access capability-->
<mutexAbility opt="WDR"/>
</CaptureModelIndex53>

<CaptureModelIndex87>
<!--1280*960@50fps -->
<!--req mutually exclusive capability WDR access capability-->
<mutexAbility opt="WDR"/>
</CaptureModelIndex87>

<CaptureModelIndex88>
<!--1280*960@60fps -->
<!--req mutually exclusive capability WDR access capability-->
<mutexAbility opt="WDR"/>
</CaptureModelIndex88>

<CaptureModelIndex89>
<!--1280*960@100fps -->
<!--req mutually exclusive capability WDR access capability-->
<mutexAbility opt="WDR"/>
</CaptureModelIndex89>

<CaptureModelIndex90>
<!--1280*960@120fps -->
```

```

<!--req mutually exclusive capability WDR access capability-->
<mutexAbility opt="WDR"/>
</CaptureModelIndex90>
<WhiteBalance><!-- req, white balance -->
<WhiteBalanceMode><!-- req, white balance mode -->
<!-- req, 0-Manual, 1-AWB1, 2-AWB2, 3-Automatic Control(4~9:reserved), 11-Auto Trace,12-One Push,13-Indoor,
14-Outdoor, 15-Outdoor Auto, 16- SodiumLight Auto -->
<!-- req, 10~16: new options for speed domes -->
<Range>1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16</Range>
<Default>1</Default><!-- req, default value -->
</WhiteBalanceMode>
<WhiteBalanceModeRGain><!-- req, R gain of white balance -->
<Min>0</Min><!-- req, minimum value -->
<Max>255</Max><!-- req, maximum value -->
<Default>100</Default><!-- req, default value -->
</WhiteBalanceModeRGain>
<WhiteBalanceModeBGain><!-- req, B gain of white balance -->
<Min>0</Min><!-- req, minimum value -->
<Max>255</Max><!-- req, maximum value -->
<Default>100</Default><!-- req, default value -->
</WhiteBalanceModeBGain>
</WhiteBalance>

<!-- req, supported by IPC only -->
<Exposure> <!-- req, exposure -->
<ExposureMode> <!-- req, exposure mode, reserved currently -->
<Range>0,1</Range><!-- req, 0-manual exposure,1-auto exposure -->
<Default>0</Default><!-- req, default value -->
</ExposureMode>

<ExposureSet>
<!-- req, exposure time, 0(index): auto*8(display on the client), 40000*8 us(the actual value)-->
<!-- req,0-auto*8(40000*8us),1-auto*5(40000*5us),2-auto*4(40000*4us),3-auto*2(40000*2us),-->
<!-- req,4-1/25(40000us),5-1/50(20000us),6-1/100(10000us),7-1/250(4000us),8-1/500(2000us),-->
<!-- req,9-1/750(1333us),10-1/1000(1000us),11-1/2000(500us),12-1/4000(250us),-->
<!-- req,13-1/10,000(100us),14-1/100,000(10us), 17-1/150, 18-1/200, 20-1-1000000us, 21-1/75, 22-1/125,
23-1/175, 24-1/225,25-1/300, 26-1/400 -->
<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14</Range>
<Default>4</Default><!-- req, default value-->
<DynamicAbility>
<!--req, IPC 5.1.0 supports to get or set abilities dynamically-->
<dynamicAbilityLinkTo opt="wdrEnable,irisType"/>
<!--req, Dynamic related items, WDR enable, the structure used for defining the type of len is
NET_DVR_CAMERAPARAMCFG_EX, the parameters is struWdr.byWDREnabled and struWdr.byIrisMode -->
</DynamicAbility>
</ExposureSet>
<exposureUSERSET><!--req, customized exposure time-->
<Min>1</Min><!--req, minimum value-->
<Max>40000</Max><!--req, maximum value-->
<Default>20000</Default><!--req, default value-->
</exposureUSERSET>
<ExposureTarget> <!--req, reserved-->

```

```

<Min>0</Min><!-- req, minimum value -->
<Max>2000000</Max><!-- req, maximum value -->
<Default>1000000</Default><!-- req, default value -->
</ExposureTarget>
</Exposure>
<IrisMode><!--req, Lens mode-->
  <!--req, 0- auto iris, 1- manual iris, 2- Piris1 "Tamron 2.8-8mm F1.2 (M13VP288-IR)", 3- Union 3-9mm F1.6-2.7
  (T5280-PQ1), 4- Union 2.8-12mm F1.6-2.7 (T5289-PQ1), 5- Private 3.8-16mm F1.5 (HV3816P-8MPIR), 6-Private
  11-40mm F1.7, 7- Private 2.7-12mm F1.2 (TV2712P-MPIR), 8- MZ5721D-12MPIR, 9- MZ1555D-12MPIR, 10-
  MZ5721D-12MPIR(RS485), 11- MZ1555D-12MPIR(RS485)-->
  <Range>0,1,2,3,4,5,6,7,8,9,10,11</Range>
  <Default>1</Default><!-- req, default value -->
<Piris>
  <!--req valid when IrisMode>=2-->
  <Piris1><!--req, Tamron 2.8-8mm F1.2 (M13VP288-IR)-->
    <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
    <PirisAperture min="" max="" /><!--req, level range: 1 to 100 (can be configured under the manual mode)-->
  </Piris1>
  <Piris2><!--req, Union 3-9mm F1.6-2.7 (T5280-PQ1)-->
    <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
    <PirisAperture min="" max="" /><!--req, level range: 1 to 100 (can be configured under the manual mode)-->
  </Piris2>
  <Piris3><!--req, Union 2.8-12mm F1.6-2.7 (T5289-PQ1)-->
    <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
    <PirisAperture min="" max="" /><!--req, level range: 1 to 100 (can be configured under the manual mode)-->
  </Piris3>
  <Piris4><!--req, Private 3.8-16mm F1.5 (HV3816P-8MPIR)-->
    <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
    <PirisAperture min="" max="" /><!--req, level range: 1 to 100 (can be configured under the manual mode)-->
  </Piris4>
  <Piris6><!--req, Private 11-40mm F1.7 (HV1140P-8MPIR)-->
    <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
    <PirisAperture min="" max="" /><!--req, level range: 1 to 100 (can be configured under the manual mode)-->
  </Piris6>
  <Piris7><!--req, Private 2.7-12mm F1.2 (TV2712P-MPIR)-->
    <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
    <PirisAperture min="" max="" /><!--req, level range: 1 to 100 (can be configured under the manual mode)-->
  </Piris7>
</Piris>
</IrisMode>

<AutoApertureLevel> <!-- req, auto aperture sensitivity -->
<Min>0</Min><!-- req, minimum value -->
<Max>15</Max><!-- req, maximum value -->
<Default>7</Default><!-- req, default value -->
</AutoApertureLevel>

<FocusMode> <!--req, reserved-->
  <!--req, 0- manual focus; 1-auto focus; 2- auto back focus-->
  <Range>0,1,2</Range>
  <Default>0</Default><!--req, default value-->
  <GainLevel><!--req, gain, ranges from 0 to 100-->

```

```

<Min>0</Min><!--req, minimum value-->
<Max>100</Max><!--req, maximum value-->
<Default>50</Default><!-- req, default value-->
</GainLevel>
<BrightnessLevel><!--req, brightness, ranges from 0 to 100 -->
<Min>0</Min><!-- req, minimum value -->
<Max>100</Max><!-- req, maximum value -->
<Default>50</Default><!-- req, default value -->
</BrightnessLevel>
<ContrastLevel><!--req, contrast, ranges from 0 to 100-->
<Min>0</Min><!--req, minimum value-->
<Max>100</Max><!--req, maximum value-->
<Default>50</Default><!--req, default value-->
</ContrastLevel>
<SharpnessType><!--sharpness type-->
<Range><!--sharpness type range (for speed dome): 0-automatic, 1-manual--></Range>
<Default><!--default value--></Default>
</SharpnessType>
<SharpnessLevel><!--req, sharpness-->
<!--req, IPC(min/max); speed dome/zoom camera(Range), from 0 to 100-->
<Min>0</Min><!-- req, minimum value -->
<Max>100</Max><!-- req, maximum value -->
<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17</Range>
<!-- req, speed dome: 0- auto, 1- manual, 2-1, 3-2, 4-3, 5-4, 6-5, 7-6, 8-7, 9-8, 10-9, 11-10, 12-11, 13-12, 14-13,
15-14, 16-15, 17-16 -->
<Default>50</Default><!-- req, default value -->
</SharpnessLevel>
<HorizonAperture> <!-- req, horizontal sharpness -->
<!-- req, special for speed dome: 1~64 -->
<Min>1</Min><!-- req, minimum value -->
<Max>64</Max><!-- req, maximum value -->
<Default>10</Default><!-- req, default value -->
</HorizonAperture>
<VerticalAperture> <!-- req, vertical sharpness -->
<!-- req, special for speed dome: 1~64 -->
<Min>1</Min><!-- req, minimum value -->
<Max>64</Max><!-- req, maximum value -->
<Default>10</Default><!-- req, default value -->
</VerticalAperture>

<LaserConfig>
<controlMode opt="auto,manual"/>
<autoMode><!--req,Control Mode-->
<sensitivity min="0" max="100"/><!--req,laser light sensitivity-->
<triggerMode opt="cameraModuleTrigger, photoresistanceTrigger"/>
<!--req laser light triggering mode-->
<limitBrightness min="0" max="100"/>
<!--req,laser light brightness limitation-->
<angle min="1" max="36"/><!--req,laser light angle-->
<enable opt="true,false" />
<!-- dep, enable manual control laser: 0- No, 1- Yes -->
<illumination min="0" max="100" /><!-- dep, laser light strength limit-->

```

```

<lightAngle min="0" max="100" /><!-- dep, light angle -->
</autoMode>

<manualMode>
<sensitivity min="0" max="100"/><!--req,laser light sensitivity-->
<triggerMode opt=" cameraModuleTrigger, photoresistanceTrigger"/>
<!--req laser light triggering mode-->
<brightness min="0" max="255"/><!--req,laser light brightness-->
<angle min="1" max="36"/><!--req,laser light angle-->
</manualMode>
</LaserConfig>

<ChromaSuppress><!-- req, color suppression -->
<!--req, special for speed dome: 0~100 -->
<Min>0</Min><!--req, minimum value -->
<Max>100</Max><!--req, maximum value -->
<Default>50</Default><!--req, default value -->
</ChromaSuppress>
<SaturationLevel><!--req, saturation, from 0 to 100 -->
<Min>0</Min><!--req, minimum value -->
<Max>100</Max><!--req, maximum value -->
<Default>50</Default><!--req, default value -->
</SaturationLevel>
<HueLevel><!--req, hue, from 0 to 100-->
<Min>0</Min><!--req, minimum value -->
<Max>100</Max><!--req, maximum value -->
<Default>50</Default><!--req, default value -->
</HueLevel>
<GammaCorrection><!--req, gamma correction-->
<GammaCorrectionEnabled><!--req, 0-disable, 1-enable-->
<Range>0,1</Range>
<Default>0</Default><!--req, default value-->
</GammaCorrectionEnabled>
<GammaCorrectionLevel><!--req, the level of Gamma correction-->
<Min>0</Min>
<Max>100</Max>
<Default>50</Default><!--req, default value -->
</GammaCorrectionLevel>
</GammaCorrection>
<WDR><!--req, wide dynamic range-->
<WDREnabled><!--req, 0-disable, 1-enable, 2-auto-->
<Range>0,1,2</Range>
<Default>0</Default><!--req, default value-->
</WDREnabled>
<isNotSupportDigitalChanCfg opt="true,false"/><!--optional, whether setting digital channels is not supported: true-yes (not supported), false-no (supported). If this field is not returned, it indicates that setting digital channels is supported-->
<WDRLevel1><!--req, level 1 of wide dynamic range, from 0 to 15-->
<Min>0</Min><!-- req, minimum value-->
<Max>15</Max><!-- req, maximum value-->
<Range>0,1,2</Range><!--req, speed dome: 0- low, 1- medium, 2- high-->
<Default>0</Default><!--req, default value-->

```

```

</WDRLevel1>
<WDRLevel2><!--req, level 2 of wide dynamic range, from 0 to 15-->
<Min>0</Min><!--req, minimum value-->
<Max>15</Max><!--req, maximum value-->
<Default>0</Default><!--req, default value-->
</WDRLevel2>
<WDRContrastLevel><!--req, contrast of wide dynamic range, from 0 to 100 -->
<Min>0</Min><!-- req, minimum value -->
<Max>100</Max><!-- req, maximum value -->
<Default>50</Default><!-- req, default value -->
</WDRContrastLevel>
</WDR>
<DayNightFilter><!--req, day and night switch -->
<DayNightFilterType><!--req, day and night switch mode -->
<!--req, 0- day, 1- night, 2- auto, 3- timing, 4- triggered by alarm input , 5- Auto mode 2(no photosensitivity)-->
<Range>0,1,2,3,4</Range>
<Default>2</Default><!--req, default value -->
</DayNightFilterType>
<SwitchSchedule>
<SwitchScheduleEnabled><!--req, reserved -->
<Range>0,1</Range><!--req, 0- disable 1- enable -->
<Default>1</Default><!--req, default value -->
</SwitchScheduleEnabled>
<DayToNightFilterLevel><!--req, sensitivity of switching day to night -->
<Range>0,1,2,3,4,5,6,7,8,9,10,11,12</Range>
<!--req, 0, 1, 2, 3, 4, 5, 6, 7, 10-low, 11-medium, 12-high -->
<!-- req, (10~12: new options for speed domes) -->
<Default>3</Default><!-- req, default value -->
</DayToNightFilterLevel>
<NightToDayFilterLevel><!-- req, sensitivity of switching night to day -->
<Range>0,1,2,3,4,5,6,7,8,9,10,11,12</Range>
<!-- req, 0, 1, 2, 3, 4, 5, 6, 7, 10-low, 11-medium, 12-high -->
<!-- req, (10~12: new options for speed domes) -->
<Default>3</Default><!-- req, default value -->
</NightToDayFilterLevel>
<DayNightFilterTime><!-- req, filtering time of switching day to night -->
<!-- req, IPC(min/max); speed dome/zoom camera(Range) -->
<Min>10</Min><!-- req, minimum value -->
<Max>120</Max><!-- req, maximum value -->
<Range>0,1,2,3,4,5,6,7</Range>
<!-- req, zoom camera/speed dome: 0-2S, 1-3S, 2-5S, 3-10S, 4-15S, 5-30S, 6-45S, 7-60S -->
<Default>55</Default><!-- req, default value -->
</DayNightFilterTime>
<NightDayFilterTime>
<!-- req, IPC(min/max); speed dome/zoom camera(Range) -->
<Min>0</Min><!-- req, minimum value -->
<Max>120</Max><!-- req, maximum value -->
<Range>0,1,2,3,4,5,6,7</Range>
<!-- req, zoom camera/speed dome: 0-2S, 1-3S, 2-5S, 3-10S, 4-15S, 5-30S, 6-45S, 7-60S -->
<Default>55</Default><!-- req, default value -->
</NightDayFilterTime>
<TimeSchedule><!--2012-08-29-->
```

```

<BeginTime>1</BeginTime><!--req, 1 means it supports the beginning time-->
<EndTime>1</EndTime><!--req, 1 means it supports the ending time-->
</TimeSchedule>
</SwitchSchedule>
<AlarmInTrigType><!--2012-08-29-->
<Range>0,1</Range><!--req, triggered status of alarm input: 0- day, 1- night-->
</AlarmInTrigType>
<DayNightFilterandGain>
    <!--opt, whether to support setting day/night auto-switch and gain simultaneously-->
    <enabled><!--req, if this function is supported, this node must exist and be set to "true"--></enabled>
</DayNightFilterandGain>
</DayNightFilter>
<Backlight><!-- req, backlight compensation -->
<BacklightMode><!-- req, option of backlight compensation -->
    <!-- req, 0-closed, 1-UP, 2-DOWN, 3-LEFT, 4-RIGHT, 5-MIDDLE, 6-customized, 10-open, 11-auto, 12- multi-zone
backlight compensation -->
    <!-- req, (10~12: new options for speed domes, when the value is 10(open), it supports to adjust the
compensation level) -->
<Range>0,1,2,3,4,5,6,7,8,9,10,11,12</Range>
<Default>0</Default><!-- req, default value -->
</BacklightMode>
<BacklightLevel><!-- req, backlight compensation level -->
    <!-- req, IPC, 0-15 -->
    <Min>0</Min><!-- req, minimum value -->
    <Max>15</Max><!-- req, maximum value -->
<Range>0,1,2</Range>
    <!-- req, speed dome/zoom camera: 0-low, 1-medium, 2-high -->
<Default>0</Default><!-- req, default value -->
</BacklightLevel>
</Backlight>
<LowLightLimit><!--req, low illumination electronic shutter-->
<LowLightLimitEnabled><!--req, enable: 0-closed, 1-open-->
<Range>0,1</Range>
<Default>0</Default><!-- req, default value -->
</LowLightLimitEnabled>
<LowLightLimitLevel><!-- req, the level of low illumination electronic shutter -->
    <!-- req, speed dome 0- slow shutter*2, 1-slow shutter*3, 2-slow shutter*4, 3-slow shutter*8, 4-slow
shutter*16, 5-slow shutter*32, 6-1, 7-2, 8-3, 9-4, 10-5, 11-6, 12-low, 13-medium, 14-high -->
<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14</Range>
<Default>0</Default><!-- req, default value -->
</LowLightLimitLevel>
</LowLightLimit>
<ImageStabilize>
    <ImageStabilizeLevel><!--req, image stabilization level-->
        <Range>0,1,2</Range><!--req, speed dome: 0-low, 1-medium, 2-high-->
        <Default>0</Default><!--req, default value -->
    </ImageStabilizeLevel>
</ImageStabilize>
<CameralRCorrection><!--req, the movement infrared correction function-->
    <Range>0,1,2</Range><!--req, speed dome: 0-auto, 1-open, 2-closed-->
    <Default>0</Default><!--req, default value-->
</CameralRCorrection>

```

```
<HighSensitivitySupport>
  <!--req, whether supports setting high sensitivity: 1- suport, no this node if not support-->
  <Range>1</Range>
</HighSensitivitySupport>
<InitializeLensSupport>
  <!--req, whether supports initializing the Lens: 1-support, no this node if not support-->
  <Range>1</Range>
</InitializeLensSupport>
<CameraResetSupport>
  <!--req, whether supports rebooting movement: 1-support, no this node if not support-->
  <Range>1</Range>
</CameraResetSupport>
<CameraRestoreSupport>
  <!--req, whether supports resuming movement to the factory settings: 1-support, no this node if not support-->
  <Range>1</Range>
</CameraRestoreSupport>

<Mirror><!--req, mirror-->
  <Range>0,1,2,3</Range><!--req, 0-off, 1-leftright, 2-updown, 3-center-->
  <Default>0</Default><!-- req, default value -->
</Mirror>

<EPTZ><!-- req, E-PTZ -->
  <!-- req, 1-support, and there is no this node if not support -->
  <Range>1</Range>
</EPTZ>

<LOCALOUTPUT><!--req, local output-->
  <!--req, 0-not support, 1-support-->
  <!--req, 365: mini-dome and cube camera don't support local output -->
  <!--req, 6467:BNC-0,1, 10-closed, 11-scaling output, 12-cropping output,-->
  <!--req, 13-cropping and scaling output (10~13: special for speed dome);-->
  <!--req, HDMI®-0:not support,20:HDMI®(720P50),21:HDMI®(720P60),22:HDMI®(1080I60)-->
  <!--req, 23 : HDMI®(1080I50), 24 : HDMI®(1080P24), 25 : HDMI®(1080P25),-->
  <!--req, 26:HDMI®(1080P30), 27 : HDMI®(1080P50), 28 : HDMI®(1080P60)-->
  <Range>0,1,10,11,12,13,20,21,22,23,24,25,26,27,28</Range>
  <Default>1</Default><!-- req, default value -->
</LOCALOUTPUT>

<DigitalNoiseReduction><!--req, noise reduction-->
  <DigitalNoiseReductionEnable>
    <!-- req, 0-closed,1-normal mode,2-expert mode,(3~9:reserved),10-open -->
    <!-- req, (10: new options for speed domes, when the value is 10(open), it supports to adjust noise reduction level (that is, DigitalNoiseReductionLevel is valid) ) -->
    <Range>0,1,2,3,4,5,6,7,8,9,10</Range>
    <Default>0</Default><!-- req, default value -->
  </DigitalNoiseReductionEnable>
  <DigitalNoiseReductionLevel>
    <!--req, digital noise reduction level in normal mode, from 0 to 100-->
    <Min>0</Min><!-- req, minimum value -->
    <Max>100</Max><!-- req, maximum value -->
```

```

<!-- req, speed dome: 0-low,1-medium,2-high,
3-1,4-2,5-3,6-4,7-5,8-6,9-7,10-8,11-9,12-10,13-11,14-12,15-13,16-14,17-15 -->
<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17</Range>
<Default>50</Default><!-- req, default value -->
</DigitalNoiseReductionLevel>
<DigitalNoiseSpectralLevel>
    <!-- req, spatial digital noise reduction level in expert mode, from 0 to 100-->
    <Min>0</Min><!-- req, minimum value -->
    <Max>100</Max><!-- req, maximum value -->
    <Default>50</Default><!-- req, default value -->
</DigitalNoiseSpectralLevel>
<DigitalNoiseTemporalLevel>
    <!-- req, temporal digital noise reduction level in expert mode, from 0 to 100-->
    <Min>0</Min><!-- req, minimum value -->
    <Max>100</Max><!-- req, maximum value -->
    <Default>50</Default><!-- req, default value -->
</DigitalNoiseTemporalLevel>
<DigitalNoiseRemove2DEnable><!-- whether to enable 2D noise reduction for captured frames: 0-disable, 1-enable-->
    <Range>0,1</Range>
    <Default>0</Default><!-- default value -->
</DigitalNoiseRemove2DEnable>
<DigitalNoiseRemove2DLevel><!-- 2D noise reduction level for captured frames, which is between 0 and 100-->
    <Min>0</Min><!-- minimum value -->
    <Max>100</Max><!-- maximum value -->
    <Default>50</Default><!-- default value -->
</DigitalNoiseRemove2DLevel>
<mutexAbility opt="1920*1080@50fps,1920*1080@60fps"/>
<!--req, if 1920*1080@50fps or 1920*1080@60fps needs to be enabled after 3D noise reduction, SMD, rotation, or WDR (wide dynamic range) is enabled, users will be prompted to disable 3D noise reduction, SMD, rotation, and WDR in the self-adaptive mode and schedule mode first. This node is mutually exclusive with
CaptureMode-->
</DigitalNoiseReduction>
<SceneMode><!--req, scene mode: 0-outdoor, 1-indoor-->
    <Range>0,1</Range><!--req, 0-outdoor, 1-indoor, 2-default, 3-low light-->
    <Default>0</Default><!--req, default value -->
</SceneMode>
<ColorRange><!--req, color scale range-->
    <Range>0,1</Range><!--req, 0:16-235, 1:0-255-->
    <Default>0</Default><!--req, default value -->
</ColorRange>
<DigitalZoom><!--req, digital zoom, special for thermal network camera-->
    <Range>0,1,2,3,4,5</Range>
        <!-- req, digital zoom: 0-closed, 1-x2, 2-x4, 3-x8, 4-x16, 5-x32 -->
    <Default>0</Default><!--req, default value -->
</DigitalZoom>
<DeadPixelDetect><!--req, dead pixel detection, 1-support, and there is no this node if not support-->
    <Range>1</Range>-->
</DeadPixelDetect>

<LINEENCODE><!--req, whether it supports line coding capacity: 1-support, and there is no this node if not
support-->
```

```

<Range>1</Range>
</LINEENCODE>
<!--req, whether it supports one-key focus or not: 1- support, and there is no this node if not support-->
<OnepushFocus>1</OnepushFocus>

<Dehaze><!--req, de-haze-->
<DehazeEnable>0,1</DehazeEnable>
<!--req, enable de-haze mode or not: 0-no, 1-adaptive mode-->
</Dehaze>

<!--req, the following from dimmer mode to auto shutter compensation are special for thermal network camera-->
<DimmerMode>
<!--req, dimmer mode: 0-semiautomatic, 1-automatic-->
<Range>0,1</Range>
<Default>0</Default><!--req, default value-->
</DimmerMode>

<PaletteMode>
<!-- req, palette: 0- white heat, 1-black heat, 2-palette2, ..., 8-palette8, 9-fusion 1, 10-rainbow, 11-fusion 2, 12-iron red 1, 13-iron red 2, 14-sepia, 15-color 1, 16-color 2, 17-ice & fire, 18-rain, 19-red hot, 20-green hot, 21-dark blue, 22-color 3-->
<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22</Range>
<Default>0</Default><!-- req, default value -->
</PaletteMode>

<EnhancedMode>
<!-- req, enhanced mode(detection object surrounding): 0- not enhanced, 1-1, 2-2, 3-3, 4-4 -->
<Range>0,1,2,3,4</Range>
<Default>0</Default><!-- req, default value -->
</EnhancedMode>

<FilterSwitch>
<!-- req, filter switch: 1-support -->
<Range>1</Range>
</FilterSwitch>

<FocusSpeed>
<!-- req, focus speed: 0~10 -->
<Min>0</Min><!-- req, minimum value -->
<Max>10</Max><!-- req, maximum value -->
<Default>5</Default><!-- req, default value -->
</FocusSpeed>

<AutoCompensationInterval><!--req, time interval of auto shutter compensation-->
<!-- req, timing auto shutter compensation: 1~120, unit: minute -->
<Min>1</Min><!-- req, minimum value -->
<Max>120</Max><!-- req, maximum value -->
<Default>60</Default><!-- req, default value -->
</AutoCompensationInterval>

<SmartIR><!--2012-08-29-->

```

```

<Range>0,1</Range><!--req,SMART IR: 0-closed, 1-open-->
<modeType opt="automatic, manual"><!--req,0-auto 1-manual--><-req valid when switch is on>
<IRDistance min="" max="" /><!--req,level 1-100(can be set in manaul mode)-->
<ShortIRDistance min="" max="" />
<!--req,the level of short light 1-100(can be set in manaul mode)-->
<LongIRDistance min="" max="" />
<!--req,the level of long light 1-100(can be set in manaul mode)-->
</SmartIR>

<Illumination><!--2012-08-29-->
<Range>0,1</Range><!--req,low light: 0-closed, 1-open-->
</Illumination>

<LightInhibit><!--2012-08-29-->
<LightInhibitEnable opt="true,false"/>
<!--req, enable high light compensation: 0-closed, 1-open-->
<isNotSupportDigitalChanCfg opt="true,false"/><!--optional, whether setting digital channels is not supported: true-yes (not supported), false-no (supported). If this field is not returned, it indicates that setting digital channels is supported-->
<level min="0" max="100"/>
<!--req,high light compensation level-->
</LightInhibit>

<GrayLevel><!--2012-08-29-->
<Range>0,1</Range><!--req,grayscale value range,0- [0,255], 1- [16,235]-->
</GrayLevel>

<AutoFocusMode><!--req, focus mode of zoom camera and speed dome-->
<FocusModeSet>
<!-- req, focus mode: 0-auto,1-manual,2-once,3-semiautomatic -->
<Range>0,1,2,3</Range>
<Default>0</Default><!-- req, default value -->
</FocusModeSet>
<AFModeChoose>
<!-- req, auto focus mode: 0-closed, 1-mode A, 2-mode B, 3-mode AB, 4-mode C -->
<Range>0,1,2,3,4</Range>
<Default>0</Default><!-- req, default value -->
</AFModeChoose>
<MinFocusDistance>
<!-- req, minimum focusing distance: 0- automatic, 0xffff- unlimited -->
<Range>0,1,2,5,10,30,50,100,150,200,300,500,600,800,1000,2000,0xffff</Range>
<Default>0</Default><!-- req, default value -->
</MinFocusDistance>
<ZoomSpeedLevel> <!-- req, zoom speed -->
<!-- req, 0-0, 1-1, 2-2, 3-3, 4-4, 10-low, 11-medium, 12-high -->
<!-- req, (0-4: special for zoom camera, 10-12: special for speed dome) -->
<Range>0,1,2,3,4,5,6,7,8,9,10,11,12</Range>
<Default>0</Default><!-- req, default value -->
</ZoomSpeedLevel>
<FocusSpeedLevel>
<!-- req, focus speed: 0-low, 1-medium, 2-high -->
<Range>0,1,2</Range>

```

```

<Default>0</Default><!-- req, default value -->
</FocusSpeedLevel>
</AutoFocusMode>
<assistFocus opt="true"/><!--req, whether to enable assist zoom: 0-No, 1-Yes -->
<focusSensitivity min="0" max="2" def="1"/>
<!--opt, focus sensitivity, ranges from 0 to 2, it is valid when the focus mode is auto or semi-auto-->
<relativeFocusPos min="0" max="4000" def="" />
<!--opt, xs:integer, relative focus sensitivity, low 16 bytes indicate focus value (ranges from 0 to 4000), and high
16 bytes indicate temperature value under current focus, it is valid when the focus mode is manual or semi-auto-->
</FocusMode>

<AutoExposureMode>
<!-- req, exposure and gain control of zoom camera and speed dome-->
<ExposureSet>
<!-- req, exposure mode: 0-manual mode, 1-auto exposure, 2-aperture priority, 3-shutter priority, 4-gain priority
-->
<Range>0,1,2,3,4</Range>
<Default>0</Default><!-- req, default value -->
</ExposureSet>
<IrisSet> <!-- req, aperture -->
<!-- req, 0-F1.2,1-F1.4,2-F1.6,3-F1.67,4-F1.8,5-F1.85,6-F1.96,7-F2.0,8-F2.11 -->
<!-- req, 9-F2.2,10-F2.4,11-F2.41,12-F2.64,13-F2.8,14-F2.86,15-F3.13,16-F3.2 -->
<!-- req, 17-F3.4,18-F3.53,19-F3.7,20-F3.95,21-F4.0,22-F4.4,23-F4.49,24-F4.8 -->
<!-- req, 25-F5.35,26-F5.6,27-F6.38,28-F6.4,29-F6.8,30-F7.90,31-F8.0,32-F8.8 -->
<!-- req, 33-F9.6,34-F11.35-F11.06,36-F12,37-F14,38-F16,39-F16.60,40-F17 -->
<!-- req, 41-F19,42-F22,43-F24,44-F33.19,45-F34 -->

<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,3
9,40,41,42,43,44,45</Range>
<Default>0</Default><!-- req, default value -->
</IrisSet>
<ShutterSet> <!-- req, shutter -->
<!-- req,0-closed, 1-auto x1,2-auto x2,3-auto x4,4-auto x8,5-auto x16,6-auto x32,-->
<!-- req,7-auto x64, 8-auto x128, 9-1/1, 10-1/2, 11-1/3, 12-1/4, 13-1/6, 14-1/8,-->
<!-- req,15-1/12, 16-1/15, 17-1/25, 18-1/30, 19-1/50, 20-1/60, 21-1/75, 22-1/90,-->
<!-- req,23-1/100, 24-1/120, 25-1/125, 26-1/150, 27-1/180, 28-1/200, 29-1/215,-->
<!-- req,30-1/250, 31-1/300, 32-1/350, 33-1/425, 34-1/500, 35-1/600, 36-1/725,-->
<!-- req,37-1/1000, 38-1/1250, 39-1500, 40-1/1750, 41-1/2000, 42-1/2500, 43-3000,-->
<!-- req,44-1/3500, 45-1/4000, 46-1/6000, 47-1/10000, 48-1/30000, 49-1/100000 -->

<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,
39,40,41,42,43,44,45,46,47,48,49</Range>
<Default>0</Default><!-- req, default value -->
</ShutterSet>
<GainSet><!-- req, gain: 0~100 -->
<Min>0</Min>
<Max>100</Max>
<!-- req, : 0-closed, 1-low, 2-medium, 3-high, 4-0, 5-1, 6-2, 7-3, 8-4, 9-5, 10-6, 11-7, 12-8, 13-9, 14-10, 15-11,
16-12, 17-13, 18-14, 19-15 -->
<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19</Range>
<!-- req, (0~19: special for speed dome) -->
<Default>50</Default><!-- req, default value -->

```

```

</GainSet>
<GainLimit><!-- req, gain limit -->
<!-- req, speed dome -->
<Min>0</Min><!-- req, minimum value -->
<Max>0x0f</Max><!-- req, maximum value -->
<Default>0</Default><!-- req, default value -->
</GainLimit>
<ExposureComp>
<!-- req, exposure compensation: 0~100 -->
<Min>0</Min>
<Max>100</Max>
<Default>50</Default><!-- req, default value -->
<!-- req, : 0-closed, 1-low, 2-medium, 3-high, 4-0, 5-1, 6-2, 7-3, 8-4, 9-5, 10-6, 11-7, 12-8, 13-9, 14-10, 15-11,
16-12, 17-13, 18-14 -->
<!-- req, (0~18: special for speed dome) -->
<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19</Range>
</ExposureComp>
</AutoExposureMode>

<ZoomPara>
<!-- req, zoom parameter of zoom camera/speed dome -->
<ZoomDisplay>
<!-- req, zoom display: 0-closed, 1-open -->
<Range>0,1</Range>
<Default>0</Default><!-- req, default value -->
</ZoomDisplay>
<ZoomLimit>
<!-- req, zoom camera(Range); speed dome(min,max)-->
<!-- req, zoom limit of zoom camera: 0-10,1-18,2-20,3-22,4-23,5-30,6-36,7-37,8-38,9-39,10-40,11-unlimited -->
<Range>0,1,2,3,4,5,6,7,8,9,10,11</Range>
<!-- req, optical zoom of speed dome: 1~50 -->
<Min>1</Min>
<Max>50</Max>
<Default>1</Default><!-- req, default value -->
</ZoomLimit>
<DigitalZoom>
<!-- req, zoom camera(Range);speed dome(min,max) -->
<!-- req, digital zoom of zoom camera: 0-closed, 1- *2, 2-*4, 3-*8, 4-*10, 5-*12 -->
<Range>0,1,2,3,4,5</Range>
<!-- req, digital zoom of speed dome: 0~30 -->
<Min>1</Min>
<Max>30</Max>
<Default>1</Default><!-- req, default value -->
</DigitalZoom>
</ZoomPara>

<SnapExposure>
<!-- req, exposure control general triggered snapshot -->
<SnapMode>
<!-- req, snapshot mode: 0- snapshot mode 1, 1- snapshot mode 2, 2- snapshot mode 3 -->
<Range>0,1,2</Range>
<Default>0</Default><!-- req, default value -->

```

```

</SnapMode>
<SnapGain1>
    <!-- req, snapshot gain 1: 0-100 -->
    <Min>0</Min>
    <Max>100</Max>
    <Default>50</Default><!-- req, default value -->
</SnapGain1>
<SnapGain2>
    <!-- req, snapshot gain 2: 0-100 -->
    <Min>0</Min>
    <Max>100</Max>
    <Default>50</Default><!-- req, default value -->
</SnapGain2>
</SnapExposure>

<!--req, dynamic contrast ratio level of intelligent traffic camera-->
<DynamicContrast>
    <DynamicContrastLevel>
        <Min>0</Min>
        <Max>100</Max>
        <Default>50</Default><!--req, default value-->
    </DynamicContrastLevel>
</DynamicContrast>

<!--req, Rotation Mode-->
<CorridorMode>
    <corridorModeFunEnable opt="true,false"/>
    <!--req,Enable or not, true-Enable, false-Disable-->
    <!--req,If enable 1080p50/1080p60 after the 3D DNR, SMD, rotation or WDR is enabled, the prompt will pop up
    "Close the 3D DNR, SMD, rotation and WDR under the self-adaptive and continuous mode first."-->
    <!--req,If enable 720p50/720p60 after the rotation or WDR is enabled, the prompt will pop up "Close rotation and
    WDR under the self-adaptive and continuous mode first."-->
    <mutexAbility opt="19-1920*1080@50fps,20-1920*1080@60fps,8-1280*720@60fps,7-1280*720@50fps"/>
    <!--req Mutex among the CaptureMode 1920*1080@50fps, 1920*1080@60fps, 1280*720@60fps and
    1280*720@50fps, prompt will pop up when enabled any of the modes.-->
</CorridorMode>

<ISPAdvanceCfg><!--req,ISP supports return or not.-->
<ISPSupportMode opt="dayMode,nightMode"/>
<!--req Mode supported by ISP-->
<workMode opt="auto,schedule"/>
<!--req,0-Auto,1-Scheduled switch-->
<TimeSchedule>
    <beginTime opt="hour,min,sec,millisecond"/>
    <!--req Type of start time period-->
    <endTime opt="hour,min,sec,millisecond"/>
    <!--req Type of end time period-->
</TimeSchedule>
<ISPCfgSupport opt="whiteBalanceMode,whiteBalanceModeRGain,whiteBalanceModeBGain,exposureSet,
    exposureUserSet,gainLevel,brightnessLevel,contrastLevel,sharpnessLevel,WDREnabled,
    WDRLevel1,WDRLevel2,WDRContrastLevel,backlightMode,backlightLevel,position1,
    position2,imageStabilizeEnable,imageStabilizeLevel,digitalNoiseReductionEnable,>

```

```
    digitalNoiseReductionLevel,digitalNoiseSpectralLevel,digitalNoiseTemporalLevel,  
    dehazeEnable,dehazeLevel,lightInhibitEnable,lightInhibitLevel,grayLevel"/>  
</ISPAdvanceCfg>  
  
<supportCCDFunc opt="whiteBalance,exposure,WDR,dayNightFilter,gammaCorrection,digitalNoiseReduction,  
    backLight,lowLight,focus,infrared,domeAemode,dehaze,parkAction,elecStab,other,ISP,laser"/>  
<!--req supported front-end parameter capabilities.-->  
  
<!--req, Illumination Enhancement Capture Camera v3.5-->  
<BrightCompensate>  
    <brightCompensate min="0" max="100"/>  
</BrightCompensate>  
  
<!--req, Exposure Control Capture Camera v3.5-->  
<ExposureSegment>  
    <exposureSegmentEnable opt="true,false"/>  
    <!--req,Enable or not, true-Enable, false-Disable-->  
</ExposureSegment>  
  
<LensDistortionCorrection>  
    <enable opt="false,true" default="false"/>  
    <!--req,Lens Distortion Correction (0-Disable/1-Enable)-->  
    <mutexAbility opt="WDR"/>  
    <!--req,mutex in WDR -->  
    <level min="1" max="3">  
        <!--opt, xs:integer, Distortion Correction Level, 1-3-->  
    </level>  
    <zoomedInDistantView>  
        <!--dep, xs:integer, Remote zoom, takes effect when Distortion Correction is enabled.-->  
        <enabled>  
            <!--req, xs:boolean, "true,false"-->  
        </enabled>  
        <level min="1" max="3">  
            <!--opt, xs:integer, Correction level of remote zoom, 1-3-->  
        </level>  
    </zoomedInDistantView>  
    <horizontalFOV min="0" max="100">  
        <!--opt, xs:integer, Horizontal FOV[0-100]-->  
    </horizontalFOV>  
    <verticalFOV min="0" max="100">  
        <!--opt, xs:integer, Vertical FOV[0,100]-->  
    </verticalFOV>  
</LensDistortionCorrection>  
<BrightnessSuddenChangeSuppressionCap>  
    <enabled opt="true,false">  
        <!--req, xs:boolean, brightness sudden change suppression-->  
    </enabled>  
</BrightnessSuddenChangeSuppressionCap>  
  
<DPCParam>  
    <!--req,DPC-->  
    <ctrltype
```

```

opt="correct,cancelCorrect,crossDisplayOpen,crossDisplayClose,point,up,down,right,left,allCorrect,save"/>
    <!--req,correction, cancel correction, enable/disable DPC cross display, DPC coordinate, up-forward offset of DPC
coordinate,
    down-forward offset of DPC coordinate, right-forward offset of DPC coordinate, left-forward offset of DPC
coordinate, DPC all, save defective Pixel-->
<dpcMode opt="manual,auto" def="auto"/>
    <!--req,xs:string,"manual-Manual Correction, auto-Auto Correction, if device does not support this node, all will
be handled manually"-->
</DPCParam>

<FFCParam>
<mode opt="schedule,temperature "/>
    <!--req,1-Continuous mode, 2-Temperature difference mode, 3-Close-->
<ScheduleMode>
    <compensateTime opt="10,20,30,40,50,60,120,180,240"/>
    <compensateTimeUnit opt="min"/>
    <!--req,min-->
</ScheduleMode>
<FFCManualCtrl opt="true"/>
    <!--req,FFC Manual Control-->
<FFCBackCompCtrl opt="true"/>
    <!--req,FFC Background Compensation control-->
</FFCParam>

<DDEParam>
<mode opt="off,normal,expert"/>
    <!--req,1-Close, 2-Normal Mode, 3-Expert Mode-->
<normalLevel min="1" max="100"/>
    <!--req,Level settings under normal mode-->
<expertLevel min="1" max="100"/>
    <!--req,Level settings under expert mode-->
</DDEParam>

<AGCParam>
<scene opt="normal,highlight,manual"/>
    <!--req,1-Normal scene, 2-Highlight scene, 3- Manual scene-->
<ManualMode>
    <lightLevel min="1" max="100"/>
    <!--req,Brightness level-->
    <gainLevel min="1" max="100"/>
    <!--req,Gain level-->
</ManualMode>
</AGCParam>
<fusionMode/><!--opt, xs:string, visual and thermal image fusion mode: "thermal"-thermal mode, "fusion"-fusion
mode, "PIP"-picture in picture mode, "Visible"-visible mode, "fusionB/W"-black and white fusion mode, "city",
"jungle", "deset", "sea", "snow"-->
<ThermometryAGC>
<mode opt = "close,auto,manual">
    <!--opt, xs:string-->
</mode>
<highTemperature min="-273" max="10000">
    <!--dep, xs:integer-->

```

```

</highTemperature>
<lowTemperature min="-273" max="10000">
    <!--dep, xs:integer-->
</lowTemperature>
</ThermometryAGC>

<isSupportGPSControl>
    <!--optional, boolean, whether the device supports GPS control capability-->
</isSupportGPSControl >

<gearRange>
    <!--optional, xs:integer, the number of ranges supported by the device, e.g., when the value is 3,it indicates
supported three ranges-->
    <gearRange>

    </ChannelEntry>
</ChannelList>
</CAMERAPARA>

```

F.165 XML_Cap_AdminAccessProtocolList

AdminAccessProtocol capability message in XML format

```

<AdminAccessProtocolList version="2.0" xmlns="http://www.isapi.com/ver20/XMLSchema">
    <AdminAccessProtocol><!--multiple <AdminAccessProtocol> nodes are allowed-->
        <id><!--req, xs: string, ID--></id>
        <enabled opt="true,false"><!--opt, xs: boolean--></enabled>
        <protocol
            opt="HTTP,HTTPS,SDK,RTSP,DEV_MANAGE,WebSocket,WebSocketS,SDK_OVER_TLS,SRTP,MATRIX_GATEWAY,IOT,Bonjour,104Proto,SRTP">
            <!--req, xs: string-->
            </protocol>
        <portNo min="2000" max="65535" default="8443"><!--req, xs:integer --></portNo>
        <redirectToHttps opt="true,false">
            <!--opt, xs: boolean, whether to automatically go to HTTPS when connecting to HTTP port and HTTPS is enabled.
This node is valid when protocol is "HTTPS"-->
            </redirectToHttps>
            <streamOverTls opt="true,false">
                <!--opt, xs: boolean, whether to enable TLS link encryption when the streaming mode is SDK_OVER_TLS, true-yes,
false-no-->
            </streamOverTls>
            <TLS1_1Enable opt="true,false"><!--dep, xs: boolean, whether to enable TLS version 1.1, it is valid when protocol is
"HTTPS", "true"-yes, "false"-no--></TLS1_1Enable>
            <TLS1_2Enable opt="true,false"><!--dep, xs: boolean, whether to enable TLS version 1.2, it is valid when protocol is
"HTTPS", "true"-yes, "false"-no--></TLS1_2Enable>
        </AdminAccessProtocol>
        <TLS1_0Enable opt="true,false">
            <!--dep, xs: boolean, whether to enable TLS version 1.0, it is valid when protocol is "HTTPS", "true"-yes, "false"-no-->
        </TLS1_0Enable>
        <TLS1_1Enable opt="true,false">

```

```
<!--dep, xs: boolean, whether to enable TLS version 1.1, it is valid when protocol is "HTTPS", "true"-yes, "false"-no-->
</TLS1_1Enable>
<TLS1_2Enable opt="true,false">
  <!--dep, xs: boolean, whether to enable TLS version 1.2, it is valid when protocol is "HTTPS", "true"-yes, "false"-no-->
</TLS1_2Enable>
</AdminAccessProtocolList>
```

F.166 XML_Cap_AudioVideoCompressInfo

AudioVideoCompressInfo capability message in XML format

```
<AudioVideoCompressInfo version="2.0">
  <AudioCompressInfo>
    <Audio><!--req, audio-->
      <ChannelList>
        <ChannelEntry>
          <ChannelNumber><!--req channel No.1--></ChannelNumber>
          <MainAudioEncodeType><!--req, encoding type of main audio stream: 0-G722_1, 1-G711_MU, 2-G711_A, 5-MP2L2, 6-726, 7-AAC, 8-PCM, 12-AAC_LC, 13-AAC_LD, 14-Opus, 15-MP3-->
          <Range>0</Range>
          <OggVorbisAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
          <!--req, G722 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps -->
          <G711UAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
          <!--req, G711-U audio bit rate 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
          <G711AAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
          <!--req, G711-A audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
          <MP2L2AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps,40kbps,48kbps,56kbps,
80kbps,96kbps,112kbps,144kbps,160kbps" default="32kbps"/>
          <!--req, MP2L2 audio bit rate 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps,7-40Kbps,
8-48Kbps,9-56Kbps, 10-80Kbps,11-96Kbps,12-112Kbps,13-144Kbps,14-160Kbps-->
          <MP2L2SamplingRate opt="default,16kHz,32kHz,48kHz,44.1kHz" default="16kHz"/>
          <!--req audio sampling rate: 0-default, 1-16kHz, 2-32kHz, 3-48kHz, 4-44.1kHz-->
          <MP2L2SamplingRate16kHz>
            <audioBitRate opt="32kbps, 40kbps, 48kbps, 56kbps, 64kbps, 80kbps, 96kbps, 112kbps, 128kbps, 144kbps,
160kbps"/>
          </MP2L2SamplingRate16kHz>
          <MP2L2SamplingRate32kHz>
            <audioBitRate opt="32kbps, 48kbps, 56kbps, 64kbps, 80kbps, 96kbps, 112kbps, 128kbps, 160kbps,
192kbps"/>
          </MP2L2SamplingRate32kHz>
          <MP2L2SamplingRate44.1kHz>
            <audioBitRate opt="32kbps, 48kbps, 56kbps, 64kbps, 80kbps, 96kbps, 112kbps, 128kbps, 160kbps,
192kbps"/>
          </MP2L2SamplingRate44.1kHz>
          <MP2L2SamplingRate48kHz>
            <audioBitRate opt="32kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,160kbps,192kbps"/>
          </MP2L2SamplingRate48kHz>
```

```

<G726AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G726 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps;ipc5.1.0
Default: 0-64Kbps-->
<AACAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps" default="32kbps"/>
<!--req, AAC audio bit rate: 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps -->
<AACSamplingRate opt="default,16kHz,32kHz,48kHz,44.1kHz" default="16kHz"/>
<!--req, AAC audio sampling rate: 0-default,1-16kHz,2-32kHz,3-48kHz,4-44.1kHz -->
<PCMSamplingRate opt="default,8kHz,16kHz,32kHz,48kHz,44.1kHz" default="16kHz"/>
<!--req audio sampling rate: 0-default, 1-16kHz,2-32kHz,3-48kHz,4-44.1kHz,5-8kHz-->
<MP3AudioBitRate opt="8kbps,16kbps,32kbps,40kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,
160kbps,192kbps,224kbps,256kbps,320kbps" default="64kbps"/>
<!--req, audio bit rate->
<MP3SamplingRate opt="8kHz,16kHz,32kHz,44.1kHz,48kHz" default="16kHz"/><!--req, audio sampling rate-->
</MainAudioEncodeType>
<!--req, encoding type of sub stream: 0- G722_1, 1- G711_MU, 2- G711_A, 5- MP2L2, 6- G726, 7- AAC, 8- PCM,
12-AAC_LC, 13-AAC_LD, 14-Opus, 15-MP3-->
<SubAudioEncodeType>
<Range>0</Range>
<OggVorbisAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G722 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps -->
<G711UAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G711-U audio bit rate 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps; ipc5.1.0
Default: 0-64Kbps-->
<G711AAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G711-A audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps; ipc5.1.0
Default: 0-64Kbps-->
<MP2L2AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps,40kbps,48kbps,56kbps,
80kbps,96kbps,112kbps,144kbps,160kbps" default="32kbps"/>
<!--req, MP2L2 audio bit rate: 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps,7-40Kbps,
8-48Kbps,9-56Kbps,
10-80Kbps,11-96Kbps,12-112Kbps,13-144Kbps,14-160Kbps-->
<MP2L2SamplingRate opt="default,16kHz,32kHz,48kHz,44.1kHz" default="16kHz"/>
<!--req audio samping rate: 0-default, 1-16kHz,2-32kHz,3-48kHz,4-44.1kHz-->
<MP2L2SamplingRate16kHz>
<audioBitRate opt="32kbps,40kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,144kbps,
160kbps"/>
</MP2L2SamplingRate16kHz>
<MP2L2SamplingRate32kHz>
<audioBitRate opt="32kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,160kbps,192kbps"/>
</MP2L2SamplingRate32kHz>
<MP2L2SamplingRate44.1kHz>
<audioBitRate opt="32kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,160kbps,192kbps"/>
</MP2L2SamplingRate44.1kHz>
<MP2L2SamplingRate48kHz>
<audioBitRate opt="32kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,160kbps,192kbps"/>
</MP2L2SamplingRate48kHz>
<G726AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G726 audio bit rate: 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps;ipc5.1.0
Default: 0-64Kbps-->
<AACAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps" default="32kbps"/>
<!--req, AAC audio bit rate: 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps -->

```

```

<AACSamplingRate opt="default,16kHz,32kHz,48kHz"/>
<!--audio sampling rate: 0-default, 1-16kHz,2-32kHz,3-48kHz/ -->
<!--req, AAC audio sampling rate: 0-default,1-16kHz,2-32kHz,3-48kHz,4-44.1kHz -->
<PCMSamplingRate opt="default,8kHz,16kHz,32kHz,48kHz,44.1kHz" default="16kHz"/>
<!--req audio sampling rate: 0-default, 1-16kHz,2-32kHz,3-48kHz,4-44.1kHz,5-8kHz-->
<MP3AudioBitRate opt="8kbps,16kbps,32kbps,40kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,
160kbps,192kbps,224kbps,256kbps,320kbps" default="64kbps"/>
    <!--req, audio bit rate-->
    <MP3SamplingRate opt="8kHz,16kHz,32kHz,44.1kHz,48kHz" default="16kHz"/><!--req, audio sampling rate-->
>
</SubAudioEncodeType>

<EventAudioEncodeType><!--req, event stream audio encoding type: 0- G722_1, 1- G711_MU, 2- G711_A -->
<Range>0</Range>
<OggVorbisAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
    <!--req, G.722 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps -->
    <G711UAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
        <!--req, G711-U audio bit rate 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
    <G711AAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
        <!--req, G711-A audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
    <MP2L2AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
        <!--req, MP2L2 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
    <G726AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
        <!--req, G726 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
    <MP2L2SamplingRate opt="default,16kHz,32kHz,48kHz,44.1kHz" default="16kHz"/>
    <!--req, audio sampling rate: 0-default, 1-16kHz, 2-32kHz, 3-48kHz, 4-44.1kHz-->
    <MP2L2SamplingRate16kHz>
        <audioBitRate opt="32kbps,40kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,144kbps,
160kbps"/>
    </MP2L2SamplingRate16kHz>
    <MP2L2SamplingRate32kHz>
        <audioBitRate opt="32kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,160kbps,192kbps"/>
    </MP2L2SamplingRate32kHz>
    <MP2L2SamplingRate44.1kHz>
        <audioBitRate opt="32kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,160kbps,192kbps"/>
    </MP2L2SamplingRate44.1kHz>
    <MP2L2SamplingRate48kHz>
        <audioBitRate opt="32kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,160kbps,192kbps"/>
    </MP2L2SamplingRate48kHz>
    <G726AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
        <!--req, audio bit rate: 0-default, 1-8Kbps, 2-16Kbps, 3-32Kbps, 4-64Kbps, 5-128Kbps, 6-192Kbps, for network
camera with version 5.1.0, the default value is "0"-->
    <AACAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps" default="32kbps"/>
        <!--req, audio bit rate: 0-default, 1-8Kbps, 2-16Kbps, 3-32Kbps, 4-64Kbps, 5-128Kbps, 6-192Kbps-->
    <ACCSamplingRate opt="default,16kHz,32kHz,48kHz,44.1kHz" default="16kHz"/>
        <!--req, audio sampling rate: 0-default, 1-16kHz, 2-32kHz, 3-48kHz, 4-44.1kHz-->
    <PCMSamplingRate opt="default,8kHz,16kHz,32kHz,48kHz,44.1kHz" default="16kHz"/>

```

```

<!--req, audio sampling rate: 0-default, 1-16kHz, 2-32kHz, 3-48kHz, 4-44.1kHz, 5-8kHz-->
<MP3AudioBitRate opt="8kbps,16kbps,32kbps,40kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,
160kbps,192kbps,224kbps,256kbps,320kbps" default="64kbps"/>
<!--req, audio bit rate-->
<MP3SamplingRate opt="8kHz,16kHz,32kHz,44.1kHz,48kHz" default="16kHz"/><!--req, audio sampling rate-->
>
</EventAudioEncodeType>
<Stream3AudioEncodeType><!--req, audio compression type of the third stream: 0-G722_1, 1-G711_MU, 2-
G711_A-->
<Range>0</Range>
<OggVorbisAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G.722 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps -->
<G711UAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G711-U audio bit rate 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<G711AAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G711-A audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<MP2L2AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, MP2L2 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<G726AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G726 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64Kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<MP2L2SamplingRate opt="default,16kHz,32kHz,48kHz,44.1kHz" default="16kHz"/>
<!--req, audio sampling rate: 0-default, 1-16kHz, 2-32kHz, 3-48kHz, 4-44.1kHz-->
<MP2L2SamplingRate16kHz>
<audioBitRate opt="32kbps,40kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,144kbps,
160kbps"/>
</MP2L2SamplingRate16kHz>
<MP2L2SamplingRate32kHz>
<audioBitRate opt="32kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,160kbps,192kbps"/>
</MP2L2SamplingRate32kHz>
<MP2L2SamplingRate44.1kHz>
<audioBitRate opt="32kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,160kbps,192kbps"/>
</MP2L2SamplingRate44.1kHz>
<MP2L2SamplingRate48kHz>
<audioBitRate opt="32kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,160kbps,192kbps"/>
</MP2L2SamplingRate48kHz>
<G726AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, audio bit rate: 0-default, 1-8Kbps, 2-16Kbps, 3-32Kbps, 4-64Kbps, 5-128Kbps, 6-192Kbps, for network
camera with version 5.1.0, the default value is "0"-->
<AACAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps" default="32kbps"/>
<!--req, audio bit rate: 0-default, 1-8Kbps, 2-16Kbps, 3-32Kbps, 4-64Kbps, 5-128Kbps, 6-192Kbps-->
<ACASamplingRate opt="default,16kHz,32kHz,48kHz,44.1kHz" default="16kHz"/>
<!--req, audio sampling rate: 0-default, 1-16kHz, 2-32kHz, 3-48kHz, 4-44.1kHz-->
<PCMSamplingRate opt="default,8kHz,16kHz,32kHz,48kHz,44.1kHz" default="16kHz"/>
<!--req, audio sampling rate: 0-default, 1-16kHz, 2-32kHz, 3-48kHz, 4-44.1kHz, 5-8kHz-->
<MP3AudioBitRate opt="8kbps,16kbps,32kbps,40kbps,48kbps,56kbps,64kbps,80kbps,96kbps,112kbps,128kbps,
160kbps,192kbps,224kbps,256kbps,320kbps" default="64kbps"/>
<!--req, audio bit rate-->
```

```

<MP3SamplingRate opt="8kHz,16kHz,32kHz,44.1kHz,48kHz" default="16kHz"/><!--req, audio sampling rate-->
</Stream3AudioEncodeType>
<AudioInType><!--req, audio input type: 0-mic in,1-line in -->
<Range>0</Range>
</AudioInType>
<AudioInVolume><!--opt,volume of audio input--><!--req, 2012-08-29-->
<Min>0</Min>
<Max>100</Max>
</AudioInVolume>
</ChannelEntry>
<ChannelEntry>
<ChannelNumber>2</ChannelNumber><!--req, channel 2 -->
<!--req, audio compression type of main stream: 0- G722_1, 1- G711_MU, 2- G711_A-->
<MainAudioEncodeType>
<Range>0</Range>
<OggVorbisAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G.722 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps -->
<G711UAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G711-U audio bit rate 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<G711AAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G711-A audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<G726AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G726 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<MainAudioEncodeType>
<!--req, audio compression type of sub stream: 0- G722_1, 1- G711_MU, 2- G711_A -->
<SubAudioEncodeType>
<Range>0</Range>
<OggVorbisAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G.722 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps -->
<G711UAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G711-U audio bit rate 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<G711AAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G711-A audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<MP2L2AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, MP2L2 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<G726AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G726 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<SubAudioEncodeType>
<!--req, audio compression type of event stream: 0- G722_1, 1- G711_MU, 2- G711_A -->

```

```

<EventAudioEncodeType>
<Range>0</Range>
<OggVorbisAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G.722 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps -->
<G711UAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G711-U audio bit rate 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<G711AAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G711-A audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<MP2L2AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, MP2L2 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<G726AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G726 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
</EventAudioEncodeType>
<!--req, audio compression type of the third stream: 0- G722_1, 1- G711_MU, 2- G711_A -->
<Stream3AudioEncodeType>
<Range>0</Range>
<OggVorbisAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G.722 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps -->
<G711UAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G711-U audio bit rate 0-default,1-8Kbps,2-16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<G711AAudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G711-A audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<MP2L2AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, MP2L2 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
<G726AudioBitRate opt="default,8kbps,16kbps,32kbps,64kbps,128kbps,192kbps"/>
<!--req, G726 audio bit rate 0-default,1-8Kbps,2- 16Kbps,3-32Kbps,4-64kbps,5-128Kbps,6-192Kbps??ipc5.1.0
Default: 0-64Kbps-->
</Stream3AudioEncodeType>
<!--req, audio input type: 0-mic in,1-line in -->
<AudioInType>
<Range>0</Range>
</AudioInType>
<AudioInVolume><!--opt,volume of audio input--><!--req, 2012-08-29-->
<Min>0</Min>
<Max>100</Max>
</AudioInVolume>
</ChannelEntry>
</ChannelList>
</Audio>
<VoiceTalk><!--req, two-way aduio-->
<ChannelList>
<ChannelEntry>
<ChannelNumber>1</ChannelNumber><!--req, channel 1 -->

```

```

<!--req, compression type of voice talk: 0- G722_1, 1- G711_MU, 2- G711_A -->
<VoiceTalkEncodeType><!--two-way audio encoding type: 0-G.722, 1-G711_U, 2-G711_A, 5- MP2L2, 6-G726,
12-AAC_LC, 13-AAC_LD, 14-Opus, 15-MP3-->
<Range>0</Range>
</VoiceTalkEncodeType>
<!--req, input type of voice talk: 0-mic in, 1-line in -->
<VoiceTalkInType>
<Range>0</Range>
</VoiceTalkInType>
</ChannelEntry>
</ChannelList>
</VoiceTalk>
</AudioCompressInfo>

<VideoCompressInfo><!--req, video encoding type-->
<ChannelList>
<ChannelEntry>
<ChannelNumber>1</ChannelNumber><!--req, channel number -->
<MainChannel>
<VideoEncodeType>
<!--req, 0-private 264,1-standard h264,2-standard mpeg4,7-M-JPEG,8-MPEG2 -->
<Range>0,1,2,7,8</Range>
</VideoEncodeType>
<VideoEncodeEfficiency>
<!--req, the complexity of video compression: 0-low,1-medium,2-high -->
<Range>0</Range>
</VideoEncodeEfficiency>
<!--req, whether it supports CABAC or not -->
<VideoCabac>1</VideoCabac>
<VideoResolutionList> <!--req, resolution list -->
<VideoResolutionEntry>
<!--req, DCIF -->
<Index>0</Index>
<Name>DCIF</Name><!--req, for reference only, the device may return different value -->
<Resolution>528*384</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame

```

rate supported by P standard and N standard is same-->

<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->

```

<VideoBitrate>
  <!--req,video bit rate-->
  <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
  <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
  <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K, -->
  <!--req, 10-224K,11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K, -->
  <!--req, 19-1024K,20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K-->
  <Range>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</Range>
</VideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
  <!--req, CIF -->
  <Index>1</Index>
  <Name>CIF</Name>
  <Resolution>352*288</Resolution>
  <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
  12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
  27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

```

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRate>

<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->

<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRateN>

<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->

<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->

```

<VideoBitrate>
  <!--req,video bit rate-->
  <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
  <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
  <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K, -->
  <!--req, 10-224K,11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
  <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
  <Range>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</Range>
</VideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
  <!--req, QCIF -->
  <Index>2</Index>
  <Name>QCIF</Name>

```

```

<Resolution>176*144</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
    rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
    rate supported by P standard and N standard is same-->
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
    minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
        <VideoBitrate>
            <!--req,video bit rate-->
            <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
            <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
            <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K, -->
            <!--req, 10-224K,11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
            <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
            <Range>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</Range>
        </VideoBitrate>
    </VideoResolutionEntry>
    <VideoResolutionEntry>
        <!--req, 4CIF -->
        <Index>3</Index>
        <Name>4CIF</Name>
        <Resolution>704*576</Resolution>
        <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
    rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>

```

```

<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
<VideoBitrate>
<!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
<!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
<!--req, 2CIF -->
<Index>4</Index>
<Name>2CIF</Name>
<Resolution>704*288</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
<VideoBitrate>
<!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,20-1280K, -->
<!--req, 21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
<!--req, VGA -->
<Index>16</Index>

```

```

<Name>VGA</Name>
<Resolution>640*480</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
        <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
        12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
        26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
        <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
            <VideoBitrate>
                <!--req,video bit rate-->
                <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
                <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
                <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
                <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,20-1280K, -->
                <!--req, 21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
                <Range>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</Range>
            </VideoBitrate>
        </VideoResolutionEntry>
        <VideoResolutionEntry>
            <!--req, UXGA -->
            <Index>17</Index>
            <Name>UXGA</Name>
            <Resolution>1600*1200</Resolution>
            <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
            12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
            27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
        <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
        12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
        26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</

```

VideoFrameRateN

```

<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
<VideoBitrate>
<!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,20-1280K, -->
<!--req, 21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
<!--req, SVGA -->
<Index>18</Index>
<Name>SVGA</Name>
<Resolution>800*600</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
<VideoBitrate>
<!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
<!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
<!--req, 720P -->

```

```
<Index>19</Index>
<Name>720P</Name>
<Resolution>1280*720</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
<VideoBitrate>
<!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,20-1280K, -->
<!--req, 21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
<!--req, XVGA -->
<Index>20</Index>
<Name>XVGA</Name>
<Resolution>1280*960</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->
```

```
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
```

<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->

<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->

```
<VideoBitrate>
```

<!--req,video bit rate-->

```
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
```

```
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
```

```
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
```

```
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
```

```
<!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
```

```
<Range>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</Range>
```

```
</VideoBitrate>
```

```
</VideoResolutionEntry>
```

```
<VideoResolutionEntry>
```

<!--req, HD900 -->

```
<Index>21</Index>
```

```
<Name>HD900</Name>
```

```
<Resolution>1600*912</Resolution>
```

<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;

11-12;

12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

```
<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
```

<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->

<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;

11-12;

12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;

26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

```
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
```

<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->

<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->

```
<VideoBitrate>
```

<!--req,video bit rate-->

```
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
```

```
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
```

```
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
```

```
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
```

```
<!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
```

```
<Range>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</Range>
```

```
</VideoBitrate>
```

```
</VideoResolutionEntry>
```

```
<VideoResolutionEntry>
```

```

<!--req, SXGA -->
<Index>22</Index>
<Name>SXGA</Name>
<Resolution>1280*1024</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
    rate supported by P standard and N standard is same-->
        <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
        12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
        26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
    rate supported by P standard and N standard is same-->
        <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
        minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
            <VideoBitrate>
                <!--req,video bit rate-->
                <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
                <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
                <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
                <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
                <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
                <Range>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</Range>
            </VideoBitrate>
        </VideoResolutionEntry>
        <VideoResolutionEntry>
            <!--req, 1080P -->
            <Index>27</Index>
            <Name>HD1080P</Name>
            <Resolution>1920*1080</Resolution>
            <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
            12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
            27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
    rate supported by P standard and N standard is same-->
        <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
        12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
        26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

```

```
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
```

<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->

<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->

```
<VideoBitrate>
<!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
<!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
</VideoResolutionEntry>
</VideoResolutionEntry>
<!--req, 2560*1920 -->
<Index>28</Index>
<Name>2560*1920</Name>
<Resolution>2560*1920</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->
```

```
<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
```

<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->

<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

```
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
```

<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->

<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->

```
<VideoBitrate>
<!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
<!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
</VideoResolutionEntry>
```

```

<VideoResolutionEntry>
<!--req, 1600*304 -->
<Index>29</Index>
<Name>1600*304</Name>
<Resolution>1600*304</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
        <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
        12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
        26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
        <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
            <VideoBitrate>
                <!--req,video bit rate-->
                <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
                <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
                <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
                <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
                <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
                <Range>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</Range>
            </VideoBitrate>
        </VideoResolutionEntry>
        <VideoResolutionEntry>
            <!--req, 2048*1536 -->
            <Index>30</Index>
            <Name>2048*1536</Name>
            <Resolution>2048*1536</Resolution>
            <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
            12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
            27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
        <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
        12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;

```

26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

```

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
    rate supported by P standard and N standard is same-->
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
    minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
    <VideoBitrate>
        <!--req,video bit rate-->
        <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
        <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
        <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
        <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
        <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
        <Range>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</Range>
    </VideoBitrate>
    <AverageVideoBitrate>
        <!--req, average video bit rate, it is valid when SmartCodec is enabled-->
        <Min>32</Min>
        <!--req, min. bit rate, unit: kb-->
        <Max>16384</Max>
        <!--req, max. bit rate, unit: kb-->
        <!--req, 0-OK, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
    >
        <!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K,
23-2048K-->
        <!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
        <Range>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</Range>
    <Default>
        <!--req, default value for different products are different-->
    </Default>
    </AverageVideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
    <!--req, 1920*1536 or 2048*1536 -->
    <Index>133</Index>
    <Name>3MP</Name>
    <Resolution>1920*1536</Resolution>
    <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
    rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

```

```

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
    rate supported by P standard and N standard is same-->
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
    minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
        <VideoBitrate>
            <!--req,video bit rate-->
            <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
            <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
            <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
            <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
            <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
            <Range>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</Range>
        </VideoBitrate>
        <AverageVideoBitrate>
            <!--req, average video bit rate, it is valid when SmartCodec is enabled-->
            <Min>32</Min>
            <!--req, min. bit rate, unit: kb-->
            <Max>16384</Max>
            <!--req, max. bit rate, unit: kb-->
            <!--req, 0-0K, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
        >
            <!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K,
23-2048K-->
            <!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
            <Range>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</Range>
        <Default>
            <!--req, default value for different products are different-->
        </Default>
        </AverageVideoBitrate>
    </VideoResolutionEntry>
    <VideoResolutionEntry>
        <!--req, 2560*1944 -->
        <Index>134</Index>
        <Name>5MP</Name>
        <Resolution>2560*1944</Resolution>
        <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->
    </VideoResolutionEntry>
<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
    rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
        12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

```

```

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
    rate supported by P standard and N standard is same-->
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
    minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
    <VideoBitrate>
        <!--req,video bit rate-->
        <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
        <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
        <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
        <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
        <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
        <Range>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</Range>
    </VideoBitrate>
    <AverageVideoBitrate>
        <!--req, average video bit rate, it is valid when SmartCodec is enabled-->
        <Min>32</Min>
        <!--req, min. bit rate, unit: kb-->
        <Max>16384</Max>
        <!--req, max. bit rate, unit: kb-->
        <!--req, 0-0K, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
    >
        <!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K,
23-2048K-->
        <!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
        <Range>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</Range>
    <Default>
        <!--req, default value for different products are different-->
    </Default>
    </AverageVideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
    <!--req, 2720*1920 -->
    <Index>137</Index>
    <Name>2720*1920</Name>
    <Resolution>2720*1920</Resolution>
    <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->
<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
    rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</

```

VideoFrameRateN

```
<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame  
rate supported by P standard and N standard is same-->  
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and  
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->  
<VideoBitrate>  
<!--req,video bit rate-->  
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->  
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->  
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->  
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->  
<!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->  
<Range>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</Range>  
</VideoBitrate>  
<AverageVideoBitrate>  
<!--req, average video bit rate, it is valid when SmartCodec is enabled-->  
<Min>32</Min>  
<!--req, min. bit rate, unit: kb-->  
<Max>16384</Max>  
<!--req, max. bit rate, unit: kb-->  
<!--req, 0-0K, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->  
<!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K,  
23-2048K-->  
<!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->  
<Range>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</Range>  
<Default>  
<!--req, default value for different products are different-->  
</Default>  
</AverageVideoBitrate>  
</VideoResolutionEntry>  
<VideoResolutionEntry>  
<!--req, 4096*1200 -->  
<Index>138</Index>  
<Name>4096*1200</Name>  
<Resolution>4096*1200</Resolution>  
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;  
11-12;  
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;  
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->  
  
<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</  
VideoFrameRate>  
<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame  
rate supported by P standard and N standard is same-->  
<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;  
11-12;  
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;  
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->  
  
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</  
VideoFrameRateN>
```

```
<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame  
rate supported by P standard and N standard is same-->  
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and  
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->  
<VideoBitrate>  
  <!--req,video bit rate-->  
  <Min>16</Min><!--req, the min bitrate(unit: Kb) -->  
  <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->  
  <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->  
  <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->  
  <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->  
  <Range>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</Range>  
</VideoBitrate>  
<AverageVideoBitrate>  
  <!--req, average video bit rate, it is valid when SmartCodec is enabled-->  
  <Min>32</Min>  
  <!--req, min. bit rate, unit: kb-->  
  <Max>16384</Max>  
  <!--req, max. bit rate, unit: kb-->  
  <!--req, 0-OK, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->  
>  
  <!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K,  
23-2048K-->  
  <!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->  
  <Range>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</Range>  
<Default>  
  <!--req, default value for different products are different-->  
</Default>  
</AverageVideoBitrate>  
</VideoResolutionEntry>  
<VideoResolutionEntry>  
  <!--req, 3840*1080 -->  
  <Index>139</Index>  
  <Name>3840*1080</Name>  
  <Resolution>3840*1080</Resolution>  
  <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;  
11-12;  
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;  
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->  
  
<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</  
VideoFrameRate>  
  <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame  
rate supported by P standard and N standard is same-->  
  <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;  
11-12;  
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;  
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->  
  
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</  
VideoFrameRateN>  
  <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
```

rate supported by P standard and N standard is same-->

<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->

```
<VideoBitrate>
<!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
<!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
<AverageVideoBitrate>
<!--req, average video bit rate, it is valid when SmartCodec is enabled-->
<Min>32</Min>
<!--req, min. bit rate, unit: kb-->
<Max>16384</Max>
<!--req, max. bit rate, unit: kb-->
<!--req, 0-0K, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
<!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K, 23-2048K-->
<!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
<Range>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</Range>
<Default>
<!--req, default value for different products are different-->
</Default>
</AverageVideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
<!--req, 2720*800 -->
<Index>140</Index>
<Name>2720*800</Name>
<Resolution>2720*800</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRate>
<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->
<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRateN>
<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
```

```

<VideoBitrate>
    <!--req,video bit rate-->
    <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
    <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
    <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
    <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
    <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
    <Range>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</Range>
</VideoBitrate>
<AverageVideoBitrate>
    <!--req, average video bit rate, it is valid when SmartCodec is enabled-->
    <Min>32</Min>
    <!--req, min. bit rate, unit: kb-->
    <Max>16384</Max>
    <!--req, max. bit rate, unit: kb-->
    <!--req, 0-OK, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
    <!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K, 23-2048K-->
    <!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
    <Range>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</Range>
<Default>
    <!--req, default value for different products are different-->
</Default>
</AverageVideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
    <!--req, 3840*1080 -->
    <Index>139</Index>
    <Name>3840*1080</Name>
    <Resolution>3840*1080</Resolution>
    <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
    <VideoBitrate>
        <!--req,video bit rate-->
        <Min>16</Min><!--req, the min bitrate(unit: Kb) -->

```

```

<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
<!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
<AverageVideoBitrate>
    <!--req, average video bit rate, it is valid when SmartCodec is enabled-->
    <Min>32</Min>
    <!--req, min. bit rate, unit: kb-->
    <Max>16384</Max>
    <!--req, max. bit rate, unit: kb-->
    <!--req, 0-OK, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
    <!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K, 23-2048K-->
    <!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
    <Range>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</Range>
    <Default>
        <!--req, default value for different products are different-->
    </Default>
    </AverageVideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
    <!--req, 512*232 -->
    <Index>141</Index>
    <Name>512*232</Name>
    <Resolution>512*232</Resolution>
    <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->
<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
    <VideoBitrate>
        <!--req,video bit rate-->
        <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
        <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
        <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
        <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->

```

```

<!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
<AverageVideoBitrate>
    <!--req, average video bit rate, it is valid when SmartCodec is enabled-->
    <Min>32</Min>
    <!--req, min. bit rate, unit: kb-->
    <Max>16384</Max>
    <!--req, max. bit rate, unit: kb-->
    <!--req, 0-OK, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
    <!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K, 23-2048K-->
    <!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
    <Range>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</Range>
    <Default>
        <!--req, default value for different products are different-->
    </Default>
</AverageVideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
    <!--req, 704*200 -->
    <Index>142</Index>
    <Name>704*200</Name>
    <Resolution>704*200</Resolution>
    <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->
<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
    <VideoBitrate>
        <!--req,video bit rate-->
        <Min>16</Min><!--req, the min bitrate(unit: Kb)-->
        <Max>16384</Max><!--req, the max bitrate(unit: Kb)-->
        <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K,-->
        <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,-->
        <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K-->
        <Range>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</Range>
    </VideoBitrate>

```

```

<AverageVideoBitrate>
    <!--req, average video bit rate, it is valid when SmartCodec is enabled-->
    <Min>32</Min>
    <!--req, min. bit rate, unit: kb-->
    <Max>16384</Max>
    <!--req, max. bit rate, unit: kb-->
    <!--req, 0-0K, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
    <!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K, 23-2048K-->
    <!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
    <Range>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</Range>
    <Default>
        <!--req, default value for different products are different-->
    </Default>
</AverageVideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
    <!--req, 512*152 -->
    <Index>143</Index>
    <Name>512*152</Name>
    <Resolution>512*152</Resolution>
    <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
    <VideoBitrate>
        <!--req,video bit rate-->
        <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
        <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
        <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K,-->
        <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,-->
        <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
        <Range>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</Range>
    </VideoBitrate>
    <AverageVideoBitrate>
        <!--req, average video bit rate, it is valid when SmartCodec is enabled-->
        <Min>32</Min>

```

```

<!--req, min. bit rate, unit: kb-->
<Max>16384</Max>
<!--req, max. bit rate, unit: kb-->
<!--req, 0-OK, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
>
<!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K, 23-2048K-->
<!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
<Range>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</Range>
<Default>
    <!--req, default value for different products are different-->
</Default>
</AverageVideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
    <!--req, 2048*900 -->
    <Index>144</Index>
    <Name>2048*900</Name>
    <Resolution>2048*900</Resolution>
    <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
        <VideoBitrate>
            <!--req,video bit rate-->
            <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
            <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
            <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
            <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
            <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
            <Range>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</Range>
        </VideoBitrate>
        <AverageVideoBitrate>
            <!--req, average video bit rate, it is valid when SmartCodec is enabled-->
            <Min>32</Min>
            <!--req, min. bit rate, unit: kb-->
            <Max>16384</Max>
            <!--req, max. bit rate, unit: kb-->

```

```

<!--req, 0-OK, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
<!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K,
23-2048K-->
<!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
<Range>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</Range>
<Default>
    <!--req, default value for different products are different-->
</Default>
</AverageVideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
    <!--req, 2048*600 -->
    <Index>145</Index>
    <Name>2048*600</Name>
    <Resolution>2048*600</Resolution>
    <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7;
28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
        <VideoBitrate>
            <!--req,video bit rate-->
            <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
            <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
            <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
            <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
            <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
            <Range>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</Range>
        </VideoBitrate>
        <AverageVideoBitrate>
            <!--req, average video bit rate, it is valid when SmartCodec is enabled-->
            <Min>32</Min>
            <!--req, min. bit rate, unit: kb-->
            <Max>16384</Max>
            <!--req, max. bit rate, unit: kb-->
            <!--req, 0-OK, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
        </AverageVideoBitrate>
    </VideoResolutionEntry>

```

```
<!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K,  
23-2048K-->  
    <!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->  
    <Range>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</Range>  
    <Default>  
        <!--req, default value for different products are different-->  
    </Default>  
    </AverageVideoBitrate>  
    </VideoResolutionEntry>  
    <VideoResolutionEntry>  
        <!--req, 1280*376 -->  
        <Index>146</Index>  
        <Name>1280*376</Name>  
        <Resolution>1280*376</Resolution>  
        <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;  
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5; 27-7;  
28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->  
  
<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</  
VideoFrameRate>  
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame  
rate supported by P standard and N standard is same-->  
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;  
11-12;  
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;  
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->  
  
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</  
VideoFrameRateN>  
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame  
rate supported by P standard and N standard is same-->  
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and  
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->  
    <VideoBitrate>  
        <!--req,video bit rate-->  
        <Min>16</Min><!--req, the min bitrate(unit: Kb) -->  
        <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->  
        <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->  
        <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->  
        <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->  
        <Range>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</Range>  
    </VideoBitrate>  
    <AverageVideoBitrate>  
        <!--req, average video bit rate, it is valid when SmartCodec is enabled-->  
        <Min>32</Min>  
        <!--req, min. bit rate, unit: kb-->  
        <Max>16384</Max>  
        <!--req, max. bit rate, unit: kb-->  
        <!--req, 0-0K, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->  
    <!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K,  
23-2048K-->
```

```

<!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
<Range>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</Range>
<Default>
    <!--req, default value for different products are different-->
</Default>
</AverageVideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
    <!--req, 8208*3072 or 4096*1536 -->
    <Index>150</Index>
    <Name>8208*3072</Name>
    <Resolution>8208*3072</Resolution>
    <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5; 27-7;
28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
        12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
        26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
    <VideoBitrate>
        <!--req,video bit rate-->
        <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
        <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
        <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
        <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
        <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
        <Range>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</Range>
    </VideoBitrate>
    <AverageVideoBitrate>
        <!--req, average video bit rate, it is valid when SmartCodec is enabled-->
        <Min>32</Min>
        <!--req, min. bit rate, unit: kb-->
        <Max>16384</Max>
        <!--req, max. bit rate, unit: kb-->
        <!--req, 0-OK, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
    >
        <!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K,
23-2048K-->
        <!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
        <Range>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</Range>

```

```

<Default>
    <!--req, default value for different products are different-->
</Default>
</AverageVideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
    <!--req, 4096*1536 -->
    <Index>151</Index>
    <Name>4096*1536</Name>
    <Resolution>4096*1536</Resolution>
    <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
    <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
    <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
    <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
    <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
        <VideoBitrate>
            <!--req,video bit rate-->
            <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
            <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
            <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
            <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
            <!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
            <Range>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</Range>
        </VideoBitrate>
        <AverageVideoBitrate>
            <!--req, average video bit rate, it is valid when SmartCodec is enabled-->
            <Min>32</Min>
            <!--req, min. bit rate, unit: kb-->
            <Max>16384</Max>
            <!--req, max. bit rate, unit: kb-->
            <!--req, 0-OK, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
        <!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K,
23-2048K-->
            <!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
            <Range>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</Range>
        <Default>

```

```

<!--req, default value for different products are different-->
</Default>
</AverageVideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
<!--req, 1280*1440 -->
<Index>189</Index>
<Name>4MPLite</Name>
<Resolution>1280*1440</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
<VideoBitrate>
<!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
<!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K, -->
<!--req, 20-1280K,21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
<AverageVideoBitrate>
<!--req, average video bit rate, it is valid when SmartCodec is enabled-->
<Min>32</Min>
<!--req, min. bit rate, unit: kb-->
<Max>16384</Max>
<!--req, max. bit rate, unit: kb-->
<!--req, 0-0K, 1-16K, 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K-->
<!--req, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K,
23-2048K-->
<!--req, 24-3072K, 25-4096K, 26-8192K, 27-16384K-->
<Range>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</Range>
<Default>
<!--req, default value for different products are different-->

```

```

        </Default>
    </AverageVideoBitrate>
</VideoResolutionEntry>
<DynamicAbility>
    <dynamicAbilityLinkTo opt="captureMode, vbrAverageCap"/>
    <!--req, dymatic related items, WDR enable, the related structure of type of lens is
NET_DVR_CAMERAPARAMCFG_EX byCaptureMode -->

    </DynamicAbility>
</VideoResolutionList>
<IntervalBPFrame>
    <Range>0,1,2</Range> <!--req, 0-BBP; 1-BP;2-single P-->
</IntervalBPFrame>
<SVCFunEnable opt="true,false"/><!--req,enable or disable, true-enable, false-disable-->
<SVCSupportEncodeType opt="0,1"/> /*0-Private264, 1-H264, 2-mpeg4, 7-M-JPEG??8-MPEG2 */
<EFrame>0</EFrame> //the largest encoding ability of E frame
<intervalIframe min="0" max="25"/><!--req Interval of I Frame-->
<StreamSmooth>
    <level min="" max="" />
<SmartCodecCap><!--req, high performance encoding-->
    <supportCodeType opt="H.264,H.265">
        <!--req, xs:string= "supported encoding type, when video encoding type is other, Smart264 and Smart265
(high performance encoding field) switch is hidden" -->
    </supportCodeType>
    <smartCodec opt="true,false" default="false"><!--req, xs:boolean="" , "high encoding switch"--></smartCodec>
    <H264><!--function needs to be hidden, after smart encoding is enabled after device is restarted, the capability
corresponding to hidden function will not be returned-->
        <readOnlyAbility opt="keyFrameInterval,Profile,SVC,videoBitrate">
            <!-- req, xs:string, "items need to be grayed out: I frame interval, encoding complexity, SVC"-->
        </readOnlyAbility>
        <BitrateType>
            <!--req, when Smart 264 is enabled, in VBR mode, a line is added under the bitrate upper limit, and the title is
average bitrate while the bitrate upper limit is grayed out which cannot be modified; default average bitrate is
converted based on bitrate upper limit, the range is [0, bitrate upper limit]. The average bitrate is saved independently
and bitrate upper limit is not reused; when bitrate type is CBR, and average bitrate is required to be hidden, the
bitrate upper limit is configurable.-->
            <Constant><!--req,CBR-->
                <support opt="videoBitrate">
                    <!--req, xs:string, "averageVideoBitrate(average bitrate),videoBitrate(bitrate upper limit)"-->
                </support>
                <hiddenAbility opt="averageVideoBitrate">
                    <!--req, xs:string, "averageVideoBitrate(average bitrate),videoBitrate(bitrate upper limit)"-->
                </hiddenAbility>
            </Constant>
            <Variable><!--req,VBR-->
                <support opt="averageVideoBitrate">
                    <!--req, xs:string, "averageVideoBitrate(average bitrate),videoBitrate(bitrate upper limit)"-->
                </support>
                <readOnlyAbility opt="videoBitrate">
                    <!--req, xs:string, "averageVideoBitrate(average bitrate),videoBitrate(bitrate upper limit)"-->
                </readOnlyAbility>
            </Variable>
        </BitrateType>
    </smartCodec>
</H264>

```

```

</BitrateType>
<smart264EnabledPrompt opt="prompt1,prompt2,prompt3,prompt13">
    <!--opt,wo,xs:string,"Smart264 enable prompt"-->
</smart264EnabledPrompt>
</H264>
<H265>
    <!--function needs to be hidden, after smart encoding is enabled after device is restarted, the capability
corresponding to hidden function will not be returned-->
    <readOnlyAbility opt="keyFrameInterval,Profile,SVC,fixedQuality">
        <!--req, xs:string, "items need to be grayed out: I frame interval, encoding complexity, SVC, video quality"-->
    </readOnlyAbility>
    <BitrateType>
        <Constant><!--req,CBR-->
            <support opt="videoBitrate">
                <!--req, xs:string, "averageVideoBitrate(average bitrate),videoBitrate(video bitrate)"-->
            </support>
            <hiddenAbility opt="averageVideoBitrate">
                <!--req, xs:string, "averageVideoBitrate(average bitrate),videoBitrate(video bitrate)"-->
            </hiddenAbility>
        </Constant>
        <Variable><!--req,VBR-->
            <support opt="averageVideoBitrate">
                <!--req, xs:string, "averageVideoBitrate(average bitrate),videoBitrate(video bitrate)"-->
            </support>
            <readOnlyAbility opt="videoBitrate">
                <!--req, xs:string, "averageVideoBitrate(average bitrate),videoBitrate(video bitrate)"-->
            </readOnlyAbility>
        </Variable>
    </BitrateType>
    <smart265EnabledPrompt opt="prompt4">
        <!--opt,wo,xs:string,"Smart265 enable prompt"-->
    </smart265EnabledPrompt>
</H265>
<rateType opt="constant,variable"/>
<!--opt,constant(CBR)??variable(VBR)-->
<!--opt,field byBitrateType in structure NET_DVR_COMPRESSION_INFO_V30 corresponding to rateType, all
supported by default when the node <rateType> is not returned. If the node is returned, the node content will be
resolved and displayed (sub-stream is similar to third stream definition)-->
</SmartCodecCap>
<!--req 1-Clear, 100-Smooth-->
    /*0-Private264, 1-H264, 2-mpeg4, 7-M-JPEG, 8-MPEG2*/
<supportEncodeType opt="1"/>
<supportRateType opt="constant,variable"/><!--req constant, variable-->
</StreamSmooth>
</MainChannel>
<SubChannelList>
    <SubChannelEntry>
        <index>1</index><!--req, sub channel -->
        <VideoEncodeType><!--req, 0- private 264, 1- standard h264, 2- standard mpeg4, 7- M-JPEG, 8- MPEG2 -->
            <Range>0,1,2,7,8</Range>
        </VideoEncodeType>
        <VideoEncodeEfficiency><!--req, the complexity of video compression: 0-low,1-medium,2-high -->

```

```

<Range>0</Range>
</VideoEncodeEfficiency>
<VideoCabac>1</VideoCabac><!--req, whether it supports CABAC or not -->
<VideoResolutionList>
  <VideoResolutionEntry>
    <Index>1</Index>
    <Name>CIF</Name>
    <Resolution>352*288</Resolution>
    <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
  <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
  <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
  <!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
  <!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
    <VideoBitrate><!--req,video bit rate-->
      <Min>16</Min><!--req, the min bitrate(unit: Kb) -->
      <Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
      <!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K, -->
      <!--req, 11-256K,12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,20-1280K, -->
      <!--req, 21-1536K,22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
      <Range>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</Range>
    </VideoBitrate>
    </VideoResolutionEntry>
    <VideoResolutionEntry>
      <Index>2</Index>
      <Name>QCIF</Name>
      <Resolution>176*144</Resolution>
      <!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5; 27-7;
28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
  <!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
  <!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7;
28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

```

```
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
```

<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same--></p>

<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format--></p>

```
<VideoBitrate><!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K,11-256K, -->
<!--req, 12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,20-1280K,21-1536K, -->
<!--req, 22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
```

```
</VideoResolutionEntry>
```

```
<VideoResolutionEntry>
```

```
<Index>6</Index>
```

```
<Name>QVGA</Name>
```

```
<Resolution>320*240</Resolution>
```

<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3--></p>

```
<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
```

<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same--></p>

<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10; 11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3--></p>

```
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
```

<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same--></p>

<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format--></p>

```
<VideoBitrate><!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K,11-256K, -->
<!--req, 12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,20-1280K,21-1536K, -->
<!--req, 22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
```

```
</VideoResolutionEntry>
```

```
<DynamicAbility>
```

<dynamicAbilityLinkTo opt="captureMode"/><!--req, dymatic related items, WDR enable, the related structure of type of lens is NET_DVR_CAMERAPARAMCFG_EX byCaptureMode -->

```
</DynamicAbility>
```

```
</VideoResolutionList>
```

```

<IntervalBPFrame><!--req, 0-BBP; 1-BP;2-single P -->
<Range>0,1,2</Range>
</IntervalBPFrame>
<EFrame>0</EFrame><!--req, the max encoding capability of E frame -->
</SubChannelEntry>
</SubChannelList>
<EventChannel>
<VideoEncodeType><!--req, 0- private 264, 1- standard h264, 2- standard mpeg4, 7- M-JPEG, 8- MPEG2 -->
<Range>0,1,2,7,8</Range>
</VideoEncodeType>
<VideoEncodeEfficiency><!--req, the complexity of video compression: 0-low,1-medium,2-high -->
<Range>0</Range>
</VideoEncodeEfficiency>
<VideoCabac>1</VideoCabac><!--req, whether it supports CABAC or not -->
<VideoResolutionList>
<VideoResolutionEntry>
<Index>1</Index>
<Name>CIF</Name>
<Resolution>352*288</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5; 27-7;
28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7;
28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
<VideoBitrate><!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160K,9-192K,10-224K,11-256K, -->
<!--req, 12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,20-1280K,21-1536K, -->
<!--req, 22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
</VideoResolutionEntry>
<VideoResolutionEntry>
<Index>2</Index>
<Name>QCIF</Name>
<Resolution>176*144</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5; 27-7;
```

28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

```
<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
```

<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->

```
<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7;
28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->
```

```
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
```

<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->

```
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
```

```
<VideoBitrate><!--req,video bit rate-->
```

```
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
```

```
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
```

```
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K,11-256K, -->
```

```
<!--req, 12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,20-280K,21-1536K, -->
```

```
<!--req, 22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
```

```
<Range>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</Range>
```

```
</VideoBitrate>
```

```
</VideoResolutionEntry>
```

```
<VideoResolutionEntry>
```

```
<Index>6</Index>
```

```
<Name>QVGA</Name>
```

```
<Resolution>320*240</Resolution>
```

```
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7;
28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->
```

```
<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
```

<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->

```
<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12; 12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3; 26-5; 27-7;
28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->
```

```
<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
```

<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame rate supported by P standard and N standard is same-->

```
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
```

```
<VideoBitrate><!--req,video bit rate-->
```

```
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
```

```
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
```

```
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K,11-256K, -->
```

```
<!--req, 12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,20-1280K,21-1536K, -->
```

```

<!--req, 22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
</VideoResolutionEntry>
<DynamicAbility>
<dynamicAbilityLinkTo opt="captureMode"/><!--req, dymatic related items, WDR enable, the related structure
of type of lens is NET_DVR_CAMERAPARAMCFG_EX byCaptureMode -->
</DynamicAbility>
</VideoResolutionList>
<IntervalBPFrame><!--req, 0-BBP; 1-BP;2-single P -->
<Range>0,1,2</Range>
</IntervalBPFrame>
<EFrame>0</EFrame><!--req, the max encoding capability of E frame -->
</EventChannel>
<Stream3>
<VideoEncodeType><!--req, 0- private 264, 1- standard h264, 2- standard mpeg4, 7- M-JPEG, 0xff-invalid -->
<Range>0,1,2,7</Range>
</VideoEncodeType>
<VideoEncodeEfficiency><!--req, the complexity of video compression: 0-low,1-medium,2-high -->
<Range>0</Range>
</VideoEncodeEfficiency>
<VideoCabac>1</VideoCabac><!--req, whether it supports CABAC or not -->
<VideoResolutionList>
<VideoResolutionEntry>
<Index>1</Index>
<Name>CIF</Name>
<Resolution>352*288</Resolution>
<!--req, Frame rate supported by P standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;26-5;
27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRate>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRate>
<!--req, frame rate supported by P standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, Frame rate supported by N standard: 0-Full;1-1/16; 2-1/8; 3-1/4; 4-1/2; 5-1; 6-2; 7-4; 8-6; 9-8; 10-10;
11-12;
12-16; 13-20; 14-15; 15-18; 16-22; 17-25; 18-30; 19-35; 20-40; 21-45; 22-50; 23-55; 24-60; 25-3;
26-5; 27-7; 28-9; 29-100; 30-120; 31-24; 32-48; 33-8.3-->

<VideoFrameRateN>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33</
VideoFrameRateN>
<!--req, frame rate supported by N standard, if there is no VideoFrameRateN node, it indicates that the frame
rate supported by P standard and N standard is same-->
<!--req, encoding bit rate node: if the device supports custom bit rate, it will be return in maximum and
minimum format; if the device does not support the custom bit rate, it will be returned in Range format-->
<VideoBitrate>
<!--req,video bit rate-->
<Min>16</Min><!--req, the min bitrate(unit: Kb) -->
<Max>16384</Max><!--req, the max bitrate(unit: Kb) -->
<!--req, 0-reserved,1-16K,2-32K,3-48k,4-64K,5-80K,6-96K,7-128K,8-160k,9-192K,10-224K,11-256K, -->

```

```
<!--req, 12-320K,13-384K,14-448K,15-512K,16-640K,17-768K,18-896K,19-1024K,20-1280K,21-1536K,-->
<!--req, 22-1792K,23-2048K,24-3072K,25-4096K,26-8192K,27-16384K -->
<Range>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</Range>
</VideoBitrate>
</VideoResolutionEntry>
</VideoResolutionList>
<IntervalBPFrame>
<!--req, 0-BBP; 1-BP;2-single P -->
<Range>0,1,2</Range>
</IntervalBPFrame>
<EFrame>0</EFrame><!--req, the max encoding capability of E frame -->
</Stream3>
</ChannelEntry>
</ChannelList>
</VideoCompressInfo>
</AudioVideoCompressInfo>
```

F.167 XML_Cap_Color

Color capability message in XML format

```
<?xml version="1.0" encoding="utf-8"?>
<Color version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<brightnessLevel min="" max=""><!--opt, xs:integer, brightness--></brightnessLevel>
<contrastLevel min="" max=""><!--opt, xs:integer, contrast--></contrastLevel>
<saturationLevel min="" max=""><!--opt, xs:integer, saturation--></saturationLevel>
<hueLevel min="" max=""><!--opt, xs:integer, hue--></hueLevel>
<grayScale><!--opt, gray scale-->
<grayScaleMode><!--opt, xs:string, gray scale mode: "indoor,outdoor"--></grayScaleMode>
</grayScale>
<nightMode opt="true,false">
<!--opt, xs:boolean, enable night mode, when its value is "true", the saturation can be adjusted, otherwise, the
saturation cannot be adjusted-->
</nightMode>
</Color>
```

F.168 XML_Cap_DefaultParam

DefaultParam capability message in XML format

```
<DefaultParam version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<DialNum min="1" max="32"><!--opt, xs:string--></DialNum>
<Username min="1" max="32"><!--opt, xs:string--></Username>
<Password min="1" max="32"><!--opt, xs:string--></Password>
<APNname min="1" max="32"><!--opt, xs:string--></APNname>
<VerifyProto opt="auto,CHAP,PAP"><!--req, xs:string, verification protocol: "auto, CHAP, PAP"--></VerifyProto>
</DefaultParam>
```

F.169 XML_Cap_DeviceInfo

MXL message about capability of device information

```
<DeviceInfo version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <deviceName min="" max=""><!--required, xs:string, device name--></deviceName>
  <DockStation>
    <!--optional, dock station configuration-->
    <Platform>
      <!--optional, platform configuration-->
      <type opt="none, 9533, 8618, ISAPI"><!--required, xs:string, platform type--></type>
      <ip><!--optional, xs:string, IP address --></ip>
      <port><!--optional, xs:integer, communication port--></port>
      <userName><!--required, xs:string, user name, which is used for the dock station to log in to platform--></userName>
      <password><!--required, xs:string, password, which is used for the dock station to log in to platform, it should be encrypted--></password>
    </Platform>
    <centralStorageBackupEnabled opt="true, false"><!--optional, xs:boolean, whether to enable central storage backup--></centralStorageBackupEnabled>
  </DockStation>
  <languageType
opt="chinese,english/spanish/portuguese/italian/french/russian/german/polish/turkish/greek/czech/brazilianPortuguese,
slovenian/swedish/norwegian/slovak/serbian/dutch/hungarian/irish/bulgarian/hebrew"/><!--optional, xs:string-->
  <deviceID min="0" max="128"><!--read-only, required, xs:string, uuid--></deviceID>
  <deviceDescription min="0" max="16"><!--optional, xs:string--></deviceDescription>
  <deviceLocation opt="STD-CGI, hangzhou"><!--optional, xs:string--></deviceLocation>
  <systemContact opt="STD-CGI"><!--optional, required, xs:string--></systemContact>
  <model min="0" max="64"><!--read-only, required, xs:string--></model>
  <serialNumber min="0" max="48"><!--read-only, required, xs:string--></serialNumber>
  <macAddress min="0" max="64"><!--read-only, required, xs:string--></macAddress>
  <firmwareVersion min="0" max="64"><!--read-only, required, xs:string--></firmwareVersion>
  <firmwareReleasedDate min="0" max="64"><!--read-only, optional, xs:string--></firmwareReleasedDate>
  <bootVersion min="0" max="16"><!--read-only, optional, xs:string--></bootVersion>
  <bootReleasedDate min="0" max="16"><!--read-only, optional, xs:string--></bootReleasedDate>
  <hardwareVersion min="0" max="16"><!--read-only, optional, xs:string--></hardwareVersion>
  <encoderVersion min="0" max="64"><!--read-only, optional, xs:string--></encoderVersion>
  <encoderReleasedDate min="0" max="64"><!--read-only, optional, xs:string--></encoderReleasedDate>
  <decoderVersion min="0" max="64"><!--read-only, optional, xs:string--></decoderVersion>
  <decoderReleasedDate min="0" max="64"><!--read-only, optional, xs:string--></decoderReleasedDate>
  <deviceType opt="IPCamera, IPDome, DVR, HybirdNVR, NVR, DVS, IPZoom"><!--read-only, required, xs:string--></deviceType>
  <telecontrolID min="1" max="255"><!--optional, xs:integer, "1-255"--></telecontrolID>
  <supportBeep><!--optional, xs:boolean: "true/false"--></supportBeep>
  <firmwareVersionInfo><!--read-only, optional, xs:string--></firmwareVersionInfo>
  <subChannelEnabled><!--optional, xs:boolean: "true/false"--></subChannelEnabled>
  <thrChannelEnabled><!--optional, xs:boolean: "true/false"--></thrChannelEnabled>
  <actualFloorNum><!--required, xs:integer, "1-128"--></actualFloorNum>
  <radarVersion><!--optional, xs:string, radar version--></radarVersion>
  <powerOnMode opt="button,adapter" def="button"><!--optional, xs:string, device startup mode: "button"-press button to power on (default), "adapter"-connect adapter to power on--></powerOnMode>
```

```
<webVersion><!--optional, read-only, xs:string, web version No., it is the current value by default--></webVersion>
<deviceRFProgramVersion><!--optional, read-only, xs:string, version No. of the device's RF (Radio Frequency)
program, it is the current value by default--></deviceRFProgramVersion>
<securityModuleSerialNo><!--optional, read-only, xs:string, serial No. of the security module, it is the current value
by default--></securityModuleSerialNo>
<securityModuleVersion><!--optional, read-only, xs:string, version No. of the security module, it is the current value
by default--></securityModuleVersion>
<securityChipVersion><!--optional, read-only, xs:string, version No. of the security chip, it is the current value by
default--></securityChipVersion>
<securityModuleKeyVersion><!--optional, read-only, xs:string, version No. of the security module key, it is the current
value by default--></securityModuleKeyVersion>
<UIDLampRecognition><!--optional, information of the UID lamp recognition device-->
<enabled opt="true,false"><!--optional, xs:boolean, whether to enable--></enabled>
</UIDLampRecognition>
<bootTime><!--optional, xs:string, read-only, system boot time, ISO 8601 format; the maximum length is 32 bytes--></bootTime>
<isSupportNewVersionDevlanguageSwitch><!--optional, xs:boolean, whether it supports switching language by new
version of the protocol, related URI: /ISAPI/System/DeviceLanguage/capabilities. If the device does not return this
node, it does not indicate that the device does not support switching language by new version of the protocol, and
whether the device supports this function is determined by the obtained capability--></
isSupportNewVersionDevlanguageSwitch>
<ZigBeeVersion min="0",max="16"><!--optional, xs:string, ZigBee module version--></firmwareVersion>
<R3Version min="0",max="16"><!--optional, xs:string, R3 module version--></R3Version>
<RxVersion min="0",max="16"><!--optional, xs:string, Rx module version--></RxVersion>
<isResetDeviceLanguage opt="true,false">
    <!--optional, boolean, whether it supports switching the device language on the HikConnect and the Portal after the
    security control panel is upgraded-->false
</isResetDeviceLanguage>
<bspVersion min="1" max="1"><!--optional, xs:string, BSP software version, read-only--></bspVersion>
<dspVersion min="1" max="1"><!--optional, xs:string, DSP software version, read-only--></dspVersion>
<localUIVersion min="1" max="1"><!--optional, xs:string, local UI version, read-only--></localUIVersion>
</DeviceInfo>
```

F.170 XML_Cap_DeviceLanguage

DeviceLanguage capability message in XML format

```
<DeviceLanguage version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <language
        opt="SimChinese,TraChinese,English,Russian,Bulgarian,Hungarian,Greek,German,Italian,Czech,Slovakia,French,Polish,D
        utch,Portuguese,Spanish,Romanian,Turkish,Japanese,Danish,Swedish,Norwegian,Finnish,Korean,Thai,Estonia,Vietname
        se,Hebrew,Latvian,Arabic,Sovenian,Croatian,Lithuanian,Serbian,BrazilianPortuguese,Indonesian,Ukrainian">
        <!--req, xs:string, language supported by the device: "SimChinese"-simplified Chinese, "TraChinese"-traditional
        Chinese, "English", "Russian", "Bulgarian", "Hungarian", "Greek", "German", "Italian", "Czech", "Slovakia", "French",
        "Polish", "Dutch", "Portuguese", "Spanish", "Romanian", "Turkish", "Japanese", "Danish", "Swedish", "Norwegian",
        "Finnish", "Korean", "Thai", "Estonia", "Vietnamese", "Hebrew", "Latvian", "Arabic", "Sovenian"-Slovenian, "Croatian",
        "Lithuanian", "Serbian", "BrazilianPortuguese"-Brazilian Portuguese, "Indonesian", "Ukrainian"-->
    </language>
    <upgradeFirmWareEnabled><!--optional, xs:boolean, whether to upgrade the firmware: true (the firmware of the
    device needs to be updated when the language is switched because there is no upgrading package for the language in
```

```
the local device)--></upgradeFirmWareEnabled>
</DeviceLanguage>
```

F.171 XML_Cap_DiagnosedDataParameter

XML message about capability of exporting diagnose information

```
<?xml version="1.0" encoding="utf-8"?>
<DiagnosedDataParameter version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled opt="true,false">
    <!--required, xs:boolean, whether to enable exporting diagnose information: true-yes, false-no-->
  </enabled>
  <HardwareInfo opt="true,false">
    <!--optional, xs:boolean,opt="true,false", whether to enable exporting hardware information, such as running
information of motor, fan, heater, compressor, supplement light, wiper, and so on-->
  </HardwareInfo>
  <SoftwareInfo opt="true,false">
    <!--optional, xs:boolean, whether to enable exporting software information, such as software status, running log,
system information-->
  </SoftwareInfo>
  <logInfo opt="true,false"><!--optional, xs:boolean, whether to enable exporting startup log--></logInfo>
  <isSupportServer><!--optional, xs:boolean, whether the device supports diagnostic server configuration (related
URI: /ISAPI/System/diagnosedData/server?format=json)--></isSupportServer>
</DiagnosedDataParameter>
```

F.172 XML_Cap_Dial

Dial capability message in XML format

```
<Dial version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled opt="true,false" def="false"><!--req, xs:boolean--></enabled>
  <DialMethod opt="auto,manual">
    <!--req, xs:string, "auto, manual", set it to manual when accessing the private network, and you can configure dial-
up parameters(including access number, user name, password, APN, and verification protocol-->
  </DialMethod>
  <SwitchMethod opt="auto,4GFirst,3GFirst,manualto2G,manualto3G,manualto4G,cableFirst">
    <!--req, xs:string, "auto,4GFirst,3GFirst, manualto2G, manualto3G,manualto4G,cableFirst"-->
  </SwitchMethod>
  <OfflineTime min="30" max="65535"><!--opt, xs:integer, unit: second--></OfflineTime>
  <UIMCardNum min="1" max="32"><!--opt, xs:string--></UIMCardNum>
  <DialNum min="1" max="32"><!--opt, xs:string--></DialNum>
  <Username min="1" max="32"><!--opt, xs:string--></Username>
  <Password min="1" max="32"><!--opt, xs:string--></Password>
  <APNname min="1" max="32"><!--opt, xs:string--></APNname>
  <SIMNum min="" max=""><!--opt, xs:string, SIM card No. (mobile phone number)--></SIMNum>
  <MTU min="100" max="1500"><!--opt, xs: integer--></MTU>
  <VerifyProto opt="auto,CHAP,PAP"><!--req, xs:string, verification protocol: "auto, CHAP, PAP"--></VerifyProto>
  <DefaultParam/><!--opt, default parameters, see details in the message of XML_Cap_DefaultParam-->
  <netAPN min="" max=""><!--opt, xs:string, APN configuration for the private network-->
```

```
<Flow><!--opt, flow configuration-->
<limitEnabled opt="true,false"><!--opt, xs:boolean, whether to enable flow limitation--></limitEnabled>
<consumeFlow opt="true,false"><!--opt, xs:boolean, whether supports displaying flow usage, unit: MB--></
consumeFlow>
<threshold min="" max=""><!--opt, threshold of flow, unit: MB--></threshold>
</Flow>
<enabled4G opt="true,false" def="true"><!-- opt, xs:boolean, whether to enable 4G--></enabled4G>
<isSupportNetworkKeepAlive><!--opt, xs:boolean, whether it supports network keepalive --></
isSupportNetworkKeepAlive>
<enabledDNS opt="true,false" def="true"><!-- opt, xs:boolean, whether to enable DNS manual configuration--></
enabledDNS>
<pinCode min="" max=""><!--optional, write-only, xs:string, PIN code-->
<ISPName min="" max=""><!--optional, xs:string, ISP (Internet Service Provider) name--></ISPName>
<IMEINo min="" max=""><!--optional, xs:string, IMEI code--></IMEINo>
<ICCID><!--optional, xs:string, ICCID code--></ICCID>
<netType opt="2G,3G,4G"><!--optional, xs:string, network type--></netType>
</Dial>
```

See Also

[XML_Cap_DefaultParam](#)

F.173 XML_Cap_EagleFocusing

EagleFocusing capability message in XML format

```
<?xml version="1.0" encoding="utf-8"?>
<EagleFocusing version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id><!--req,xs:string--></id>
<enabled opt="true,false"><!--req, xs:boolean, "true"-start calibration, "false"-stop calibration--></enabled>
<normalizedScreenSize><!--req,ro-->
<normalizedScreenWidth><!--req,ro,xs:integer--></normalizedScreenWidth>
<normalizedScreenHeight><!--req,ro,xs:integer--></normalizedScreenHeight>
</normalizedScreenSize>
<EagleFocusingRegionList><!--opt-->
<EagleFocusingRegion><!--list-->
<id><!--req, xs:string--></id>
<sid><!--opt,xs:string, scene ID--></sid>
<rate><!--opt,xs:integer, rate--></rate>
<spotNum><!--opt,xs:integer, the number of calibration points--></spotNum>
<type><!--req,xs:string, "line", region type, now only supports "line"--></type>
<Region><!--dep, depends on type-->
<RegionCoordinatesList size="4"><!--opt-->
<RegionCoordinates><!--list-->
<positionX><!--req,xs:integer,coordinate--></positionX>
<positionY><!--req,xs:integer,coordinate--></positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</Region>
</EagleFocusingRegion>
</EagleFocusingRegionList>
```

```

<sceneNum min="" max=""><!--opt, xs:integer, the maximum number of supported scenes--></sceneNum>
<isSupportSceneGoto opt="true,false"><!--opt, xs:boolean, whether device supports turning to specified scene--></isSupportSceneGoto>
<rate min="" max=""><!--opt, xs:integer, rate--></rate>
<spotNum min="" max=""><!--opt, xs:integer, the number of calibration points on line--></spotNum>
<onlyReadParam opt="rate,spotNum"><!--opt, xs:string, "rate,spotNum"--></onlyReadParam>
<isSupportAuto opt="true,false"><!--req, xs:boolean, whether device supports automatic calibration of rapid focus--></isSupportAuto>
</EagleFocusing>

```

F.174 XML_Cap_EHome

XML message about the configuration capability of accessing servers via ISUP

```

<Ehome version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<enabled opt="true,false"><!--optional, boolean, whether to enable--></enabled>
<id min="1" max="1"><!--opt, xs:string, ISUP center ID--></id>
<GPRSAddressingFormatType opt="ipaddress"><!--opt, xs:string, GPRS address type. If this node is not returned, the address type will be determined by addressingFormatType which indicates that the address types configured by different networks do not need to be distinguished--></GPRSAddressingFormatType>
<addressingFormatType opt="ipaddress,hostname"><!--req, xs:string--></addressingFormatType>
<hostName min="" max=""><!--dep, xs:string, domain name--></hostName>
<ipAddress min="" max=""><!--dep, xs:string--></ipAddress>
<ipv6Address min="" max=""><!--dep, xs:string--></ipv6Address>
<portNo min="" max=""><!--opt, xs:integer--></portNo>
<deviceID min="" max=""><!--req, xs:string--></deviceID>
<registerStatus min="" max=""><!--ro, xs:boolean--></registerStatus>
<ServerAddressList><!--optional, array, information list of servers-->
<ServerAddress><!--optional, object, server information-->
  <id><!-required, string, server type: "alarm", "picture", "ntp" (NTP time synchronization), "backup", "enforce" (enforcement), "bayonet" (checkpoint)--></id>
  <ipAddress><!--required, string, IP address of the server--></ipAddress>
  <portNo><!--required, int, port No. of the server--></portNo>
</ServerAddress>
</ServerAddressList>
<key min="" max=""><!--opt, xs:string, encryption key of ISUP version 5.0--></key>
<version min="" max=""><!--ro, xs:string--></version>
<netWork opt="0,1,2,3,4"/><!--opt, xs:integer, 0-make no sense, 1-automatic, 2-wired network preferred, 3-wired network, 4-3G/4G/GPRS-->
<voiceDeviceType opt="bluetooth,client,local"><!--opt, xs:string, two-way audio device type: "bluetooth"-bluetooth device, "client", "local"--></voiceDeviceType>
<protocolVersion opt="v2.0,v2.6,v4.0,v5.0"><!--opt, xs:string, protocol version: "v2.0,v2.6,v4.0,v5.0". If this node is set to "v2.0", the device can only use protocol v2.0 to register; if this node is set to "v2.6", "v4.0" or "v5.0", the device will firstly use this protocol to register. If this node is not returned, the protocol version will be determined by <version>--></protocolVersion>
<pictureServerID min="1" max="64"><!--optional, xs:integer, ID of the linked picture storage server (/ISAPI/System/PictureServer)--></pictureServerID>
<ISUPID min="1" max="64"><!--optional, xs:integer, ISUP multi-platform ID--></ISUPID>
<isSupportTest><!--optional, xs:boolean, whether it supports ISUP (EHome) server test--></isSupportTest>
<periodicTestEnabled opt="true,false"><!--optional, xs:boolean, whether to enable periodic test. After this function
```

```
is enabled, the host will send a Test event to the server every configured interval for checking whether the link is valid  
for interaction--></periodicTestEnabled>  
<periodicTestTime min="10" max="86400"><!--optional, xs:integer, periodic test interval, unit: second. This node is  
valid when <b>periodicTestEnabled</Ehome>
```

F.175 XML_Cap_ExternalDevice

XML message about external device configuration capability

```
<?xml version="1.0" encoding="utf-8"?>  
<ExternalDevice version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  
<SupplementLight/><!--opt, whether device supports configuring supplement light, see XML_Cap_SupplementLight  
for details-->  
<THScreen/><!--opt, capability of peripheral screen configuration, corresponding url: /ISAPI/System/externalDevice/  
THScreen-->  
<isSupportSupplementLightChannelSetting>  
<!--opt, xs:boolean, whether device supports configuring peripheral supplement light of specified channel,  
corresponding url: /ISAPI/System/channel/<ID>/externalDevice/supplementLight-->  
</isSupportSupplementLightChannelSetting>  
</ExternalDevice>
```

See Also

[XML_Cap_SupplementLight](#)

F.176 XML_Cap_EZVIZ

XML message about the EZVIZ access configuration capability

```
<?xml version="1.0" encoding="utf-8"?>  
<EZVIZ version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  
<!--redirect: whether to enable redirecting server address-->  
<!--verificationCode: only admin users are allowed to edit the verification code-->  
<enabled opt="true,false"><!--required, xs:boolean--></enabled>  
<registerStatus opt="true,false"><!--read-only, optional, xs:boolean--></registerStatus>  
<redirect opt="true,false"><!--optional, xs:boolean--></redirect>  
<serverAddress><!--optional-->  
<addressingFormatType min="" max=""><!--required, xs:string, "ipaddress,hostname"--></addressingFormatType>  
<hostName min="" max=""><!--dependent, xs:string--></hostName>  
<ipAddress min="" max=""><!--dependent, xs:string--></ipAddress>  
<ipv6Address min="" max=""><!--dependent, xs:string--></ipv6Address>  
</serverAddress>  
<verificationCode min="" max=""><!--optional, xs:string--></verificationCode>  
<offlineStatus opt="secretKeyInvalid,netUnreachable,blocklist,unknown">  
<!--read-only, dependent, xs:string, device offline status: "secretKeyInvalid"-invalid verification code,  
"netUnreachable"-network is unreachable, "blocklist"-blocklist, "unknown"-unknown error; it is valid when the value  
of registerStatus is "false"-->  
</offlineStatus>
```

```
<streamEncrypteEnabled opt="true,false"><!--optional, xs:boolean, whether to enable stream encryption--></streamEncrypteEnabled>
<isSupportUpISAPI><!--optional, xs:boolean, whether the device supports uplink transmitting ISAPI message-->true</isSupportUpISAPI>
<isSupportDownISAPI><!--optional, xs:boolean, whether the device supports downlink transmitting ISAPI message-->true</isSupportDownISAPI>
<isSupportEZVIZQRCode><!--optional, xs:boolean, whether the device supports getting the Guarding Vision QR code (related URI: /ISAPI/System/Network/EZVIZ/QRCode?format=json)--></isSupportEZVIZQRCode>
<periodicTestEnabled opt="true,false">
    <!--optional, xs:boolean, whether to enable periodic test. After this function is enabled, the host will send a Test event to the EZVIZ platform every configured interval for checking whether the link is valid for interaction-->true
</periodicTestEnabled>
<periodicTestTime min="10" max="86400">
    <!--optional, xs:integer, periodic test interval, unit: second, range: [10,86400]-->5
</periodicTestTime>
</EZVIZ>
```

F.177 XML_Cap_FocusConfiguration

FocusConfiguration capability message in XML format

```
<FocusConfiguration>
<focusStyle opt="AUTO,MANUAL,SEMIAUTOMATIC"><!--req, xs:string--></focusStyle>
<focusLimited opt="50,100,300,600,1000,2000,5000,10000,15000" def="1000"><!--opt, xs:integer--></focusLimited>
<focusPosition/><!--dep, xs:integer, depends on FocusStyle-->
<focusSpeed><!--opt, xs:intger--></focusSpeed>
<focusSensitivity min="0" max="2" def="1"><!--optional, xs:intger, sensitivity of focus, ranging from 0 to 2. It is valid when the focus mode is automatic or semi-automatic--></focusSensitivity>
<temperatureChangeAdaptEnabled opt="true,false"><!--optional, xs:boolean--></temperatureChangeAdaptEnabled>
<relativeFocusPos min="0" max="4000" def=""><!--optional, xs:intger--></relativeFocusPos>
<highTemperaturePriority opt="true,false"><!--optional, xs:boolean, enable high temperature priority mode--></highTemperaturePriority>
<focusStatus opt="success,failed,working"><!--optional, xs:string, focus status: success, failed, working (focusing)--></focusStatus>
<motorMovementType opt="manual,auto,armingLocation"><!--optional, xs:string, motor movement type: manual, auto (automatic), armingLocation (arming location)--></motorMovementType>
<focusRange opt="min,medium,max"><!--optional, xs:string, focus adjustment range: min (the minimum), medium (the medium), max (the maximum)--></focusRange>
</FocusConfiguration>
```

F.178 XML_Cap_FTPNotification

XML message about the capability of a specific FTP

```
<FTPNotification version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id>
    <!--required, xs:string, FTP ID-->
</id>
<enabled>
```

```
<!--required, xs:boolean, whether to enable the FTP server-->
</enabled>
<useSSL>
  <!--optional, xs:boolean-->
</useSSL>
<addressingFormatType opt="ipaddress,hostname">
  <!--required, xs:string, this node can only be set to "ipaddress"-->
</addressingFormatType>
<hostName>
  <!--dependent, xs:string-->
</hostName>
<ipAddress>
  <!--dependent, xs:string-->
</ipAddress>
<ipv6Address>
  <!--dependent, xs:string-->
</ipv6Address>
<portNo>
  <!--optional, xs:integer, FTP port No.-->
</portNo>
<userName>
  <!--required, xs:string, user name-->
</userName>
<password>
  <!--wo, xs:string, password-->
</password>
<passiveModeEnabled>
  <!--optional, xs:boolean-->
</passiveModeEnabled>
<annoyftp>
  <!--optional, xs:boolean-->
</annoyftp>
<uploadPicture>
  <!--optional, xs:boolean-->
</uploadPicture>
<uploadVideoClip>
  <!--optional, xs:boolean-->
</uploadVideoClip>
<uploadPath>
  <!--req-->
<pathDepth>
  <!--required, xs:integer, directory levels, up to 16 levels can be supported-->
</pathDepth>
<topDirNameRule opt="none, devName, devId, devIp, positionInfo, time_month, time_date, illegalType, direction, site, chanName, chanId, lanId, customize ,time, buildUnitNo">
  <!--dependent, xs:string, parent directory name: "none", "devName"-device name, "devId"-device ID, "devIp"-device IP address, "positionInfo"-camera 1, "time_month"-usage date (YYYY-MM), "time_date"-usage date (YYYY-MM-DD), "illegalType"-violation type, "direction"-direction, "site"-place, "chanName"-channel name, "chanId"-channel No., "lanId"-lane No., "customize"-custom, "time", "buildUnitNo"-building No. and unit No. This node is set to NULL by default-->
</topDirNameRule>
<topDirName min="" max="">
```

```
<!--dependent, xs:string, custom string for parent directory, the maximum string length is 32 bytes. This node is valid when <topDirNameRule> is "customize". If attributes of "min" and "max" are not obtained after parsing, the default value will be returned without error message-->
<topDirName><!--dependent, xs:string, custom string for parent directory, the maximum string length is 32 bytes. This node is valid when <topDirNameRule> is "customize"-->
<subDirNameRule opt="none, devName, devId, devIp, positionInfo, time_month, time_date, illegalType, direction, site, chanName, chanId, lanId, customize ,time, buildUnitNo">
    <!--dependent, xs:string, child directory name-->
</subDirNameRulesubDirName min="" max="">
    <!--dependent, xs:string, custom string for child directory, the maximum string length is 32 bytes. This node is valid when <subDirNameRule> is "customize". If attributes of "min" and "max" are not obtained after parsing, the default value will be returned without error message-->
<subDirName><!--dependent, xs:string, custom string for child directory, the maximum string length is 32 bytes. This node is valid when <subDirNameRule> is "customize"-->
<threeDirNameRule opt="none, devName, devId, devIp, positionInfo, time_month, time_date, illegalType, direction, site, chanName, chanId, lanId, customize ,time, buildUnitNo">
    <!--dependent, xs:string, the third directory name-->
</threeDirNameRulethreeDirName><!--required, xs:string, custom string for the third directory, the maximum string length is 32 bytes. This node is valid when <threeDirNameRule> is "customize"-->
<fourDirNameRule opt="none, devName, devId, devIp, positionInfo, time_month, time_date, illegalType, direction, site, chanName, chanId, lanId, customize ,time, buildUnitNo">
    <!--dependent, xs:string, the fourth directory name-->
</fourDirNameRulefourDirName><!--required, xs:string, custom string for the fourth directory, the maximum string length is 32 bytes. This node is valid when <fourDirNameRule> is "customize"-->
</uploadPath>
<FtpUploadvehiclePicNamemode opt="default,custom">
    <!--required, xs:string, ro, mode-->test
</modeNameRuleTypeRuleTypeItemList size="8">
    <!--optional, xs:object, ro, rule type list-->
<RuleTypeItemididitem opt="capture_time,plate_No,alarm_type,camera_name,vehicleLogo,carDirectionType,countryOrRegion">
    <!--required, xs:string, ro, rule item: "capture_time"-capture time, "plate_No"-license plate No., "alarm_type"-alarm type, "camera_name"-camera name, "vehicleLogo"-vehicle brand, "carDirectionType"-driving direction (upward, downward), "countryOrRegion"-country/region-->capture_time
</itemcameraName min="1" max="64">
    <!--optional, xs:string, ro, camera name-->test
</cameraName
```

```
</RuleTypeItem>
</RuleTypeItemList>
</NameRuleType>
</vehiclePicName>
</FtpUpload>
<picArchivingInterval min="" max="">
  <!--optional, xs:integer, the value is between 1 and 30, 0-close-->
</picArchivingInterval>
<picNameRuleType opt="default,prefix">
  <!--optional, xs:string-->
</picNameRuleType>
<picNamePrefix min="0" max="32">
  <!--dependent, xs:string-->
</picNamePrefix>
<ftpPicNameRuleType opt="videoIntercom,ITC">
  <!--required, xs:string, type of FTP picture name rule: "videoIntercom"-rule used by video intercom products, "ITC"-rule used by traffic cameras-->
</ftpPicNameRuleType>
<FTPPicNameRule>
  <!--dependent, picture name rule of a specific FTP-->
  <ItemList/><!--required, see details in the message of XML_Cap_ItemList-->
  <delimiter>
    <!--required, xs:string, delimiter, which is a single character and the default value is "_"-->
  </delimiter>
  <customStr min="1" max="128">
    <!--required, xs:string, custom string-->
  </customStr>
</FTPPicNameRule>
<upDataType opt="0,1,2">
  <!--optional, xs:integer, picture uploading type: 0-all, 1-checkpoint, 2-violation. When only one FTP server is enabled, this node can only be set to 0. When two FTP servers are both enabled, you should set 1 for one FTP server and set 2 for another FTP server, which means that two FTP servers cannot be set to the same type-->
</upDataType>
<uploadPlateEnable>
  <!--optional, xs:boolean, whether to enable uploading license plate thumbnail-->
</uploadPlateEnable>
<site min="1" max="128">
  <!--required, xs:string, place, the maximum string length is 128 bytes-->
</site>
<roadNum min="1" max="32">
  <!--required, xs:string, intersection No., the maximum string length is 32 bytes-->
</roadNum>
<instrumentNum min="1" max="32">
  <!--required, xs:string, device No., the maximum string length is 32 bytes-->
</instrumentNum>
<direction min="1" max="32">
  <!--required, xs:string, direction No., the maximum string length is 32 bytes-->
</direction>
<directionDesc min="1" max="32">
  <!--required, xs:string, direction description, the maximum string length is 32 bytes-->
</directionDesc>
<monitoringInfo1 min="1" max="44">
```

```
<!--required, xs:string, camera 1 information, the maximum string length is 44 bytes-->
</monitoringInfo1>
<uploadAttachedInfomation>
  <!--required, xs:boolean, whether to upload additional information-->
</uploadAttachedInfomation>
<BrokenNetHttp><!--optional, whether it supports ANR (automatic network replenishment)-->
  <enabled opt="true,false"><!--optional, xs:boolean, whether to enable ANR (automatic network replenishment)-->
</enabled>
  <supportEventType opt="personQueueCounting"/><!--optional, xs:string, event types supporting FTP ANR:>
"personQueueCounting"-person queue counting detection, "personQueueTime"-person queue time detection,
"personQueueRealTimeData"-upload real-time data of person queue counting detection, "faceCapture"-face capture
and recognition, "fieldDetection"-intrusion, "attendedBaggage"-object removal, "unattendedBaggage"-unattended
baggage, "regionExiting"-region exiting, "regionEntrance"-region entrance, "lineDetection"-line crossing detection-->
</BrokenNetHttp>
<uploadProtocolType opt="FTP,SFTP"><!--optional, xs:string, type of the uploaded protocol--></uploadProtocolType>
</FTPNotification>
```

See Also

[XML_Cap_ItemList](#)

F.179 XML_Cap_FTPNotificationList

XML message about the FTP capability

```
<FTPNotificationList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <FTPNotification/><!--optional, see details in the message of XML_Cap_FTPNotification-->
</FTPNotificationList>
```

See Also

[XML_Cap_FTPNotification](#)

F.180 XML_Cap_HardwareService

HardwareService capability message in XML format

```
<HardwareService version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <IrLightSwitch><!--opt-->
    <mode opt="open,close"><!--req, xs:string --></mode>
  </IrLightSwitch>
  <ABF><!--opt-->
    <enabled><!--req, xs:boolean --></enabled>
  </ABF>
  <LED><!--opt-->
    <enabled><!--req, xs:boolean --></enabled>
  </LED>
  <Defog><!--opt-->
    <enabled><!--req, xs:boolean --></enabled>
  </Defog>
```

```

<SupplementLight><!--opt-->
<enabled><!--req, xs:boolean --></enabled>
<isSupportFireLaserLight opt="true,false"><!--opt, xs:boolean --></isSupportFireLaserLight>
<isSupportSupplementLightWord opt="true,false"><!--opt, xs:boolean --></isSupportSupplementLightWord>
<captureWithSupplimentLightEnabled opt="true,false">
    <!--opt, xs:boolean, enable snapshot supplement light or not-->
</captureWithSupplimentLightEnabled>
</SupplementLight>
<Deicing><!--opt-->
<enabled><!--req, xs:boolean --></enabled>
</Deicing>
<ManualDeicing><!--opt-->
<enabled><!--req, xs:boolean --></enabled>
</ManualDeicing>
<mutexAbility opt="laserLight, deicing">
    <!--req, mutual exclusion ability, the laser light and deicing (including manual deicing and automatic deicing) are
mutual exclusive-->
</mutexAbility>
<HighTemperatureProtection><!--opt-->
<enabled><!--req, xs:boolean --></enabled>
<temperatureType opt="90,100,110" def="90"><!--dep, xs:string, unit:°C --></temperatureType>
</HighTemperatureProtection>
</HardwareService>

```

F.181 XML_Cap_hddList

hddList capability message in XML format

```

<hddList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema" size="">
<hdd><!--list-->
<id><!--read-only, required, xs: string; ID--></id>
<hddName><!--read-only, required, xs: string--></hddName>
<hddPath><!--optional, read-only, xs: string--></hddPath>
<hddType opt="IDE,SATA,eSATA,NFS,iSCSI,Virtual Disk"><!--read-only, required, xs: string--></hddType>
<status
opt="ok,unformatted,error,idle,mismatch,offline,smartFailed,reparing,formating,notexist,unRecordHostFormatted">
    <!--read-only, required, xs: string, unRecordHostFormatted (unformatted in education sharing system)-->
</status>
<capacity><!--read-only, required, xs: float, unit: MB--></capacity>
<freeSpace><!--read-only, required, xs: float, unit: MB--></freeSpace>
<property opt="RW,read-only,Redund"><!--required, xs: string, HDD properties--></property>
<group><!--optional, xs: string; HDD group ID--></group>
<reservedDayCfg min="0" max="1000"><!--optional, xs:integer, video saving duration, it is between 0 and 1000,
unit: day--></reservedDayCfg>
<DataModeList><!--optional, read-only, current HDD allocation mode-->
<DataMode>
    <type opt="recordStorage,pictureCloudStorage,fileStorage">
        <!--required, xs:string, storage application type: recordStorage (video storage), pictureCloudStorage (picture to
be saved in cloud storage), fileStorage (file-storage)-->
    </type>

```

```
<occupancyRate><!--required, xs: integer, HDD usage, range: [0,100]--></occupancyRate>
</DataMode>
</DataModeList>
<formatType opt="FAT32,EXT4" def="FAT32">
    <!--optional, read-only, xs: string, formatting type, this node is only available for SD card; if this node does not
exist, the default formatting type is FAT32-->
</formatType>
<Encryption>
    <passwordLen min="6" max="64"/>
    <encryptionStatus opt="unencrypted,encrypted,verfyFailed"><!--optional, read-only, xs:string, encryption status:
"unencrypted", "encrypted", "verfyFailed"-verification failed--></encryptionStatus>
        <encryptFormatType opt="FAT32,EXT4"><!--optional, read-only, xs:string--></encryptFormatType>
    </Encryption>
    <isSupportLogStorage><!--optional, xs:boolean, whether it can be used as a log HDD--></isSupportLogStorage>
</hdd>
</hddList>
```

F.182 XML_Cap_HiddenInformation

XML message about hidden configuration capability of POS information

```
<HiddenInformation version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id>
    <!--required, xs:integer, device channel No.-->
</id>
<funcType opt="POS">
    <!--required, xs:string, functional type: "POS"-hide POS information-->
</funcType>
<PosCofig>
    <!--dependent, this node is valid only when the value of node <funcType> is set to "POS"-->
    <keyWordOne min="0" max="32">
        <!--required, xs:string, keyword 1-->
    </keyWordOne>
    <keyWordTwo min="0" max="32">
        <!--required, xs:string, keyword 2-->
    </keyWordTwo>
    <keyWordThree min="0" max="32">
        <!--required, xs:string, keyword 3-->
    </keyWordThree>
</PosCofig>
</HiddenInformation>
```

F.183 XML_Cap_ImageChannel

XML message about image channel capability

```
<ImageChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id><!--req, xs:integer--></id>
<enabled><!--req, xs:boolean--></enabled>
```

```
<videoInputID><!--req, xs:integer--></videoInputID>
<FocusConfiguration/><!--opt, refer to the message XML_Cap_FocusConfiguration for details-->
<LensInitialization/><!--opt-->
<ImageFlip/><!--opt-->
<ImageFreeze/><!--opt-->
<proportionalpan/><!--opt-->
<WDR>
  <mode opt="close,open,auto"><!--req, xs:string, WDR mode: "open,close,auto"--></mode>
  <WDRLevel max="100" min="0"><!--opt, xs:integer--></WDRLevel>
  <WDRContrastLevel max="" min="0"><!--opt, xs:integer--></WDRContrastLevel>
  <WDRLevel1 max="" min="0"><!--opt, xs:integer--></WDRLevel1>
</WDR>
<BLC/><!--opt-->
<NoiseReduce/><!--opt-->
<ImageEnhancement/><!--opt-->
<SlowShutter/><!--opt-->
<DSS/><!--opt-->
<WhiteBlance/><!--opt-->
<Exposure/><!--opt, exposure configuration parameters, refer to the message XML_Exposure for details-->
<DayNightGate/><!--opt-->
<BrightEnhance/><!--opt-->
<Sharpness>
  <SharpnessLevel max="100" min="0"><!--req, xs:integer--></SharpnessLevel>
</Sharpness>
<gammaCorrection/><!--opt-->
<powerLineFrequency/><!--opt-->
<Color><!--opt, image adjustment capability-->
  <brightnessLevel max="100" min="0"><!--opt, xs:integer, brightness--></brightnessLevel>
  <contrastLevel max="100" min="0"><!--opt, xs:integer, contrast--></contrastLevel>
  <saturationLevel max="100" min="0"><!--opt, xs:integer, saturation--></saturationLevel>
  <grayScale>
    <grayScaleMode opt="outdoor,indoor"><!--opt, xs:string, gray scale mode: "indoor,outdoor"--></grayScaleMode>
  </grayScale>
</Color>
<lrcutFilter/><!--opt, configuration parameters of day/night auto-switch, refer to the message XML_IrcutFilter for details-->
<Scene/><!--opt-->
<EPTZ/><!--opt-->
<EIS/><!--opt-->
<HLC/> <!--opt-->
<ZoomLimit/> <!--opt-->
<corridor/><!--opt-->
<Dehaze/><!--opt-->
<ImageMode opt="standard, indoor, outdoor, dimLight"/><!--opt, xs:string-->
<enableImageLossDetection><!--opt, boolean--></enableImageLossDetection>
<CaptureMode/><!--opt-->
<IrLight/><!--opt-->
<LensDistortionCorrection/><!--opt-->
<SupplementLight/><!--opt, supplement light configuration capability, refer to the message XML_Cap_SupplementLight for details-->
<OpticalDehaze/><!--opt-->
<ManualRanging/><!--opt-->
```

```

<OIS/><!--opt-->
<isSupportLaserSpotManual><!--opt, boolean--></isSupportLaserSpotManual>
<isSupportLaserSpotAdjustment><!--opt, boolean--></isSupportLaserSpotAdjustment>
<DigitalZoom>
  <ZoomRatio opt="1x,2x,4x,8x,16x,32x"><!--req, xs:string--></ZoomRatio>
</DigitalZoom>
<Palettes>
  <mode
opt="WhiteHot,BlackHot,Fusion1,Rainbow,Fusion2,Ironbow1,Ironbow2,Sepia,Color1,Color2,IceFire,Rain,RedHot,Green
Hot,DeepBlue,Color3"><!--
opt, xs:string, "WhiteHot,BlackHot,Fusion1,Rainbow,Fusion2,Ironbow1,Ironbow2,Sepia,Color1,Color2,IceFire,Rain,RedH
ot,GreenHot,DeepBlue,Color3"--></mode>
  <ColorateTarget><!--dep, mode="WhiteHot", "target coloration"-->
    <ColorateTargetModeList>
      <ColorateTargetMode><!--list-->
        <id><!--req, xs:integer "index, which starts from 1"--></id>
        <mode opt="colorateHotAreae,colorateIntervalArea,colorateColdArea"><!--
req, xs:string, "colorateHotAreae,colorateIntervalArea,colorateColdArea"--></mode>
          <enabled opt="true,false"><!--req, xs:boolean, "true,false"--></enabled>
          <TemperatureLimit><!--req "temperature range"-->
            <minTemperature><!--dep <mode> is colorateHotAreae/colorateIntervalArea, xs:float--></minTemperature>
            <maxTemperature><!--dep <mode> is colorateColdArea/colorateIntervalArea, xs:float--></maxTemperature>
          </TemperatureLimit>
          <Color><!--req, "area color"-->
            <R><!--req, xs:integer--></R>
            <G><!--req, xs:integer--></G>
            <B><!--req, xs:integer--></B>
          </Color>
        </ColorateTargetMode>
      </ColorateTargetModeList>
    </ColorateTarget>
    <supportColorateTargetMode opt="WhiteHot,BlackHot"><!--opt, xs:string, palette modes that support target
coloration--></supportColorateTargetMode>
  </Palettes>
<ExposureSync opt="true,false" def="false">
  <enabled><!--req, xs:boolean--></enabled>
</ExposureSync>
<BrightnessSuddenChangeSuppressionCap><!--opt-->
<isSupportIcr><!--opt, boolean--></isSupportIcr>
<isSupportMultishut><!--opt, boolean--></isSupportMultishut>
<isSupportPlateBright><!--opt, boolean--></isSupportPlateBright>
<isSupportJPEGParam><!--opt, boolean--></isSupportJPEGParam>
<isSupportDarkEnhance><!--opt, boolean--></isSupportDarkEnhance>
<isSupportHdr><!--opt, boolean--></isSupportHdr>
<isSupportLse><!--opt, boolean--></isSupportLse>
<isSupportMce><!--opt, boolean--></isSupportMce>
<isSupportScve><!--opt, boolean--></isSupportScve>
<isSupportSectionCtrl><!--opt, boolean--></isSupportSectionCtrl>
<isSupportAutoContrast><!--opt, boolean--></isSupportAutoContrast>
<isSupportGrayRange><!--opt, boolean--></isSupportGrayRange>
<isSupportLSEDetail><!--opt, boolean--></isSupportLSEDetail>
<isSupportCapture><!--opt, boolean--></isSupportCapture>

```

```
<isSupportBrightEnhance><!--opt, boolean--></isSupportBrightEnhance>
<isSupportRecord><!--opt, xs:boolean--></isSupportRecord>
<isSupportDefog><!--opt, xs:boolean--></isSupportDefog>
<isSupportGeneral><!--opt, xs:boolean, whether to display general configuration page on the interface--></isSupportGeneral>
<isSupportIa><!--opt, xs:boolean, whether to display video configuration page on the interface--></isSupportIa>
<NoiseReduce2D>
  <noiseReduce2DEnable><!--req, xs:boolean--></noiseReduce2DEnable>
  <noiseReduce2DLevel min="0" max="100"><!--dep,xs:integer--></noiseReduce2DLevel>
</NoiseReduce2D>
<Shutter>
  <ShutterLevel min="100" max="40000"><!--req, xs:integer--></ShutterLevel>
</Shutter>
<Gain>
  <GainLevel min="0" max="100"><!--req, xs:integer--></GainLevel>
</Gain>
<TempRange/><!--opt, temperature range capability, see XML_Cap_tempRange for details-->
</ImageChannel>
```

Remarks

The nodes **<DigitalZoom>** and **<ExposureSync>** are not supported by thermographic automation thermal camera (DS-2TA03-15SVI, DS2TA06-25SVI).

F.184 XML_Cap_Integrate

Message about the configuration capability of the access protocol in XML format.

```
<Integrate version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <CGI>
    <enable opt="true, false"><!--optional, xs:boolean--></enable>
    <certificateType opt="digest,digest/baisc" def="digest"><!--required, xs:string--></certificateType>
  </CGI>
  <ONVIF>
    <enable opt="true, false"><!--optional, xs:boolean--></enable>
    <certificateType opt="digest,digest/WSSE,WSSE" def="digest/WSSE"><!--required, xs:string--></certificateType>
  </ONVIF>
  <ISAPI>
    <enable opt="true, false"><!--optional, xs:boolean, "true,false"--></enable>
  </ISAPI>
  <IPCAddStatus>
    <enable opt="true, false"><!--optional, xs:boolean, "true,false"--></enable>
  </IPCAddStatus>
  <Bonjour>
    <enable opt="true, false"><!--optional, xs:boolean, "true,false"--></enable>
  </Bonjour>
  <Megaeyes>
    <enable opt="true, false"><!--optional, xs:boolean, "true,false"--></enable>
  </Megaeyes>
  <WebSocket>
    <enable><!--optional, xs:boolean, "true,false"--></enable>
  </WebSocket>
```

```
</WebSocket>
<WebSocketS>
<enable><!--optional, xs:boolean, "true,false"--></enable>
</WebSocketS>
</Integrate>
```

F.185 XML_Cap_IPFilter

IPFilter capability message in XML format

```
<IPFilter version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<enabled opt="true, false"><!--req, xs:boolean--></enabled>
<permissionType opt="deny, allow">
<!--req, xs:string, IP address filter status: "deny, allow". If this node is configured, it will overwrite value of all <permissionType> in <IPFilterAddress>-->
</permissionType>
<IPFilterAddressList size="32"><!--opt, the attribute "size" refers to the maximum number of IP addresses supported by the device. If the attribute "size" is not returned, the default supported number of IP addresses is 48-->
<IPFilterAddress>
<id min="" max=""><!--req, xs:string, ID--></id>
<permissionType opt="deny, allow"><!--dep, ro, xs:string--></permissionType>
<addressFilterType opt="mask,range">
<!--dep, ro, xs:string, filter type: "mask"-single address, "range"-address range-->
</addressFilterType>
<AddressRange><!--dep, this node is valid when <addressFilterType> contains "range"-->
<startIPAddress min="" max=""><!--dep, xs:string--></startIPAddress>
<endIPAddress min="" max=""><!--dep, xs:string--></endIPAddress>
<startIPv6Address min="" max=""><!--dep, xs:string--></startIPv6Address>
<endIPv6Address min="" max=""><!--dep, xs:string--></endIPv6Address>
</AddressRange>
<AddressMask><!--dep, it is valid when <addressFilterType> contains "mask"-->
<ipAddress min="" max=""><!--dep, xs:string--></ipAddress>
<ipv6Address min="" max=""><!--dep, xs:string--></ipv6Address>
<bitMask min="" max=""><!--opt, xs:string--></bitMask>
<bitMaskIPV6 min="" max=""><!--req, xs:string, IPv6 prefix length--></bitMaskIPV6>
</AddressMask>
<describeMsg min="" max=""><!--opt, xs:string--></describeMsg>
</IPFilterAddress>
</IPFilterAddressList>
</IPFilter>
```

F.186 XML_Cap_ItemList

ItemList capability message in XML format

```
<ItemList size="15">
<Item>
<itemID min="1" max="15">
<!--req, xs:string, item ID, which is between 1 and 15-->
```

```
</itemID>
<itemOrder
opt="none,devlp,time,buildUnitNo,outDoorDevNo,unlockType,devName,deviceNo,channelName,channelNo,
plateNo,plateColor,laneNo,carSpeed,
positionInfo1,pictureNo,CarNo,speedLimit,illegalCode,siteNo,directionNo,carColor,platePosition,carType,illegalType,cu
stom">
    <!--req, xs:string, name element: "none", "devlp"-device IP address, "time", "buildUnitNo"-building No. and unit
No., "outDoorDevNo"-door station, "unlockType"-unlocking type, "devName"-device name, "deviceNo"-device No.,
"channelName"-channel name, "channelNo"-channel No., "plateNo"-license plate number, "plateColor"-license plate
color, "laneNo"-lane No., "carSpeed"-vehicle speed, "positionInfo1"-camera 1, "pictureNo"-picture No., "CarNo"-_
vehicle No., "speedLimit"-speed limit, "illegalCode"-violation code, "siteNo"-intersection No., "directionNo"-direc
tion No., "carColor"-vehicle color, "platePosition"-license plate coordinates, "carType"-vehicle type, "illegalType"-violation
type, "custom"-->
    </itemOrder>
    <itemCustomStr min="1" max="32">
        <!--req, xs:string, element custom string, which is between 1 and 32, unit: bytes. This node is valid only when
<itemOrder> is "custom". Currently traffic cameras only support one custom name-->
        </itemCustomStr>
    </Item>
</ItemList>
```

F.187 XML_Cap_LensDistortionCorrection

LensDistortionCorrection capability message in XML format

```
<LensDistortionCorrection version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<enabled><!--req, xs: boolean--></enabled>
<level opt="level1,level2,level3,custom">
    <!--opt, xs: string, distortion correction level: level 1, level 2, level 3, custom-->
</level>
<accurateLevel min="0" max="100">
    <!--opt, xs: integer, correction strength, value range: [0,100]-->
</accurateLevel>
<zoomedInDistantViewLevel min="0" max="100">
    <!--opt, xs: integer, zooming in range, value range: [0-100], this node is valid when distortion correction is enabled-->
</zoomedInDistantViewLevel>
<horizontalFOV min="0" max="100"><!--opt, xs: integer, horizontal FOV, value range: [0,100]--></horizontalFOV>
<verticalFOV min="0" max="100"><!--opt, xs: integer, vertical FOV, value range: [0,100]--></verticalFOV>
</LensDistortionCorrection>
```

F.188 XML_Cap_MACFilter

MACFilter capability message in XML format

```
<MACFilter version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<enabled opt="true,false"><!--req, xs:boolean--></enabled>
<permissionType opt="deny, allow"><!-- req, xs:string, "deny, allow"--></permissionType>
<MACFilterAddressList size="">
    <MACFilterAddress>
```

```
<id> <!--req, xs:string,id--></id>
<MACAddress max=""><!--req, xs:string--></MACAddress>
</MACFilterAddress>
</MACFilterAddressList>
</MACFilter>
```

F.189 XML_Cap_MaxElevation

MaxElevation capability message in XML format

```
<MaxElevation version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <mElevation min="-20" max="0"><!--req, xs:integer, the lower limit of max. tilt-angle--> </mElevation>
  <mVerElevation><!--req, xs:integer, the upper limit of max. tilt-angle, normally it is 90°--></mVerElevation>
</MaxElevation>
```

F.190 XML_Cap_MountingScenario

MountingScenario capability message in XML format

```
<MountingScenario version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <mode
opt="indoor,outdoor,day,night,morning,nightfall,mode1,mode2,mode3,mode4,street,lowIllumination,custom1,custom
2,normal,road,faceSnap,highway,road2,backlight,frontlight">
  <!--req, xs:string, "lowIllumination"-low illumination, "backlight"-back light, "frontlight"-front light, "faceSnap"-face
picture capture-->
  </mode>
</MountingScenario>
```

F.191 XML_Cap_SlaveCamera

SlaveCamera capability message in XML format

```
<SlaveCamera version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
<SlaveCameraInfoList>
  <SlaveCameraInfo>
    <id min="1" max="4">
      <!--req, xs:integer-->
    </id>
    <serverAddress>
      <addressingFormatType opt="ipaddress,hostname">
        <!--req, xs:string, "ipaddress,hostname"-->
      </addressingFormatType>
      <hostName max="">
        <!--dep, xs:string-->
      </hostName>
      <ipAddress>
        <!--dep, xs:string-->
      </ipAddress>
```

```

</ipAddress>
<ipv6Address>
  <!--dep, xs:string-->
</ipv6Address>
</serverAddress>
<userName max="">
  <!--req, xs:string-->
</userName>
<passWord max="">
  <!--req, xs:string-->
</passWord>
<portNo min="" max="">
  <!--req, xs:integer-->
</portNo>
<loginStatus opt="login/logout">
  <!--req,xs:string-->
</loginStatus>
</SlaveCameraInfo>
</SlaveCameraInfoList>
</SlaveCamera>

```

F.192 XML_Cap_Schedule

Schedule capability message in XML format

```

<Schedule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--req, xs:string, ID--></id>
  <eventType
    opt="IO,VMD,videoloss,PIR,linedetection,fielddetection,audioexception,facedetection,regionEntrance,regionExiting,loitering,group,rapidMove,parking,unattendedBaggage,attendedBaggage,storageDetection,shipsDetection,HUMANATTRIBUTION,humanAttribute,faceContrast,faceSnap, mixedTargetDetection">
    <!--opt, xs:string, event type-->
  </eventType>
  <inputIOPortID><!--ro, dep, xs:string, alarm input ID--></inputIOPortID>
  <outputIOPortID><!--ro, dep, xs:string, alarm output ID--></outputIOPortID>
  <videoInputChannelID><!--ro, dep, xs:string, video input channel ID--></videoInputChannelID>
  <TimeBlockList size="8"><!--req-->
    <TimeBlock>
      <dayOfWeek opt="1,2,3,4,5,6,7">
        <!--opt, xs:integer, days of the week in ISO8601 time format, 1-Monday, 2-Tuesday, 3-Wednesday, 4-Thursday, 5-Friday, 6-Saturday, 7-Sunday-->
      </dayOfWeek>
      <TimeRange><!--req, time periods in arming schedule-->
        <beginTime><!--req, xs:time, start time of time period, in ISO8601 time format--></beginTime>
        <endTime><!--req, xs:time, end time of time period, in ISO8601 time format--></endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
  <HolidayBlockList><!--opt, holiday arming period-->
    <TimeBlock>

```

```
<TimeRange><!--req, time period-->
<beginTime><!--req, xs:time, start time, in ISO8601 time format--></beginTime>
<endTime><!--req, xs:time, end time, in ISO8601 time format--></endTime>
</TimeRange>
</TimeBlock>
</HolidayBlockList>
</Schedule>
```

F.193 XML_Cap_storage

XML message about configuration capability of storage working mode

```
<?xml version="1.0" encoding="utf-8"?>
<storage version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<hddList size=""><!--optional, the contents are same to the message XML_hddList (related URI: /ISAPI/ContentMgmt/Storage/hdd)--></hddList>
<nasList size=""><!--optional, the contents are same to the message XML_nasList (related URI: /ISAPI/ContentMgmt/Storage/nas)--></nasList>
<workMode opt="group,quota,extract"><!--optional, xs:string, working mode: "group", "quota", "extract"--></workMode>
<supportDataReconstruction opt="all,record">
<!--optional, xs:string, whether the device supports disk data reconstruction, and the supported data type for reconstruction: "all", "record"-->
</supportDataReconstruction>
<isSupportSSDManagement>
<!--optional, xs:string, whether the device supports SSD management (related URI: /ISAPI/ContentMgmt/storage/ssd/<ID>/capabilities?format=json)-->
</isSupportSSDManagement>
<ssdList size=""><!--optional, the contents are same to the message XML_ssdlList (related URI: /ISAPI/ContentMgmt/Storage/ssd)--></ssdList>
</storage>
```

F.194 XML_Cap_storageExtension

XML message about storage strategy capability.

```
<storageExtension xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0" >
<LoopEnable opt="true,false">
<!--optional, xs:boolean, whether to enable overwriting-->
</LoopEnable>
<enableDormant>
<!--optional, xs:boolean-->
</enableDormant>
<packDuration max="300" min="1">
<!--optional, xs:integer, unit: minute-->
</packDuration>
<logStorageMode opt="system,custom">
<!--optional, xs:string, log storage mode: "system"-system default mode, "custom"-custom mode-->
</logStorageMode>
```

```

<logStorageId>
  <!--dependent, xs:integer, disk ID (saving logs in a disk) or array number (saving logs in disk arrays), which are valid
only when logStorageMode is "custom"; -->
</logStorageId>
<logStorageCycle min="0" max="90" def="90">
  <!--dependent, xs:integer, log storage period, which is valid only when logStorageMode is "custom"-->
</logStorageCycle>
</storageExtension>

```

F.195 XML_Cap_StreamingChannel

StreamingChannel capability message in XML format

```

<StreamingChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id opt="111,222,333,444"><!--req, xs:string, ID--></id>
  <channelName min="0" max="64"><!--req, xs:string, channel name--></channelName>
  <enabled opt="true,false" def="true"><!--req, xs:boolean--></enabled>
  <Transport><!--req-->
    <rtpPortNo min="0" max="65535" def="554">554</rtpPortNo>
    <maxPacketSize min="0" max="1500"><!--optional, xs:integer--></maxPacketSize>
    <audioPacketLength min="0" max="5000"/>
    <audioInboundPacketLength min="0" max="5000"/><!--optional, xs:integer-->
    <audioInboundPortNo min="0" max="65535"/><!--optional, xs:integer-->
    <videoSourcePortNo min="0" max="65535"/><!--optional, xs:integer-->
    <audioSourcePortNo min="0" max="65535"/><!--optional, xs:integer-->
  <ControlProtocolList><!--req-->
    <ControlProtocol><!--list-->
      <streamingTransport opt="RTSP/RTP,HTTP"><!--req, xs:string, "HTTP,RTSP,SHTTP,SRTP"--></streamingTransport>
    </ControlProtocol>
  </ControlProtocolList>
  <Unicast><!--opt-->
    <enabled opt="true,false" def="false"/><!--req, xs:boolean-->
    <rtpTransportType opt="RTP/UDP,RTP/TCP"/><!--optional, xs:string, "RTP/UDP,RTP/TCP"-->
  </Unicast>
  <Multicast><!--opt-->
    <enabled opt="true,false" def="false"/><!--req, xs:boolean-->
    <userTriggerThreshold/><!--optional, xs:integer-->
    <videoDestPortNo min="1" max="65535" def="8860"/><!--optional, xs:integer-->
    <audioDestPortNo min="1" max="65535" def="8860"/><!--optional, xs:integer-->
    <destIpAddress min="8" max="16"/><!--dep, xs:string-->
    <destIPv6Address min="15" max="39"/><!--dep, xs:string-->
    <ttl min="0" max="127" def="1"/><!--optional, xs:integer-->
    <activeMulticastEnabled>
      <!--optional, xs: boolean, "true, false", whether to enable active multicast, which is mutually exclusive with
passive multicast-->
    </activeMulticastEnabled>
    <packagingFormat opt="RTP,TS,ES,PS"><!--optional, xs: string, encapsulation format--></packagingFormat>
    <FecInfo><!--opt-->
      <fecRatio min="0" max="100">
        <!--req, read-only, xs: integer, extra bandwidth occupation ratio of forward error correction (FEC) data, the value

```

is between 0 and 100, the default value is 0-->

```

</fecRatio>
<fecDestPortNo>
  <!--optional, xs: integer, port No. of FEC multicast, the default port No. is specified by device-->
</fecDestPortNo>
</FecInfo>
</Multicast>
<Security><!--opt-->
  <enabled opt="true,false" def="false"/><!--req, xs:boolean-->
  <certificateType opt="digest,digest/basic" def="digest"><!--req, xs:string--></certificateType>
  <SecurityAlgorithm><!--dependency, this node is valid when certificateType is "digest" or "digest/basic"-->
    <algorithmType opt="MD5,SHA256,MD5/SHA256"><!--optional, xs:string, algorithm type: "MD5", "SHA256",
    "MD5/SHA256"--></algorithmType>
  </SecurityAlgorithm>
</Security>
<SRTPMulticast><!--opt-->
  <SRTPVideoDestPortNo min="" max=""><!--optional, xs:integer--></SRTPVideoDestPortNo>
  <SRTPAudioDestPortNo min="" max=""><!--optional, xs:integer--></SRTPAudioDestPortNo>
</SRTPMulticast>
</Transport>
<Video>
  <enabled opt="true,false">true</enabled>
  <videoInputChannelID opt="1,2,3,4">2</videoInputChannelID>
  <videoCodecType opt="MJPEG,MPEG4">MPEG4</videoCodecType>
  <videoScanType opt="interlaced,progressive">progressive</videoScanType>
  <videoResolutionWidth min="0" max="640">640</videoResolutionWidth>
  <videoResolutionHeight min="0" max="480">480</videoResolutionHeight>
  <videoDiffResolutionList><!--optional, list, if this node exists, it indicates that the resolution depends on the node
<videoCodecType>, and <videoResolutionWidth> and <videoResolutionHeight> are invalid. For forward
compatibility, the maximum set of <videoCodecType>, <videoResolutionWidth>, and <videoResolutionHeight>-->
    <videoDiffResolution>
      <videoCodecType>
        <!--req, xs:string, "MPEG4,MJPEG,3GP,H.264,HEVC,MPNG,SVAC"-->
      </videoCodecType>
      <videoResolutionWidth><!--req, xs:integer--></videoResolutionWidth>
      <videoResolutionHeight><!--req, xs:integer--></videoResolutionHeight>
    </videoDiffResolution>
  </videoDiffResolutionList>
  <videoPositionX min="0" max="640">0</videoPositionX>
  <videoPositionY min="0" max="480">0</videoPositionY>
  <videoQualityControlType opt="CBR,VBR">CBR</videoQualityControlType>
  <constantBitRate min="50" max="4000" dynamic="true">2000</constantBitRate>
  <maxFrameRate opt="2500,1250,625,312,156,78, 830">2500</maxFrameRate>
  <keyFrameInterval min="0" max="10000">1000</keyFrameInterval>
  <rotationDegree opt="0,90,180,270" def="0">0</rotationDegree>
  <mirrorEnabled opt="true,false" def="false">false</mirrorEnabled>
  <snapShotImageType opt="JPEG" def="JPEG">JPEG</snapShotImageType>
  <intelligentInfoDisplayMethod opt="player,non-player"><!--optional, xs:string, method of displaying intelligent
information: "player", "non-player"--></intelligentInfoDisplayMethod>
  <minimumResolutionSupportedBySmartCode>
    <!--optional, xs:string, the minimum resolution supported by smart coding (smart264 and smart265), e.g.,
    "640*512". If this node is returned, it indicates the smart code of current device (H7, H5) is limited by the minimum

```

resolution, and the upper layer can get the supported minimum resolution for smart coding of current device. For the resolution lower than the minimum resolution does not support smart coding. If this node is not returned, it indicates the smart coding of current device (H3) is not limited by minimum resolution-->

```
</minimumResolutionSupportedBySmartCode>
</Video>
<Audio>
  <enabled opt="true,false" def="false">false</enabled>
  <audioInputChannelID opt="1,2,3,4">2</audioInputChannelID>
  <audioCompressionType opt="G.726,G.711ulaw" def="G.726">G.726</audioCompressionType>
  <audioBitRate opt="16,24,32,40" def="32" dynamic="true">24</audioBitRate>
  <audioSamplingRate opt="8" dynamic="true">8</audioSamplingRate>
  <audioResolution opt="3,4,5,6" dynamic="true"/>
</Audio>
<isSpprtDynamicCapWithCondition><!--optional, xs:boolean, whether to support dynamic capability with conditions--></isSpprtDynamicCapWithCondition>
<enableCABAC><!--optional, xs:boolean, whether it supports compressing stream to improve performance--></enableCABAC>
<subStreamRecStatus><!--optional, xs:boolean--></subStreamRecStatus>
<isSupportRefreshFrame><!--optional, xs:boolean, whether it supports refreshing frames when Smart264 is enabled--></isSupportRefreshFrame>
<isSupportBareDataOverlay><!--optional, xs:boolean--></isSupportBareDataOverlay>
<isSupportRTCPCfg><!--optional, xs:boolean--></isSupportRTCPCfg>
<customStreamEnable><!--optional, xs:boolean, whether the stream is custom stream: "true"-yes, this node is not returned-no--></customStreamEnable>
<isSupportPictureByUrl>
  <!--optional, xs: boolean, whether the device supports capture picture in URL format (related URI: /ISAPI/ContentMgmt/StreamingProxy/channels/<ID>/PictureByUrl/capabilities?format=json)-->
</isSupportPictureByUrl>
</StreamingChannel>
```

F.196 XML_Cap_SoftwareService

SoftwareService capability message in XML format

```
<SoftwareService version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ThirdStream><!--opt, third stream configuration-->
    <enabled opt="true,false"><!--req, xs: boolean, whether to enable third stream configuration--></enabled>
    <thirdStreamPrompt opt="prompt5">
      <!--req, xs: string, set audible prompt for third stream, prompt5-"When third stream is enabled, functions such as intrusion detection, line crossing detection, region entrance detection, region exiting detection, face detection, HDMI, H.264+ and H265+, are not supported. New settings will take effect after a reboot."-->
    </thirdStreamPrompt>
  </ThirdStream>
  <PanoramaDisplay><!--opt, display the image range on panorama view-->
    <enabled opt="true,false" default="false">
      <!--req, xs: boolean, whether to enable image range display on panorama view-->
    </enabled>
  </PanoramaDisplay>
  <MotionDetect>
    <enabled opt="true,false"><!--whether to enable motion detetion--></enabled>
```

```

<MotionDetectPrompt opt="prompt1">
    <!--req, xs: string, motion detection audible prompt, prompt1-->"When motion detection is enabled, functions such
as intrusion detection, line crossing detection, region entrance detection, and region exiting detection, are not
supported. New settings will take effect after a reboot."-->
</MotionDetectPrompt>
</MotionDetect>
</SoftwareService>

```

F.197 XML_Cap_SupplementLight

SupplementLight capability message in XML format

```

<SupplementLight><!--opt-->
    <mode opt="schedule,off,on,auto"><!--opt, xs: string, adjustment mode of supplement light: "schedule,off,on,auto"-->
    </mode>
    <Schedule>
        <TimeRange><!--req-->
            <beginTime><!--req, xs: time, ISO8601 time--></beginTime>
            <endTime><!-- req, xs: time, ISO8601 time--></endTime>
        </TimeRange>
    </Schedule>
    <brightnessLimit min="0" max="100">
        <!--dep, xs: integer, brightness of supplement light, which is between 0 and 100; the brightness can be adjust when
<mode> is set to "on"-->
    </brightnessLimit>
    <supplementLightMode opt="mixed,whiteLight,close">
        <!--opt, xs: string, illumination mode: "mixed"-hybrid, "whitelight"-white light, "close"-disabled-->
    </supplementLightMode>
    <irLightBrightness min="0" max="100">
        <!--dep, xs: integer, brightness of IR supplement light, which is between 0 and 100; this node is valid only when
<supplementLightMode> is set to "mixed" and <mixedLightBrightnessRegulatMode> is set to "manual"-->
    </irLightBrightness>
    <mixedLightBrightnessRegulatMode opt="manual,auto">
        <!--dep, xs: string, brightness adjustment mode of hybrid supplement light; this node is valid only when
<supplementLightMode> is set to "mixed"-->
    </mixedLightBrightnessRegulatMode>
    <mixedModeSupportType opt="irLight,whiteLight">
        <!--dep, xs: string, light type of hybrid illumination mode, "irLight"-IR light, "whiteLight"-white light-->
    </mixedModeSupportType>
    <highIrLightBrightness min="0" max="100">
        <!--dep, xs: integer, brightness of far IR light, which is between 0 and 100; this node cannot be configured together
with node <irLightBrightness>-->
    </highIrLightBrightness>
    <highWhiteLightBrightness min="0" max="100">
        <!--dep, xs: integer, brightness of far white light, which is between 0 and 100; this node cannot be configured
together with node <brightnessLimit>-->
    </highWhiteLightBrightness>
    <lowIrLightBrightness min="0" max="100">
        <!--dep, xs: integer, brightness of near IR light, which is between 0 and 100; this node cannot be configured together
with node <irLightBrightness>-->
    </lowIrLightBrightness>

```

```
</lowIrLightBrightness>
<lowWhiteLightBrightness min="0" max="100">
    <!--dep, xs: integer, brightness of near white light, which is between 0 and 100; this node cannot be configured together with node <brightnessLimit>-->
</lowWhiteLightBrightness>
<whiteLightBrightness min="0" max="100"><!--dep, xs: integer, white light brightness, which is between 0 and 100-->
</whiteLightBrightness>
<irLightbrightnessLimit min="0" max="100"><!--dep, xs:integer, IR light brightness limit, range: [0,100]; this node is valid when the value of mode "auto"--></irLightbrightnessLimit>
<whiteLightbrightnessLimit min="0" max="100"><!--dep, xs:integer, white light brightness limit, range: [0,100]; this node is valid when the value of mode "auto"--></whiteLightbrightnessLimit>
<unscheduledBrightness min="" max=""><!--optional, xs:integer, brightness in the unscheduled time, by default it is 0 (the supplement light is disabled) and the value is between 0 and 100--></unscheduledBrightness>
</SupplementLight>
```

F.198 XML_Cap_tempRange

tempRange capability message in XML format

```
<?xml version="1.0" encoding="utf-8"?>
<TempRange version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <mode opt="automatic,manual">
        <!--req, xs:string, temperature range mode: "automatic"-automatic mode, "manual"-manual mode-->
    </mode>
    <temperatureUpperLimit min="-20" max="550" def="150">
        <!--dep, xs:float, maximum temperature, corrects to one decimal, the minimum value is "-20", the maximum value is "550", unit: Celsius degree; its value should be larger than temperatureLowerLimit; it is valid when the value of mode is "manual"-->
    </temperatureUpperLimit>
    <temperatureLowerLimit min="-20" max="550" def="0">
        <!--dep, xs:float, minimum temperature, corrects to one decimal, the minimum value is "-20", the maximum value is "550", unit: Celsius degree; its value should be smaller than temperatureUpperLimit; it is valid when the value of mode is "manual"-->
    </temperatureLowerLimit>
</TempRange>
```

F.199 XML_Cap_Time

Time capability message in XML format

```
<Time version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <timeMode opt="NTP,manual,satellite,timecorrect,platform"><!--required, xs:string--></timeMode>
    <localTime min="0" max="256">
        <!--dependent, xs:datetime, this node is required when <timemode> is "manual" or "local"-->
    </localTime>
    <timeZone min="0" max="256">
        <!--dependent, xs:string, POSIX time zone string, this node is required when <timemode> is "manual", "local" or "NTP"-->
    </timeZone>
```

```
<satelliteInterval min="" max="">
  <!--dependent, xs:integer, minutes, time synchronization interval of locating by satellite, this node is valid only when
<timemode> is "satellite"-->
</satelliteInterval>
<timeType opt="local, UTC">
  <!--optional, xs: string, time type, it can be local, or UTC, or local and UTC. If this node is not returned, it indicates
that device does not support getting or setting time type-->
</timeType>
<platformType opt="EZVIZ">
  <!--dependent, xs: string, platform type: "EZVIZ"-Guarding Vision; it is valid only when the value of timeMode is
"platform"-->
</platformType>
</Time>
```

F.200 XML_Cap_Track

XML message about recording schedule configuration capability

```
<Track version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--xs:integer, the value of <id> and <Channel> is the same. For example, 201 refers to the main stream of
channel 1, 202 refers to the sub-stream of channel 2, and so on--></id>
  <Channel><!--req, xs:integer--></Channel>
  <Enable><!--req, xs:boolean--></Enable>
  <Description><!--req, xs:string, e.g., trackType=standard,sourceTag=AXIS210a,contentType=video,codecType=MPEG4-
SP,resolution=640x480,frameRate=20 fps,bitrate=6000 kbps--></Description>
  <TrackGUID><!--req, xs:string, GUID generated by the client, e.g., A01AAAAA-BBBB-CCCC-DDDD-033595353625--></
TrackGUID>
  <Size><!--opt, xs:integer--></Size>
  <Duration min="0" max="750"><!--opt, xs:string, video expiry date, e.g., P10DT15H indicates that the video will
expire after 10 days and 15 hours--></Duration>
  <DefaultRecordingMode
    opt="CMR,MOTION,ALARM,EDR,ALARMANDMOTION,AllEvent,POS,PECCANCY,SMART,temperatureIntervalMeasureme
nt">
    <!--req, xs:string, default record type: temperatureIntervalMeasurement (interval temperature measurement), CMR
(continuously record according to schedule), MOTION (record triggered by motion detection), ALARM (record
triggered by alarm), EDR (record triggered by alarm/motion detection), ALARMANDMOTION (record triggered by
alarm and motion detection), Command (record by command), SMART (smart record), AllEvent (all events)--></
DefaultRecordingMode>
  <LoopEnable><!--opt, xs:boolean, whether to support recurrently overwriting--></LoopEnable>
  <SrcDescriptor><!--video source description-->
  <SrcGUID><!--req, xs:string, GUID generated by the device, e.g., E800A543-9D53-4520-8BB8-9509062C692D--></
SrcGUID>
  <SrcChannel><!--req, xs:integer, source channel--></SrcChannel>
  <StreamHint><!--req, xs:string, e.g., "video, mp4, 640x480, 20 fps, 6000 kbps"--></StreamHint>
  <SrcDriver><!--req, xs:string, stream executable driver name, e.g., RTP/RTSP--></SrcDriver>
  <SrcType><!--opt, xs:string, source type, e.g., "mp4", "video"--></SrcType>
  <SrcUrl opt="rtsp://localhost/PSIA/Streaming/channels/101,rtsp://localhost/PSIA/Streaming/channels/102,rtsp://
localhost/PSIA/Streaming/channels/103,,rtsp://localhost/PSIA/Streaming/channels/107"><!--req, xs:string, e.g.,
rtsp://10.3.2.26/mpeg4/media.amp--></SrcUrl>
  <SrcUrlMethods><!--req, xs:string, methods supported by the source: "DESCRIBE,SETUP,PLAY,TEARDOWN"--></
```

```

SrcUrlMethods>
  <SrcLogin><!--req, xs:string, login source password--></SrcLogin>
</SrcDescriptor>
<TrackSchedule><!--recording schedule-->
  <ScheduleBlockList>
    <ScheduleBlock><!--list, recording schedule list-->
      <ScheduleBlockGUID><!--xs:string, schedule GUID, e.g., ABC12345-CDEF-4520-8BB8-7135789C8790--></
ScheduleBlockGUID>
      <ScheduleBlockType><!--req, xs:string--></ScheduleBlockType>
      <ScheduleAction><!--list-->
        <id><!--req, xs:integer--></id>
        <ScheduleActionStartTime><!--start time of the schedule-->
          <DayOfWeek><!--req, xs:string, "Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday"--></
DayOfWeek><!--inclusive-->
          <TimeOfDay><!--xs:time, 00:00:00--></TimeOfDay>
        </ScheduleActionStartTime>
        <ScheduleActionEndTime><!--end time of the schedule-->
          <DayOfWeek><!--req, xs:string, "Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday"--></
DayOfWeek><!--exclusive-->
          <TimeOfDay><!--xs:time, 08:00:00--></TimeOfDay>
        </ScheduleActionEndTime>
        <ScheduleDSTEnable><!--req, xs:boolean, whether to enable DST (daylight saving time)--></ScheduleDSTEnable>
        <Description><!--req, xs:string, PreMorning (midnight to 8 a.m., local time)--></Description>
        <Actions><!--alarm or motion detection that triggers recording-->
          <Record><!--opt, xs:boolean--></Record>
          <Log><!--opt, xs:boolean--></Log>
          <SaveImg><!--opt, xs:boolean--></SaveImg>
          <ActionRecordingMode><!--req, xs:string, record type: temperatureIntervalMeasurement (interval temperature
measurement), CMR (continuously record according to schedule), MOTION (record triggered by motion detection),
ALARM (record triggered by alarm), EDR (record triggered by alarm/motion detection), ALARMANDMOTION (record
triggered by alarm and motion detection), Command (record by command), SMART (smart record), AllEvent (all
events)--></ActionRecordingMode>
          <PreRecordTimeSeconds><!--opt, xs:integer, pre-record time, it is between 0 and 10 and the default value is 5,
unit: second--></PreRecordTimeSeconds>
          <PostRecordTimeSeconds><!--opt, xs:integer, post-record time, it is between 0 and 20 and the default value is
5, unit: second--></PostRecordTimeSeconds>
        </Actions>
      </ScheduleAction>
    </ScheduleBlock>
  </ScheduleBlockList>
</TrackSchedule>
<CustomExtensionList>
  <CustomExtension>
    <CustomExtensionName><!--opt, xs:string, example: www.isapi.com/RaCM/trackExt/ver10--></
CustomExtensionName>
    <enableSchedule><!--opt, xs:boolean, whether to enable recording schedule configuration--></enableSchedule>
    <SaveAudio><!--opt, xs:boolean, whether to enable recording audio--></SaveAudio>
    <PreRecordTimeSeconds><!--opt, xs:integer, pre-record time, unit: second--></PreRecordTimeSeconds>
    <PostRecordTimeSeconds><!--opt, xs:integer, post-record time, unit: second--></PostRecordTimeSeconds>
    <HolidaySchedule>
      <ScheduleBlock>
        <ScheduleBlockGUID><!--req, xs:string, example: 00000000-0000-0000-0000-000000000000--></

```

```
ScheduleBlockGUID>
    <ScheduleBlockType><!--opt, xs:string, e.g., www.isapi.com/racm/schedule/ver10--></ScheduleBlockType>
    </ScheduleBlock>
    </HolidaySchedule>
    </CustomExtension>
</CustomExtensionList>
<IntelligentRecord><!--opt, xs:boolean, whether to enable VCA recording function: 0-no, 1-yes--></IntelligentRecord>
<delayTime opt="0,3,4,5,10,30,60,120,300"><!--opt, xs:integer, capture delay time, unit: second--></delayTime>
<durationEnabled opt="true,false"><!--opt, xs:boolean, whether to enable video expiry time. If this function is not supported, this node will not be returned. If this function is supported, the video expiry date will be set by the node <Duration>--></durationEnabled>
</Track>
```

F.201 XML_Cap_Tracking

Tracking capability message in XML format

```
<Tracking version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
    <enabled>
        <!--req, xs:boolean-->
    </enabled>
    <mode opt="auto,manual">
        <!--dep, xs:string, "auto,manual"-->
    </mode>
    <trackingTime min="0" max="60" def="5">
        <!--dep, xs:integer, tracking time-->
    </trackingTime>
    <normalizedScreenSize>
        <!--req, ro-->
        <normalizedScreenWidth>
            <!--req, ro, xs:integer-->
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!--req, ro, xs:integer-->
        </normalizedScreenHeight>
    </normalizedScreenSize>
    <ManualRegionCoordinatesList min="4" size="4">
        <!--dep, tracking area-->
        <RegionCoordinates>
            <!--opt-->
            <positionX>
                <!--req, xs:integer, X-coordinate-->
            </positionX>
            <positionY>
                <!--req, xs:integer, Y-coordinate-->
            </positionY>
        </RegionCoordinates>
    </ManualRegionCoordinatesList>
    <isNotSupportTrackingDisplay>
        <!--req, xs:boolean, whether to support displaying smart linkage on the interface: "true"-yes, "false"-no-->
```

```
</isNotSupportTrackingDisplay>
<Schedule>
  <!--opt ,dep, schedule, this node is valid only when enabled is "true". By default the schedule is from 00:00 to 24:00
if no schedule is configured-->
  <TimeRange >
    <beginTime>
      <!--UTC time, e.g.: "2018-03-13T19:42:27+08:00"-->
    </beginTime>
    <endTime>
      <!--UTC time, e.g.: "2018-03-13T19:42:27+08:00"-->
    </endTime>
  </TimeRange >
</Schedule >
</Tracking>
```

F.202 XML_Cap_User

XML message about user configuration capability

```
<User xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0" >
<id>
  <!--required, xs:integer, user ID, value range: [1,16]-->
</id>
<userName>
  <!--required, xs:string -->
</userName>
<password>
  <!--required, xs:string, read-only-->
</password>
<bondIpAddressList>
<bondIpAddress>
  <id>
    <!--required, xs:integer -->
  </id>
  <ipAddress>
    <!--dependent, xs:string -->
  </ipAddress>
  <ipv6Address>
    <!--dependent, xs:string -->
  </ipv6Address>
</bondIpAddress>
</bondIpAddressList>
<bondMacAddressList>
<bondMacAddress>
  <id>
    <!--required, xs:integer -->
  </id>
  <macAddress>
    <!--optional, xs:string -->
  </macAddress>
```

```
</bondMacAddress>
</bondMacAddressList>
<userLevel>
  <!--optional, xs:string, user type: "Administrator", "Operator", "Viewer"-->
</userLevel>
<attribute>
  <!--optional-->
  <inherent>
    <!--xs:boolean -->
  </inherent>
</attribute>
<belongUserId>
  <!--optional, xs:integer, upper-level user ID, "0"-->there is no upper-level user-->
</belongUserId>
<belongOrgNo>
  <!--optional, xs:integer, organization No.-->
</belongOrgNo>
<cardList>
  <!--optional, object, list of cards that belong to the user-->
<card>
  <!--optional, object, card-->
  <id>
    <!--optional, int, card ID, range:[1,16]-->1
  </id>
  <name>
    <!--optional, string, card name, the maximum length is 32 bytes-->test
  </name>
</card>
</cardList>
<remoteCtrlList>
  <!--optional, object, list of keyfobs that belong to the user-->
<remoteCtrl>
  <!--optional, object, keyfob-->
  <id>
    <!--optional, int, keyfob ID, range:[1,16]-->1
  </id>
  <name>
    <!--optional, string, keyfob name, the maximum length is 32 bytes-->test
  </name>
</remoteCtrl>
</remoteCtrlList>
<userNo min="501" max="null">
  <!--optional, int, displayed No. corresponding to the user ID-->1
</userNo>
<adminType opt="Cloud,LAN">
  <!--optional, string, admin type, "Cloud" (user account created in the cloud), "LAN" (user account created on LAN)-->
</adminType>
<installerType opt="Cloud,LAN">
  <!--optional, string, installer type, "Cloud" (user account created in the cloud), "LAN" (user account created on LAN)-->Cloud
</installerType>
```

```
</installerType>
</User>
```

F.203 XML_Cap_UserPermissionList

XML message about the configuration capability of all users' permission

```
<UserPermissionList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<UserPermission><!--required-->
  <id min="" max="" /><!--required, xs:string-->
  <userID min="" max="" /><!--required, xs:string, user ID-->
  <userType optional="admin,operator,viewer"/><!--required, xs:string, user type: "admin, operator, viewer"-->
  <localPermission><!--optional-->
    <backup optional="true,false"/><!--optional, xs:boolean, whether it supports local backup. If this node is returned, it indicates that the device supports this function. You can configure whether to assign this permission to users-->
    <record optional="true,false"/><!--optional, xs:boolean, whether it supports local manual operation-->
    <playBack optional="true,false"/><!--optional, xs:boolean, local playback-->
    <preview optional="true,false"/><!--optional, xs:boolean, local live view-->
    <videoChannelPermissionList><!--optional-->
      <videoChannelPermission><!--optional-->
        <id min="" max="" /><!--required, xs:string-->
        <playBack optional="true,false"/><!--optional, xs:boolean-->
        <preview optional="true,false"/><!--optional, xs:boolean-->
        <record optional="true,false"/><!--optional, xs:boolean-->
        <backup optional="true,false"/><!--optional, xs:boolean-->
      </videoChannelPermission>
    </videoChannelPermissionList>
  </localPermission>
</UserPermission>
<ptzControl optional="true,false"/><!--optional, xs:boolean, local PTZ control-->
<ptzChannelPermissionList> <!--optional-->
  <ptzChannelPermission><!--required-->
    <id min="" max="" /><!--required, xs:string-->
    <ptzControl optional="true,false"/><!--optional, xs:boolean, local PTZ control-->
  </ptzChannelPermission>
</ptzChannelPermissionList>
<logOrStateCheck optional="true,false"/><!--optional, xs:boolean, whether it supports view logs locally-->
<parameterConfig optional="true,false"/><!--optional, xs:boolean, whether it supports setting parameters locally-->
<restartOrShutdown optional="true,false"/><!--optional, xs:boolean, whether it supports shutting down and rebooting the device locally-->
<upgrade optional="true,false"/><!--optional, xs:boolean, whether it supports upgrading and formatting locally-->
<manageChannel optional="true,false"/><!--optional, xs:boolean, whether it supports managing channels locally-->
<AIModelManagement>
  <!--read-only, optional, xs:boolean, whether to enable permission for operation on the AI model packet-->true
</AIModelManagement>
<AITaskManagement>
  <!--read-only, optional, xs:boolean, whether to enable permission for management and operation on AI tasks-->true
</AITaskManagement>
<subSysOrZoneArm>
  <!--read-only, optional, xs:boolean, arm permission of partition/alarm zone-->true
</subSysOrZoneArm>
```

```
<subSysOrZoneDisarm>
  <!--read-only, optional, xs:boolean, disarm permission of partition/alarm zone-->true
</subSysOrZoneDisarm>
<operateOutput>
  <!--read-only, optional, xs:boolean, permission for operation on relay-->true
</operateOutput>
<supportLinkageSubSystemList size="64">
  <!--read-only, optional, list of the partitions that support linkage-->
  <subSystem>
    <!--read-only, optional, xs:integer, the partition that supports linkage, value range:[1,64]-->1
  </subSystem>
</supportLinkageSubSystemList>
<subSystemList>
  <!--read-only, optional, xs:object, list of partitions-->
  <subSystem>
    <!--read-only, optional, xs:integer, partition, range:[1,64]-->1
  </subSystem>
</subSystemList>
</localPermission>
<remotePermission><!--optional-->
<record optional="true,false"/><!--optional, xs:boolean, whether it supports recording remotely and manually-->
<playBack optional="true,false"/><!--optional, xs:boolean, whether it supports remote playback-->
<preview optional="true,false"/><!--optional, xs:boolean, whether it supports remote live view-->
<videoChannelPermissionList><!--optional -->
  <videoChannelPermission><!--optional -->
    <id min="" max="/"><!--required, xs:string !-->
    <preview optional="true,false"/><!--optional, xs:boolean, whether it supports remote live view-->
    <playBack optional="true,false"/><!--optional, xs:boolean, whether it supports remote playback-->
    <record optional="true,false"/><!--optional, xs:boolean, whether it supports recording remotely and manually-->
  </videoChannelPermission>
</videoChannelPermissionList>
<ptzControl optional="true,false"/><!--optional, xs:boolean, whether it supports remote PTZ control-->
<ptzChannelPermissionList><!--optional -->
  <ptzChannelPermission><!--optional -->
    <id><!--required, this ID must correspond to the PTZ ID in /ISAPI/PTZCtrl/channels/<ID>--></id>
    <ptzControl optional="true,false"/><!--optional, xs:boolean, whether it supports controlling the PTZ remotely-->
  </ptzChannelPermission>
</ptzChannelPermissionList>
<logOrStateCheck optional="true,false"/><!--optional, xs:boolean, whether it supports viewing logs remotely-->
<parameterConfig optional="true,false"/><!--optional, xs:boolean, whether it supports setting parameters
remotely-->
<restartOrShutdown optional="true,false"/><!--optional, xs:boolean, whether it supports shutting down and
rebooting remotely-->
<upgrade optional="true,false"/><!--optional, xs:boolean, whether it supports upgrading and formatting remotely-->
<voiceTalk optional="true,false"/><!--optional, xs:boolean, whether it supports remote two-way audio-->
<transParentChannel optional="true,false"/><!--optional, xs:boolean, whether it supports controlling the serial
port remotely-->
<controlLocalOut optional="true,false"/><!--optional, xs:boolean, whether it supports controlling the local output
remotely-->
<manageChannel optional="true,false"/><!--optional, xs:boolean, whether it supports managing channels
remotely-->
```

```
<alarmOutOrUpload optional="true,false"/><!--optional, xs:boolean, whether it supports request for uploading alarms and alarm output remotely-->
<IRaid>
  <rapidConfiguration optional="true,false"/><!--optional, xs:boolean, whether it supports configuring IRAID permissions rapidly-->
    <reset optional="true,false"/><!--optional, xs:boolean, whether it supports configuring IRAID to reset configuration permissions-->
  </IRaid>
  <subSysOrZoneArm>
    <!--read-only, optional, xs:boolean, arm permission of partition/alarm zone-->true
  </subSysOrZoneArm>
  <subSysOrZoneDisarm>
    <!--read-only, optional, xs:boolean, disarm permission of partition/alarm zone-->true
  </subSysOrZoneDisarm>
  <subSysOrZoneClearArm>
    <!--read-only, optional, xs:boolean, clear alarm permission of partition/alarm zone-->true
  </subSysOrZoneClearArm>
  <zoneBypass>
    <!--read-only, optional, xs:boolean, bypass permission in alarm zone-->true
  </zoneBypass>
  <zoneBypassRecover>
    <!--read-only, optional, xs:boolean, bypass recovery permission in alarm zone-->true
  </zoneBypassRecover>
  <operateOutput>
    <!--read-only, optional, xs:boolean, permission for operation on relay-->true
  </operateOutput>
  <supportLinkageSubSystemList>
    <!--read-only, optional, list of the partitions that support linkage-->
    <subSystem>
      <!--read-only, optional, xs:integer, the partition that supports linkage, value range:[1,64]-->1
    </subSystem>
  </supportLinkageSubSystemList>
  <subSystemList>
    <!--read-only, optional, xs:object, list of partitions-->
    <subSystem>
      <!--read-only, optional, xs:integer, partition, range:[1,64]-->1
    </subSystem>
  </subSystemList>
  <factoryReset>
    <!--read-only, optional, xs:boolean, recover default parameters-->true
  </factoryReset>
  <arm>
    <!--read-only, optional, xs:boolean, arm-->true
  </arm>
  <disarm>
    <!--read-only, optional, xs:boolean, disarm-->true
  </disarm>
  <accessControl>
    <!--read-only, optional, xs:boolean, access control-->true
  </accessControl>
  <restoreTamper opt="true,false">
    <!--optional, boolean, whether to restore tampering, this node is only valid for installer-->true
  </restoreTamper>

```

```
</restoreTamper>
<restoreConfirmedAlarm opt="true,false">
    <!--optional, boolean, whether to restore alarm acknowledgment, this node is for linked zones-->true
</restoreConfirmedAlarm>
</remotePermission>
<passwordValidity min="" max="" /><!--required, xs:string, valid duration of the password, value range: [0, 365], unit: day. Only the admin user can edit this parameter-->
</UserPermission>
</UserPermissionList>
```

F.204 XML_Cap_VCAResource

Capability message of intelligent resources switch in XML format

```
<VCAResource version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <type
        opt="basicBehavior,fullBehavior,facesnapBehavior,facesnap,TFS,smartVehicleDetection,smartHVTDetection,smart,judicial,smart264AndRoadDetection,smart264AndFaceDetection,smart264AndHeatMap,smartVehicleIllegalParkingDetection,smartIntelligentMonitor,smartTrafficDataCollection, roadDetection,humanRecognition, perimeterCapture, vehicleDetection,HVTDetection,mixedTargetDetection,trackingCaptureMode,nonTrackingCaptureMode,close,faceHumanModelingContrast,cityManagement,teacherBehavior, 12MPLiveView,personQueueDetection,verticalPeopleCounting,safetyHelmet,faceCounting,AIOpenPlatform">
        <!--required, xs: string, intelligent resource types, "trackingCaptureMode"-panoramic capture mode, "nonTrackingCaptureMode"-single target capture mode, "faceHumanModelingContrast"-resources of face/human body+face modeling+face picture comparison, "verticalPeopleCounting"-vertical people counting, "safetyHelmet"-hard hat detection, "faceCounting"-face counting, "personArming"-person arming, "AIOpenPlatform"-AI Open Platform-->
    </type>
    <PromptList><!--optional, prompt for switching intelligent resources-->
        <Prompt><!--list-->
            <resourceType opt="roadDetection">
                <!--optional, xs: string, intelligent resource to prompt, currently, only "roadDetection" is supported-->
            </resourceType>
            <resourcePrompt opt="smartCode">
                <!--optional, xs: string, prompt contents-->
            </resourcePrompt>
        </Prompt>
    </PromptList>
</VCAResource>
```

F.205 XML_Cap_WirelessServer

WirelessServer capability message in XML format

```
<WirelessServer version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <wifiApEnabled opt="true,false"><!--opt,xs: boolean,--></wifiApEnabled>
    <broadcastEnabled opt="true,false"><!--opt,xs: boolean--></broadcastEnabled>
    <wlanShareEnabled opt="true,false"><!--opt,xs: boolean--></wlanShareEnabled>
    <ssid min="" max="" /><!--opt, xs: string--></ssid>
```

```

<WirelessSecurity><!--req-->
<securityMode opt="disable,WEP,WPA-personal,WPA2-personal,WPA-RADIUS,WPA-enterprise,WPA2-enterprise">
  <!--opt, xs: string-->
</securityMode>
<WEP><!--dep, depends on <securityMode>-->
<authenticationType opt="open,sharedkey,auto">
  <!--req, xs: string-->
</authenticationType>
<defaultTransmitKeyIndex min="" max="">
  <!--req, xs: integer-->
</defaultTransmitKeyIndex>
<wepKeyLength opt="64,128"><!--opt, xs: integer--></wepKeyLength>
<EncryptionKeyList size="">
  <encryptionKey>
    <!--req, xs: hexBinary, WEP encryption key in hexadecimal format-->
  </encryptionKey>
</EncryptionKeyList>
</WEP>
<WPA><!--dep, depends on <securityMode>-->
<algorithmType opt="TKIP,AES,TKIP/AES">
  <!--req, xs: string, "TKIP,AES,TKIP/AES"-->
</algorithmType>
<sharedKey><!--opt, xs: string, shared key used in WPA--></sharedKey>
<wpaKeyLength min="" max=""><!-- opt, xs: integer, the key length is between 8 and 63--></wpaKeyLength>
<defaultPassword><!--opt, xs: boolean--></defaultPassword>
</WPA>
</WirelessSecurity>
<DHCPEnabled opt="true,false"><!--opt, xs: boolean--></DHCPEnabled>
<ipVersion opt="v4,v6"><!--opt, xs:string--></ipVersion>
<HostIpAddress><!--opt-->
  <ipAddress><!--dep, xs:string--></ipAddress>
  <ipv6Address><!--dep, xs:string--></ipv6Address>
</HostIpAddress>
<IPMask><!--opt-->
  <subnetMask><!--dep, xs:string, subnet mask for IPv4 address--></subnetMask>
  <bitMask><!--dep, xs:integer, bitmask IPv6 address--></bitMask>
</IPMask>
<AddressPool><!--opt-->
  <startIPV4Address><!--dep, xs:string--></startIPV4Address>
  <endIPV4Address><!--dep, xs:string--></endIPV4Address>
  <startIPV6Address><!-- dep, xs:string--></startIPV6Address>
  <endIPV6Address><!--dep, xs:string--></endIPV6Address>
</AddressPool>
<DNSAddressList size="2"><!--opt-->
  <DNSAddress><!--opt-->
    <id><!--opt,xs:string,start from 1--></id>
    <ipAddress><!--dep, xs: string--></ipAddress>
    <ipv6Address><!--dep, xs: string--></ipv6Address>
  </DNSAddress>
</DNSAddressList>
<GatewayAddress>
  <ipAddress><!--dep, xs:string--></ipAddress>

```

```

<ipv6Address><!--dep, xs:string--></ipv6Address>
</GatewayAddress>
<wifiApModeType opt="true,false,auto">
    <!--opt, xs: string, current wireless access point (AP) mode, if this node is returned, it indicates that the device
supports auto mode-->
</wifiApModeType>
</WirelessServer>

```

Remarks

For Client supports auto AP, the node <>wifiApModeType<> is valid; for Client does not supports auto AP, the node <>wifiApEnabled<> is valid; the values of these two nodes will effect each other, and when <>wifiApModeType<> equals to "auto", the value of <>wifiApEnabled<> is "true".

F.206 XML_ChannelEventCap

XML message about event capability of a single channel

```

<ChannelEventCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<eventType opt="">
    <!--required, xs:string, event types supported by the channel; see remarks for details-->
</eventType>
<shieldEventType opt="behavior,faceSnap,humanRecognition,faceCapture,targetCapture">
    <!--required, xs:string, event types that support area shield-->
</shieldEventType>
<channelID><!--optional, xs:integer, current device channel No.--></channelID>
<id><!--required, xs:integer--></id>
<passThroughEventType
opt="faceSnapModeling,cityManagement,strudentsStoodUp,dredgerDetection,reverseEntrance,teacherBehaviorDetec
t,personQueueDetection">
    <!--optional, xs:string, event types can be transmitted-->
</passThroughEventType>
</ChannelEventCap>

```

Remarks

The values of node <>eventType<> are shown in the table below.

eventType	Description
VMD (motionDetection)	Motion detection
cityManagement	Intelligent city management
framesPeopleCounting	Regional people counting
Shelteralarm (tamperDetection)	Video tampering alarm
videoLoss	Video loss
ROI	Region of interest

eventType	Description
facedetection (faceDetection)	Face detection
intelliTrace	Auto-track
fielddetection (fieldDetection)	Intrusion detection
defocus (defocusDetection)	Defocus detection
audioexception (audioDetection)	Audio exception detection
scenechangedetection (sceneChangeDetection)	Sudden scene change detection
linedetection (lineDetection)	Line-crossing detection
regionEntrance	Region entrance detection
regionExiting	Region exiting detection
loitering	Loitering detection
group	People gathering detection
rapidMove	Fast moving detection
parking	Parking detection
unattendedBaggage	Unattended baggage
attendedBaggage	Object removal detection
peopleDetection	Human detection
storageDetection	HDD health detection
behavior	Behavior analysis
faceCapture(faceSnap)	Face capture
ANPR(vehicleDetection)	Vehicle detection
fireDetection	Fire detection
shipsDetection	Ship detection
PIR	PIR alarm
targetCapture (humanRecognition)	Target capture
alarmResult (faceContrast)	Face comparison
framesPeopleCounting	People counting in single frame
hide	Tampering detection

eventType	Description
audioLoss	Audio loss
cameraAngleAnomaly	Image exception
violentMotion	Violent motion
trial	Tailing
humanEnter	Human entrance
operateOverTime	Operation timed out
stickUp	Sticking scrip
installScanner	Installing scanner
faceDetect	ATM face detection
temperature (temperatureDetection)	Temperature difference detection
PeopleCounting	People counting
personQueueDetection	People queuing-up detection
heatmap	Heat map
mixedTargetDetection	Multi-target-type detection
MTD_face	Supported face attributes of multi-target-type detection (MTD)
MTD_human	Supported human attributes of multi-target-type detection (MTD)
MTD_vehicle	Supported motor vehicle attributes of multi-target-type detection (MTD)
MTD_nonMotor	Supported non-motor vehicle attributes of multi-target-type detection (MTD)
faceSnapModeling	Face capture modeling
HVTVehicleDetection	Mixed-traffic detection
PictureCaptureComparision	N:1 face comparison
IO	Sensor alarm
smokeDetection	Smoke detection
smokeAndFireDetection	Smoke and fire detection
diskfull	HDD is full

eventType	Description
diskerror	HDD error
nicbroken	Network disconnected
ipconflict	IP address conflicted
illaccess	Illegal access
personDensityDetection	People density detection
vehicleControl	Upload alarm of vehicle in allowlist and blocklist
vehicleRcogResult	Vehicle secondary recognition
cardMatch	Authentication
overSpeed	Overspeed alarm
highTempAlarm	High temperature alarm
abnormalAcceleration	Abnormal accelerated speed alarm
failDown	People falling down
leavePosition	Absence detection
peopleNumChange	The number of people changed
retention	Overstay detection
running	Running
thermometry	Temperature measurement
heatmapPDC	Heat map people counting
heatmapDuration	Heat map people staying duration
intersectionAnalysis	Intersection analysis
AID_abandonedObject	Thrown object in traffic events
AID_pedestrian	Pedestrian detection in traffic events
AID_congestion	Congestion detection in traffic events
AID_roadBlock	Roadblock detection in traffic events
AID_construction	Construction detection in traffic events
AID_trafficAccident	Traffic accident detection in traffic events
AID_fogDetection	Fog detection in traffic events
TFS_illegalParking	Illegal parking of enforcement events

eventType	Description
TFS_wrongDirection	Wrong-way driving of enforcement events
TFS_crossLane	Driving on lane line of enforcement events
TFS_laneChange	Lane change of enforcement events
TFS_vehicleExist	Vehicle existing of enforcement events
TFS_turnRound	Turning round of enforcement events
TFS_parallelParking	Parallel parking of enforcement events
TPS	Traffic data collection configuration
luma	Brightness exception
chroma	Color cast detection
snow	Snow noise
streak	Stripe noise
freeze	Image freeze
sigLose	Signal loss
clarity	Clarity exception
jitter	Image flutter
block	Video tampering
flowers	Blurred screen detection
noise	Image noise
ghost	Abnormal light spot detection
purple	Image purple edge detection
ICR	ICR exception detection
protectiveFilm	Protective film unremoved
certificateRevocation	Certificate is expired
dataPreAlarm	Traffic pre-alarm
vibrationDetection	Vibration detection
radarFieldDetection	Radar intrusion detection
radarLineDetection	Radar line crossing detection
personArmingTrack	Person arming

eventType	Description
manualPersonArmingTrack	Manual person arming
fireEscapeDetection	Fire engine access detection
takingElevatorDetection	Elevator detection
temperatureIntervalMeasurement	Interval temperature measurement
containerDetection	Container detection  Note If this node is not returned, you can refer to the value returned by URI (/ISAPI/Traffic/ContentMgmt/InputProxy/channels/<ID>/ocrScene/capabilities) to find whether the device supports container detection.
thermalVehicleDetection	Thermal vehicle detection
wasteGasDetection	Waste gas detection



The value in the bracket is the old version, and considering the compatibility, both new and old value will be returned.

F.207 XML_ChannelEventCapList

Message about event capability of all channels in XML format

```
<ChannelEventCapList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ChannelEventCap>
    <eventType opt="">
      <!--required, xs:string, event types supported by the channel; see remarks for details-->
    </eventType>
    <shieldEventType opt="behavior,faceSnap,humanRecognition,faceCapture,targetCapture">
      <!--required, xs:string, event types that support area shield-->
    </shieldEventType>
    <channelID><!--optional, xs:integer, current device channel No.--></channelID>
    <id><!--required, xs:integer--></id>
    <passThroughEventType
      opt="faceSnapModeling,cityManagement,studentsStoodUp,dredgerDetection,reverseEntrance,teacherBehaviorDetec
      t,personQueueDetection">
      <!--optional, xs:string, event types can be transmitted-->
    </passThroughEventType>
  </ChannelEventCap>
</ChannelEventCapList>
```

F.208 XML_ChannelInfo

ChannelInfo message in XML format.

```
<ChannelInfo version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id><!--required, xs:integer, channel number--></id>
<EagleEye>
    <!--optional, the channel supports the functions of PanoVu series camera-->
    <StreamFusion>
        <!--optional, "panoramic + PTZ camera streaming channel, it is not returned when not support-->
        <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
    </StreamFusion>
    <PanoramicMetaData>
        <!--optional, panoramic camera streaming channel, it is not returned when not support-->
        <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
    </PanoramicMetaData>
</EagleEye>
<FishEye>
    <!--optional, the channel supports the functions of fisheye camera-->
    <FishEyeStream>
        <!--optional, fisheye camera stream, it is used for software decoding of play library, it is not returned when not support-->
        <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
    </FishEyeStream>
</FishEye>
<Thermal>
    <!--optional, the channel supports the functions of thermal products-->
    <TemperatureMetaData>
        <!--optional, temperature measurement metadata, it is not returned when not support-->
        <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
    </TemperatureMetaData>
    <ThermometryDetection>
        <!--optional, temperature measurement detection, it is not returned when not support-->
        <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
    </ThermometryDetection>
    <ThermometryDiffDetection>
        <!--optional, temperature difference detection, it is not returned when not support-->
        <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
    </ThermometryDiffDetection>
    <FireDetection>
        <!--optional, fire detection, it is not returned when not support-->
        <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
    </FireDetection>
    <SmokeDetection>
        <!--optional, smoke detection, it is not returned when not support-->
        <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
    </SmokeDetection>
    <ShipsDetection>
        <!--optional, ship detection, it is not returned when not support-->
        <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
    </ShipsDetection>
```

```
<IntelligentRuleDisplay>
    <!--optional, intelligent rule display parameters of thermal application, it is not returned when not support-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</IntelligentRuleDisplay>
<IntelligentBehaviorRule>
    <!--optional, intelligent behavior rule configuration, it is not returned when not support-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</IntelligentBehaviorRule>
<DPC>
    <!--optional, defective pixel correction, it is not returned when not support-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</DPC>
<Metadata>
    <!--optional, metadata extraction, it is not returned when not support-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</Metadata>
<ThermometryShieldMask>
    <!--optional, temperature measurement shielded area, it is not returned when not support-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</ThermometryShieldMask>
<FaceThermometry>
    <!--optional, face thermography, it is not returned when not support-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</FaceThermometry>
<BlackBody>
    <!--optional, black body, it is not returned when not support-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</BlackBody>
<BodyTemperatureCompensation>
    <!--optional, body temperature compensation, it is not returned when not support-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</BodyTemperatureCompensation>
<LensCorrection>
    <!--optional, lens correction, when this node is not returned, it indicates not support-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</LensCorrection>
<BurningPrevention>
    <!--optional, burning prevention, when this node is not returned, it indicates not support-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</BurningPrevention>
<FaceSnapThermometry>
    <!--optional, upload the captured face picture with temperature information, it is not returned when this function is not supported-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</FaceSnapThermometry>
<TemperatureCorrect><!--optional, temperature correction; this node is not returned when this function is not supported-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</TemperatureCorrect>
<TemperatureIntervalMeasurement><!--optional, interval temperature measurement; this node is not returned when the function is not supported-->
    <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
```

```
</TemperatureIntervalMeasurement>
</Thermal>
<isSupportChangedUpload/><!--optional, xs: boolean, whether supports uploading status changes-->
<PanoramaCamera><!--optional, PTZ camera channel-->
<PTZCtrl>
  <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</PTZCtrl>
</PanoramaCamera>
<GlobalCamera><!--optional, Panoramic camera channel-->
  <InsertChanNo><!--optional, xs:integer, access channel number--></InsertChanNo>
</GlobalCamera>
</ChannelInfo>
```

F.209 XML_ChannelInfoList

ChannelInfoList message in XML format

```
<ChannelInfoList>version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ChannelInfo/><!--see details in the message of XML_ChannelInfo-->
</ChannelInfoList>
```

See Also

[XML_ChannelInfo](#)

F.210 XML_CheckInfo

CheckInfo message in XML format

```
<CheckInfo version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <randomString><!--req, xs: string, random string--></randomString>
  <verificationCode>
    <!--opt, xs: string, verification code to be sent, which will be encrypted by sha256, and then convert the
    hexadecimal code to a string, finally it will be encoded by Base64. The format is first 32-byte of randomString + actual
    verification code-->
  </verificationCode>
</CheckInfo>
```

F.211 XML_CMSearchProfile

CMSearchProfile message in XML format

```
<CMSearchProfile version="1.0" xmlns="ISAPIllianxce.org:resourcedescription">
  <searchProfile><!--req, xs: string, video search conditions--></searchProfile>
  <textSearch>
    <!--req, xs: string, text type to be searched, such as recording information text and metadata text-->
  </textSearch>
```

```
<maxSearchTimespans>
  <!--req, xs: integer, maximum time period of results returned in each search-->
</maxSearchTimespans>
<maxSearchTracks><!--req, xs: integer, maximum tracks can be searched--></maxSearchTracks>
<maxSearchSources>
  <!--req, xs: integer, maximum recording channels can be searched-->
</maxSearchSources>
<maxSearchMetadatas>
  <!--req, xs: integer, maximum number of metadata can be searched-->
</maxSearchMetadatas>
<maxSearchMatchResults>
  <!--req, xs: integer, maximum search results can be returned-->
</maxSearchMatchResults>
<maxSearchTimeout><!--req, xs: integer, maximum search timeout--></maxSearchTimeout>
<maxConcurrentSearches>
  <!--req, xs: integer, maximum concurrent search operations-->
</maxConcurrentSearches>
</CMSearchProfile>
```

F.212 XML_Color

XML message about color adjustment parameters

```
<Color version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <brightnessLevel><!--opt, xs:integer, brightness--></brightnessLevel>
  <contrastLevel><!--opt, xs:integer, contrast--></contrastLevel>
  <saturationLevel><!--opt, xs:integer, saturation--></saturationLevel>
  <hueLevel><!--opt, xs:integer, hue--></hueLevel>
  <grayScale>
    <grayScaleMode><!--opt, xs:string, gray scale mode: "indoor,outdoor"--></grayScaleMode>
  </grayScale>
  <nightMode>
    <!--opt, xs:boolean, enable night mode, when its value is "true", the saturation can be adjusted, otherwise, the
    saturation cannot be adjusted-->
  </nightMode>
</Color>
```

F.213 XML_DefaultParam

DefaultParam message in XML format

```
<DefaultParam version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <DialNum><!--opt, xs:string--></DialNum>
  <Username><!--opt, xs:string--></Username>
  <Password><!--opt, xs:string--></Password>
  <APNname><!--opt, xs:string--></APNname>
  <VerifyProto><!--req, xs:string, "auto, CHAP, PAP"--></VerifyProto>
</DefaultParam>
```

F.214 XML_Desc_AccessProtocolAbility

Input description message for getting protocol access capability.

```
<AccessProtocolAbility version="2.0">
    <channelNO><!--req, channel No.--></channelNO>
</AccessProtocolAbility>
```

F.215 XML_Desc_AudioVideoCompressInfo

Input description message for getting video and audio encoding capability.

```
<AudioVideoCompressInfo>
    <AudioChannelNumber><!--req, audio channel No.--></AudioChannelNumber>
    <VoiceTalkChannelNumber><!--req, two-way audio channel No.--></VoiceTalkChannelNumber>
    <VideoChannelNumber><!--req, video channel No.--></VideoChannelNumber>
    <StreamAttachInfo><!--stream additional information-->
        <streamWithVca><!--whether to overlay intelligent information on stream--></streamWithVca>
        <mutexAbility opt="PDC"/><!--opt-->
    </StreamAttachInfo>
</AudioVideoCompressInfo>
```

F.216 XML_Desc_EventAbility

Input description message for getting event capability.

```
<?xml version="1.0" encoding="utf-8"?>
<!--req, input description message for getting event capability.-->
<EventAbility version="2.0">
    <channelNO><!--req, channel No.--></channelNO>
</EventAbility>
```

F.217 XML_Desc_JpegCaptureAbility

Input description message for getting device capability of capturing JPEG picture.

```
<jpegCaptureAbility version="2.0">
    <channelNO>
        <!--req, channel No.-->
    </channelNO>
    <!--opt, the node that responds to the request can be specified, and it can be the child node of the root node-->
    <ManualCapture/>
</jpegCaptureAbility>
```

F.218 XML_Desc_NetAppAbility

Input description message for getting network application capability.

```
<?xml version="1.0" encoding="utf-8"?>
<!--req, input description message for getting network application capability-->
<NetAppAbility version="2.0">
</NetAppAbility>
```

F.219 XML_Desc_POSAbility

Input description message for getting POS capability

```
<?xml version="1.0" encoding="utf-8"?>
<POSAbility version="2.0">
  <!--optional, you can specify the nodes under the root node to return-->
</POSAbility>
```

F.220 XML_Desc_UserAbility

Input description message to get user configuration capability.

```
<UserAbility version="2.0"></UserAbility>
```

F.221 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-->true
</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>
<isSupportGPSCalibration>
  <!--optional, xs:boolean, whether it supports GPS calibration capability-->
</isSupportGPSCalibration>
<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>
<AIDEventSupport 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"-driving in low speed, "dragRacing"-street racing-->
</AIDEventSupport>
<TFSEventSupport
opt="illegalParking ,wrongDirection,crossLane,laneChange,vehicleExist,turnRound,parallelParking,notKeepDistance,not
SlowZebraCrossing,overtakeRightSide,lowSpeed,dragRacing,changeLaneContinuously,SSharpDriving,largeVehicleOccup
yLine,jamCrossLine">
  <!--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,notSupport"/><!--
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 <b><moduleType></b> in the URI (/
```

```
ISAPI/System/debugLog?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.222 XML_DeviceInfo

XML message about device information

```
<?xml version="1.0" encoding="utf-8"?>
<DeviceInfo version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<deviceName><!--required, xs:string--></deviceName>
<deviceID><!--required, read-only, xs:string, uuid--></deviceID>
<deviceDescription>
    <!--optional, xs:string, description about the device defined in RFC1213. For network camera, this node is set to "IPCamera"; for network speed dome, this node is set to "IPDome"; for DVR or DVS, this node is set to "DVR" or "DVS"-->
</deviceDescription>
<deviceLocation><!--optional, xs:string, actual location of the device--></deviceLocation>
<deviceStatus><!--optional, read-only, xs:string, device status: "normal", "abnormal"--></deviceStatus>
<DetailAbnormalStatus>
    <!--dependent, error status details, it is valid only when deviceStatus is "abnormal"-->
    <hardDiskFull>
        <!--optional, read-only, xs: boolean, whether the error of "HDD full" occurred: "true"-yes,"false"-no-->
    </hardDiskFull>
    <hardDiskError>
        <!--optional, read-only, xs:boolean, whether the error of "HDD error" occurred: "true"-yes,"false"-no-->
    </hardDiskError>
    <ethernetBroken>
        <!--optional, read-only, xs: boolean, whether the error of "network disconnected" occurred: "true"-yes,"false"-no-->
    </ethernetBroken>
    <ipaddrConflict>
        <!--optional, read-only, xs: boolean, whether the error of "IP address conflicted" occurred: "true"-yes,"false"-no-->
    </ipaddrConflict>
    <illegalAccess>
        <!--optional, read-only, xs: boolean, whether the error of "illegal login" occurred: "true"-yes,"false"-no-->
    </illegalAccess>
    <recordError>
        <!--optional, read-only, xs: boolean, whether the error of "recording exception" occurred: "true"-yes,"false"-no-->
    </recordError>
    <raidLogicDiskError>
        <!--optional, read-only, xs: boolean, whether the error of "RAID exception" occurred: "true"-yes,"false"-no-->
    </raidLogicDiskError>
    <spareWorkDeviceError>
        <!--optional, read-only, xs: boolean, whether the error of "working device exception" occurred: "true"-yes,"false"-no-->
    </spareWorkDeviceError>
</DetailAbnormalStatus>
<systemContact><!--optional, xs:string, contact information of the device--></systemContact>
<model><!--required, read-only, xs:string--></model>
<serialNumber><!--required, read-only, xs:string--></serialNumber>
<macAddress><!--required, read-only, xs:string--></macAddress>
<firmwareVersion><!--required, read-only, xs:string--></firmwareVersion>
<firmwareReleasedDate><!--optional, read-only, xs:string--></firmwareReleasedDate>
<bootVersion><!--optional, read-only, xs:string--></bootVersion>
```

```

<bootReleasedDate><!--optional, read-only, xs:string--></bootReleasedDate>
<hardwareVersion><!--optional, read-only, xs:string--></hardwareVersion>
<encoderVersion><!--optional, read-only, xs:string--></encoderVersion>
<encoderReleasedDate><!--optional, read-only, xs:string--></encoderReleasedDate>
<decoderVersion><!--optional, read-only, xs:string--></decoderVersion>
<decoderReleasedDate><!--optional, read-only, xs:string--></decoderReleasedDate>
<softwareVersion><!--optional, read-only, xs:string, software version--></softwareVersion>
<capacity><!--optional, read-only, xs:integer, unit: MB, device capacity--></capacity>
<usedCapacity><!--optional, read-only, xs:integer, unit: MB, capacity usage--></usedCapacity>
<deviceType>
    <!--required, read-only, xs:string, device type: "IPCamera", "IPDome", "DVR", "HybirdNVR", "NVR", "DVS", "IPZoom",
    "CVR", "Radar", "PerimeterRadar"--><!--perimeter radar, "ACS", "PHA"--><!--Axiom hybrid security control panel-->
</deviceType>
<telecontrolID><!--optional, xs:integer, keyfob control ID, the value is between 1 and 255--></telecontrolID>
<supportBeep><!--optional, xs:boolean--></supportBeep>
<supportVideoLoss><!--optional, xs:boolean, whether it supports video loss detection--></supportVideoLoss>
<firmwareVersionInfo><!--optional, read-only, xs:string, firmware version information--></firmwareVersionInfo>
<actualFloorNum>
    <!--required, xs: integer, actual number of floors, which is between 1 and 128-->
</actualFloorNum>
<subChannelEnabled><!--optional, xs:boolean, whether to support sub-stream live view: "true"-yes, "false"-no--></
subChannelEnabled>
<thrChannelEnabled><!--optional, xs:boolean, whether to support third stream live view: "true"-yes, "false"-no--></
thrChannelEnabled>
<radarVersion><!--optional, xs:string, radar version--></radarVersion>
<cameraModuleVersion><!--read-only, xs:string, camera module version--></cameraModuleVersion>
<mainversion><!--optional, xs:integer, main version No. which is between 1 and 255--></mainversion>
<subversion><!--optional, xs:integer, sub version No. which is between 1 and 255--></subversion>
<upgradeversion><!--optional, xs:integer, upgraded version No. which is between 1 and 255--></upgradeversion>
<customizeversion><!--optional, xs:integer, customized version No. which is between 1 and 255--></
customizeversion>
<companyName><!--optional, xs:string, the manufacturing company's abbreviation--></companyName>
<copyright><!--optional, xs:string, copyright information--></copyright>
<systemName><!--optional, xs:string , storage system name: "storageManagement"-storage management system,
"distributedStorageManagement"-distributed storage management system--></systemName>
<systemStatus><!--optional, xs:string, system status: "configured"-configured, "unConfigured"-not configured--></
systemStatus>
<isLeaderDevice><!--optional, xs:boolean, whether it is the corresponding device of the resource IP address--></
isLeaderDevice>
<clusterVersion><!--dependent, xs:string, system cluster version. This node is valid when the value of isLeaderDevice
is true--></clusterVersion>
<manufacturer><!--optional, xs:string, manufacturer information: "hikvision"-Hikvision devices; for neutral devices,
this node should be empty--></manufacturer>
<customizedInfo><!--optional, xs:string, order No. of the customization project. For baseline devices, this node is
empty; for custom devices, the order No. of the customization project will be returned by this node--></
customizedInfo>
<localZoneNum><!--optional, xs:integer, number of local zones--></localZoneNum>
<alarmOutNum><!--optional, xs:integer, number of alarm outputs--></alarmOutNum>
<distanceResolution><!--optional, xs:float, resolution of distance, unit: meter--></distanceResolution>
<angleResolution><!--optional, xs:float, resolution of angle, unit: degree--></angleResolution>
<speedResolution><!--optional, xs:float, resolution of speed, unit: m/s--></speedResolution>
<detectDistance><!--optional, xs:float, detection distance, unit: meter--></detectDistance>

```

```
<languageType><!--optional, xs:string, language type: Chinese, English, Spanish, Portuguese, Italian, French, Russian, German, Polish, Turkish, Greek, Czech, Brazilian, Portuguese, Slovenian, Swedish, Norwegian, Slovak, Serbian, Dutch, Hungarian, Irish, Bulgarian, Hebrew, Thai, Indonesian, Arabic, Traditional Chinese--></languageType>
<relayNum><!--optional, xs:integer, number of local relays--></relayNum>
<electroLockNum><!--optional, xs:integer, number of local electronic locks--></electroLockNum>
<RS485Num><!--optional, xs:integer, number of local RS-485--></RS485Num>
<powerOnMode><!--optional, xs:string, device startup mode: "button"-press button to power on (default), "adapter"-connect adapter to power on--></powerOnMode>
<DockStation>
    <!--optional, dock station configuration-->
<Platform>
    <!--optional, platform configuration-->
    <type><!--required, xs:string, platform type: none, 9533, 8618, ISAPI--></type>
    <ip><!--optional, xs:string, IP address --></ip>
    <port><!--optional, xs:integer, communication port--></port>
    <userName><!--required, xs:string, user name, which is used for the dock station to log in to platform--></userName>
        <password><!--required, xs:string, password, which is used for the dock station to log in to platform, it should be encrypted--></password>
    </Platform>
    <centralStorageBackupEnabled><!--optional, xs:boolean, whether to enable central storage backup--></centralStorageBackupEnabled>
</DockStation>
<webVersion><!--optional, read-only, xs:string, web version No.--></webVersion>
<deviceRFProgramVersion><!--optional, read-only, xs:string, version No. of the device's RF (Radio Frequency) program--></deviceRFProgramVersion>
<securityModuleSerialNo><!--optional, read-only, xs:string, serial No. of the security module--></securityModuleSerialNo>
<securityModuleVersion><!--optional, read-only, xs:string, version No. of the security module--></securityModuleVersion>
<securityChipVersion><!--optional, read-only, xs:string, version No. of the security chip--></securityChipVersion>
<securityModuleKeyVersion><!--optional, read-only, xs:string, version No. of the security module key--></securityModuleKeyVersion>
<UIDLampRecognition><!--optional, information of the UID lamp recognition device-->
    <enabled><!--optional, xs:boolean, whether to enable--></enabled>
</UIDLampRecognition>
<bootTime><!--optional, xs:string, read-only, system boot time, ISO 8601 format; the maximum length is 32 bytes--></bootTime>
<ZigBeeVersion min="0" max="16"><!--optional, xs:string, ZigBee module version--></ZigBeeVersion>
<R3Version min="0" max="16"><!--optional, xs:string, R3 module version--></R3Version>
<RxVersion min="0" max="16"><!--optional, xs:string, Rx module version--></RxVersion>
<bspVersion><!--optional, xs:string, BSP software version--></bspVersion>
<dspVersion><!--optional, xs:string, DSP software version--></dspVersion>
<localUIVersion><!--optional, xs:string, local UI version--></localUIVersion>
<isResetDeviceLanguage>
    <!--optional, boolean, whether it supports resetting the device language (only for Admin and Installer)-->false
</isResetDeviceLanguage>
</DeviceInfo>
```

F.223 XML_DeviceLanguage

DeviceLanguage message in XML format

```
<DeviceLanguage version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <language>
    <!--required, xs:string, language supported by the device: "SimChinese"-simplified Chinese, "TraChinese"-traditional Chinese, "English", "Russian", "Bulgarian", "Hungarian", "Greek", "German", "Italian", "Czech", "Slovakia", "French", "Polish", "Dutch", "Portuguese", "Spanish", "Romanian", "Turkish", "Japanese", "Danish", "Swedish", "Norwegian", "Finnish", "Korean", "Thai", "Estonia", "Vietnamese", "Hebrew", "Latvian", "Arabic", "Sovenian"-Slovenian, "Croatian", "Lithuanian", "Serbian", "BrazilianPortuguese"-Brazilian Portuguese, "Indonesian", "Ukrainian"-->
  </language>
</DeviceLanguage>
```

F.224 XML_DeviceStatus

DeviceStatus message in XML format

```
<DeviceStatus version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <currentDeviceTime><!--opt, xs: datetime--></currentDeviceTime>
  <deviceUpTime><!--opt, xs: integer, seconds--></deviceUpTime>
  <TemperatureList><!--opt-->
    <Temperature>
      <tempSensorDescription><!--req, xs: string--></tempSensorDescription>
      <temperature><!--req, xs: float--></temperature>
    </Temperature>
  </TemperatureList>
  <FanList><!--opt-->
    <Fan>
      <fanDescription><!--req, xs: string--></fanDescription>
      <speed><!--req, xs: integer--></speed>
    </Fan>
  </FanList>
  <PressureList><!--opt-->
    <Pressure>
      <pressureSensorDescription><!--req, xs: string--></pressureSensorDescription>
      <pressure><!--req, xs: integer--></pressure>
    </Pressure>
  </PressureList>
  <TamperList><!--opt-->
    <Tamper><tamperSensorDescription><!--req, xs: string--></tamperSensorDescription>
      <tamper><!--req, xs: boolean--></tamper>
    </Tamper>
  </TamperList>
  <CPUList><!--opt-->
    <CPU>
      <cpuDescription><!--req, xs: string--></cpuDescription>
      <cpuUtilization><!--req, xs: integer, percentage, which is between 0 and 100--></cpuUtilization>
    </CPU>
  </CPUList>
```

```
</CPUList>
<MemoryList><!--opt-->
<Memory>
  <memoryDescription><!--req, xs: string--></memoryDescription>
  <memoryUsage><!--req, xs: float, unit: MB--></memoryUsage>
  <memoryAvailable><!--req, xs: float, unit: MB--></memoryAvailable>
</Memory>
</MemoryList>
<openFileHandles><!--opt, xs: integer--></openFileHandles>
<CameraList><!--opt-->
<Camera>
  <zoomReverseTimes><!--req, xs: integer--></zoomReverseTimes>
  <zoomTotalSteps><!--req, xs: integer--></zoomTotalSteps>
  <focusReverseTimes><!--req, xs: integer--></focusReverseTimes>
  <focusTotalSteps><!--req, xs: integer--></focusTotalSteps>
  <irisShiftTimes><!--req, xs: integer--></irisShiftTimes>
  <irisTotalSteps><!--req, xs: integer--></irisTotalSteps>
  <icrShiftTimes><!--req, xs: integer--></icrShiftTimes>
  <icrTotalSteps><!--req, xs: integer--></icrTotalSteps>
  <lensIntirTimes><!--req, xs: integer--></lensIntirTimes>
  <cameraRunTotalTime><!--req, xs: integer--></cameraRunTotalTime>
</Camera>
</CameraList>
<DomeInfoList><!--opt-->
<DomeInfo>
  <domeRunTotalTime><!--opt, xs: integer--></domeRunTotalTime >
  <runTimeUnderNegativetwenty><!--opt, xs: integer--></runTimeUnderNegativetwenty>
  <runTimeBetweenNtwentypforty><!--opt, xs: integer--></runTimeBetweenNtwentypforty>
  <runtimeOverPositiveforty><!--opt, xs: integer--></runtimeOverPositiveforty>
  <panTotalRounds><!--opt, xs: integer--></panTotalRounds>
  <tiltTotalRounds><!--opt, xs: integer--></tiltTotalRounds>
  <heatState><!--opt, xs: integer--></heatState>
  <fanState><!--opt, xs: integer--></fanState>
</DomeInfo>
</DomeInfoList>
<deviceStatus><!--req, xs: string, device working status: "normal, abnormal" --></deviceStatus>
<dialSignalStrength>
  <!--optional, xs:integer, 4G signal strength, it is between 1 and 3-->
</dialSignalStrength>
<USBStatusList>
<USBStatus>
  <!--list-->
  <id>
    <!--required, xs:integer-->
  </id>
  <state>
    <!--optional, xs:string, USB status: "connected", "fullCapacity", "deviceException", "disconnected"-->
  </state>
</USBStatus>
</USBStatusList>
<WifiStatusList>
<WifiStatus>
```

```
<!--list-->
<id>
  <!--required, xs:integer-->
</id>
<state>
  <!--optional, xs:string, Wi-Fi status: "disabled"-Wi-Fi is disabled, "enable"-Wi-Fi is enabled, "disconnected"-Wi-Fi
is disconnected, "enableHotspot"-Wi-Fi hot spot is enabled-->
</state>
</WifiStatus>
</WifiStatusList>
<AlertStreamServerList><!--opt, user of arming device-->
<AlertStreamServer><!--list-->
<id><!--opt, xs:integer--></id>
<protocolType><!--opt, xs:string, "SDK,ISAPI", arming protocol type--></protocolType>
<ipAddress><!--dep, xs:string, IP address of platform, server, NVR/CVR, and so on--></ipAddress>
</AlertStreamServer>
</AlertStreamServerList>
</DeviceStatus>
```

F.225 XML_DiagnosedDataParameter

XML message about parameters of exporting diagnose information

```
<?xml version="1.0" encoding="utf-8"?>
<DiagnosedDataParameter version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled>
    <!--required, xs:boolean, whether to enable exporting diagnose information: true-yes, false-no-->
  </enabled>
  <HardwareInfo>
    <!--optional, xs:boolean, opt="true,false", whether to enable exporting hardware information, such as running
information of motor, fan, heater, compressor, supplement light, wiper, and so on-->
  </HardwareInfo>
  <SoftwareInfo>
    <!--optional, xs:boolean, opt="true,false", whether to enable exporting software information, such as software
status, running log, system information-->
  </SoftwareInfo>
  <logInfo>
    <!--optional, xs:boolean, opt="true,false", whether to enable exporting startup log-->
  </logInfo>
</DiagnosedDataParameter>
```

F.226 XML_Dial

Dial message in XML format

```
<Dial version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled><!--req, xs:boolean-->
  </enabled>
  <DialMethod><!--req, xs:string, "auto, manual"-->
  </DialMethod>
  <SwitchMethod>
    <!--req, xs:string, "auto,4GFirst,3GFirst, manualto2G, manualto3G, manualto4G"-->
  </SwitchMethod>
</Dial>
```

```
</SwitchMethod>
<OfflineTime><!--opt, xs:integer--></OfflineTime>
<UIMCardNum><!--opt, xs:string--></UIMCardNum>
<DialNum><!--opt, xs:string--></DialNum>
<Username><!--opt, xs:string--></Username>
<Password><!--opt, xs:string--></Password>
<APNname><!--opt, xs:string--></APNname>
<SIMNum><!--opt, xs:string, SIM card No. (mobile phone number)--></SIMNum>
<MTU><!--opt, xs: integer--></MTU>
<VerifyProto><!--req, xs:string, "auto, CHAP, PAP"--></VerifyProto>
<DefaultParam/><!--opt, default parameters, see details in the message of XML_DefaultParam-->
<netAPN><!--opt, xs:string, APN configuration of the private network--></netAPN>
<Flow><!--opt, network traffic flow configuration-->
  <limitEnabled><!--opt, xs:boolean, whether to enable traffic flow limitation--></limitEnabled>
  <consumeFlow><!--opt, xs:float, unit: MB--></consumeFlow>
  <threshold><!--opt, xs:integer, unit: MB--></threshold>
</Flow>
<pinCode><!--opt, wo, xs:string--></pinCode>
<enabled4G><!-- opt, xs:boolean, whether to enable 4G--></enabled4G>
<enabledDNS><!-- opt, xs:boolean, whether to enable DNS manual configuration--></enabledDNS>
<ISPName><!--optional, xs:string, ISP (Internet Service Provider) name--></ISPName>
<IMEINo><!--optional, xs:string, IMEI code--></IMEINo>
<ICCID><!--optional, xs:string, ICCID code--></ICCID>
<netType><!--optional, xs:string, network type--></netType>
</Dial>
```

See Also

[XML_DefaultParam](#)

F.227 XML_DownloadAbility

DownloadAbility message in XML format

```
<DownloadAbility version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <isSupportDownloadbyTime>
    <!--opt, xs:boolean, whether it supports download by time-->
  </isSupportDownloadbyTime>
  <isSupportDownloadbyFileName>
    <!--opt, xs:boolean, whether it supports download by file name -->
  </isSupportDownloadbyFileName>
  <isSupportDownloadToUSB>
    <!--opt, xs:boolean, whether it supports exporting files to devices via USB-->
  </isSupportDownloadToUSB>
</DownloadAbility>
```

F.228 XML_downloadRequest

XML message about downloading request

```
<downloadRequest version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <playbackURI>
    <!--required, xs:string, playback URL, returned by the search service. It contains the information of file name and size, e.g., <playbackURI>rtp://IpAddress/Streaming/tracks/101?starttime=2016-07-18 00:00:00&endtime=2016-07-18 00:25:04Z&name=00000000721000000&size=1065437356</playbackURI>-->
  </playbackURI>
  <downloadCourseFile>
    <!--optional, xs:boolean, whether to download the course files-->
  </downloadCourseFile>
</downloadRequest>
```

Remarks

When download file by name, the **playbackURI** inputs the file start time, end time, file name, and size; when download file by time, the **playbackURI** inputs the start time and end time.

F.229 XML_Ehome

XML message about the parameters of accessing a server via ISUP

```
<Ehome version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>
    <!--required, xs:integer, ID-->1
  </id>
  <enabled><!--optional, xs: boolean--></enabled>
  <addressingFormatType>
    <!--required, xs: string, address format type: "ipaddress"-IP address, "hostname"-host name-->
  </addressingFormatType>
  <hostName><!--dependent, xs: string, host name, this node is valid only when <addressingFormatType> is "hostname"--></hostName>
  <ipAddress><!--dependent, xs: string, IP address, this node is valid only when <addressingFormatType> is "ipaddress"--></ipAddress>
  <ipv6Address><!--dependent, xs: string, IPv6 address, this node is valid only when <addressingFormatType> is "ipaddress"--></ipv6Address>
  <portNo><!--optional, xs: integer, port No.--></portNo>
  <deviceID><!--required, xs: string, device ID--></deviceID>
  <registerStatus><!--ro, xs: boolean, registration status--></registerStatus>
  <ServerAddressList><!--optional, array, information list of servers-->
    <ServerAddress><!--optional, object, server information-->
      <id><!--required, string, server type: "alarm", "picture", "ntp" (NTP time synchronization), "backup", "enforce" (enforcement), "bayonet" (checkpoint)--></id>
        <ipAddress><!--required, string, IP address of the server--></ipAddress>
        <portNo><!--required, int, port No. of the server--></portNo>
      </ServerAddress>
    </ServerAddressList>
```

```
<key><!--optional, xs: string, encryption key of ISUP version 5.0, which is required for ISUP version 50--></key>
<version><!--ro, xs: string, version No.--></version>
<voiceDeviceType><!--optional, xs:string, two-way audio device type: "bluetooth"-bluetooth device, "client", "local"--></voiceDeviceType>
<protocolVersion><!--optional, xs:string, protocol version: "v2.0,v2.6,v4.0,v5.0". If this node is set to "v2.0", the device can only use protocol v2.0 to register; if this node is set to "v2.6", "v4.0" or "v5.0", the device will firstly use this protocol to register. If this node is not returned, the protocol version will be determined by <version>--></protocolVersion>
<netWork><!--optional, xs:integer, 0-make no sense, 1-automatic, 2-wired network preferred, 3-wired network, 4-3G/4G/GPRS--></netWork>
<pictureServerID>
  <!--optional, xs:integer, ID of the linked picture storage server (/ISAPI/System/PictureServer)-->1
</pictureServerID>
<periodicTestEnabled>
  <!--optional, xs:boolean, whether to enable periodic test. After this function is enabled, the host will send a Test event to the server every configured interval for checking whether the link is valid for interaction-->true
</periodicTestEnabled>
<periodicTestTime>
  <!--optional, xs:integer, periodic test interval, unit: second, range: [10,86400]-->10
</periodicTestTime>
</Ehome>
```

F.230 XML_EhomeList

XML message about the parameters of accessing all servers via ISUP

```
<EhomeList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--required, array-->
  <Ehome>
    <id>
      <!--required, xs:integer, ID-->1
    </id>
    <enabled><!--optional, xs: boolean--></enabled>
    <addressingFormatType>
      <!--required, xs: string, address format type: "ipaddress"-IP address, "hostname"-host name-->
    </addressingFormatType>
    <hostName><!--dependent, xs: string, host name, this node is valid only when <addressingFormatType> is "hostname"--></hostName>
    <ipAddress><!--dependent, xs: string, IP address, this node is valid only when <addressingFormatType> is "ipaddress"--></ipAddress>
      <ipv6Address><!--dependent, xs: string, IPv6 address, this node is valid only when <addressingFormatType> is "ipaddress"--></ipv6Address>
        <portNo><!--optional, xs: integer, port No.--></portNo>
        <deviceID><!--required, xs: string, device ID--></deviceID>
        <registerStatus><!--ro, xs: boolean, registration status--></registerStatus>
        <ServerAddressList><!--optional, array, information list of servers-->
          <ServerAddress><!--optional, object, server information-->
            <id><!--required, string, server type: "alarm", "picture", "ntp" (NTP time synchronization), "backup", "enforce" (enforcement), "bayonet" (checkpoint)--></id>
            <ipAddress><!--required, string, IP address of the server--></ipAddress>
```

```
<portNo><!--required, int, port No. of the server--></portNo>
</ServerAddress>
</ServerAddressList>
<key><!--optional, xs: string, encryption key of ISUP version 5.0, which is required for ISUP version 50--></key>
<version><!--ro, xs: string, version No.--></version>
<voiceDeviceType><!--optional, xs:string, two-way audio device type: "bluetooth"-bluetooth device, "client", "local"--></voiceDeviceType>
<protocolVersion><!--optional, xs:string, protocol version: "v2.0,v2.6,v4.0,v5.0". If this node is set to "v2.0", the device can only use protocol v2.0 to register; if this node is set to "v2.6", "v4.0" or "v5.0", the device will firstly use this protocol to register. If this node is not returned, the protocol version will be determined by <version>--></protocolVersion>
<netWork><!--optional, xs:integer, 0-make no sense, 1-automatic, 2-wired network preferred, 3-wired network, 4-3G/4G/GPRS--></netWork>
<pictureServerID>
  <!--optional, xs:integer, ID of the linked picture storage server (/ISAPI/System/PictureServer)-->1
</pictureServerID>
<periodicTestEnabled>
  <!--optional, xs:boolean, whether to enable periodic test. After this function is enabled, the host will send a Test event to the server every configured interval for checking whether the link is valid for interaction-->true
</periodicTestEnabled>
<periodicTestTime>
  <!--optional, xs:integer, periodic test interval, unit: second, range: [10,86400]-->10
</periodicTestTime>
</Ehome>
</EhomeList>
```

F.231 XML_EPTZ

EPTZ message in XML format

```
<EPTZ version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled><!--req, xs:string, whether to enable e-PTZ: "true"-yes, "false"-no--></enabled>
  <streamType><!--opt, xs:integer, stream type: 1-main stream, 2-sub-stream, 3-third stream, 4-fourth stream--></streamType>
</EPTZ>
```

F.232 XML_EventAbility

EventAbility capability message in XML format

```
<EventAbility version="2.0">
  <channelNO><!--req, channel No.--></channelNO>
  <ExceptionAlarm>
    <exceptionType
      opt="diskFull,diskError,nicBroken,ipConflict,illAccess,videoMismatch,badVideo,recordingFailure,raid,resolutionMismatch,spareException,POEPowerException"/>
    <alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,SMS"/>
    <!--req, exception function linkage detailing, if resolved to the capability below, it will be processed as exception alarm shown below; if not, it will be processed according to alarmHandleType node-->
```

```
<alarmRelateAct opt="ftp,email"/>
<!--req, alarm linkage action: "ftp"- upload captured picture to FTP, "email"-upload captured JPEG picture to email-->
</DetailedExceptionAlarm>
<DiskFull><!--req, HDD full-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</DiskFull>
<DiskError><!--req, HDD error-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</DiskError>
<NicBroken><!--req, network cable broken-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</NicBroken>
<IPConflict><!--req, IP conflict-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</IPConflict>
<IIIAccess><!--req, illegal access-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</IIIAccess>
<BadVideo><!--req, video loss-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</BadVideo>
<VideoMismatch><!--req, video not matched-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</VideoMismatch>
<RecordingFailure><!--req, video exception-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</RecordingFailure>
<Raid><!--req, Array exception-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</Raid>
<ResolutionMismatch><!--req, resolution not matched-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</ResolutionMismatch>
<SpareException><!--req, hot spare exception-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</SpareException>
<TemperatureException><!--req, temperature exception-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</TemperatureException>
<SubsystemException><!--req, sub system exception-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</SubsystemException>
<FanException><!--req, fan exception-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
</FanException>
</DetailedExceptionAlarm>
</ExceptionAlarm>
<AlarmIn>
<alarmTime>8</alarmTime>
<!--req, number of arming interval -->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,uploadcloud"/>
```

```
<!--req, handle -->
<notSupportPTZLinkage>true</notSupportPTZLinkage>
<!--req, dynamic display based on current capability-->
<RelatePTZ>
    <!--req, dynamic display based on current capability-->
    <presetNo min="" max="" />
    <!--req, link supported preset No.-->
    <CruiseNo min="" max="" />
    <!--req, link supported patrol path No.-->
    <ptzTrackNo min="" max="" />
    <!--req, link supported PTZ pattern No.-->
    <presetDurationTime min="0" max="20"/>
    <!--opt,preset dwell time -->
</RelatePTZ>
<inputType opt="switch,signal"/>
<!--opt, alarm input type: switch- relay, signal- pulse signal-->
<eventTypeNum min="1" max="" /><!--opt, supported number of composite event types-->
<eventType
opt="VMD,shelteralarm,facedetection,faceCapture,fielddetection,linedetection,regionEntrance,regionExitings,loitering
,group,rapidMove,parking,unattendedBaggage,attendedBaggage,ANPR,audioexception"/>
    <!--opt,composite event types, VMD-motion detection, shelteralarm-tampering alarm, facedetection-face
detection, faceCapture-face capture, fielddetection-intrusion, linedetection-line crossing, regionEntrance-region
entrance, regionExitings-region exiting, loitering-lotering detection, group-people gathering, rapidMove-fast moving,
parking-parking detection, unattendedBaggage-unattended baggage detection, attendedBaggage-object removal
detection, ANPR-license plate recognition, audioexception-audio exception detection-->
    <alarmRelateAct opt="ftp,email"/>
    <!--req, alarm linkage action: ftp- upload captured picture to FTP, email- upload captured JPEG picture to email-->
</Alarmln>
<AlarmOut>
    <alarmTime>8</alarmTime>
    <!--req, the number of output acyivate time-->
    <pulseDuration opt="5,10,30,60,120,300,600,manual"/>
    <!--req, the alarm output delay, unit: s-->
</AlarmOut>
<FaceDetection>
    <!--req, face detection-->
    <detectFaceEnable opt="true,false"/>
    <!--req, enable face detection: true-yes, false-no -->
    <detectFaceSensitive min="0" max="9"/>
    <!--req,sensitivity -->
    <alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,uploadcloud "/>
    <!--req, handle-->
    <triggerRecord>true</triggerRecord>
    <!--req,enable or disable, true-support, false-not support-->
    <mutexAbility opt="traversingVirtualPlane,fieldDetection"/>
    <!--req mutex ability for line crossing detection and intrusion detection-->
    <alarmRelateAct opt="ftp,email"/>
    <!--req, alarm linkage action: ftp- upload captured picture to FTP, email- upload captured JPEG picture to email-->
</FaceDetection>
<Track>
    <!--req, intelligent track -->
    <trackEnable opt="true,false"/>
```

```
<!--req, enable the configuration,true-enable, false- disable -->
<stopTrackWhenFindFace opt="true,false"/>
<!--req find face, whether stop tracking -->
<stopTrackThreshold min="1" max="100"/><!--req stop threshold value, the function is supported when the node exits -->
<ManualTrace>
  <!--opt,support manual trace, the corresponding structure is NET_DVR_PTZ_MANUALTRACE-->
  <enabled>true</enabled>
  <!--req, the function is supported when the node exit-->
  <trackType opt="normalTrack,highSpeedRoadTrack,urbanRoadTrack"/>
  <!--req trace type, the function is supported when the node exits, 0- normally track, 1- highSpeedRoadTrack, 2- urbanRoadTrack-->
<linkageTrack opt="ipd"/>
  <!--opt,"ipd means link to dome camera tracking"-->
</ManualTrace>
</Track>
<VoiceDetection>
  <!--req, voice exception detection -->
  <enable opt="true,false"/>
  <!--req, enable voice detection: true- yes, false- no -->
<Abnormal>
  <sensitivityLevel min="1" max="100"/>
  <!--req,sensitivity -->
  <audioMode opt="0,1,2"/>
  <!--req, audio detection mode: 0- enable sensitivity detection, 1- enable db threshold detection, 2- enable both -->
  <enable opt="true,false"/>
  <!--req,enable audio detection: true-yes, false- no -->
  <threshold min="1" max="100"/>
  <!--req, audio threshold -->
</Abnormal>
<alarmTime>8</alarmTime>
<!--req, number of output activate time interval -->
<alarmHandleType opt="monitor, audio, center, alarmout, picture, wirelesslight, uploadftp, uploadcloud"/>
<!--req, handle-->
<audioSteepDrop>
  <!--sound sharp rise or fall-->
  <enable opt="true,false"/><!--req, whether to enable, true- enable, false- disable-->
  <sensitivityLevel min="1" max="100"/>
  <!--req, sensitivity-->
</audioSteepDrop>
</VoiceDetection>
<TraversingVirtualPlane>
  <!--req, line crossing detection capability-->
  <enable opt="true,false"/>
  <!--req, false: disable; true: enable -->
<enableDualVca opt="true,false"/>
  <!--req enable supporting smart post search: true- enable, false- disable-->
<ALERTLINENum>8</ALERTLINENum>
<!--req, the maximum number of warning line that traversing supported -->
<AlertLine>
  <!--req, the node has the number according to the ALERTLINENum -->
  <ID>1</ID>
```

```
<crossDirection opt="0,1,2"/>
<!--req, cross direction: 0-both way, 1- from left to right, 2- from right to left -->
<sensitivityLevel min="0" max="100"/>
<!--req,sensitivity -->
</AlertLine>
<alarmTime>8</alarmTime>
<!--req, number of arming time interval -->
<alarmHandleType
opt="monitor, audio, center, alarmout, picture, wirelesslight, uploadftp, ptztrack, uploadcloud, SMS"/>
<!--req, handle, picture: capture and send by e-mail(added because the lack of uploadftp in the early time; if there is no string, judge the UploadFTP node of devices soft hardware to find whether support capture and then uploaded to FTP -->
<supportLogNotCfg>true</supportLogNotCfg><!--req, support log not configuration, no return means support configuration and log-->
<holidayTimeSlotNum>8</holidayTimeSlotNum><!--req,the number of holiday time for traversing plane detection -->
<mutexAbility opt="faceDetection,19-1920*1080@50fps,
20-1920*1080@60fps,PDC,videoFrameRate50,videoFrameRate60,recordPlan"/>
<!--req, mutually exclusive capability, face detection, CaptureMode 1920*1080@50fps, CaptureMode 1920*1080@60fps, PDC, videoFrameRate50, videoFrameRate60, scheduled recording-->
<detectionTarget opt="all,human,vehicle,others"/>
<recogRuleType opt="vectorMode,slopeMode"/>
<!--req, line crossing detection direction: vector or slope-->
<triggerRecord>true</triggerRecord>
<!--req, whether to enable triggered recording: true- supported, false- not supported-->
<alarmRelateAct opt="ftp,email"/>
<!--req, alarm linkage action: ftp- upload captured picture to FTP, email- upload captured JPEG picture to email-->
<isSupportHumanMisinfoFilter>
<!--opt, xs:boolean, "support preventing false human detection alarm or not, if support, return true, if not, no return"-->
</isSupportHumanMisinfoFilter>
<isSupportVehicleMisinfoFilter>
<!--opt, xs:boolean, "support preventing false vehicle detection alarm or not, if support, return true, if not, no return"-->
</isSupportVehicleMisinfoFilter>
<isSupportAllDayUpload><!--opt, xs:boolean, whether to support uploading all-day events: "true,false"--></isSupportAllDayUpload>
</TraversingVirtualPlane>
<FieldDetection>
<!--req, intrusion detection-->
<enable opt="true,false"/>
<!--req,false: disable; true: enable -->
<IntrusiongionNum>8</IntrusiongionNum>
<!--req,the maximum warning line that field detection supported -->
<Intrusiongion>
<!--req,INTRUSIONREGIONNum, the node has the corresponding number -->
<ID>1</ID>
<regionNum min="3" max="10"/>
<!--req,the valid point number of each field detection supported -->
<duration min="1" max="100"/>
<!--req, time triggered -->
<sensitivityLevel min="0" max="100"/>
```

```
<!--req, sensitivity -->
<rate min = "0" max = "100"/>
<!--req, rate: no alarm target size accounted for the proportion of alarm area -->
</Intrusioniong>
<alarmTime>8</alarmTime>
<!--req, number of arming time interval -->
<alarmHandleType
opt="monitor, audio, center, alarmout, picture, wirelesslight, uploadftp, ptztrack, uploadcloud, SMS"/>
<!--req, handle, picture: capture and send by e-mail(added because the lack of uploadftp in the early time; if there is
no string, judge the UploadFTP node of devices soft hardware to find whether support capture and then uploaded to
FTP -->
<holidayTimeSlotNum>8</holidayTimeSlotNum><!--req, the number of holiday time for traversing plane detection-->
<mutexAbility opt="faceDetection,19-1920*1080@50fps,
20-1920*1080@60fps,PDC,videoFrameRate50,videoFrameRate60,recordPlan"/>
<!--req, mutually exclusive capability, face detection, CaptureMode 1920*1080@50fps, CaptureMode
1920*1080@60fps, PDC, videoFrameRate50, videoFrameRate60, scheduled recording-->
<detectionTarget opt="all,human,vehicle,others"/>
<notSupportTriggerRecord><!--req, "true"-not support--></notSupportTriggerRecord>
<!--req, whether to enable triggered recording: true- supported, false- not supported-->
<alarmRelateAct opt="ftp,email"/>
<!--req, alarm linkage action: ftp- upload captured picture to FTP, email- upload captured JPEG picture to email-->
<isSupportHumanMisinfoFilter>
  <!--opt, xs:boolean, "support preventing false human detection alarm or not, if support, return true, if not, no
return"-->
  </isSupportHumanMisinfoFilter>
<isSupportVehicleMisinfoFilter>
  <!--opt, xs:boolean, "support preventing false vehicle detection alarm or not, if support, return true, if not, no
return"-->
  </isSupportVehicleMisinfoFilter>
<isSupportAllDayUpload><!--opt, xs:boolean, whether to support uploading all-day events: "true,false"--></
isSupportAllDayUpload>
</FieldDetection>
<DefousDetection>
  <!--req, defocus detection -->
  <enable opt="true,false"/>
  <!--req, false: disable; true: enable -->
  <alarmHandleType opt="monitor, audio, center, alarmout, picture, wirelesslight, uploadftp, focus"/>
  <!--req, handle, picture: capture and send by e-mail(added because the lack of uploadftp in the early time; if there is
no string, judge the UploadFTP node of devices soft hardware to find whether support capture and then uploaded to
FTP -->
    <sensitivityLevel min="1" max="100"/><!--req, sensitivity-->
  <alarmRelateAct opt="ftp,email"/>
  <!--req, alarm linkage action: ftp- upload captured picture to FTP, email- upload captured JPEG picture to email-->
</DefousDetection>
<SceneChangeDetection>
  <!--req, scene change detection -->
  <enable opt="true,false"/>
  <!--req, false: disable; true: enable -->
  <sensitiveLevel min="1" max="100"/>
  <!--req, sensitivity -->
<alarmTime>8</alarmTime>
```

```
<!--req,number of arming time interval -->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,uploadcloud"/>
<!--req,handle -->
<sceneChangeDetectionRecord>true</sceneChangeDetectionRecord><!--req video linkage -->
<alarmRelateAct opt="ftp,email"/>
<!--req, alarm linkage action: ftp- upload captured picture to FTP, email- upload captured JPEG picture to email-->
</SceneChangeDetection>
<AudioLimitAlarm>
<!--req, sound limit alarm-->
<enable opt="true,false"/><!--req,false: disable, true: enable-->
<!--req, alarm threshold, unit: dB-->
<decibelLimit min="50" max="70"/>
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
<!--req, process type, picture: capture and email-->
</AudioLimitAlarm>
<ButtonDownAlarm>
<enable opt="true,false"/>
<!--req,false: disable, true: enable-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
<!--req, process type, picture: capture and email-->
</ButtonDownAlarm>
<VCADetection>
<!--req, VCA detection-->
<enable opt="true,false"/>
<!--req,false: disable, true: enable-->
<alarmTime>8</alarmTime>
<!--req, number of arming periods-->
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
<!--req, process type, picture: capture and email-->
<preset>true</preset>
<!--preset, no return if not supported-->
<cruise>true</cruise>
<!--patrol point, no return if not supported-->
<ptzTrack>true</ptzTrack>
<!--PTZ track, no return if not supported-->
<holidaySched>true</holidaySched>
<!--holiday plan, no return if not supported-->
<alarmRelateAct opt="ftp,email"/>
<!--req, alarm linkage action: ftp- upload captured picture to FTP, email- upload captured JPEG picture to email-->
</VCADetection>
<VehicleDetection>
<DetectVehicleCalibration><!--req, detect vehicle calibration ability-->
<widePixels>400</widePixels>
<highPixels>400</highPixels>
</DetectVehicleCalibration>
<TriggerCond>
<channel min="" max="" /><!--req, channel No.-->
<detSencelD min="" max="" /><!--req, scene No.-->
<triggerMode
opt="postIOSpeed,postSingleIO,postRS485,postRS485Radar,postVirtualcoil,epoliceloTrafficLights,epoliceRS485,peRS48
5,videoPolice,postMPR,viaVtCoil,ipcHVT"/>
<!--req, triggering mode-->
```

```
</TriggerCond>
<TriggerCfg>
    <enable opt="disable,enable"/><!--req, enable this triggering mode or not-->
    <triggerType
opt="postIOSpeed,postSingleIO,postRS485,postRS485Radar,postVirtualcoil,epoliceloTrafficLights,epoliceRS485,peRS48
5,videoEpolicy,postMPR,viaVtCoil,ipCHVT"/>
    <!--req,triggering mode-->
    <TriggerParam>
        <PostMPR>
            <enable opt="disable,enable"/>
            <!--req, enable or disable-->
            <laneNum min="" max="" />
            <!--req, lane number-->
            <sourceType opt="MPR,IO,RS485"/>
            <!--req, signal source-->
        <LaneBoundaryLine>
            <Line>
                <Start>
                    <x min="0.001" max="1"/>
                    <!--req, X-axis, 0.001~1-->
                    <y min="0.001" max="1"/>
                    <!--req, Y-axis, 0.001~1-->
                </Start>
                <End>
                    <x min="0.001" max="1"/>
                    <!--req, X-axis, 0.001~1-->
                    <y min="0.001" max="1"/>
                    <!--req, Y-axis, 0.001~1-->
                </End>
            </Line>
        </LaneBoundaryLine>
        <PlateRecogParam>
            <defaultCHNLen min="0" max="3"/>
            <!--req, the string length of the chinese abbreviations of the provience where the device is running-->
            <province
opt="1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,0xff"/>
            <!--req, provience index-->
            <enable opt="disable,enable"/>
            <!--req, enable the licensee plate recognition of this area or not-->
            <!--req, recognition type-->
            <RecogMode>
                <plateRecogPos opt="fromFront,fromBack"/>
                <!--req, licensee plate recognition position, fromFront-Front Plate, fromBack-Rear Plate-->
                <plateRecogType opt="big,little"/>
                <!--req, licensee plate recognition type, big-Big Plate, little-Small Plate-->
                <vechileColorRecog opt="true,false"/>
                <!--req, vehicle color recognition: 0-Disable vehicle color recognition when real plate or small plate
recognition, 1-Enable vehicle color recognition-->
                <farmVehicleRecog opt="true,false"/>
                <!--req, farm vehicle recognition-->
                <blurRecog opt="true,false"/>
                <!--req, fuzzy recognition-->
```

```
<locationType opt="frameLocation,sceneLocation"/>
<!--req, position type, frameLocation-Frame Positioning, sceneLocation-Scene Positioning-->
<recogType opt="frameRecognition,sceneRecognition"/>
<!--req, recognition type, frameRecognition-Frame Recognition ,sceneRecognition-Scene Recognition-->
<recogTime opt="daytime,night"/>
<!--req, recognition time, daytime-Daytime, night-Night-->
<motorCarRecog opt="true,false"/>
<!--req, motorcycle recognition-->
<sceneType opt="epolice,gate"/>
<!--req, scene type, EPolice-E-police/Multiframe, gate-Monitoring Point-->
<microPlateRecog opt="true,false"/>
<!--req, smaller license plate recognition-->
<region opt="Res,EU,ER,EUAndCIS"/>
<!--req, area index: 0-Reserved, 1-Europe(EU), 2-Russia(ER), 3-Europe&Russia(EU&CIS)-->
<platePixelWidthMin min="" max="" />
<!--req, the pixel width of license plate, corresponding to the wPlatePixelWidthMin in struct
NET_ITC_PLATE_RECOG_PARAM-->
<platePixelWidthMax min="" max="" />
<!--req,the pixel width of license plate, corresponding to the wPlatePixelWidthMax in struct
NET_ITC_PLATE_RECOG_PARAM-->
</RecogMode>
<vehicleLogoRecog opt="disable,enable"/>
<!--req, vehicle logo recognition-->
</PlateRecogParam>
<maxLaneLineNum min="0" max="6"/>
<!--req, the max. lane line number supported-->
<LaneParam>
<laneNO min="" max="" />
<!--req, related lane No.-->
<TssParamInfo>
<IO>
<no min="" max="" />
<triggerType opt="fallEdge,riseEdge"/>
</IO>
<RS485>
<relateChan min="" max="" />
</RS485>
</TssParamInfo>
<carDriveDirection opt="unknown,uptodown,downtoup"/>
<!--vehicle driving direction, unknown-Unknow, uptodown-Down Direction,downtoup-Up Direction-->
<LaneLine>
<Line>
<Start>
<x min="0.001" max="1"/>
<!--req,X-axis, 0.001~1-->
<y min="0.001" max="1"/>
<!--req,Y-axis, 0.001~1-->
</Start>
<End>
<x min="0.001" max="1"/>
<!--req,X-axis, 0.001~1-->
```

```
<y min="0.001" max="1"/>
<!--req,Y-axis, 0.001~1-->
</End>
<lineType
opt="unknown,ordinaryLanes,busLanes,fastLane,slowLane,motorcycleLane,nonmotorVehicleLane,reversedLane,banTrucksLane,mixLane"/>
<!--req, lane line type, 0-Unknown, 1-The white solid line between lanes, 2-Lane Stop Line, 3-Single Yellow Line, 4-Double Yellow Line,5-Guardrails on the Lane Line, 6-The lane line the vehicle cannot cross, 7-Dotted Line-->
</Line>
</LaneLine>
<PlateRecog>
<pointNum min="3" max="20"/>
<!--req, efficient point, great than or equal to 3 points, if three points are in the same line, the area is invalid. If the lines cross, the area is invalid. Support up to 20 points.-->
<Pos>
<x min="0.001" max="1"/>
<!--req, X-axis, 0.001~1-->
<y min="0.001" max="1"/>
<!--req, Y-axis, 0.001~1-->
</Pos>
</PlateRecog>
</LaneParam>
<autoBuildRecogArea>true</autoBuildRecogArea>
<!--opt, client generates the recognition area automatically, SDK interface has no corresponding field information-->
<brokenNetHttp>true</brokenNetHttp>
<!--opt, support breakpoint resume or not-->
<RodeType>
<!--opt,Road Type-->
<type opt="entrance,city,custom,alarmInput"/>
<!--opt, road type: entrance- entrance, city- city road, custom- custom, alarmInput- alarm input post, publicSecurity-->
<Custom>
<delayTime min="0" max="15000"/>
<delayTimeUnit opt="ms"/>
</Custom>
</RodeType>

<brokenNetHttp>true</brokenNetHttp>
<!--opt, whether to support ANR for vehicle recognition-->

<SnapLine>
<!--opt, capture line-->
<Line>
<Start>
<x min="0.001" max="1"/>
<!--req, X coordinate, 0.001~1-->
<y min="0.001" max="1"/>
<!--req, Y coordinate, 0.001~1-->
</Start>
<End>
```

```

<x min="0.001" max="1"/>
<!--req, X coordinate, 0.001~1-->
<y min="0.001" max="1"/>
<!--req, Y coordinate, 0.001~1-->
</End>
</Line>
</SnapLine>
</PostMPR>
</TriggerParam>
</TriggerCfg>

<!--req, capture triggering mode supports linkage of arming time periods and uploading center-->
<GuardCond>
    <channel min="" max="" />
    <!--req, channel No.-->
    <relateType opt="MPR"/>
    <!--req, relation type-->
</GuardCond>
<GuardCfg>
    <detSencelD min="" max="" />
    <!--req, scene No.-->
    <alarmSchedDays min="0" max="7"/>
    <!--req, arming days, up to 7 days supported for one week -->
    <alarmSchedTimes min="0" max="8"/>
    <!--req, arming time periods, up to 8 periods supported for one day-->
    <AlarmSched>
        <startHour min="0" max="24"/>
        <!--req, arming start time-->
        <startMin min="0" max="60"/>
        <!--req, arming start time-->
        <stopHour min="0" max="24"/>
        <!--req, arming end time-->
        <stopMin min="0" max="60"/>
        <!--req, arming end time-->
    </AlarmSched>
    <alarmHandleType
opt="monitor, audio, center, alarmout, picture, wirelesslight, uploadftp, focus, PTZ, uploadcloud"/>
        <!--req, handling type-->
        <maxAlarmOutChannelNum min="0" max="64"/>
        <!--req, the max. alarm output channel to be triggered supported by the device-->
        <alarmOutChannelNum min="" max="" />
        <!--req, the currently set channel No. to be triggered-->
        <alarmOutTriggeredChannel min="" max="" />
        <!--req, alarm channel to be triggered-->
<direction>
    <!--opt,xs:string,"both,forward,reverse",triggering direction-->
</direction>
</GuardCfg>

<!--req, OSD test overlay(support overlay of independent and composite graph)-->
<OverlapCond>
    <channel min="" max="" />

```

```

<!--req, channel No.-->
<configMode opt="terminal,camera"/>
<!--req, configuration mode, 0-Terminal, 1-Front-end(directly connect to the front-end device or connect the
terminal and front-end device)-->
<picModeType opt="smallPicture,bigPicture"/>
<!--req, picture type, 0-Small Picture(Independent Graph), 1-Large Picture(Composite Graph)-->
<relateType opt="MPR,HVT"/>
<!--req, MPR by default when no node is returned-->
</OverlapCond>
<OverlapCfg>
<enable opt="disable,enable"/>
<!--req, enable or not, 0-Disable, 1-Enable-->
<OverlapItemParam>
<overlapItemNum min="0" max="50"/>
<!--req, up to 50 characters-->
<SingleItem>
<itemType
opt="unknown,place,crossingNo,deviceNo,directionNo,direction,laneNo,lane,captureTime,captureTimeWithMS,plateN
o,vehicleColor,vehicleType,vehicleBrand,vehicleSpeed,speedLimitSign,vehicleLength,illegalcode,monitorInfo,illegalActi
vities,superSpeedRatio,redStartTime,redStopTime,redBrightTime,securityCode,captureNo,safeBelt,monitorNo,sunViso
r,laneDirection,licensePlateColor,sceneNumber, sceneName, yellowSignCar, dangerousCar, carSubBrand,
vehicleDirection, validity, country,plateType,plateColor,plateCategory"/>
    <!--req, text type, 0-Unknow, 1-Place, 2-Intersection No., 3-Device No., 4-Direction No., 5-Direction, 6-Lane
No., 7-Lane, 8-Capture Time(without millisecond),9-Capture Time(with millisecond), 10-License Plate Number, 11-
Vehicle Color, 12-Vehicle Type, 13-Vehicle Brand, 14-Vehicle Speed, 15-Speed Limit Sign, 16-Vehicle Length(1 to 99m),
17-Violation Code(traffic violation information is more useful than code, e.g. Normal, Low Speed, Overspeed, Opposite
Direction, Run the Red Light, Occupying Lane, Over Yellow Lane Line),18-Monitoring Site Information, 19-Traffic
Violation, 20-Overspeed Ratio, 21-Red Light Start Time, 22-Red Light End Time, 23-Red Light Time, 24-Security Code,
25-Capture No.,26-Seat Belt, 27-Monitoring Site No., 28-Sun Shield, 29-Lane Direction, 30-License Plate Color, 31-
Scene No., 32-Scene Name, 33- Yellow Label Car, 34- Dangerous Goods Vehicle, 35- Vehicle Sub Brand, 36- vehicle
direction, 38- confidence 40- Country, 41- License plate type 42-License plate size, 43- License plate color, 44- License
plate additional information-->
    <changeLineNum min="0" max="10"/>
    <!--req, line feeds number after overlay[0 to 10](by default is 0)-->
    <spaceNum min="0" max="255"/>
    <!--req, space number after overlay [0 to255](by default is 0)-->
    <enablePos opt="0,1 "/>
    <!--req, enable coordinate display or not, 0-Disable, 1-Enable-->
    <startPosTop min="0" max="2448"/>
    <!--req, start top coordinate, only valid in picture internal overlay [0 to 2448](by default is 0)-->
    <startPosLeft min="0" max="2448"/>
    <!--req, start left coordinate, only valid in picture internal overlay [0 to 2448](by default is 0)-->
</SingleItem>
<linePercent min="0" max="100"/>
<!--req, percentage of overlay lines(0 to 100),(by default is 100)-->
<itemsStyle opt="0,1"/>
<!--req, overlay mode, 0-Horizontal, 1-Vertical (by default is horizontal)-->
<startPosTop min="0" max="2448"/>
<!--req, start top coordinate, only valid in picture internal overlay[0 to 2448](by default is 0)-->
<startPosLeft min="0" max="2448"/>
<!--req, start left coordinate, only valid in picture internal overlay [0 to 2448](by default is 0)-->
<charStyle opt="0,1"/>

```

```

<!--req, font type, 0-Song Typeface, 1-Wei Typeface(default)-->
<charSize min="" max="" />
<!--req, character length-->
<charInterval min="0" max="" />
<!--req, character pitch, [0 to 16], Unit: Pixel(default)-->
<foreClorRGB opt="B,G,R,White"/>
<!--req, RGB of foreground color, bit0-7:(B) bit8-15:(G) bit16-23:(R) (by default 0x00FFFFFF-White)-->
<backClorRGB opt="B,G,R,Black"/>
<!--req, RGB of background color, only valid in picture external overlay, bit0-7:(B) bit8-15:(G) bit16-23:(R) (by
default x00000000-Black-->
<colorAdapt opt="disable,enable"/>
<!--req, color self-adaptive or not, 0-No, 1-Yes-->
<paramFillZeroEnble opt="disable,enable"/>
<!--req, enable parameter zerofilling, 0-Enable, 1-Disable (annotation). Speed, speed limit, zerofill when less
than 3 digits -->
<plateLeftCornerEnable opt="disable,enable"/>
<!--req, enable overlay of license plate window in the upper-left corner, 0-Disable, 1-Enable-->
<startSPicPosTop min="0" max="2448"/>
<!--req, start top coordinate, only valid in picture internal overlay [0 to 2448](by default is 0)-->
<startSPicPosLeft min="0" max="2448"/>
<!--req, start left coordinate, only valid in picture internal overlay [0~2448](by default is 0)-->
<osdLocate opt="inside,upper,lower"/>
<!--req, OSD overlay position, 0-In the picture, 1-Picture top edge, 2-Picture lower edge (for composite
graph, out of the top edge)-->
</OverlapItemParam>
<OverlapInfoParam>
    <siteLen min="0" max="128"/>
    <!--req, position description length-->
    <roadNumLen min="0" max="32"/>
    <!--req, intersection No. length-->
    <instrumentNumLen min="0" max="32"/>
    <!--req, device No. length-->
    <directionLen min="0" max="32"/>
    <!--req, direction No. length-->
    <directionDesLen min="0" max="32"/>
    <!--req, direction description length-->
    <laneDesLen min="0" max="32"/>
    <!--req, lane description length-->
    <monitoringSite1Len min="0" max="44"/>
    <!--req, monitoring site 1 information length-->
    <monitoringSite2Len min="0" max="32"/>
    <!--req, monitoring site 2 information length-->
</OverlapInfoParam>
</OverlapCfg>

<!--req, monitoring site parameter configuration-->
<MonitorCfg>
    <deviceType opt="camera,coilTriggeringCamera,videoAnalysisCamera,ITC"/>
    <!--req, device type, 0-Camera for Monitoring, 1-Coil Triggered Camera, 2-Video Analysis Camera, 3-All-in-
one(ITC)-->
    <monitoringSiteIDLen min="0" max="48"/>
    <!--req, monitoring site No. length-->

```

```
<deviceIDLen min="0" max="48"/>
<!--req, device No. length-->
<directionNo min="" max="" />
<!--req, monitoring direction No.-->
<monitoringSInfoLen min="0" max="48"/>
<!--req, monitoring site description length-->
</MonitorCfg>

<!--req,trigger capture manually-->
<ManualSnap>
<enabled opt="disable,enable"/>
<!--req, support triggering capture manually or not. If not support, it will not display.-->
<osdEnable opt="disable,enable"/>
<!--req, enable OSD, false-Enable(default), true-Disable-->
</ManualSnap>

<!--req, captured picture parameter configuration-->
<SnapPicInfoCfg>
<picSize min="64" max="2048"/>
<!--req, picture size[64k,2048k]-->
<picQuality min="1" max="100"/>
<!--req, picture quality [1,100]-->
</SnapPicInfoCfg>
<mutexAbility opt="stream3,hvtVehicleDetection,heatMap"/>
<!--req, mutually exclusive ability:stream3(Stream 3), mixed-traffic detection -->
</VehicleDetection>
<HVTVehicleDetection>
<DetectVehicleCalibration>
<!--req, detect vehicle calibration capability-->
<widePixels>400</widePixels>
<highPixels>400</highPixels>
</DetectVehicleCalibration>
<TriggerCond>
<channel min="" max="" />
<!--req channel No.-->
<detSencelD min="" max="" />
<!--req detect scene No.-->
<triggerMode opt="postIORSpeed,postSingleIO,postRS485,postRS485Radar,postVirtualcoil,epoliceloTrafficLights,
epolicerRS485,peRS485,videoEpolicy,postMPR,viaVtCoil,ipCHVT"/>
<!--req, trigger mode-->
</TriggerCond>
<TriggerCfg>
<enable opt="disable,enable"/>
<!--req, whether to enable the trigger mode-->
<triggerType opt="postIORSpeed,postSingleIO,postRS485,postRS485Radar,postVirtualcoil,epoliceloTrafficLights,
epolicerRS485,peRS485,videoEpolicy,postMPR,viaVtCoil,ipCHVT"/>
<!--req, trigger mode -->
<TriggerParam>
<PostIPCHVT>
<enable opt="disable,enable"/>
<!--req, enable-->
<laneNum min="" max="" />
```

```

<!--req, number of lanes-->
<LaneBoundaryLine>
  <Line>
    <Start>
      <x min="0.001" max="1"/>
      <!--req, X axis, 0.001 to 1-->
      <y min="0.001" max="1"/>
      <!--req, Y axis, 0.001 to 1-->
    </Start>
    <End>
      <x min="0.001" max="1"/>
      <!--req, X coordinate, 0.001~1-->
      <y min="0.001" max="1"/>
      <!--req, Y coordinate, 0.001~1-->
    </End>
  </Line>
</LaneBoundaryLine>
<PlateRecogParam>
  <defaultCHNLen min="0" max="3"/>
  <!--req, length of the string for province name shorthand-->
  <province
opt="1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,0xff"/>
  <!--req province index-->
  <enable opt="disable,enable"/>
  <!--req, whether to enable the region sign-->
  <!--detected type-->
  <RecogMode>
    <plateRecogPos opt="fromFront,fromBack"/>
    <!--plate recognition position, fromFront- from front, fromBack- from back-->
    <plateRecogType opt="big,little"/>
    <!--plate recognition type, big- big plate, little- little plate-->
    <vechileColorRecog opt="true,false"/>
    <!--vehicle color recognition: 0- disable, cannot be enabled for recognition of small plate or from back, 1-
enable-->
    <farmVehicleRecog opt="true,false"/>
    <!--farming vehicle recognition-->
    <blurRecog opt="true,false"/>
    <!--fuzzy recognition-->
    <locationType opt="frameLocation,sceneLocation"/>
    <!--location type: frameLocation- frame location, sceneLocation- scene location-->
    <recogType opt="frameRecognition,sceneRecognition"/>
    <!--recognition type: frameRecognition- frame recognition, sceneRecognition- scene recognition-->
    <recogTime opt="daytime,night"/>
    <!--recognition time: daytime- day, night- night-->
    <motorCarRecog opt="true,false"/>
    <!--motorcycle recognition-->
    <sceneType opt="epolice,gate"/>
    <!--scene type: EPolice- e-police/multi-frame, gate- checkpoint-->
    <microPlateRecog opt="true,false"/>
    <!--micro plate recognition-->
    <region opt="Res,EU,ER"/>
    <!--req, region index: 0- reserved, 1- Europe, 2- Russia-->

```

```

</RecogMode>
<vehicleLogoRecog opt="disable,enable"/>
<!--Logo recognition-->
</PlateRecogParam>
<maxLaneLineNum min="0" max="6"/>
<!--req max. supported number of lane lines-->
<LaneParam>
<laneNO min="" max="" />
<!--req link lane No.-->
<carDriveDirection opt="unknown,uptodown,downtoup"/>
<!--vehicle driving direction, unknown- unknown, uptodown- downward, downtoup- upward-->
<LaneLine>
<Line>
<Start>
<x min="0.001" max="1"/>
<!--req X coordinate, 0.001~1-->
<y min="0.001" max="1"/>
<!--req Y coordinate, 0.001~1-->
</Start>
<End>
<x min="0.001" max="1"/>
<!--req X coordinate, 0.001~1-->
<y min="0.001" max="1"/>
<!--req Y coordinate, 0.001~1-->
</End>
</Line>
</LaneLine>
<PlateRecog>
<pointNum min="3" max="20"/>
<!--req valid point, >=3, if the three points are on the same line or the lines crosses, the region is considered invalid, up to 20 points are supported-->
<Pos>
<x min="0.001" max="1"/>
<!--req X coordinate, 0.001~1-->
<y min="0.001" max="1"/>
<!--req Y coordinate, 0.001~1-->
</Pos>
</PlateRecog>
</LaneParam>
</PostIPCHVT>
</TriggerParam>
</TriggerCfg>

<!--req, capture trigger mode supports linkage to arming time periods and uploading to center-->
<GuardCond>
<channel min="" max="" />
<!--req, channel No.-->
<relateType opt="MPR"/>
<!--req, linkage type-->
</GuardCond>
<GuardCfg>
<detSencelD min="" max="" />

```

```

<!--req, detection scene No.-->
<alarmSchedDays min="0" max="7"/>
<!--req, arming days, up to 7 days a week are supported-->
<alarmSchedTimes min="0" max="8"/>
<!--req, arming time periods, up to 8 periods a day are supported-->
<AlarmSched>
    <startHour min="0" max="24"/>
    <!--req, arming start time-->
    <startMin min="0" max="60"/>
    <!--req, arming start time-->
    <stopHour min="0" max="24"/>
    <!--req, arming end time-->
    <stopMin min="0" max="60"/>
    <!--req, arming end time-->
</AlarmSched>
<alarmHandleType opt="monitor, audio, center, alarmout, picture, wirelesslight, uploadftp, focus, PTZ"/>
<!--req, processing type-->
<maxAlarmOutChannelNum min="0" max="64"/>
<!--req, max number of triggered alarm output channels supported by device-->
<alarmOutChannelNum min="" max="" />
<!--req, current set alarm triggered channels-->
<alarmOutTriggeredChannel min="" max="" />
<!--req, trigger alarm channel-->
<direction>
    <!--opt, xs:string, "both, forward, reverse", trigger direction-->
</direction>
</GuardCfg>

<!--req, OSD character overlay(independent image or composite image overlay)-->
<OverlapCond>
    <channel min="" max="" />
    <!--req channel No.-->
    <configMode opt="terminal, camera" />
    <!--req configuration mode, 0- terminal, 1- front end(direct connection or terminal connection)-->
    <picModeType opt="smallPicture, bigPicture" />
    <!--req picture type: 0- small (independent), 1- big (composite)-->
    <relateType opt="MPR, HVT" />
    <!-- default to MPR if the node is not returned-->
</OverlapCond>
<OverlapCfg>
    <enable opt="disable, enable" />
    <!--req whether to enable, 0- disable, 1-enable-->
    <OverlapItemParam>
        <overlapItemNum min="0" max="50" />
        <!--req max. 50 characters-->
        <SingleItem>
            <itemType
                opt="unknown, place, crossingNo, deviceNo, directionNo, direction, laneNo, lane, captureTime, captureTimeWithMS, plateN
                o, vehicleColor, vehicleType, vehicleBrand, vehicleSpeed, speedLimitSign, vehicleLength, illegalcode, monitorInfo, illegalActi
                vities, superSpeedRatio, redStartTime, redStopTime, redBrightTime, securityCode, captureNo, safeBelt, monitorNo, sunViso
                r, laneDirection, licensePlateColor, sceneNumber, sceneName, vehicleDirection, validity" />
            <!--req character type: 0- unknown, 1- location, 2- crossroad No., 3- device No., 4- direction No., 5- direction, 6-

```

lane No., 7- lane, 8- capture time(without millisecond), 9- capture time (without millisecond), 10- plate No., 11- vehicle color, 12- vehicle type, 13- vehicle brand, 14- vehicle speed, 15- speed limit sign, 16- vehicle length, 17- illegal code, 18- camera information, 19- illegal conduct, 20- overspeed ratio, 21- red light start time, 22- light off time, 23- red light duration, 24- anti-counterfeiting code, 25- capture code, 26- safety belt, 27- camera No., 28- sun shield, 29- lane driving direction, 30- plate color, 31- scene No., 32- scene name, 36- vehicle direction, 38- confidence -->

```

<changeLineNum min="0" max="10"/>
<!--req lines after item overlay [0-10](default to 0)-->
<spaceNum min="0" max="255"/>
<!--req spaces after item overlay [0-255](default to 0)-->
<enablePos opt="0,1 "/>
<!--req whether to enable coordinate display, 0- no, 1- yes-->
<startPosTop min="0" max="2448"/>
<!--req start top coordinate, only valid for internal picture overlay [0-2448](default to 0)-->
<startPosLeft min="0" max="2448"/>
<!--req start left coordinate, only valid for internal picture overlay [0-2448](default to 0)-->
</SingleItem>
<linePercent min="0" max="100"/>
<!--req overlay line percentage (0-100),(default to 100)-->
<itemsStlye opt="0,1"/>
<!--req overlay type: 0- horizontal, 1- vertical(default to horizontal)-->
<startPosTop min="0" max="2448"/>
<!--req start top coordinate, only valid for internal picture overlay [0-2448](default to 0)-->
<startPosLeft min="0" max="2448"/>
<!--req start left coordinate, only valid for internal picture overlay [0-2448](default to 0)-->
<charStyle opt="0,1"/>
<!--req character type, 0- Song 1- Wei(default)-->
<charSize min="" max="" />
<!--req character length-->
<charInterval min="0" max="" />
<!--req character space, [0-16], unit: pixel(default)-->
<foreClorRGB opt="B,G,R,White"/>
<!--req foreground color RGB value bit0-7:(B) bit8-15:(G) bit16-23:(R) (default: 0x00FFFFFF-white)-->
<backClorRGB opt="B,G,R,Black"/>
<!--req background color RGB value, only valid for external picture overlay, bit0-7:(B) bit8-15:(G) bit16-23:(R)
(default: x00000000-black-->
<colorAdapt opt="disable,enable"/>
<!--req whether the color is self-adaptive 0- no, 1- yes-->
<paramFillZeroEnble opt="disable,enable"/>
<!--req enable parameter zero fill, 0- enable, 1- disable (detailed notes) speed, speed limit, zero fill to 3 places-->
<plateLeftCornerEnable opt="disable,enable"/>
<!--req enable plate small picture overlay to upper left corner, 0- disable, 1- enable-->
<startSPicPosTop min="0" max="2448"/>
<!--req start top coordinate, only valid for internal picture overlay [0-2448](default to 0)-->
<startSPicPosLeft min="0" max="2448"/>
<!--req start left coordinate, only valid for internal picture overlay [0-2448](default to 0)-->
<osdLocate opt="inside,upper,lower"/>
<!--req OSD overlay position 0- inside the picture, 1- top edge, 2- bottom edge (top edge is exclusive for
composite picture)-->
</OverlapItemParam>
<OverlapInfoParam>
<siteLen min="0" max="128"/>
<!--req, location description length-->

```

```

<roadNumLen min="0" max="32"/>
<!--req, crossroads No. length-->
<instrumentNumLen min="0" max="32"/>
<!--req, device No. length-->
<directionLen min="0" max="32"/>
<!--req, direction No. length-->
<directionDescLen min="0" max="32"/>
<!--req, direction description length-->
<laneDesLen min="0" max="32"/>
<!--req, lane description-->
<monitoringSite1Len min="0" max="44"/>
<!--req, camera 1 information length-->
<monitoringSite2Len min="0" max="32"/>
<!--req, camera 2 information length-->
</OverlapInfoParam>
</OverlapCfg>

<!--req, camera parameter configuration-->
<MonitorCfg>
    <deviceType opt="camera,coilTriggeringCamera,videoAnalysisCamera,ITC"/>
    <!--req, device type, 0- surveillance camera; 1- coil trigger camera; 2- video analysis camera, 3- ITC-->
<monitoringSiteIDLen min="0" max="48"/><!--req camera No. length-->
    <deviceIDLen min="0" max="48"/>
    <!--req, device No. length-->
    <directionNo min="" max="" />
    <!--req, surveillance direction No.-->
    <monitoringSInfoLen min="0" max="48"/>
    <!--req, camera information description length-->
</MonitorCfg>

<!--req, manual trigger capture-->
<ManualSnap>
    <enabled opt="disable,enable"/>
    <!--req, whether to support manual trigger capture. It will not be displayed if not supported-->
    <osdEnable opt="disable,enable"/>
    <!--req, enable OSD false- enable(default), true-disable-->
</ManualSnap>

<!--req, capture picture parameter configuration-->
<SnapPicInfoCfg>
    <picSize min="64" max="2048"/>
    <!--req picture size[64k,2048k]-->
    <picQuality min="1" max="100"/>
    <!--req picture quality[1,100]-->
</SnapPicInfoCfg>
<mutexAbility opt="stream3,vehicleDetection"/>
<!--req mutually exclusive capability: stream3(stream 3), mixed vehicle detection -->
</HVTVehicleDetection>
<!--req, support current trigger mode by default, device will be indicated if the node is returned, supporting detailed function, the interface display is based on capability-->
<CurVehicleDetection>
    <getCfg>true</getCfg>

```

```

<setCfg>true</setCfg>
</CurVehicleDetection>
<AlarmSearch><!--optional, capability of searching for alarm or event information-->
  <alarmComm opt="json,faceSnap,faceContrast"><!--required, xs:string, supported alarm type: "json"-alarm
transmitted in JSON format, "faceSnap"-face capture, "faceContrast"-face comparison--></alarmComm>
  <EventList><!--dependent, specific alarm type to be transmitted, this node is valid only when <alarmComm> is
"json"-->
    <Event>
      <type><!--required, xs:string, event type: "mixedTargetDetection"-multi-target-type detection--></type>
      <subEventType opt="face,human,vehicle,radar"><!--optional, xs:string, sub event type: 1-"face", 2-"human"
(human body), 3-"vehicle", 4-"radar". The index starts from 1--></subEventType>
    </Event>
  </EventList>
</AlarmSearch>
</EventAbility>

```

F.233 XML_EventCap

EventCap capability message in XML format

```

<EventCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <isSupportHDFull><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportHDFull>
  <isSupportHDError><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportHDError>
  <isSupportNicBroken><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportNicBroken>
  <isSupportIpConflict><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportIpConflict>
  <isSupportIIIAccess><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportIIIAccess>
  <isSupportViException><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportViException>
  <isSupportViMismatch><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportViMismatch>
  <isSupportRecordException><!--optional, xs:boolean, "true"-support, "false"-not support--></
isSupportRecordException>
  <isSupportTriggerFocus><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportTriggerFocus>
  <isSupportMotionDetection><!--optional, xs:boolean, "true"-support, "false"-not support--></
isSupportMotionDetection>
  <isSupportVideoLoss><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportVideoLoss>
  <isSupportTamperDetection><!--optional, xs:boolean, "true"-support, "false"-not support--></
isSupportTamperDetection>
  <isSupportStudentsStoodUp><!--optional, xs:boolean, "true"-support, "false"-not support--></
isSupportStudentsStoodUp>
  <isSupportFramesPeopleCounting><!--optional, xs:boolean, "true"-support, "false"-not support--></
isSupportFramesPeopleCounting>
  <isSupportRaidException><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportRaidException>
  <isSupportSpareException><!--optional, xs:boolean, "true"-support, "false"-not support--></
isSupportSpareException>
  <isSupportPoePowerException><!--optional, xs:boolean, "true"-support, "false"-not support--></
isSupportPoePowerException>
  <isSupportRegionEntrance><!--optional, xs:boolean, "true"-support, "false"-not support--></
isSupportRegionEntrance>
  <isSupportRegionExiting><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportRegionExiting>
  <isSupportLoitering><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportLoitering>
  <isSupportGroup><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportGroup>

```

```
<isSupportRapidMove><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportRapidMove>
<isSupportFireDetection><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportFireDetection>
<isSupportIllegalParking><!--optional, xs:boolean, whether it supports illegal parking detection: "true"-support,
"false"-not support--></isSupportIllegalParking>
<isSupportUnattendedBaggage><!--optional, xs:boolean --></isSupportUnattendedBaggage>
<isSupportAttendedBaggage><!--optional, xs:boolean, "true"-support, "false"-not support--></
isSupportAttendedBaggage>
<isSupportHumanAttribute><!--optional, xs:boolean, "true"-support, "false"-not support--></
isSupportHumanAttribute>
<isSupportFaceContrast><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportFaceContrast>
<isSupportFaceLib><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportFaceLib>
<isSupportWhiteListFaceContrast><!--opt, xs:boolean, "true"-support, "false"-not support--></
isSupportWhiteListFaceContrast>
<isSupportBlackListFaceContrast><!--opt, xs:boolean, whether it supports blocklist face comparison: "true"-support,
"false"-not support--></isSupportBlackListFaceContrast>
<isSupportFramesPeopleCounting><!--optional, xs:boolean, whether it supports regional people counting--></
isSupportFramesPeopleCounting>
<isSupportHumanRecognition><!--optional, xs:boolean, "true"-support, "false"-not support--></
isSupportHumanRecognition>
<isSupportFaceSnap><!--optional, xs:boolean, "true"-support, "false"-not support--></isSupportFaceSnap>
<isSupportPersonDensityDetection><!--optional, xs:boolean, "true"-support, "false"-not support--></
isSupportPersonDensityDetection>
<isSupportMixedTargetDetection><!--optional, xs:boolean, whether it supports multi-target-type detection alarm:
"true"-support, "false"-not support--></isSupportMixedTargetDetection>
<isSupportPedestrian><!--optional, xs:boolean, whether it supports pedestrian detection: "true"-support, "false"-not
support--></isSupportPedestrian>
<isSupportTrafficAccident><!--optional, xs:boolean, whether it supports traffic accident detection: "true"-support,
"false"-not support--></isSupportTrafficAccident>
<isSupportConstruction><!--optional, xs:boolean, whether it supports construction detection: "true"-support, "false"-not
support--></isSupportConstruction>
<isSupportRoadBlock><!--optional, xs:boolean, whether it supports roadblock detection: "true"-support, "false"-not
support--></isSupportRoadBlock>
<isSupportAbandonedObject><!--optional, xs:boolean, whether it supports thrown object detection: "true"-support,
"false"-not support--></isSupportAbandonedObject>
<isSupportParallelParking><!--optional, xs:boolean, whether it supports parallel parking detection: "true"-support,
"false"-not support--></isSupportParallelParking>
<isSupportParkingState><!--optional, xs:boolean, whether it supports parking space status detection: "true"-support,
"false"-not support, currently this node is not supported--></isSupportParkingState>
<isSupportCongestion><!--optional, xs:boolean, whether it supports congestion detection: "true"-support, "false"-not
support--></isSupportCongestion>
<isSupportVehicleStatistics><!--optional, xs:boolean, whether it supports data collection: "true"-support, "false"-not
support--></isSupportVehicleStatistics>
<isSupportWrongDirection><!--optional, xs:boolean, whether it supports wrong-way driving detection: "true"-support,
"false"-not support--></isSupportWrongDirection>
<isSupportTrunRound><!--optional, xs:boolean, whether it supports U-turning detection: "true"-support, "false"-not
support--></isSupportTrunRound>
<isSupportCrossLane><!--optional, xs:boolean, whether it supports driving on the lane line detection: "true"-support,
"false"-not support--></isSupportCrossLane>
<isSupportLaneChange><!--optional, xs:boolean, whether it supports illegal lane change detection: "true"-support,
"false"-not support--></isSupportLaneChange>
<isSupportVehicleExist><!--optional, xs:boolean, whether it supports motor vehicle on non-motor vehicle lane
detection: "true"-support, "false"-not support--></isSupportVehicleExist>
```

```
<isSupportFogDetection><!--optional, xs:boolean, whether it supports fog detection: "true"-support, "false"-not support--></isSupportFogDetection>
<isSupportIntersectionAnalysis><!--optional, xs: boolean, whether it supports configuring intersection analysis alarm: "true"-support, "false"-not support--></isSupportIntersectionAnalysis>
<isSupportVoltageInstable><!--optional, xs:boolean, whether it supports supply voltage exception alarm: "true"-support, "false"-not support--></isSupportVoltageInstable>
<isSupportSafetyHelmetDetection><!--optional, xs:boolean, whether it supports hard hat detection: "true"-support, "false"-not support--></isSupportSafetyHelmetDetection>
<isSupportCertificateRevocation><!--optional, xs:boolean, whether it supports certificate expiry alarm--></isSupportCertificateRevocation>
<isSupportNoMaskDetection><!--optional, xs:boolean, whether device supports no wearing mask detection--></isSupportNoMaskDetection>
<isSupportTMPA><!--optional, xs:boolean, whether device supports temperature measurement pre-alarm--></isSupportTMPA>
<RuleScheduleCap><!--optional, capability of setting arming schedule by rule-->
<isSupportCityManagement>
    <!--optional, xs:boolean, whether the device supports setting arming schedule by rule for intelligent city management; if supports, the value is true, otherwise, this node will not be returned-->
</isSupportCityManagement>
</RuleScheduleCap>
<isSupportThermalCalibrationFileException><!--optional, xs:boolean, whether the device supports alarm of thermography calibration file exception--></isSupportThermalCalibrationFileException>
<isSupportTemperatureIntervalMeasurement><!--optional, xs:boolean, whether the device supports interval temperature measurement--></isSupportTemperatureIntervalMeasurement>
<isSupportPTEEventCfg><!--optional, xs:boolean, whether the device supports event transmission, related URI(/ISAPI/Event/PTEEventCfg/capabilities?format=json)-->true</isSupportPTEEventCfg>
</EventCap>
```

F.234 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.235 XML_EventTrigger

Linkage parameter message in XML format

```
<EventTrigger version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:string, ID--></id>
  <eventType>
    <!--required, xs:string, see details in the "Remarks" below-->
  </eventType>
  <eventDescription><!--optional, xs:string--></eventDescription>
  <inputIOPortID><!--dependent, xs:string, alarm input ID--></inputIOPortID>
  <dynInputIOPortID><!--dependent, xs:string, dynamic alarm input ID--></dynInputPortID>
  <videoInputChannelID>
    <!--dependent, xs:string, video input channel ID, it is valid when <eventType> is "VMD, videoloss, tamperdetection, regionEntrance, regionExiting, loitering, group, rapidMove, parking, unattendedBaggage, attendedBaggage"-->
  </videoInputChannelID>
  <dynVideoInputChannelID><!--dependent, xs:string, dynamic video input channel ID--></dynVideoInputChannelID>
  <intervalBetweenEvents><!--optional, xs:integer, event time interval, unit: second--></intervalBetweenEvents>
  <WLSensorID><!--dependent, xs:string, ID--></WLSensorID>
  <EventTriggerNotificationList/><!--optional, alarm/event linkage actions, see details in the message of
XML_EventTriggerNotificationList-->
</EventTrigger>
```

Remarks

The node **<eventType>** can be the following values: IO, VMD, videoloss, raidfailure, recordingfailure, badvideo, POS, analytics, fanfailure, overheat, tamperdetection, diskfull, diskerror, nicbroken, ipconflict, illaccess, videomismatch, resolutionmismatch, radifailure, PIR, WLSensor, spareException, poePowerException, heatmap, counting, linedetection, fielddetection, regionEntrance, regionExiting, loitering, group, rapidMove, parking, unattendedBaggage, attendedBaggage, HUMANATTRIBUTE, blackList, whitelist, peopleDetection, allVehicleList, otherVehicleList, vehicledetection, storageDetection, shipsDetection, humanAttribute, faceContrast, blackListFaceContrast, whiteListFaceContrast, faceSnap, faceLib, personDensityDetection, personQueueDetecton, mixedTargetDetection, HVTVehicleDetection, illegalParking, pedestrian, trafficAccident, construction, roadblock, abandonedObject, parallelParking, parkingState, congestion, intersectionAnalysis, heatMap, thermometry, shipsFlowDetection, dredgerDetection, reverseEntrance, luma, highHDTemperature, lowHDTemperature, hdImpact, hdBadBlock, SevereHDFailure, safetyHelmetDetection, vibrationDetection, HBDLib, TMPA, faceThermometry, noMaskDetection, detectorTemp, detectorSmoke, detectorTamper, smokeFireRecognize, indoorPasswayBlock, detectorShelter, detectorMotion, fireNoRegulation.

See Also

[XML_EventTriggerNotificationList](#)

F.236 XML_EventTriggerCapType

XML message about capability of alarm linkage action types

```
<EventTriggerCapType version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<isSupportCenter><!--optional, xs:boolean--></isSupportCenter>
<isSupportRecord><!--optional, xs:boolean--></isSupportRecord>
<isSupportMonitorAlarm><!--optional, xs:boolean--></isSupportMonitorAlarm>
<isSupportBeep><!--optional, xs: boolean, whether it supports audible warning--></isSupportBeep>
<isSupportIO><!--optional, xs:boolean--></isSupportIO>
<isSupportFTP><!--optional, xs:boolean--></isSupportFTP>
<isSupportEmail><!--optional, xs:boolean--></isSupportEmail>
<isSupportLightAudioAlarm><!--optional, xs:boolean--></isSupportLightAudioAlarm>
<isSupportFocus><!--optional, xs:boolean--></isSupportFocus>
<isSupportPTZ><!--optional, xs:boolean--></isSupportPTZ>
<maxPresetActionNum>
  <!--dependent, xs:integer, it is valid only when <isSupportPTZ> is "true"-->
</maxPresetActionNum>
<maxPatrolActionNum>
  <!--dependent, xs:integer, it is valid only when <isSupportPTZ> is "true"-->
</maxPatrolActionNum>
<maxPatternActionNum>
  <!--dependent, xs:integer, it is valid only when <isSupportPTZ> is "true"-->
</maxPatternActionNum>
<isSupportTrack><!--optional, xs:boolean, whether it supports PTZ linked tracking--></isSupportTrack>
<isSupportWhiteLight>
  <!--optional, xs: boolean, whether it supports supplement light alarm linkage-->
</isSupportWhiteLight>
<isSupportCloud><!--optional, xs:boolean, whether it supports upload to the cloud--></isSupportCloud>
<targetNotificationInterval max="1000" min="0" default="30"><!--xs:integer, range: [0, 1000], the default value is 30, unit: seconds, this node is valid for <MotionDetectionTriggerCap> and <TamperDetectionTriggerCap> and this node is valid when <isSupportPTZ> is "true"--></targetNotificationInterval>
  <direction opt="both,forward,reverse"><!--xs:string, triggering direction, this node is valid for the node <BlackListTriggerCap>, <WhiteListTriggerCap>, and <VehicleDetectionTriggerCap--></direction>
  <presetDurationTime min="" max=""><!--dependent, xs:integer--></presetDurationTime>
  <isSupportSMS><!--optional, xs:boolean, whether to support SMS (Short Message Service)--></isSupportSMS>
  <maxCellphoneNum><!--dependent, xs:integer, the maximum number of cellphones, which is node is valid only when <isSupportSMS> is "true"--></maxCellphoneNum>
  <isSupportOSD><!--optional, xs:boolean--></isSupportOSD>
  <isSupportAudio><!--optional, xs:boolean, whether it supports setting audio alarm independently. If this node is set to "true", audio alarm and buzzer alarm can be linked separately, and the lineage method is audio--></isSupportAudio>
  <AudioAction><!--dependent, this node is valid when <isSupportBeep> is "true" or <isSupportAudio> is "true"-->
    <audioTypeList>
      <audioType><!--list-->
        <audioID><!--required, xs:integer, alarm sound type--></audioID>
        <audioDescription><!--required, xs:string, alarm sound description, it should correspond to the alarm sound type--></audioDescription>
      </audioType>
    </audioTypeList>
  <alarmTimes opt="0,1,2,3,4,5,6,7,8,9,255"><!--required, xs:integer, alarm times, it is between 0 and 9, 255-continuous alarm, unit: time--></alarmTimes>
```

```

</AudioAction>
<isSupportSMS><!--optional, xs:boolean --></isSupportSMS>
<maxCellphoneNum><!--dependent, if <isSupportSMS> is true, xs:integer--></maxCellphoneNum>
<isNotSupportCenterModify><!--optional, xs:boolean, whether editing configuration parameters of the monitoring center is not supported: "true"-yes (configuration parameters of the monitoring center cannot be edited), "false" or this node is not returned-no (configuration parameters of the monitoring center can be edited)--></isNotSupportCenterModify>
<isSupportMessageConfig>
  <!--optional, xs:boolean, whether it supports SMS configuration, if supports, set cellphoneNumber to null-->
</isSupportMessageConfig>
<isSupportAnalogOutput><!--optional, xs:boolean, whether it supports IO output of linkage analog channel--></isSupportAnalogOutput>
<isSupportIOOutputUnify><!--optional, xs:boolean, whether it supports configuration of IO output--></isSupportIOOutputUnify>
<isSupportFaceContrast><!--optional, xs:boolean, whether it supports face picture comparison linkage--></isSupportFaceContrast>
<isSupportSiren><!--optional, xs:boolean, whether it supports siren linkage--></isSupportSiren>
<isSupportOutput><!--optional, xs:boolean, whether it supports relay linkage--></isSupportOutput>
</EventTriggerCapType>

```

F.237 XML_EventTriggerNotification

Event linkage notification message in XML format

```

<EventTriggerNotification><!--opt-->
<id><!--required, xs:string, device ID--></id>
<notificationMethod>
  <!--required, xs:string, linkage actions, opt="email,IM,IO,syslog,HTTP,FTP,beep,ptz,record, monitorAlarm, center, LightAudioAlarm, focus, trace, cloud, SMS, whiteLight, audio, whiteLight, faceContrast, siren, output"-->
</notificationMethod>
<notificationRecurrence>
  <!--optional, xs:string, "beginning,beginningandend,recurring"-->
</notificationRecurrence>
<notificationInterval><!--dependent, xs:integer, unit: millisecond--></notificationInterval>
<outputIOPortID><!--dependent, xs:string, video output No., it is required only when notificationMethod is "IO"--></outputIOPortID>
<dynOutputIOPortID><!--dependent, xs:string, dynamic video output No., it is required only when notificationMethod is "IO"--></dynOutputIOPortID>
<videoInputID><!--dependent, xs:string, video input No., it is required only when notificationMethod is "record"--></videoInputID>
<dynVideoInputID><!--dependent, xs:string, dynamic video input No., it is required only when notificationMethod is "record"--></dynVideoInputID>
<ptzAction><!--dependent, it is required only when notificationMethod is "ptz"-->
  <ptzChannelID><!--required, xs:string, PTZ channel ID--></ptzChannelID>
  <actionName><!--required, xs:string, PTZ control type: "preset", "pattern", "patrol"--></actionName>
  <actionNum><!--dependent, xs:integer--></actionNum>
</ptzAction>
<WhiteLightAction><!--dependent, white light linkage parameters, this node is valid when notificationMethod is "whiteLight"-->
  <whiteLightDurationTime><!--required, xs:integer, white light flashing duration, it is between 1 and 60, unit:
```

```
second--></whiteLightDurationTime>
</WhiteLightAction>
<cellphoneNumber><!--dependent, xs:string, min="0" max="11", cellphone number--></cellphoneNumber-->
</EventTriggerNotification>
```

F.238 XML_EventTriggerNotificationList

EventTriggerNotificationList message in XML format

```
<EventTriggerNotificationList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <EventTriggerNotification/><!--opt, see details in the message of XML_EventTriggerNotification-->
</EventTriggerNotificationList>
```

See Also

[XML_EventTriggerNotification](#)

F.239 XML_EventTriggersCap

XML message about linkage capabilities of different alarm categories

```
<EventTriggersCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <DiskfullTriggerCap><!--optional, xs: EventTriggerCapType--></DiskfullTriggerCap>
  <DiskerrorTriggerCap><!--optional, xs: EventTriggerCapType--></DiskerrorTriggerCap>
  <NicbrokenTriggerCap><!--optional, xs: EventTriggerCapType--></NicbrokenTriggerCap>
  <IpconflictTriggerCap><!--optional, xs: EventTriggerCapType--></IpconflictTriggerCap>
  <IllaccesTriggerCap><!--optional, xs: EventTriggerCapType--></IllaccesTriggerCap>
  <BadvideoTriggerCap><!--optional, xs: EventTriggerCapType--></BadvideoTriggerCap>
  <VideomismatchTriggerCap><!--optional, xs: EventTriggerCapType--></VideomismatchTriggerCap>
  <IOTriggerCap><!--optional, xs: EventTriggerCapType--></IOTriggerCap>
  <LineDetectTriggerCap><!--optional, xs: EventTriggerCapType--></LineDetectTriggerCap>
  <RegionEntranceTriggerCap><!--optional, xs: EventTriggerCapType--></RegionEntranceTriggerCap>
  <RegionExitingTriggerCap><!--optional, xs: EventTriggerCapType--></RegionExitingTriggerCap>
  <LoiteringTriggerCap><!--optional, xs: EventTriggerCapType--></LoiteringTriggerCap>
  <GroupDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></GroupDetectionTriggerCap>
  <RapidMoveTriggerCap><!--optional, xs: EventTriggerCapType--></RapidMoveTriggerCap>
  <ParkingTriggerCap><!--optional, xs: EventTriggerCapType--></ParkingTriggerCap>
  <UnattendedBaggageTriggerCap><!--optional, xs: EventTriggerCapType--></UnattendedBaggageTriggerCap>
  <AttendedBaggageTriggerCap><!--optional, xs: EventTriggerCapType--></AttendedBaggageTriggerCap>
  <FireDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></FireDetectionTriggerCap>
  <FireDetectionCap><!--optional, xs: EventTriggerCapType--></FireDetectionCap>
  <StorageDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></StorageDetectionTriggerCap>
  <ShipsDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></ShipsDetectionTriggerCap>
  <ThermometryCap><!--optional, xs: EventTriggerCapType--></ThermometryCap>
  <VandalProofTriggerCap><!--optional, xs: EventTriggerCapType--></VandalProofTriggerCap>
  <BlackListTriggerCap><!--opt, xs: EventTriggerCapType, configuration capability of blocklist arming linkage--></BlackListTriggerCap>
  <WhiteListTriggerCap><!--opt, xs: EventTriggerCapType, configuration capability of allowlist arming linkage--></WhiteListTriggerCap>
```

```
<AllVehicleListTriggerCap><!--optional, xs: EventTriggerCapType, configuration capability of other list arming linkage-->
</AllVehicleListTriggerCap>
<OtherVehicleListTriggerCap><!--optional, xs: EventTriggerCapType--></OtherVehicleListTriggerCap>
<PeopleDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></PeopleDetectionTriggerCap>
<PIRALarmCap><!--optional, xs: EventTriggerCapType--></PIRALarmCap>
<TamperDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></TamperDetectionTriggerCap>
<DefocusDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></DefocusDetectionTriggerCap>
<FaceDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></FaceDetectionTriggerCap>
<SceneChangeDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></SceneChangeDetectionTriggerCap>
<VandalProofAlarmCap><!--optional, xs: EventTriggerCapType--></VandalProofAlarmCap>
<JudgmentTriggerCap><!--optional, xs: EventTriggerCapType--></JudgmentTriggerCap>
<FightingTriggerCap><!--optional, xs: EventTriggerCapType--></FightingTriggerCap>
<RisingTriggerCap><!--optional, xs: EventTriggerCapType--></RisingTriggerCap>
<DozingTriggerCap><!--optional, xs: EventTriggerCapType--></DozingTriggerCap>
<CountingTriggerCap><!--optional, xs: EventTriggerCapType--></CountingTriggerCap>
<VideoLossTriggerCap><!--optional, xs: EventTriggerCapType--></VideoLossTriggerCap>
<HideTriggerCap><!--optional, xs: EventTriggerCapType--></HideTriggerCap>
<AlarmInTriggerCap><!--optional, xs: EventTriggerCapType--></AlarmInTriggerCap>
<VehicleDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></VehicleDetectionTriggerCap>
<AudioExceptionCap><!--optional, xs: EventTriggerCapType--></AudioExceptionCap>
<FiledDetectTriggerCap><!--optional, xs: EventTriggerCapType--></FiledDetectTriggerCap>
<MotionDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></MotionDetectionTriggerCap>
<TemperatureCap><!--optional, xs: EventTriggerCapType--></TemperatureCap>
<IntelligentTriggerCap><!--optional, xs: EventTriggerCapType--></IntelligentTriggerCap>
<FaceContrastTriggerCap><!--optional, xs: EventTriggerCapType, face picture comparison alarm linkage--></FaceContrastTriggerCap>
<PersonDensityDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></PersonDensityDetectionTriggerCap>
<PersonQueueDetectionTriggerCap><!--optional, xs: EventTriggerCapType, queue management alarm linkage--></PersonQueueDetectionTriggerCap>
<HumanRecognitionTriggerCap><!--optional, xs: EventTriggerCapType--></HumanRecognitionTriggerCap>
<FaceSnapTriggerCap><!--optional, xs: EventTriggerCapType--></FaceSnapTriggerCap>
<isSupportWhiteLightAction>
  <!--dependent, xs: boolean, see details in EventTriggerCapType, it is valid when isSupportWhiteLight is "true"-->
</isSupportWhiteLightAction>
<isSupportAudioAction>
  <!--dependent, xs: boolean, see details in EventTriggerCapType, it is valid when isSupportBeep is "true"-->
</isSupportAudioAction>
<HFPDTriggerCap><!--optional, xs: EventTriggerCapType--></HFPDTriggerCap>
<MixedTargetDetectionCap><!--optional, xs: EventTriggerCapType--></MixedTargetDetectionCap>
<HVTVehicleDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></HVTVehicleDetectionTriggerCap>
<VCATriggerCap><!--optional, xs: EventTriggerCapType--></VCATriggerCap>
<PIRCap><!--optional, xs: EventTriggerCapType--></PIRCap>
<IllegalParkingTriggerCap><!--optional, xs: EventTriggerCapType, whether it supports illegal parking detection--></IllegalParkingTriggerCap>
<PedestrianTriggerCap><!--optional, xs: EventTriggerCapType, whether it supports pedestrian detection--></PedestrianTriggerCap>
<TrafficAccidentTriggerCap><!--optional, xs: EventTriggerCapType, whether it supports traffic accident detection--></TrafficAccidentTriggerCap>
<ConstructionTriggerCap><!--optional, xs: EventTriggerCapType, whether it supports construction detection--></ConstructionTriggerCap>
<RoadBlockTriggerCap><!--optional, xs: EventTriggerCapType, whether it supports roadblock detection--></RoadBlockTriggerCap>
```

```
<AbandonedObjectTriggerCap><!--optional, xs: EventTriggerCapType, whether it supports objects dropped down detection--></AbandonedObjectTriggerCap>
<ParallelParkingTriggerCap><!--optional, xs: EventTriggerCapType, whether it supports parallel parking detection--></ParallelParkingTriggerCap>
<ParkingStateTriggerCap><!--optional, xs: EventTriggerCapType, whether it supports parking space status detection, currently this node is not supported--></ParkingStateTriggerCap>
<CongestionTriggerCap><!--optional, xs: EventTriggerCapType, whether it supports congestion detection--></CongestionTriggerCap>
<IntersectionAnalysisCap><!--optional, xs: EventTriggerCapType, whether it supports intersection analysis--></IntersectionAnalysisCap>
<ShipsFlowDetectionTriggerCap><!--optional, xs: EventTriggerCapType, ship flow detection--></ShipsFlowDetectionTriggerCap>
<dredgerDetectionTriggerCap><!--optional, xs: EventTriggerCapType, dredger detection--></dredgerDetectionTriggerCap>
<voltageInstableTriggerCap><!--optional, xs: EventTriggerCapType, supply voltage exception--></voltageInstableTriggerCap>
<HighHDTemperatureTriggerCap><!--optional, xs: EventTriggerCapType, HDD high temperature detection--></HighHDTemperatureTriggerCap>
<LowHDTemperatureTriggerCap><!--optional, xs: EventTriggerCapType, HDD low temperature detection--></LowHDTemperatureTriggerCap>
<HDImpactTriggerCap><!--optional, xs: EventTriggerCapType, HDD impact detection--></HDImpactTriggerCap>
<HDBadBlockTriggerCap><!--optional, xs: EventTriggerCapType, HDD bad sector detection--></HDBadBlockTriggerCap>
<SevereHDFailureTriggerCap><!--optional, xs: EventTriggerCapType, HDD severe fault detection--></SevereHDFailureTriggerCap>
<HUMANATTRIBUTECap><!--optional, xs: EventTriggerCapType--></HUMANATTRIBUTECap>
<HumanAttributeTriggerCap><!--optional, xs: EventTriggerCapType, human body attribute--></HumanAttributeTriggerCap>
<BlackListFaceContrastTriggerCap><!--opt, xs: EventTriggerCapType, alarm linkage capability of blocklist face comparison--></BlackListFaceContrastTriggerCap>
<FaceLibTriggerCap><!--optional, xs: EventTriggerCapType--></FaceLibTriggerCap>
<SafetyHelmetDetectionTriggerCap><!--optional, xs: EventTriggerCapType, alarm linkage capability of hard hat detection--></SafetyHelmetDetectionTriggerCap>
<VibrationDetectionTriggerCap><!--optional, xs: EventTriggerCapType, alarm linkage capability of vibration detection--></VibrationDetectionTriggerCap>
<RadarLineDetectionTriggerCap><!--optional, xs: EventTriggerCapType, alarm linkage capability of radar line crossing detection--></RadarLineDetectionTriggerCap>
<RadarFieldDetectionTriggerCap><!--optional, xs: EventTriggerCapType, alarm linkage capability of radar intrusion detection--></RadarFieldDetectionTriggerCap>
<HBDLibTriggerCap><!--optional, xs: EventTriggerCapType, alarm linkage capability of human body picture library--></HBDLibTriggerCap>
<FaceThermometryCap><!--optional, xs: EventTriggerCapType--></FaceThermometryCap>
<NoMaskDetectionTriggerCap><!--optional, xs: EventTriggerCapType, alarm linkage capability of no wearing mask detection--></NoMaskDetectionTriggerCap>
<TMPATriggerCap><!--optional, xs: EventTriggerCapType, alarm linkage capability of temperature measurement pre-alarm--></TMPATriggerCap>
<FireEscapeDetectionTriggerCap><!--optional, xs: EventTriggerCapType, alarm linkage capability of fire engine access detection--></FireEscapeDetectionTriggerCap>
<TakingElevatorDetectionTriggerCap><!--optional, xs: EventTriggerCapType, alarm linkage capability of elevator detection--></TakingElevatorDetectionTriggerCap>
<RuleTriggerCap><!--optional, linkage capability of rule triggered alarm -->
<isSupportCityManagement>
```

```
<!--optional, xs:boolean, whether the city management supports setting linkage actions by area; if supports, the  
value is true, otherwise, this node will not be returned-->  
</isSupportCityManagement>  
</RuleTriggerCap>  
<ThermalCalibrationFileExceptionCap><!--optional, xs:EventTriggerCapType, alarm linkage capability of  
thermography calibration file exception--></ThermalCalibrationFileExceptionCap>  
</EventTriggersCap>
```

See Also

[XML_EventTriggerCapType](#)

F.240 XML_externSecurityCap

externSecurityCap message in XML format

```
<externSecurityCap>  
  <RestAdminPassWord>  
    <isSupportWithSecurityQuestion>  
      <!--opt, xs: boolean, whether supports resetting password by answering security questions, "true,false" -->  
    </isSupportWithSecurityQuestion>  
    <isSupportWithGUIDFileData>  
      <!-- opt, xs: boolean, whether supports resetting password by importing GUID file, "true,false" -->  
    </isSupportWithGUIDFileData>  
    <isSupportWithSecurityEmail>  
      <!-- opt, xs: boolean, whether supports resetting password by setting recovery email, "true,false" -->  
    </isSupportWithSecurityEmail>  
  </RestAdminPassWord>  
  <SecurityLimits><!--opt-->  
    <LoginPasswordLenLimit min="1" max="16">  
      <!--opt, minimum and maximum lengths of login password-->  
    </LoginPasswordLenLimit>  
    <SecurityAnswerLenLimit min="1" max="128">  
      <!--opt, minimum and maximum answer length of security questions-->  
    </SecurityAnswerLenLimit>  
  </SecurityLimits>  
</externSecurityCap>
```

F.241 XML_EZVIZ

XML message about the EZVIZ access configuration parameters

```
<EZVIZ version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  
  <enabled><!--required, xs:boolean--></enabled>  
  <registerStatus><!--read-only, optional, xs:boolean--></registerStatus>  
  <redirect><!--optional, xs:boolean, whether to allow the device to redirect the server address--></redirect>  
  <serverAddress><!--optional-->  
    <addressingFormatType>  
      <!--required, xs:string, "ipaddress,hostname"-->
```

```

</addressingFormatType>
<hostName><!--dependent, xs:string--></hostName>
<ipAddress><!--dependent, xs:string--></ipAddress>
<ipv6Address><!--dependent, xs:string--></ipv6Address>
</serverAddress>
<verificationCode>
    <!--optional, xs:string, verification code. Only the "admin" user can edit the verification code using the control. The verification code will be displayed for other users. If this node is returned, setting verification code is supported-->
</verificationCode>
<offlineStatus>
    <!--read-only, dependent, xs:string, it is valid when registerStatus values "false", device offline status:>
    "secretKeyInvalid"-invalid verification code, "netUnreachable"-network is unreachable, "blocklist"-blocklist,
    "unknown"-unknown error-->
</offlineStatus>
<enabledTiming>
    <!--optional, xs:boolean, whether to enable Hik-Connect timing: true-yes, false-no (default); this node is valid only when the value of parameter <b>platformType

```

F.242 XML_FocusConfiguration

FocusConfiguration message in XML format

```

<FocusConfiguration version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <focusStyle/><!--req, xs:string, focus type, "AUTO, MANUAL, SEMIAUTOMATIC"-->
    <focusLimited/><!--opt, xs:integer, the minimum focus length, unit: cm-->
    <focusPosition/><!--dep, depends on "FocusStyle", xs:integer-->
    <focusSpeed><!--opt, xs:intger--></focusSpeed>

```

```
<focusSensitivity>
  <!--opt, xs:intger, focus sensitivity, ranges from 0 to 2, it is valid when "focusStyle" is "MANUAL" or
"SEMIAUTOMATIC"-->
</focusSensitivity>
<temperatureChangeAdaptEnabled><!--opt, xs:boolean, "true,false"--></temperatureChangeAdaptEnabled>
<relativeFocusPos>
  <!--opt, xs:intger, relative focus value, it is valid when "focusStyle" is "MANUAL" or "SEMIAUTOMATIC"-->
</relativeFocusPos>
<highTemperaturePriority><!--opt, xs:boolean, whether to enable high temperature priority mode--></
highTemperaturePriority>
<focusRange><!--optional, xs:string, focus adjustment range: min (the minimum), medium (the medium), max (the
maximum)--></focusRange>
</FocusConfiguration>
```

F.243 XML_FTPNotification

XML message about the parameters of a specific FTP

```
<FTPNotification version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id>
  <!--required, xs:string, FTP ID-->
</id>
<enabled>
  <!--required, xs:boolean, whether to enable the FTP server-->
</enabled>
<useSSL>
  <!--optional, xs:boolean-->
</useSSL>
<addressingFormatType opt="ipaddress,hostname">
  <!--required, xs:string, this node can only be set to "ipaddress"-->
</addressingFormatType>
<hostName>
  <!--dependent, xs:string-->
</hostName>
<ipAddress>
  <!--dependent, xs:string-->
</ipAddress>
<ipv6Address>
  <!--dependent, xs:string-->
</ipv6Address>
<portNo>
  <!--optional, xs:integer, FTP port No.-->
</portNo>
<userName>
  <!--required, xs:string, user name-->
</userName>
<password>
  <!--wo, xs:string, password-->
</password>
<passiveModeEnabled>
```

```

<!--optional, xs:boolean-->
</passiveModeEnabled>
<annoyftp>
  <!--optional, xs:boolean-->
</annoyftp>
<uploadPicture>
  <!--optional, xs:boolean-->
</uploadPicture>
<uploadVideoClip>
  <!--optional, xs:boolean-->
</uploadVideoClip>
<uploadPath>
  <!--req-->
<pathDepth>
  <!--required, xs:integer, directory levels, up to 16 levels can be supported-->
</pathDepth>
<topDirNameRule>
  <!--dependent, xs:string, parent directory name: "none", "devName"-device name, "devId"-device ID, "devIp"-device IP address, "positionInfo"-camera 1, "time_month"-usage date (YYYY-MM), "time_date"-usage date (YYYY-MM-DD), "illegalType"-violation type, "direction"-direction, "site"-place, "chanName"-channel name, "chanId"-channel No., "lanId"-lane No., "customize"-custom, "time", "buildUnitNo"-building No. and unit No. This node is set to NULL by default-->
  </topDirNameRule>
  <!--dependent, xs:string, custom string for parent directory, the maximum string length is 32 bytes. This node is valid when <b><topDirNameRule></b> is "customize"-->
  <subDirNameRule>
    <!--dependent, xs:string, child directory name: "none", "devName"-device name, "devId"-device ID, "devIp"-device IP address, "positionInfo"-camera 1, "time_month"-usage date (YYYY-MM), "time_date"-usage date (YYYY-MM-DD), "illegalType"-violation type, "direction"-direction, "site"-place, "chanName"-channel name, "chanId"-channel No., "lanId"-lane No., "customize"-custom, "time", "buildUnitNo"-building No. and unit No. This node is set to NULL by default-->
    </subDirNameRule>
    <!--dependent, xs:string, custom string for child directory, the maximum string length is 32 bytes. This node is valid when <b><subDirNameRule></b> is "customize"-->
    <threeDirNameRule>
      <!--dependent, xs:string, the third directory name: "none", "devName"-device name, "devId"-device ID, "devIp"-device IP address, "positionInfo"-camera 1, "time_month"-usage date (YYYY-MM), "time_date"-usage date (YYYY-MM-DD), "illegalType"-violation type, "direction"-direction, "site"-place, "chanName"-channel name, "chanId"-channel No., "lanId"-lane No., "customize"-custom, "time", "buildUnitNo"-building No. and unit No. This node is set to NULL by default-->
      </threeDirNameRule>
      <!--required, xs:string, custom string for the third directory, the maximum string length is 32 bytes. This node is valid when <b><threeDirNameRule></b> is "customize"-->
      <fourDirNameRule>
        <!--dependent, xs:string, the fourth directory name: "none", "devName"-device name, "devId"-device ID, "devIp"-device IP address, "positionInfo"-camera 1, "time_month"-usage date (YYYY-MM), "time_date"-usage date (YYYY-MM-DD), "illegalType"-violation type, "direction"-direction, "site"-place, "chanName"-channel name, "chanId"-channel No., "lanId"-lane No., "customize"-custom, "time", "buildUnitNo"-building No. and unit No. This node is set to NULL by default-->
        </fourDirNameRule>
        <!--required, xs:string, custom string for the fourth directory, the maximum string length is 32 bytes. This node is valid when <b><fourDirNameRule></b> is "customize"-->

```

```
</uploadPath>
<picArchivingInterval>
    <!--optional, xs:integer, the value is between 1 and 30, 0-close-->
</picArchivingInterval>
<picNameRuleType>
    <!--optional, xs:string, picture name rule type: "default, prefix"-->
</picNameRuleType>
<picNamePrefix>
    <!--dependent, xs:string, prefix length of the picture name, which is between 0 and 32-->
</picNamePrefix>
<ftpPicNameRuleType>
    <!--required, xs:string, type of FTP picture name rule: "videoIntercom"-rule used by video intercom products, "ITC"-rule used by traffic cameras-->
</ftpPicNameRuleType>
<FTPPicNameRule>
    <!--dependent, picture name rule of a specific FTP-->
    <ItemList/><!--required, see details in the message of XML_List-->
    <delimiter>
        <!--required, xs:string, delimiter, which is a single character and the default value is "_"-->
    </delimiter>
    <customStr>
        <!--required, xs:string, custom string, its length is between 1 and 128-->
    </customStr>
</FTPPicNameRule>
<upDataType>
    <!-- optional, xs:integer, picture uploading type: 0-all, 1-checkpoint, 2-violation. When only one FTP server is enabled, this node can only be set to 0. When two FTP servers are both enabled, you should set 1 for one FTP server and set 2 for another FTP server, which means that two FTP servers cannot be set to the same type-->
</upDataType>
<uploadPlateEnable>
    <!--optional, xs:boolean, whether to enable uploading license plate thumbnail-->
</uploadPlateEnable>
<site>
    <!--required, xs:string, place, the maximum string length is 128 bytes-->
</site>
<roadNum>
    <!--required, xs:string, intersection No., the maximum string length is 32 bytes-->
</roadNum>
<instrumentNum>
    <!--required, xs:string, device No., the maximum string length is 32 bytes-->
</instrumentNum>
<direction>
    <!--required, xs:string, direction No., the maximum string length is 32 bytes-->
</direction>
<directionDesc>
    <!--required, xs:string, direction description, the maximum string length is 32 bytes-->
</directionDesc>
<monitoringInfo1>
    <!--required, xs:string, camera 1 information, the maximum string length is 44 bytes-->
</monitoringInfo1>
<uploadAttachedInfomation>
    <!--required, xs:boolean, whether to upload additional information-->
```

```
</uploadAttachedInfo>
<brokenNetHttp><!--optional, xs:boolean, whether to enable ANR (automatic network replenishment)--></brokenNetHttp>
<uploadProtocolType><!--optional, xs:string, type of the uploaded protocol: "FTP", "SFTP"--></uploadProtocolType>
</FTPNotification>
```

See Also

[XMLItemList](#)

F.244 XML_hdd

XML message about HDD parameters

```
<hdd version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, read-only, xs: string; ID--></id>
  <hddName><!--required, read-only, xs: string--></hddName>
  <hddPath><!--read only, opt, xs: string--></hddPath>
  <hddType><!--required, read-only, xs: string, "IDE, SATA, eSATA, NFS, iSCSI, Virtual Disk"--></hddType>
  <status>
    <!--required, read only, xs: string, HDD status:
    "ok,unformatted,error,idle,mismatch,offline,smartFailed,reparing,formatting,notexist,unRecordHostFormatted,synching
    ,syncError,unloaded". "unRecordHostFormatted"-unformatted in education sharing system, "synching"-synchronizing,
    "syncError"-synchronization exception-->
  </status>
  <capacity><!--required, read-only, xs: float, unit: MB--></capacity>
  <freeSpace><!--required, read-only, xs: float, unit: MB--></freeSpace>
  <property><!--required, xs: string, HDD properties: "RW,RO,Redund"--></property>
  <group><!--optional, xs: string; HDD group ID--></group>
  <reservedDayCfg><!--optional, xs:integer, video saving duration, it is between 0 and 1000, unit: day--></
reservedDayCfg>
  <DataModeList><!--opt, ro, current HDD allocation mode-->
    <DataMode>
      <type><!--required, xs: string, storage application type: "recordStorage"-video storage, "pictureCloudStorage"-picture to be saved in cloud storage, "fileStorage"-file storage--></type>
      <occupancyRate><!--required, xs: integer, HDD usage, range: [0,100]--></occupancyRate>
    </DataMode>
  </DataModeList>
  <formatType>
    <!--optional, read only, xs: string, formatting type: FAT32 (default) and EXT4, this node is only available for SD card; if
this node does not exist, the default formatting type is FAT32-->
  </formatType>
  <encryptionStatus><!--optional, read only, xs:string, encryption status: "unencrypted", "encrypted", "verifyFailed"-verification failed--></encryptionStatus>
</hdd>
```

F.245 XML_hddList

XML message about HDD list parameters

```
<hddList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0" >
<hdd>
<id>
  <!--read-only, required, xs:string, HDD ID-->
</id>
<hddName>
  <!--read-only, required, xs:string-->
</hddName>
<hddPath>
  <!--read-only, optional, xs:string-->
</hddPath>
<hddType>
  <!--read-only, required, xs:string, "IDE,SATA,eSATA,NFS,iSCSI,Virtual Disk,NTFS",etc-->
</hddType>
<status>
  <!--read-only, required,
xs:string,"ok,unformatted,error,idle,mismatch,offline,smartFailed,reparing,formating,notexist,unRecordHostFormatted
,maintaining,maintainingPercent,synching,syncError,unloaded"-->
</status>
<capacity>
  <!--read-only, required, xs:float,in MB-->
</capacity>
<freeSpace>
  <!--read-only, required, xs:float,in MB-->
</freeSpace>
<property>
  <!--required, xs:string "RW,read-only, Redund"-->
</property>
<group>
  <!--optional, xs:string; id-->
</group>
<DataModeList>
  <!--optional,ro-->
<DataMode>
  <!--list-->
<type>
  <!--required, xs:string,opt="recordStorage,pictureCloudStorage,fileStorage"-->
</type>
<occupancyRate>
  <!--required, xs:integer [0,100]-->
</occupancyRate>
</DataMode>
</DataModeList>
<formatType>
  <!--read-only, optional, xs:string; "FAT32,EXT4"-->
</formatType>
<ipAddress>
```

```
<!--optional, xs:string-->
</ipAddress>
<ipv6Address>
  <!--optional, xs:string -->
</ipv6Address>
<formatType>
  <!--read-only, optional, xs:string; "FAT32,EXT4"-->
</formatType>
<encryptionStatus>
  <!--read-only, optional, xs:string, encryption status: "unencrypted"-unencrypted, "encrypted"-encrypted,
"verfyFailed"-verification failed-->
</encryptionStatus>
<logStorageEnable>
  <!--read-only, optional, xs:boolean, whether it is a log HDD-->
</logStorageEnable>
</hdd>
</hddList>
```

F.246 XML_IllegalLoginLock

IllegalLoginLock message in XML format

```
<IllegalLoginLock version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<enabled>true</enabled>
<maxIllegalLoginTimes>
  <!--opt, xs: integer, maximum illegal login attempts, whose value is between 3 and 20, and the default value is 5-->
</maxIllegalLoginTimes>
<maxIllegalLoginLockTime><!--optional, xs:integer, lock duration when maximum illegal login attempts reached the
upper limit, range:[0,120], unit: minute--></maxIllegalLoginLockTime>
</IllegalLoginLock>
```

F.247 XML_ImageChannel

ImageChannel message in XML format

```
<ImageChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id><!--req, xs: integer--></id>
<enabled><!--req, xs: boolean--></enabled>
<videoInputID><!--req, xs: integer--></videoInputID>
<Defog/><!--opt, electronic defogging parameters, see details in the message of XML_Defog-->
<NoiseReduce2D/><!--opt, 2D noise reduction parameters, see details in the message of XML_NoiseReduce2D-->
<Focusconfiguration/><!--opt-->
<LensInitialization/><!--opt-->
<ImageFlip/><!--opt, image automatic flipping parameters, see details in the message of XML_ImageFlip-->
<ImageFreeze/><!--opt-->
<proportionalpan/><!--opt-->
<WDR/><!--opt, WDR parameters, refer to the message XML_WDR for details-->
<BLC/><!--opt, BLC parameters, refer to the message XML_BLC for details-->
<NoiseReduce/><!--opt, 3D DNR parameters, see details in the message of XML_NoiseReduce-->
```

```
<ImageEnhancement/><!--opt, image enhancement parameters, see details in the message of XML_ImageEnhancement-->
<DSS/><!--opt, low illumination electronic shutter parameters in exposure, see details in the message of XML_DSS-->
<WhiteBlance/><!--opt, WB parameters, see details in the message of XML_WhiteBlance-->
<Exposure/><!--opt, exposure parameters, see details in the message of XML_Exposure-->
<Sharpness/><!--opt, sharpness parameters, see details in the message of XML_Sharpness-->
<gammaCorrection/><!--opt, gamma correction parameters, see details in the message of XML_gammaCorrection-->
<powerLineFrequency/><!--opt, image standard parameters, refer to the message XML_powerLineFrequency for details-->
<Color/><!--opt, image color parameters, see details in the message of XML_Color-->
<IrcutFilter/><!--opt, day/night auto switch parameters, see details in the message of XML_IrxutFilter-->
<ImageModeList/><!--opt, default image mode parameters, see details in the message of XML_ImageModeList-->
<BrightEnhance/><!--opt, brightness enhancement parameters, see details in the message of XML_BrightEnhance-->
<ISPMode/><!--opt, day/night mode parameters, see details in the message of XML_ISPMode-->
<Shutter/><!--opt, shutter parameters in exposure, see details in the message of XML_Shutter-->
<Gain/><!--opt, gain parameters, see details in the message of XML_Gain-->
<ImageIcrcE/><!--opt, IR-cut filter parameters, see details in the message of XML_ImageIcrcE-->
<ImageMultishut/><!--opt, multi-shutter parameters, see details in the message of XML_ImageMultishut-->
<PlateBright/><!--opt, license plate brightness compensation parameters, see details in the message of XML_PlateBright-->
<JPEGParam/><!--opt, JPEG picture size parameters, see details in the message of XML_JPEGParam-->
<DarkEnhance/><!--opt, dark space enhancement parameters, see details in the message of XML_DarkEnhance-->
<Hdr/><!--opt, WDR (Wide Dynamic Range) parameters, see details in the message of XML_Hdr-->
<LSE/><!--opt, contrast enhancement parameters, see details in the message of XML_LSE-->
<MCE/><!--opt, memory color enhancement parameters, see details in the message of XML_MCE-->
<Svce/><!--opt, part contrast parameters, see details in the message of XML_Svce-->
<SectionCtrl/><!--opt, configuration parameters of picture exposure control by video segment, see details in the message of XML_SectionCtrl-->
<AutoContrast/><!--opt, automatic contrast parameters, see details in the message of XML_AutoContrast-->
<GrayRange/><!--opt, grayscale range parameters, see details in the message of XML_GrayRange-->
<LSEDetail/><!--opt, contrast enhancement parameters, see details in the message of XML_LSE-->
<ITCImageSnap/><!--opt, captured picture parameters, see details in the message of XML_ITCImageSnap-->
<ImageRecord/><!--opt, image parameters in the recorded video, see details in the message of XML_ImageRecord-->
<Scene/><!--opt-->
<EPTZ/><!--opt-->
<EIS/><!--opt-->
<HLC/><!--opt-->
<ZoomLimit/><!--opt-->
<corridor/><!--opt, image rotate mode parameters, refer to the message XML_corridor for details-->
<Dehaze/><!--opt, defog mode parameters, refer to the message XML_Dehaze for details-->
<ImageMode/><!--opt, xs: string, image mode: "standard, indoor, outdoor, dimLight"-->
<enableImageLossDetection><!--opt, xs: boolean--></enableImageLossDetection>
<CaptureMode/><!--opt, video input mode parameters, refer to the message XML_CaptureMode for details-->
<IrLight/><!--opt-->
<LensDistortionCorrection/><!--opt-->
<ExposureSync/><!--opt, exposure synchronization settings-->
<BrightnessSuddenChangeSuppression/><!--opt-->
<TempRange/><!--opt, temperature range, refer to the message XML_tempRange for details-->
<SupplementLight/> <!-- opt -->
</ImageChannel>
```

F.248 XML_Integrate

Message about the access protocol parameters in XML format.

```
<Integrate version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<CGI>
<enable opt="true, false"><!--optional, xs:boolean--></enable>
<certificateType opt="digest,digest/baisc" def="digest"><!--required, xs:string--></certificateType>
</CGI>
<ONVIF>
<enable opt="true, false"><!--optional, xs:boolean--></enable>
<certificateType opt="digest,digest/WSSE,WSSE" def="digest/WSSE"><!--req, xs:string --></certificateType>
</ONVIF>
<ISAPI>
<enable opt="true, false"><!--optional, xs:boolean, "true,false"--></enable>
</ISAPI>
<IPCAddStatus><!--display the network camera adding status, which indicates whether the local network camera has been added to the NVR-->
<enable opt="true, false"><!--optional, xs:boolean, "true,false"--></enable>
</IPCAddStatus>
<Bonjour><!-- bonjour service-->
<enable opt="true, false"><!--optional, xs:boolean, "true,false"--></enable>
</Bonjour>
<Megaeyes><!--Mega Eyes service-->
<enable opt="true, false"><!--optional, xs:boolean, "true,false"--></enable>
</Megaeyes>
<WebSocket>
<enable><!--optional, xs:boolean, "true,false"--></enable>
</WebSocket>
<WebSocketS>
<enable><!--optional, xs:boolean, "true,false"--></enable>
</WebSocketS>
</Integrate>
```

F.249 XML_IntelliCap

XML message about intelligent capability

```
<IntelliCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<isFaceSupport><!--optional, xs:boolean, whether it supports face detection--></isFaceSupport>
<isBehaviorSupport><!--optional, xs:boolean, whether it supports behavior analysis--></isBehaviorSupport>
<isLineDetectionSupport><!--optional, xs:boolean, whether it supports line crossing detection--></isLineDetectionSupport>
<isFieldDetectionSupport><!--optional, xs:boolean, whether it supports intrusion detection--></isFieldDetectionSupport>
<isRegionEntranceSupport><!--optional, xs:boolean, whether it supports region entrance detection--></isRegionEntranceSupport>
<isRegionExitingSupport><!--optional, xs:boolean, whether it supports region exiting detection--></isRegionExitingSupport>
```

```
<isLoiteringSupport><!--optional, xs:boolean, whether it supports loitering detection--></isLoiteringSupport>
<isGroupSupport><!--optional, xs:boolean, whether it supports people gathering detection--></isGroupSupport>
<isRapidMoveSupport><!--optional, xs:boolean, whether it supports fast moving detection--></
isRapidMoveSupport>
<isParkingSupport><!--optional, xs:boolean, whether it supports parking detection--></isParkingSupport>
<isUnattendedBaggageSupport><!--optional, xs:boolean, whether it supports unattended baggage detection--></
isUnattendedBaggageSupport>
<isAttendedBaggageSupport><!--optional, xs:boolean, whether it supports object removal detection--></
isAttendedBaggageSupport>
<isTeacherSupport><!--optional, xs:boolean, whether it supports teacher's behavior detection--></
isTeacherSupport>
<isStudentSupport><!--optional, xs:boolean, whether it supports student's behavior detection--></
isStudentSupport>
<isCombinedSupport><!--optional, xs:boolean, whether it supports combined detection--></isCombinedSupport>
<isTrafficSupport><!--optional, xs:boolean, whether it supports traffic detection--></isTrafficSupport>
<RestoreLib>
  <libName opt="studentsStoodUp,peopleCounting,shipDetection"><!--optional, xs:string,--></libName>
</RestoreLib>
<RestoreDefParamForbid>
  <!--optional, xs:boolean "true, false", forbid restoring algorithm library to default. When the node does not exist, it
indicates restoring to default is not supported. When the node exists and its value is "ture", it indicates support, when
the value is "false", it indicates not support-->
</RestoreDefParamForbid>
<RestoreAlgLibParam>
  <!--optional, the capabilities of restoring algorithm library parameters according to algorithm library name-->
  <libName opt="faceSnap,HMS,behaviorAnalysis"><!--optional, xs:string--></libName>
</RestoreAlgLibParam>
<isFaceCaptureStatisticsSupport><!--whether it supports face picture statistics--></isFaceCaptureStatisticsSupport>
<isSupportPersonQueueDetection><!--whether it supports queue management--></
isSupportPersonQueueDetection>
<isSupportIntersectionAnalysis>
  <!--optional, xs: boolean, whether it supports intersection analysis-->
</isSupportIntersectionAnalysis>
<isSupportCityManagement><!--optional, xs:boolean, whether it supports intelligent city management. Refer to
URI: /ISAPI/Intelligent/channels/<ID>/cityManagement/capabilities?format=json--></isSupportCityManagement>
<mixedTargetDetectionWithoutAttribute><!--optional, xs: boolean--></mixedTargetDetectionWithoutAttribute>
<isSupportUploadFacePictureByForm><!--optional, xs:boolean, whether it supports uploading face pictures by form--></isSupportUploadFacePictureByForm>
<isSupportUploadFacePictureByUrl><!--optional, xs: boolean, whether it supports uploading face pictures by URL--></isSupportUploadFacePictureByUrl>
<isSupportUploadHumanPictureByForm><!--optional, xs:boolean, whether it supports uploading human pictures in
form--></isSupportUploadHumanPictureByForm>
<isSupportMisinfoFilterStatisticalMode><!--optioal, xs:boolean, whether it supports statistics of filtering face
comparison--></isSupportMisinfoFilterStatisticalMode>
<isSupportFaceScore><!--optional, xs:boolean, whether it supports face grading configuration (camera)--></
isSupportFaceScore>
<HumanRecognitionModeSearchCap>
  <searchTargetsNumMax><!--optional, xs:integer, maximum number of sample pictures that can be imported for
searching by picture--></searchTargetsNumMax>
  <HumanMode>
    <searchCond opt="age_group,gender,jacet_color,glass,bag,ride,unlimit"/><!--setting multiple search conditions is
supported-->
```

```

<similarity min="0.0" max="100.0"/><!--optional, xs:float, similarity, range: [0.0,100.0]-->
</HumanMode>
<HumanInfo><!--optional, person attribute information, it is required when searching for picture by attributes-->
<age_group opt="child,infant,kid,middle,middleAged,old,prime,teenager,unknown,young">
  <!--optional, xs:string, age group: "child,infant,kid,middle,middleAged,old,prime,teenager,unknown,young"-->
</age_group>
<gender opt="female,male,unknown"><!--optional, xs:string, gender: "female,male,unknown"--></gender>
<jacet_color
opt="black,black,blue,brown,cyan,deepBlue,deepGray,gray,green,mixture,orange,pink,purple,red,silver,unknown,white,yellow">
  <!--optional, xs:string, tops color-->
</jacet_color>
<glass opt="yes,no,unknown,sunglasses"><!--optional, xs:string, whether the person is wearing glasses--></glass>
<bag opt="yes,no,unknown"><!--optional, xs:string, whether the person is carrying a bag--></bag>
<ride opt="yes,no,unknown"><!--optional, xs:string, whether the person is riding--></ride>
</HumanInfo>
<positive opt="true,false"/><!--optional, xs:boolean, whether it is false human body recognition alarm: "true"-yes, "false"-no-->
<eventType opt="unlimit,humanRecognition">
  <!--optional, xs:string, event type: "unlimit"-no limit, "humanRecognition"-human body detection alarm-->
</eventType>
<isSupportMultiChannelSearch>
  <!--optional, xs:boolean, whether it supports multi-channel search-->
</isSupportMultiChannelSearch>
<isSupportTotalSearchResult>
  <!--optional, xs:boolean, whether it supports limiting number of results that can be obtained after a single search-->
</isSupportTotalSearchResult>
</HumanRecognitionModeSearchCap>
<VehicleRecognitionModeSearchCap>
  <searchTargetsNumMax><!--optional, xs:integer, maximum number of sample pictures that can be imported for searching by picture--></searchTargetsNumMax>
  <eventType opt="unlimit,vehicleBlackList,vehicleWhiteList"/><!--xs:string, event type: "unlimit"-no limit, "vehicleBlackList"-vehicle blocklist, "vehicleWhiteList"-vehicle allowlist-->
  <VehicleMode>
    <searchCond opt="licensePlate,vehicleLogo,vehicleSubLogoRecog,vehicleType,vehicleColor,unlimit"/>
    <similarity min="0.0" max="100.0"/><!--optional, xs:float, similarity, range: [0.0,100.0]-->
  </VehicleMode>
  <isSupportMultiChannelSearch>
    <!--optional, xs:boolean, whether it supports multi-channel search-->
  </isSupportMultiChannelSearch>
  <isSupportTotalSearchResult>
    <!--optional, xs:boolean, whether it supports limiting number of results that can be obtained after a single search-->
  </isSupportTotalSearchResult>
</VehicleRecognitionModeSearchCap>
<FaceContrastAnalyzeCap><!--capability of manual comparison and analysis of face pictures-->
<isSupportTrigger opt="false,true"><!--optional, xs:boolean--></isSupportTrigger>
<SearchCondition><!--optional-->
  <startTime><!--required, xs:time, ISO8601 time--></startTime>
  <endTime><!--required, xs:time, ISO8601 time--></endTime>
</SearchCondition>
```

```
</FaceContrastAnalyzeCap>
<FramesPeopleCountingCap><!--capability of people counting statistics in a single frame-->
<Statistics>
  <SearchCondition>
    <startTime><!--required, xs:time,ISO8601 time--></startTime>
    <endTime><!--required, xs:time,ISO8601 time--></endTime>
  </SearchCondition>
</Statistics>
<MaskRegion><!--optional, shielded region-->
  <maxRegionNum><!--optional, xs:integer, number of regions--></maxRegionNum>
  <Region>
    <vertexNum min="3" max="10"><!--optional, xs:integer, number of region vertexes--></vertexNum>
  </Region>
</MaskRegion>
</FramesPeopleCountingCap>
<FaceContrastPersonInfoExtend><!--configuration capability of face comparison tag-->
<personInfoCap>
  <maxPersonInfo min="0" max="4">
    <!--optional, xs:integer, maximum number of person tags-->
  </maxPersonInfo>
</personInfoCap>
<personInfoFDlibCap>
  <maxPersonInfo min="0" max="4">
    <!--optional, xs:integer, maximum number of person tags-->
  </maxPersonInfo>
</personInfoFDlibCap>
</FaceContrastPersonInfoExtend>
<isSupportSafetyHelmetDetection>
  <!-- opt, xs:boolean, whether it supports hard hat detection-->
</isSupportSafetyHelmetDetection>
<isSupportDistanceRecognition><!--optional, xs:boolean, whether supports distance measurement--></isSupportDistanceRecognition>
<isSupportATM><!--optional, xs:boolean, ATM intelligent configuraiton, corresponds to URI: /ISAPI/Intelligent/ATM/capabilities?format=json--></isSupportATM>
<isSupportFaceTemperature><!--optional, xs:boolean, whether the device supports face thermography, corresponds to URI: /ISAPI/Intelligent/faceTemperature/capabilities?format=json--></isSupportFaceTemperature>
<isSupportFireEscapeDetectionSearch>
  <!--optional, xs:boolean, whether the device supports searching for fire engine access detection events of multiple channels (related URI: URI/ISAPI/Intelligent/fireEscapeDetection/search/capabilities?format=json)-->
</isSupportFireEscapeDetectionSearch>
<isSupportTakingElevatorDetectionSearch>
  <!--optional, xs:boolean, whether the device supports searching for elevator detection events of multiple channels (related URI: /ISAPI/Intelligent/takingElevatorDetection/search/capabilities?format=json)-->
</isSupportTakingElevatorDetectionSearch>
<isSupportCityManagementSearch>
  <!--optional, xs:boolean, whether the device supports searching for city management events of multiple channels (related URI: /ISAPI/Intelligent/cityManagement/search/capabilities?format=json)-->
</isSupportCityManagementSearch>
<isSupportSafetyHelmetDetectionSearch>
  <!--optional, xs:boolean, whether the device supports searching for hard hat detection events of multiple channels (related URI: /ISAPI/Intelligent/safetyHelmetDetection/search/capabilities?format=json)-->
</isSupportSafetyHelmetDetectionSearch>
```

```
<isSupportFramesPeopleCountingSearch>
  <!--optional, boolean, ro, whether it supports searching for the regional people counting event (related URI: /ISAPI/Intelligent/framesPeopleCounting/search/capabilities?format=json)-->true
</isSupportFramesPeopleCountingSearch>
<isSupportChannelPollingSchedules>
  <!--optional, xs:boolean, whether the device supports channel polling configuration: "true,false"-->
</isSupportChannelPollingSchedules>
<AnalysisImageFaceCap>
  <!--optional, face picture analysis type (related URI:/ISAPI/Intelligent/analysisImage/face)-->
  <modeling opt="false"><!--optional,xs:boolean, whether the device supports modeling: "true,false"--></modeling>
</AnalysisImageFaceCap>
</IntelliCap>
```

F.250 XML_IOTTriggersCap

IOTTriggersCap message in XML format

```
<?xml version="1.0" encoding="utf-8"?>
<IOTTriggersCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <AccessController><!--opt, security control panel-->
    <XXTriggerCap><!--opt,xs: EventTriggerCapType--></XXTriggerCap>
  </AccessController>
  <VideoIntercom><!--opt, video intercom-->
    <XXTriggerCap><!--opt,xs: EventTriggerCapType --></XXTriggerCap>
  </VideoIntercom>
  <GJD><!--opt, GJD security control panel-->
    <XXTriggerCap><!--opt,xs: EventTriggerCapType--></XXTriggerCap>
  </GJD>
  <Luminite><!--opt, Luminite security control panel-->
    <XXTriggerCap><!--opt,xs: EventTriggerCapType--></XXTriggerCap>
  </Luminite>
  <OPTEX><!--opt, OPTEX security control panel-->
    <XXTriggerCap><!--opt,xs: EventTriggerCapType--></XXTriggerCap>
  </OPTEX>
  <CameraDetector><!--opt, detector-->
    <XXTriggerCap><!--opt,xs: EventTriggerCapType--></XXTriggerCap>
  </CameraDetector>
</IOTTriggersCap>
```

Remarks

The XX in the node **<XXTriggerCap>** corresponds to detailed event type. E.g., if the event type is `humanRecognition`, then the returned node is **<HumanRecognitionTriggerCap>**.

See Also

[XML_EventTriggerCapType](#)

F.251 XML_IPFilter

IPFilter message in XML format

```
<IPFilter version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<enabled><!--req, xs:boolean--></enabled>
<permissionType><!--opt, xs:string, "deny,allow"--></permissionType>
<IPFilterAddressList size = "32"/><!--opt, the character size indicates the max. number of supported IP address. See XML_IPFilterAddressList for details-->
</IPFilter>
```

See Also

[XML_IPFilterAddressList](#)

F.252 XML_IPFilterAddress

IPFilterAddress message in XML format

```
<IPFilterAddress version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id><!--req, xs:string,id--></id>
<permissionType><!--dep, xs:string, "deny,allow" --></permissionType>
<addressFilterType><!--ro, xs:string, "mask, range"--></addressFilterType>
<AddressRange><!--dep, it is valid when <addressFilterType> is "range"-->
<startIPAddress><!--dep, xs:string--></startIPAddress>
<endIPAddress><!--dep, xs:string--></endIPAddress>
<startIPv6Address><!--dep, xs:string--></startIPv6Address>
<endIPv6Address><!--dep, xs:string--></endIPv6Address>
</AddressRange>
<AddressMask><!--dep, it is valid when <addressFilterType> is "mask"-->
<ipAddress><!--dep, xs:string--></ipAddress>
<ipv6Address><!--dep, xs:string--></ipv6Address>
<bitMask><!--opt, xs:string--></bitMask>
</AddressMask>
</IPFilterAddress>
```

F.253 XML_IPFilterAddressList

IPFilterAddressList message in XML format

```
<IPFilterAddressList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<IPFilterAddress/><!--opt, see XML_IPFilterAddress for details-->
</IPFilterAddressList>
```

See Also

[XML_IPFilterAddress](#)

F.254 XML_InputProxyChannel

InputProxyChannel message in XML format

```
<?xml version="1.0" encoding="utf-8"?>
<InputProxyChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--req, xs:string, starts from 1--></id>
  <name><!--opt, xs:string--></name>
  <sourceInputPortDescriptor><!--req-->
    <adminProtocol><!--req, xs:string, "HIKVISION,SONY,ISAPI,ONVIF,..."--></adminProtocol>
    <addressingFormatType><!--req, xs:string, "ipaddress,hostname"--></addressingFormatType>
    <hostName><!--dep, xs:string, domain name--></hostName>
    <ipAddress><!--dep, xs:string, IP address--></ipAddress>
    <ipv6Address><!--dep, xs:string, IPv6 address--></ipv6Address>
    <managePortNo><!--req, xs:integer--></managePortNo>
    <srcInputPort><!--req, xs:string, channel No.--></srcInputPort>
    <userName><!--req, xs:string, user name, which should be encrypted--></userName>
    <password><!--req, wo, xs:string, password, which should be encrypted--></password>
    <streamType><!--opt, xs:string, opt="auto,tcp,udp"--></streamType>
    <deviceID><!--dep, xs:string--></deviceID>
    <deviceTypeName><!--ro, opt, xs:string, device type name--></deviceTypeName>
    <serialNumber><!--ro, opt, xs:string, device serial No.--></serialNumber>
    <firmwareVersion><!--ro, opt, xs:string, firmware version--></firmwareVersion>
    <firmwareCode><!--ro, opt, xs:string, firmware code--></firmwareCode>
  </sourceInputPortDescriptor>
  <enableAnr>
    <!--opt, xs:boolean, whether enables ANR funtion-->
  </enableAnr>
  <NVRInfo>
    <ipAddressNVR>
      <!--opt, xs:string, IP address of NVR-->
    </ipAddressNVR>
    <portNVR>
      <!--opt, xs:integer, port No. of NVR-->
    </portNVR>
    <ipcChannelNo>
      <!--opt, xs:integer, channel No. of the network camera in NVR-->
    </ipcChannelNo>
  </NVRInfo>
</InputProxyChannel>
```

F.255 XML_IpViewDevAbility

XML message about the video intercom device capability

```
<IpViewDevAbility version="2.0">
  <!--video intercom device capability-->
  <SipServerLogin><!--required, registration capability of SIP server-->
  <AutoLogin><!--required, whether to support automatic registration-->
```

```

<enable></enable>
</AutoLogin>
<loginStatus opt="registered,unregistered"/><!--required, registration status-->
<sip>LoginNameLen min="1" max="32"/><!--required, length of registered user name-->
<sip>LoginPasswordLen min="1" max="16"/><!--required, length of registered password-->
<displayNameLen min="1" max="128"/><!--required, length of displayed device name-->
<localNumber min="1" max="32"/><!--required, length of local station number-->
<loginCycle min="1" max="99"/><!--required, registration period, unit: minute-->
<serverSipPort min="1024" max="65535"/>
<localPort min="1024" max="65535"><!--optional, local port No.-->
<isNotSupportLocalPort></isNotSupportLocalPort><!--optional, whether the local port is not supported: true-yes
(the local port is not supported), this field is not supported-no (the local port is supported)-->
<domainNameLen min="" max="" /><!--optional, the meaning of this field is the same as that of the field
<b>sipServerDomaindomainNameLen

```

```
</CallAbility>
</LocalAbility>
<VideoIntercom>
    <!--video intercom, indoor station, door station, main station, door station (V series), doorphone-->
    <monitorChannelNo min="1" max="100"/><!--required, camera channel of main station or indoor station-->
    <DeviceID><!--device No.-->
        <enabled>true</enabled><!--supported device No. configuration-->
        <supportUnitType opt="" /><!--required, supported video intercom type-->
        <InDoorDevice><!--required, indoor station-->
            <floorNum min="1" max="16"/><!--required, floor No.-->
            <roomNum min="1" max="16"/><!--required, room No.-->
            <devIndex min="0" max="10"/><!--required, indoor station No.-->
        </InDoorDevice>
        <OutDoorDevice><!--required, door station/intelligent access control device-->
            <period min="1" max="16"/><!--required, community No.-->
            <buildingNum min="1" max="16"/><!--required, building No.-->
            <unitNum min="0" max="10"/><!--required, unit No.-->
            <floorNum min="1" max="16"/><!--required, floor No.-->
            <devIndex min="0" max="10"/><!--required, door station No.-->
        </OutDoorDevice>
        <ManageUnitDevice><!--required, main station-->
            <period min="1" max="16"/><!--required, community No.-->
            <devIndex min="0" max="10"/><!--required, main station No.-->
        </ManageUnitDevice>
        <OutDoorFenceDevice><!--required, outer door station-->
            <period min="1" max="16"/><!--required, community No.-->
            <devIndex min="0" max="10"/><!--required, outer door station No.-->
        </OutDoorFenceDevice>
        <VillaOutDoorDevice><!--required, door station (V series)-->
            <period min="1" max="16"/><!--required, community No.-->
            <buildingNum min="1" max="16"/><!--required, building No.-->
            <unitNum min="0" max="10"/><!--required, unit No.-->
            <floorNum min="1" max="16"/><!--required, floor No.-->
            <devIndex min="0" max="10"/><!--required, door station No.-->
        </VillaOutDoorDevice>
        <AgainDevice><!--required, doorphone--></AgainDevice>
    </DeviceID>
    <PrivilegePwd><!--permission password-->
        <pwdType opt="engineering,setupAlarm,householderUnlock,antiHijacking"/><!--password type: "engineering"-project password, "setupAlarm"-arming and disarming password, "householderUnlock"-resident unlocking password, "antiHijacking"-duress password-->
        <pwdLen min="6" max="16"/><!--password length-->
    </PrivilegePwd>
    <OperationTime><!--operation time-->
        <monitoringTime min="10" max="60"/><!--maximum live view duration, unit: second-->
        <ringTime min="15" max="60"/><!--maximum ringing duration, unit: second-->
        <messageTime min="30" max="60"/><!--maximum messaging duration, unit: second-->
        <talkTime min="90" max="120"/><!--maximum calling duration, unit: second-->
        <callForwardingTime min="0" max="20"/><!--calling forwarding timeout, unit: second-->
        <dwRingDurationTime min="30" max="60"/><!--ringing time duration, unit: second-->
    </OperationTime>
    <RelateDevice>
```

```

<outdoorUnitIP opt="ipv4,ipv6"/><!--IP address of main door station-->
<manageUnitIP opt="ipv4,ipv6"/><!--IP address of main station-->
<sipServerIP opt="ipv4,ipv6"/><!--IP address of SIP server-->
<centerIP opt="ipv4,ipv6"/><!--center IP address-->
<centerPort min="" max="" /><!--center port No.-->
<indoorUnitIP opt="ipv4,ipv6"/><!--IP address of indoor station-->
<notSupportAgainUnitIP opt="true,false"/><!--required, whether configuring doorphone IP address is not supported: "true"-yes, "false"-no-->
    <againUnitIP opt="ipv4,ipv6"/><!--doorphone IP address-->
    <notSupportOutDoorType opt="true,false"/><!--required, whether configuring main door station type is not supported: "true"-yes, "false"-no-->
        <outDoorType opt="unitOutdoor,villaOutDoor"/><!--main door station type: "unitOutdoor"-main door station (D series), "villaOutDoor"-main door station (V series)-->
            <outInConnectMode opt="sameLan,diffLan"/><!--network mode of door station and sub indoor station: "sameLan"-in the same LAN, "diffLan"-in different LANs-->
                <indoorConnectMode opt="wireless,wired"/><!--network mode of main indoor station and sub indoor station: "wireless"-wireless NIC, "wired"-wired NIC-->
            </RelateDevice>
        <NoticeData>
            <noticeTime><!--notice time--></noticeTime>
            <noticeNumberLen min="0" max="32"/><!--notice No.-->
            <noticeThemeLen min="0" max="64"/><!--notice theme-->
            <noticeDetailLen min="0" max="1024"/><!--notice details-->
            <noticeLevel opt="advertisement,propertyMgmt,alarm,notification"/><!--notice level: "advertisement"-advertisement information, "propertyMgmt"-property information, "alarm"-alarm information, "notification"-notice information-->
                <maxPicNum><!--number of pictures--></maxPicNum>
                <maxSinglePicSize><!--maximum size of a picture, unit: KB--></maxSinglePicSize>
        </NoticeData>
        <ControlGateway><!--unlock remotely-->
            <gatewayIndex min="1"/><!--access control No.-->
            <command opt="close,open"/><!--control command-->
            <controlSrc>true</controlSrc><!--operation source information-->
            <controlType opt="monitor,calling"/><!--unlocking scene type: "monitor", "calling"-->
            <lockType opt="normal,smartLock"/><!--lock type-->
            <lockID min="" max="" /><!--lock ID-->
            <password min="" max="" /><!--password length of the smart lock-->
        </ControlGateway>
        <Zone><!--zone configuration-->
            <ZoneConfig>
                <enabled><!--whether to support configuring zone parameters (alarm input parameters)--></enabled>
                <delayInParam><!--method of setting delay: "true"-the client sets the delay time by dwParam in the structure NET_DVR_ALARMIN_PARAM (this method is used by power and environment monitoring system and ATM security control panel), "false"-the client sets the delay time by wEnterDelay and wExitDelay in the structure NET_DVR_ALARMSUBSYSTEMPARAM--></delayInParam>
                    <detectorType
                    opt="panicButton,magneticContact,smokeDetector,activeInfraredDetector,passiveInfraredDetector,glassBreakDetector,vibrationDetector,dualTechnologyPirDetector,tripleTechnologyPirDetector,humidityDetector,temperatureDetector,com bustibleGasDetecto,dynamicSwitch,controlSwitch,smartLock,waterDetector,otherDetector"/><!--detector type supported by the device: "panicButton"-panic switch, "magneticContact"-magnetic contact, "smokeDetector"-smoke detector, "activeInfraredDetector"-active infrared detector, "passiveInfraredDetector"-passive infrared detector, "glassBreakDetector"-glass break detector, "vibrationDetector"-vibration detector, "dualTechnologyPirDetector"-dual

```

```
technology motion detector, "tripleTechnologyPirDetector"-triple technology detector, "humidityDetector"-humidity
detector, "temperatureDetector"-temperature detector, "combustibleGasDetecto"-gas detector, "dynamicSwitch"-follow-up switch, "controlSwitch"-control switch, "smartLock"-smart lock, "waterDetector"-water detector,
"otherDetector"-other detector type-->
<zoneType opt="instantZone,
24hourAudibleZone,delayZone,interiorWithDelayZone,keysplashZone,supervisedFireZone,perimeterZone,
24hourSlientZone,disable"/><!--zone type supported by the device-->
<LimitedDetectorType>
<Detector>
<name>smokeDetector</name>
<zoneType opt="24hourAudibleZone"/>
</Detector>
<Detector>
<name>glassBreakDetector</name>
<zoneType opt="24hourAudibleZone"/>
</Detector>
</LimitedDetectorType>
<uploadAlarmRecoveryReport>true</uploadAlarmRecoveryReport><!--whether to support report configuration
of uploading alarm recovery-->
<zoneDelayTime min="" max="" /><!--delayed zone delay-->
<sensitivity opt="10ms,250ms,500ms,750ms" /><!--sensitivity-->
<arrayBypass>true</arrayBypass><!--whether to support zone group bypass configuration-->
<moduleStatus attri="readonly" opt="offline,online"/><!--module status-->
<moduleAddress min="" max="" /><!--module address-->
<moduleChannel>true</moduleChannel><!--module channel-->
<moduleType opt="localZone, 1zoneExpander,2zoneExpander,8ZoneExpander,8sensorZoneExpander,
1Zone&Trigger"/><!--supported zone type-->
<zoneNo attri="readonly" min="" max="" /><!--zone No. which can only be obtained-->
<subsystemNo attri="readonly"><!--No. of the partition that the zone belongs to, it can only be obtained--></
subsystemNo>
<alarmType opt="open,close" default="open"/><!--required, alarm device type: "open"-remain open, "close"-remain closed-->
<InDelayTime min="0" max="255"/><!--required, entrance delay, unit: second-->
<OutDelayTime min="0" max="255"/><!--required, exiting delay, unit: second-->
</ZoneConfig>
<GetZoneList>
<enabled><!--whether to support getting zone list--></enabled>
</GetZoneList>
<ZoneArmDisarm>
<enabled><!--whether to support arming and disarming the zone--></enabled>
</ZoneArmDisarm>
<ZoneGroupBypass>
<enabled><!--whether to support zone group bypass--></enabled>
</ZoneGroupBypass>
</Zone>
<IOIn>
<IOInNo attri="readonly" min="" max="" /><!--IP input No. which can only be obtained-->
<useType opt="disabled,openDoorBtn,doorStatus,custom"/><!--purpose: "disabled", "openDoorBtn"-door exit
button, "doorStatus"-door status, "custom"-->
</IOIN>
<IOOut>
<IOOutNo attri="readonly" min="" max="" /><!--IO output No. which can only be obtained-->
```

```

<useType opt="disabled,electricLock,custom"/><!--purpose: "disabled", "electricLock"-electric lock, "custom"-->
</IOOut>
<ElevatorControl>
<elevatorNo attr="readonly" min="" max="" /><!--ro, elevator No.-->
<interfaceType opt="RS485,network"/><!--interface type: "RS485"-RS-485, "network"-->
<RS485Protocol opt="private,custom"/><!--RS-485 protocol type: "private", "custom"-->
<networkProtocol opt="private,custom"/><!--NIC protocol type: "private", "custom"-->
<serverIP opt="ipv4,ipv6"/><!--optional, IP address of the elevator control server-->
<serverPort min="" max="" /><!--optional, port No. of the elevator control server-->
</ElevatorControl>
<RS485Config>
<!--RS-485 configuration, this node will not be returned if RS-485 configuration is not supported-->
<deviceNameLength min="0" max="32"/><!--RS-485 name-->
<deviceType>true</DeviceType><!--whether to support configuring device type-->
<deviceProtocol>true</deviceProtocol><!--whether to support configuring device protocol-->
</RS485Config>
<supportDevInfo opt="true,false"/><!--required, whether to support getting complete indoor station No. This node will not be returned if this function is not supported, and this node is valid only when the device is an indoor station-->
<supportRegisterInfo opt="true,false"/><!--required, whether the door station supports getting registration information. This node will not be returned if this function is not supported, and this node is valid only when the device is an door station-->
<CallRoomConfig><!--configuration of calling resident by pressing button for door station (V series)-->
<keyNo min="" max="" /><!--villa button No.-->
<floorNo min="" max="" /><!--villa floor No.-->
<roomNo min="" max="" /><!--villa room No.-->
<callManageCenter opt="true,false"/><!--required, whether to set it to the calling management center-->
<calledName min="" max="" /><!--optional, user name to be called, it supports letters, digits, @, and dot. This node is valid in standard SIP mode-->
</CallRoomConfig>
<VideoCall><!--optional, video intercom capability-->
<enabled opt="true,false"/><!--required, whether supports starting video intercom-->
<supportCmd opt="callRequest,Cancel,Answer,Decline,Timeout,Bye,deviceCalling,clientCalling"/>
<!--required, command type supported by the device: "callRequest"-request for call, "Cancel"-cancel the call, "Answer"-answer, "Decline"-decline, "Timeout"-time out, "Bye"-end the call, "deviceCalling"-device is in call, "clientCalling"-client is in call-->
</VideoCall>
<CallerDevice><!--required, calling device information-->
<period min="1" max="16"/><!--required, community No.-->
<buildingNum min="1" max="16"/><!--required, building No.-->
<unitNum min="0" max="10"/><!--required, unit No.-->
<floorNum min="1" max="16"/><!--required, floor No.-->
<devIndex min="0" max="10"/><!--required, device No.-->
<devType min="1" max="9"/><!--required, device type-->
</CallerDevice>
<CallStatus><!--required, calling status-->
<callingStatus min="1" max="3"/><!--required, calling status-->
</CallStatus>
<EzvizDeviceInfo><!--required, EZVIZ device information-->
<deviceNum min="1" max="32"/><!--required, number of devices-->
<DeviceCfg size="16"/>
<deviceNameLen min="1" max="32"/><!--required, device name-->
<deviceType min="1" max="3"/><!--required, device type-->

```

```

<deviceID min="0" max="10"/><!--required, device No.-->
</EzvizDeviceInfo>
<VideoCallParam><!--required, signal interaction command (non-persistent connection)-->
  <cmdType min="0" max="4"/><!--required, command-->
</VideoCallParam>
<VideoIntercomStream><!--optional, video source-->
  <sourceType opt="IPC,DVR/DVS/NVR,OutDoorDevice,OutDoorFenceDevice,AgainDevice"/><!--optional, video
source type-->
    <againDeviceNumber min="" max="" /><!--optional, number of doorphones-->
</VideoIntercomStream>
<indoorDevChangeEnabled opt="true,false"/><!--required, whether to support switching between main indoor
station and sub indoor station-->
<indoorDevChangeReboot opt="true,false"/><!--required, whether the device will reboot after switching devices of
main indoor station and sub indoor station-->
</VideoIntercom>
<UploadAlarmCfg><!--configuration of video intercom alarm, it corresponds to the structure
NET_DVR_VIDEO_INTERCOM_ALARM_CFG-->
  <UploadDoorNotCloseAlarm opt="true,false"/><!--required, whether to upload alarms of unlocking the door-->
</UploadAlarmCfg>
<ZoneList size="64"><!--special zone configuration, which is the capability of distinguishing different zones according
to the zone ID-->
  <ZoneConfig>
    <zoneID min="" max="" /><!--zone ID--></zoneID>
    <enabled><!--whether to support configuring zone parameters (alarm input parameters)--></enabled>
    <delayInParam><!--method of setting delay: "true"-the client sets the delay time by dwParam in the structure
NET_DVR_ALARMIN_PARAM (this method is used by power and environment monitoring system and ATM security
control panel), "false"-the client sets the delay time by wEnterDelay and wExitDelay in the structure
NET_DVR_ALARMSUBSYSTEMPARAM--></delayInParam>
      <detectorType
opt="panicButton,magneticContact,smokeDetector,activeInfraredDetector,passiveInfraredDetector,glassBreakDetector,
vibrationDetector,dualTechnologyPirDetector,tripleTechnologyPirDetector,humidityDetector,temperatureDetector,combustibleGasDetector,dynamicSwitch,controlSwitch,smartLock,waterDetector,otherDetector"/><!--detector type
supported by the device: "panicButton"-panic switch, "magneticContact"-magnetic contact, "smokeDetector"-smoke
detector, "activeInfraredDetector"-active infrared detector, "passiveInfraredDetector"-passive infrared detector,
"glassBreakDetector"-glass break detector, "vibrationDetector"-vibration detector, "dualTechnologyPirDetector"-dual
technology motion detector, "tripleTechnologyPirDetector"-triple technology detector, "humidityDetector"-humidity
detector, "temperatureDetector"-temperature detector, "combustibleGasDetector"-gas detector, "dynamicSwitch"-follow-up
switch, "controlSwitch"-control switch, "smartLock"-smart lock, "waterDetector"-water detector,
"otherDetector"-other detector type-->
      <zoneType opt="instantZone,
24hourAudibleZone,delayZone,interiorWithDelayZone,keysplashZone,supervisedFireZone,perimeterZone,
24hourClientZone,disable"/><!--zone type supported by the device-->
      <uploadAlarmRecoveryReport><!--whether to support report configuration of uploading alarm recovery--></
uploadAlarmRecoveryReport>
        <zoneDelayTime min="" max="" /><!--delayed zone delay-->
        <sensitivity opt="10ms,250ms,500ms,750ms"/><!--sensitivity-->
        <arrayBypass><!--whether to support zone group bypass configuration--></arrayBypass>
        <moduleStatus attri="readonly" opt="offline,online"/><!--module status-->
        <moduleAddress min="" max="" /><!--module address-->
        <moduleChannel><!--module channel--></moduleChannel>
        <moduleType opt="localZone, 1zoneExpander,2zoneExpander,8ZoneExpander,8sensorZoneExpander,
1ZoneAndTrigger"/><!--supported zone type-->

```

```

<zoneNo attr="readonly" min="" max="" /><!--zone No. which can only be obtained-->
<subsystemNo attr="readonly"><!--No. of the partition that the zone belongs to, it can only be obtained--></
subsystemNo>
<alarmType opt="open,close" default="open"/><!--required, alarm device type: "open"-remain open, "close"-remain closed-->
<InDelayTime min="0" max="255"/><!--required, entrance delay, unit: second-->
<OutDelayTime min="0" max="255"/><!--required, exiting delay, unit: second-->
</ZoneConfig>
<GetZoneList>
<enabled><!--whether the device supports getting zone list--></enabled>
</GetZoneList>
<ZoneArmDisarm>
<enabled><!--whether the device supports arming and disarming the zone--></enabled>
</ZoneArmDisarm>
<ZoneGroupBypass>
<enabled><!--whether the device supports zone group bypass--></enabled>
</ZoneGroupBypass>
</ZoneList>
<NoNeedReboot>
<videoResolutionChange opt="true,false"/><!--optional, whether the device does not reboot after changing video resolution-->
<videoFrameRateChange opt="true,false"/><!--optional, whether the device does not reboot after changing video frame rate-->
</NoNeedReboot>
</IpViewDevAbility>

```

F.256 XML_ItemList

ItemList message in XML format

```

<ItemList>
<Item>
<itemID>
<!--req, xs:string, item ID, which is between 1 and 15-->
</itemID>
<itemOrder>
<!--req, xs:string, name element: "none", "devIp"-device IP address, "time", "buildUnitNo"-building No. and unit No., "outDoorDevNo"-door station, "unlockType"-unlocking type, "devName"-device name, "deviceNo"-device No., "channelName"-channel name, "channelNo"-channel No., "plateNo"-license plate number, "plateColor"-license plate color, "laneNo"-lane No., "carSpeed"-vehicle speed, "positionInfo1"-camera 1, "pictureNo"-picture No., "CarNo"-vehicle No., "speedLimit"-speed limit, "illegalCode"-violation code, "siteNo"-intersection No., "directionNo"-direction No., "carColor"-vehicle color, "platePosition"-license plate coordinates, "carType"-vehicle type, "illegalType"-violation type, "custom"-->
</itemOrder>
<itemCustomStr>
<!--req, xs:string, element custom string, which is between 1 and 32, unit: bytes. This node is valid only when <itemOrder> is "custom". Currently traffic cameras only support one custom name-->
</itemCustomStr>
</Item>
</ItemList>

```

F.257 XML_JpegCaptureAbility

JPEG picture capture capability message in XML format

```

<JpegCaptureAbility version="2.0">
  <channelNO>
    <!--required, channel No-->
  </channelNO>
  <FindPicInfo><!--required, picture information-->
    <supportFileType opt=
      "CMR,MOTION,ALARM,EDR,ALARMANDMOTION,manual,intelligentPic,pir,wlsensor,callhelp,previewScreenshot,facedetection,LineDetection,FieldDetection,scenedechangedetection,lockPlaybackScreenshot,INTELLIGENT,regionEntrance,regionExiting,loitering,group,rapidMove,parking,unattendedBaggage,attendedBaggage,VehicleDetection,HvtVehicleDetection,faceSnap evidence, vcaEventGetUp, vcaEventAdvReachHeight, vcaEventToiletTarry, unregisteredStreetVendor,illegalParking, wrongdirection,crosslane,vehicleexist,lanechange, turnround, pedestrian, roadblock, abandonedObject,fogDetection, construction, congestion, trafficAccident,parallelParking, evidence, personQueueCounting, personQueueTime, vcaEventSafetyHelmet, vibrationDetection, allType"/>
    <!--required, supported capture type: 0-scheduled capture, 1-motion detection capture, 2-alarm capture, 3-alarm or motion detection capture, 4-alarm and motion detection capture, 6-manual capture, 9-VCA picture, 10-PIR alarm, 11-wireless alarm, 12-emergency alarm, 0xa-capture in live view, 0xd-face detection, 0xe-line crossing detection, 0xf-intrusion detection, 0x10-scene change detection, 0x11-capture in local playback, 0x12-VCA detection, 0x13-region entrance detection, 0x14-region exiting detection, 0x15-loitering detection, 0x16-people gathering detection, 0x17-fast moving detection, 0x18-parking detection, 0x19-unattended baggage detection, 0x1a-object removal detection, 0x1b-vehicle detection, 0x1c-mixed-traffic detection (enforcement event), 0x25-face capture, 0x2a-getting up detection, 0x2b-climbing alarm, 0x2c-in-toilet overtime detection, 0x35-unregistered street vendor, 0x36-people density detection (number of people pre-alarm), 0x37-absence detection, 0x38-number of people exception detection, 0x39-violent motion detection, 0x3a-illegal parking detection, 0x3b-wrong-way driving detection, 0x3c-driving on the lane line detection, 0x3d-motor vehicle on non-vehicle lane detection, 0x3e-illegal lane change detection, 0x3f-U-turning detection, 0x40-pedestrian detection, 0x41-roadblock detection, 0x42-thrown object detection, 0x43-fog detection, 0x44-construction detection, 0x45-congestion detection, 0x46-traffic accident detection, 0x47-parallel parking detection, 0x48-trigger alarm manually, 0x52-people queuing-up detection, 0x53-waiting time detection, 0x2d-hard hat detection, 0x58-vibration detection, 0xff-all-->
    <enableNeedCard opt="disable,able"/>
    <province
      opt="1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,0xff"/>
      <!--required, province index-->
    <cardNumberLen min="" max="" />
    <StartTime>
      <year min="" max="" />
      <month min="" max="" />
      <day min="" max="" />
      <hour min="" max="" />
      <minute min="" max="" />
      <second min="" max="" />
    </StartTime>
    <StopTime>
      <year min="" max="" />
      <month min="" max="" />
      <day min="" max="" />
      <hour min="" max="" />
      <minute min="" max="" />
    </StopTime>
  </FindPicInfo>
</JpegCaptureAbility>
```

```

<second min="" max="" />
</StopTime>
<trafficType opt="license,vehicleType,illegalType"/>
<vehicleType opt="smallCar,bigCar,bus,truck,car,minibus,smallTruck"/>
<subHvtType opt="all,motorVehicle,nonMotorVehicle,pedestrian"/>
  <!--required, "all", "motorVehicle"-motor vehicle, "nonMotorVehicle"-non-motor vehicle, "pedestrian"-->
<illegalType
opt="postPic,lowSpeed,highSpeed,retrograde,rushRedLight,pressLane,violateGuide,roadStrand,vehicleillegal,roadStan
d,changeLane,dirveillegalLane,violate,crossParking,greenParking"/>
<region opt="Res,EU,ER,All"/>
  <!--required, region index: 0-reserved, 1-Europe (EU), 2-Russia (ER), 0xff-all-->
<country opt="0,1,2,3,4,5,6,7,8,9,10,11,12,0xfe,0xff "/>
  <!--required, country index: 0-license plate recognition for this country is not supported by the algorithm library, 1-
Czech Republic (CZ), 2-France (FRA), 3-Germany (DE), 4-Spain (E), 5-Italy (IT), 6-Netherlands (NL), 7-Poland (PL), 8-
Slovakia (SVK), 9-Belorussia (BY), 10-Moldova (MDA), 11-Russia (RU), 12-Ukraine (UA), 13-Belgium (BEL), 14-Bulgaria,
15-Denmark, 16-Finland, 17-United Kingdom, 18-Greece, 19-Croatia, 20-Hungary, 21-Israel, 22-Luxembourg, 23-
Republic of Macedonia, 24-Norway, 25-Portugal, 26-Romania, 27-Serbia, 28-Republic of Azerbaijan, 29-Georgia, 30-
Kazakhstan, 31-Republic of Lithuania, 32-Turkmenistan, 33-Uzbekistan, 34-Latvia, 35-Estonia, 36-Albania, 37-Austria,
38-Bosnia and Herzegovina, 39-Ireland, 40-Iceland, 41-Vatican, 42-Malta, 43-Sweden, 44-Switzerland, 45-Cyprus, 46-
Turkey, 47-Slovenia, 48-Republic of Montenegro, 0xfe-unrecognized, 0xff-all-->
<licenseLen min="" max="" />
  <!--optional, the maximum and minimum length of the license plate number-->
<supportAIDTFSType opt="illegalParking,wrongdirection crosslane,vehicleexist,lanechange, turnaround, evidence"/>
  <!--optional, xs:string, event types for searching pictures both supported by AID and TFS-->
</FindPicInfo>
<SmartPicSearchInfo>
  <!--required-->
  <supportFileType opt=
"vehicleDetection,faceFeature,facePicData,LineDetection,FieldDetection,unattendedBaggage,attendedBaggage,region
Entrance,regionExiting,parking,loitering,group,rapidMove,running,violentMotion,failDown,peopleNumChange,leavePo
sition,retentionParam,situationAnalysis,allType"/>
    <!--required, supported file type, 0("vehicleDetection")-vehicle search, 1("faceFeature")-facial features,
2("facePicData")-face picture data, 3("LineDetection")-line crossing, 4("FieldDetection")-intrusion, 5-
("unattendedBaggage")unattended baggage, 6("attendedBaggage")-object removal, 7("regionEntrance")-region
entrance, 8("regionExiting")-region exiting, 9("parking")-illegal parking, 10("loitering")-loitering, 11("group")-people
gathering, 12("rapidMove")-fast moving, 13("running")-people running, 14("violentMotion")-violent motion,
15("failDown")-people falling down, 16("peopleNumChange")-people number change, 17("leavePosition")-absence
detection, 18("retentionParam")-overstay detection, 19("situationAnalysis")-situation analysis-->
<StartTime>
  <year min="" max="" />
  <month min="" max="" />
  <day min="" max="" />
  <hour min="" max="" />
  <minute min="" max="" />
  <second min="" max="" />
</StartTime>
<StopTime>
  <year min="" max="" />
  <month min="" max="" />
  <day min="" max="" />
  <hour min="" max="" />
  <minute min="" max="" />

```

```
<second min="" max="/" />
</StopTime>
<VehicleCond>
    <!--required, this node will be returned if vehicle search is supported-->
    <licenseLen min="1" max="16" />
        <!--required, license plate number length-->
    <country opt="czech,france,germany,spain,italy,netherlands,poland,slovakia,belorussia,moldova,russia,ukraine," />
        <!--required, country-->
</VehicleCond>
<FaceFeature>
    <!--required, facial features-->
    <sex opt="man,women" />
    <withGlasses opt="true,false" />
        <!--required, whether to support wearing glasses-->
</FaceFeature>
<FacePicData>
    <!--required, face picture data-->
    <faceScore min="0" max="100" />
        <!--required, similarity-->
    <picType opt="jpg" />
        <!--required, picture format-->
</FacePicData>
<isSupportFaceDataAnalysis>
    <!--optional, xs:boolean, whether to support face picture analysis: "true,false"-->
</isSupportFaceDataAnalysis>
<isSupportHumanMisinfo>
    <!--optional, xs:boolean, whether to get human body false alarms only: "true"-yes, this node is not returned-no-->
</isSupportHumanMisinfo>
</SmartPicSearchInfo>
<ManualCapture>
    <!--required, manual capture-->
    <!--required, the network camera will return the supported resolution currently-->
<ResolutionEntry>
    <!--required-->
    <resolutionName>
        <!--required, xs:string, resolution name, "CIF"...-->
    </resolutionName>
    <index>
        <!--required, xs:integer-->
    </index>
    <streamType>
        <!--optional, xs:string, "main,sub"-->
    </streamType>
</ResolutionEntry>
<ResolutionEntry>
    <!--required-->
    <resolutionName>
        <!--required, xs:string, resolution name, "4CIF"...-->
    </resolutionName>
    <index>
        <!--required, xs:integer-->
    </index>

```

```
<streamType>
  <!--optional, xs:string, "main,sub"-->
</steamType>
</ResolutionEntry>
<picQuality opt="best,better,normal"/>
  <!--picture quality: "best", "better", "normal"-->
</ManualCapture>
<SchedCapture>
  <!--required, scheduled capture-->
<TimingCap>
  <!--required, scheduled capture-->
<ResolutionEntry>
  <!--required-->
<resolutionName>
  <!--required, xs:string, resolution name, "CIF"...-->
</resolutionName>
<index>
  <!--required, xs:integer-->
</index>
<streamType>
  <!--optional, xs:string, stream type, "main,sub"-->
</steamType>
</ResolutionEntry>
<ResolutionEntry>
  <!--required-->
<resolutionName>
  <!--required, xs:string, resolution name, "4CIF"...-->
</resolutionName>
<index>
  <!--required, xs:integer-->
</index>
</ResolutionEntry>
<intervalUnit>
  <!--required, time interval unit, s/ms-->
</intervalUnit>
<interval min="" max="" opt="0"/>
  <!--required, xs:integer, capture interval-->
<RecorderDuration min="" max="" />
  <!--required, days to keep the captured pictures-->
<DayCapture>
  <!--required, all-day capture-->
<captureType opt="timing,motion,alarm,motionOrAlarm,motionAndAlarm,vca, command"/>
  <!--required, capture type-->
</DayCapture>
<TimeSlot>
  <!--required, scheduled capture-->
<slotNum><!--required, time period--></slotNum>
<captureType opt="timing,motion,alarm,motionOrAlarm,motionAndAlarm,vca, command"/>
  <!--required, capture type-->
</TimeSlot>
<HolidayDay>
  <!--required, all-day capture on holidays-->
```

```
<captureType opt="timing,motion,alarm,motionOrAlarm,motionAndAlarm,vca, command"/>
  <!--required, capture type-->
</HolidayDay>
<HolidayTimeSlot>
  <!--required, scheduled capture on holidays-->
<slotNum>
  <!--required, time period-->
</slotNum>
<captureType opt="timing,motion,alarm,motionOrAlarm,motionAndAlarm,vca, command"/>
  <!--required, capture type-->
</HolidayTimeSlot>
</TimingCap>
<EventCap>
  <eventType opt="motion,hide,loss,PIR,wireless,callhelp,vca,facedetect ,
lineDetection,fieldDetection,sceneChangeDetection"/>
    <!--required, xs:string, event type-->
<ResolutionEntry>
  <!--req-->
<resolutionName>
  <!--required, xs:string, resolution name, "CIF"...-->
</resolutionName>
<index>
  <!--required, xs:integer-->
</index>
<streamType>
  <!--optional, xs:string, stream type: "main,sub"-->
</steamType>
</ResolutionEntry>
<ResolutionEntry>
  <!--required-->
<resolutionName>
  <!--required, xs:string, resolution name, "4CIF"...-->
</resolutionName>
<index>
  <!--required, xs:integer-->
</index>
<streamType>
  <!--optional, xs:string, stream type: "main,sub"-->
</steamType>
</ResolutionEntry>
<intervalUnit>
  <!--required, time interval unit, s/ms-->
</intervalUnit>
<interval min="" max="" opt="0"/>
  <!--required, xs:integer, capture interval-->
<capTimes min="" max="" />
  <!--required, xs:integer, number of captured pictures-->
<eventCapChan opt="1,2"/>
  <!--required, xs:integer, channe that can be triggered by event to capture-->
<alarmInCapChan opt="1,2"/>
  <!--required, xs:integer, channel that can be triggered by alarm input to capture-->
</EventCap>
```

```
<AdvancedParam>
  <!--optional, advanced configuration parameters for capture-->
  <streamType opt="0-mainstream,1-substream,2-stream3,3-stream4,4-stream5">
    <!--optional, xs:string, stream type-->
  </streamType>
</AdvancedParam>
</SchedCapture>
<WindowCapture>
  <!--required, capture capability on video wall window-->
<FreeShowBoard>
  <!--required, capture resolution supported by fluent video decoder-->
<ResolutionEntry>
  <!--required-->
  <resolutionName>
    <!--required, xs:string, resolution name, "CIF"...-->
  </resolutionName>
  <index>
    <!--required, xs:integer-->
  </index>
</ResolutionEntry>
<ResolutionEntry>
  <resolutionName>
    <!--required, xs:string, resolution name, "4CIF"...-->
  </resolutionName>
  <index>
    <!--required, xs:integer-->
  </index>
</ResolutionEntry>
</FreeShowBoard>
<NormalDecBoard>
  <!--required, capture resolution supported by non-fluent video decoder-->
<ResolutionEntry>
  <!--req-->
  <resolutionName>
    <!--required, xs:string, resolution name, "CIF"...-->
  </resolutionName>
  <index>
    <!--required, xs:integer-->
  </index>
</ResolutionEntry>
<ResolutionEntry>
  <resolutionName>
    <!--required, xs:string, resolution name, "4CIF"...-->
  </resolutionName>
  <index>
    <!--required, xs:integer-->
  </index>
</ResolutionEntry>
</NormalDecBoard >
</WindowCapture>
<VcaDecCapture>
  <!--optional, VCA decoding capture capability-->
```

```
<picQuality opt="best,better,normal"/>
  <!--picture quality: "best", "better", "normal"-->
<ResolutionEntry>
  <!--req-->
  <resolutionName>
    <!--required, xs:string, resolution name, "CIF"...-->
  </resolutionName>
  <index>
    <!--required, xs:integer-->
  </index>
</ResolutionEntry>
<ResolutionEntry>
  <resolutionName>
    <!--required, xs:string, resolution name, "4CIF"...-->
  </resolutionName>
  <index>
    <!--required, xs:integer-->
  </index>
</ResolutionEntry>
</VcaDecCapture>
<CapturePicture><!--optional, capture capability-->
  <Resolution><!--required, supported resolution-->
    <value opt="218-1440*1440,255-auto"/>
  </Resolution>
  <picQuality opt="best,good,normal"/><!--required, picture quality level, 0-high, 1-medium, 2-low-->
  <picFormat opt="0-Jpeg"/><!--required, captured picture format: 0-JPEG-->
  <picType opt="normal,calibPanoramicPic,calibPanoramicPicinFlash,fishCirclePic"/>
  <!--required, captured picture type: normal, calibPanoramicPic-calibrated picture of PanoVu series camera (reboot FPJA to refresh and get picture, the longest timeout is 30 minutes), calibPanoramicPicinFlash-get existing calibrated picture of PanoVu series camera from Flash, fishCirclePic-fisheye view picture-->
</CapturePicture>
</JpegCaptureAbility>
```

F.258 XML_Language

Language message in XML format

```
<Language version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <type><!--req, xs: string, "GBK,EUC-KR", def="GBK"--></type>
</Language>
```

F.259 XML_LensDistortionCorrection

LensDistortionCorrection message in XML format

```
<PrivacyMaskRegion version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--req, xs: integer--></id>
  <enabled><!--req, xs: boolean--></enabled>
  <RegionCoordinatesList><!--req-->
```

```
<RegionCoordinates><!--list-->
  <positionX><!--req, xs: integer; coordinate--></positionX>
  <positionY><!--req, xs: integer; coordinate--></positionY>
</RegionCoordinates>
</RegionCoordinatesList>
<privacymaskName><!--opt, xs: string--></privacymaskName>
<maskType>
  <!--opt, xs: string, "gray,red,yellow,blue,orange,green,transparent,half-transparent,mosaic"-->
</maskType>
<zoomdoorlimit><!--opt, xs: integer, the value is between 10 and 1000--></zoomdoorlimit>
</PrivacyMaskRegion>
```

F.260 XML_localPermission

localPermission message in XML format

```
<localPermission version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <backup><!--opt, xs:boolean--></backup>
  <record><!--opt, xs:boolean--></record>
  <playBack><!--opt, xs:boolean--></playBack>
  <preview><!--opt, xs:boolean--></preview>
  <videoChannelPermissionList><!--opt-->
    <videoChannelPermission><!--opt-->
      <id><!--req, corresponds to the video input channel ID--></id>
      <playBack><!--opt, xs:boolean--></playBack>
      <preview><!--opt, xs:boolean--></preview>
      <record><!--opt, xs:boolean--></record>
      <backup><!--opt, xs:boolean--></backup>
      <playBackDoubleVerification>
        <!--opt, xs:boolean, whether supports secondary authentication for playback-->
      </playBackDoubleVerification>
      <backupDoubleVerification>
        <!--opt, xs:boolean, whether supports secondary authentication for backup-->
      </backupDoubleVerification>
    </videoChannelPermission>
  </videoChannelPermissionList>
  <ptzControl>
    <!--req, xs:boolean-->
  </ptzControl>
  <ptzChannelPermissionList><!--opt-->
    <ptzChannelPermission><!--req-->
      <id><!--req, corresponds to PTZ channel ID--></id>
      <ptzControl><!--opt, xs: boolean--></ptzControl>
    </ptzChannelPermission>
  </ptzChannelPermissionList>
  <logOrStateCheck><!--opt, xs: boolean--></logOrStateCheck>
  <parameterConfig><!--opt, xs: boolean--></parameterConfig>
  <restartOrShutdown><!--opt, xs: boolean--></restartOrShutdown>
  <upgrade><!--opt, xs: boolean--></upgrade>
  <AIModelManagement><!--opt, xs:boolean, operation permission of AI model packages--></AIModelManagement>
```

```
<AITaskManagement><!--opt, xs:boolean, management and operation permission of AI tasks--></AITaskManagement>
<subSysOrZoneArm>
  <!--read-only, optional, xs:boolean, arm permission of partition/alarm zone-->true
</subSysOrZoneArm>
<subSysOrZoneDisarm>
  <!--read-only, optional, xs:boolean, disarm permission of partition/alarm zone-->true
</subSysOrZoneDisarm>
<operateOutput>
  <!--read-only, optional, xs:boolean, permission for operation on relay-->true
</operateOutput>
<supportLinkageSubSystemList>
  <!--read-only, optional, list of the partitions that support linkage-->
  <subSystem>
    <!--read-only, optional, xs:integer, the partition that supports linkage, value range:[1,64]-->1
  </subSystem>
</supportLinkageSubSystemList>
<subSystemList>
  <!--read-only, optional, xs:object, list of partitions-->
  <subSystem>
    <!--read-only, optional, xs:integer, partition, range:[1,64]-->1
  </subSystem>
</subSystemList>
</localPermission>
```

F.261 XML_localPermissionCap

localPermissionCap capability message in XML format

```
<localPermissionCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <backup><!--opt, xs:boolean--></backup>
  <record><!--opt, xs:boolean--></record>
  <playBack><!--opt, xs:boolean--></playBack>
  <preview><!--opt, xs:boolean--></preview>
  <videoChannelPermissionList><!--opt-->
    <videoChannelPermission><!--opt-->
      <id><!--req, corresponds to the video input channel ID--></id>
      <playBack><!--opt, xs:boolean--></playBack>
      <preview><!--opt, xs:boolean--></preview>
      <record><!--opt, xs:boolean--></record>
      <backup><!--opt, xs:boolean--></backup>
      <playBackDoubleVerification>
        <!--opt, xs:boolean, whether supports secondary authentication for playback-->
      </playBackDoubleVerification>
      <backupDoubleVerification>
        <!--opt, xs:boolean, whether supports secondary authentication for backup-->
      </backupDoubleVerification>
    </videoChannelPermission>
  </videoChannelPermissionList>
  <ptzControl>
```

```

<!--req, xs:boolean-->
</ptzControl>
<ptzChannelPermissionList><!--opt-->
<ptzChannelPermission><!--req-->
<id><!--req, corresponds to PTZ channel ID--></id>
<ptzControl><!--opt, xs: boolean--></ptzControl>
</ptzChannelPermission>
</ptzChannelPermissionList>
<logOrStateCheck><!--opt, xs: boolean--></logOrStateCheck>
<parameterConfig><!--opt, xs: boolean--></parameterConfig>
<restartOrShutdown><!--opt, xs: boolean--></restartOrShutdown>
<upgrade><!--opt, xs: boolean--></upgrade>
</localPermissionCap>

```

F.262 XML_LockByName

XML message about parameters of locking and unlocking video by file name

```

<?xml version="1.0" encoding="utf-8"?>
<LockByName version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<TrackId><!--required, xs:string, channel ID, e.g., 101 indicates the main stream of channel No.1--></TrackId>
<recordName><!--required, xs:string, file name, the name length is between 1 and 100--></recordName>
<command><!--required, xs:string, operation command: "lock" and "unlock"--></command>
<duration>
<!--dependent, xs:integer, locking duration, unit: second; if the value is 0, it indicates that the video will be
permanently locked, this node is valid only when command is set to "lock"-->
</duration>
<recorderCode><!--optional, xs:string, body camera No.--></recorderCode>
<policeCode><!--optional, xs:string, No. of police that collects data, it should be encrypted--></policeCode>
</LockByName>

```

F.263 XML_LockCap

XML message about video locking and unlocking capability

```

<?xml version="1.0" encoding="utf-8"?>
<LockCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<LockByName><!--optional, lock or unlock video by file name-->
<TrackId><!--required, xs:string--></TrackId>
<recordName min="1" max="100"><!--required, xs:string, video file name--></recordName>
<command opt="lock,unlock"><!--required, xs:string, operation command: "lock" and "unlock"--></command>
<duration min="" max=""><!--dependent, xs:integer, locking duration, unit: second--></duration>
<recorderCode min="1" max="100"><!--optional, xs:string, body camera No.--></recorderCode>
<policeCode min="1" max="100">
<!--optional, xs:string, No. of police that collects data, it should be encrypted-->
</policeCode>
</LockByName>
<LockByTime><!--optional, lock or unlock video by time-->
<TrackId><!--required, xs:string--></TrackId>

```

```
<recordType opt= "ALL, MANUAL, CMR, MOTION, ALARM, EDR, ALARMANDMOTION, COMMAND, SMART">
  <!--required, xs:string-->
</recordType>
<startDateTime><!--dependent, xs:time, ISO8601 time format, e.g., 2004-05-03T17:30:08+08:00--></startDateTime>
<endDateTime><!-- dependent, xs:time, ISO8601 time format, e.g., 2004-05-03T17:30:08+08:00--></endDateTime>
<command opt="lock,unlock"><!--required, xs:string--></command>
<duration min="" max=""><!--dependent, xs:integer--></duration>
</LockByTime>
</LockCap>
```

F.264 XML_LogServer

XML message about log server information

```
<LogServer version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <enabled><!--required, xs: boolean, opt="true,false"--></enabled>
  <addressingFormatType>
    <!--required, xs: string, "ipaddress,hostname"-->
  </addressingFormatType>
  <hostName><!--dependent, xs: string--></hostName>
  <ipAddress><!--dependent, xs: string--></ipAddress>
  <ipv6Address><!--dependent, xs: string--></ipv6Address>
  <portNo><!--optional, xs: integer--></portNo>
  <transmissionEncryption>
    <!--optional, xs:boolean, whether to enable transmission encryption: "true"-yes, "false"-no (default). If this field is not supported, the default encryption method is TLS-->
  </transmissionEncryption>
  <checkCertificate>
    <!--optional, xs:boolean,whether to enable checking certificates-->
  </checkCertificate>
  <uploadInterval>
    <!--optional, xs:integer, log uploading interval, which is 1 hour by default, range: [1,24], unit: hour-->
  </uploadInterval>
</LogServer>
```

F.265 XML_LogServerCap

XML message about log server capability

```
<LogServerCap version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <enabled opt="true,false"></enabled>
  <addressingFormatType opt="ipaddress,hostname">
    <!--required, xs:string,"ipaddress,hostname"-->
  </addressingFormatType>
  <hostName min="" max="">
    <!--dependent, xs:string-->
  </hostName>
  <ipAddress min="" max=""><!--dep, xs:string--></ipAddress>
  <ipv6Address min="" max=""><!-dep, xs:string--></ipv6Address>
```

```
<portNo min="" max=""><!--opt, xs:integer--></portNo>
<transmissionEncryption opt="true,false">
  <!--optional, xs:boolean, whether to enable transmission encryption: "true"-yes, "false"-no (default). If this field is
not supported, the default encryption method is TLS-->
</transmissionEncryption>
<checkCertificate opt="true,false">
  <!--optional, xs:boolean, whether to enable checking certificates-->
</checkCertificate>
<uploadInterval min="1" max="24" def="1">
  <!--optional, xs:integer, log uploading interval, which is 1 hour by default, range: [1,24], unit: hour-->
</uploadInterval>
</LogServerCap>
```

F.266 XML_PrivacyMaskRegion

PrivacyMaskRegion message in XML format

```
<PrivacyMaskRegion version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id><!--req, xs: integer--></id>
<enabled><!--req, xs: boolean--></enabled>
<RegionCoordinatesList><!--req-->
  <RegionCoordinates><!--req-->
    <positionX><!--req, xs: integer; coordinate--></positionX>
    <positionY><!--req, xs: integer; coordinate--></positionY>
  </RegionCoordinates>
</RegionCoordinatesList>
<privacymaskName><!--opt, xs: string--></privacymaskName>
<maskType>
  <!--opt, xs:string "gray,red,yellow,blue,orange,green,transparent,half-transparent,mosaic,black"-->
</maskType>
<zoomdoorlimit><!--opt, xs: integer, the value is between 10 and 1000--></zoomdoorlimit>
</PrivacyMaskRegion>
```

F.267 XML_MasterSlaveTrackingCap

MasterSlaveTrackingCap capability message in XML format

```
<MasterSlaveTrackingCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<isSupportSlaveCameraCfg>
  <!--optional, xs:boolean, whether to support PTZ camera configuration-->
</isSupportSlaveCameraCfg>
<isSupportSlaveCameraStatus>
  <!--optional, xs:boolean, whether to support PTZ camera status configuration-->
</isSupportSlaveCameraStatus>
<isSupportSlaveCameraTrackingRatio>
  <!--optional, xs:boolean, whether to support PTZ camera zoom ratio configuration-->
</isSupportSlaveCameraTrackingRatio>
<isSupportSlaveCameraTracking>
  <!--optional, xs:boolean, whether to support PTZ camera tracking-->
```

```
</isSupportSlaveCameraTracking>
<isSupportLinkedTracking>
  <!--optional, xs:boolean, whether device supports advanced parameters configuration of linked tracking (related
URI: /ISAPI/MasterSlaveTracking/linkedTracking?format=json)-->
</isSupportLinkedTracking>
</MasterSlaveTrackingCap>
```

F.268 XML_MaxElevation

MaxElevation message in XML format

```
<MaxElevation version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <mElevation><!--req, xs:integer, the lower limit of max. tilt-angle--></mElevation>
</MaxElevation>
```

F.269 XML_MountingScenario

MountingScenario message in XML format

```
<MountingScenario version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <mode>
    <!--req, xs:string,
    "indoor,outdoor,day,night,morning,nightfall,mode1,mode2,mode3,mode4,highway,road2,faceSnap,backlight,frontlight
    ", "lowIllumination"-low illumination, "backlight"-back light, "frontlight"-front light, "faceSnap"-face picture capture-->
  </mode>
</MountingScenario>
```

F.270 XML_NetAppAbility

NetAppAbility capability message in XML format

```
<NetAppAbility version="2.0">
  <NTP>
    <intervalUnit><!--req, time interval unit: minute,hour--></intervalUnit>
    <serverTest><!--required, whether it supports NTP server test--></serverTest>
  </NTP>
  <Net>
    <NetworkInterface>
      <networkInterfaceNum><!--req, number of network adapters--></networkInterfaceNum>
      <NetworkInterfaceEntry>
        <id></id>
        <type opt="10Mbase-T,10MBase-T-full,100MBase-TX,100M-full,10M/100M/1000M-adapt,1000M-full"/>
        <MTU min="500" max="9676"/>
        <isSupportDNS>
          <!--opt, whether supports enabling DNS manual settings: 1-support, the node will not appear if not support-->
        </isSupportDNS>
        <EthernetPortNo min="" max="" />
```

```
<!--opt, ro, network interface number-->
</NetworkInterfaceEntry>
<NetworkInterfaceEntry>
<id></id>
<type opt="10Mbase-T,10MBase-T-full,100MBase-TX,100M-full,10M/100M/1000M-adapt,1000M-full"/>
<MTU min="500" max="9676"/>
<isSupportDNS>
  <!--opt, whether to support enabling DNS manual settings: 1- support, the node will not appear if not support-->
>
</isSupportDNS>
<EthernetPortNo min="" max="" /><!--opt,ro, network interface number-->
</NetworkInterfaceEntry>
</NetworkInterface>
<multicastIpAddr opt="IPV4,IPV6"/><!--req, multicast address type-->
</Net>
<Email>
<receiverNum>3</receiverNum><!--req, the count of recipients-->
<emailTest>true</emailTest><!--req, e-mail test-->
<enableStartTLS>true</enableStartTLS>
<!--opt, email supports enabling TSL/TLS-->
<emailTestWithParam><!--required, whether it supports e-mail test with parameters--></emailTestWithParam>
<enableTSL><!--required, whether it supports encrypting the e-mail by TSL--></enableTSL>
</Email>
<Bonjour>
  <enabled>true</enabled><!--opt, whether it supports Bonjour-->
</Bonjour>
<UPNP>
<NATType opt="manual,auto"/><!--req, mapping type: 0-manual, 1-auto-->
<friendNameLen min="0" max="64"/><!--req, UPNP service name-->
<serverPort>
  <enabled><!--req, whether supports setting mapping parameters for service port (8000)--></enabled>
</serverPort>
<HTTPPPort>
  <enabled><!--req, whether supports to setting mapping parameters for HTTP port (80)--></enabled>
</HTTPPPort>
<RTSPPort>
  <enabled><!--req, whether supports to setting mapping parameters for RTSP port (554)--></enabled>
</RTSPPort>
<HTTPSPort>
  <enabled>true</enabled><!--req, supports to setting mapping parameters for HTTPS port-->
</HTTPSPort>
<ManualMap>
  <enabled><!--req, supports manual mapping, if the UPnP is not enabled, the manually mapped port number will be assigned--></enabled>
</ManualMap>
<SDKOverTLSPort>
<enabled><!--opt, supports setting mapping parameters for SDKOverTLS port--></enabled>
</SDKOverTLSPort>
</UPNP>
<IPAddrFilter>
  <!--req, IP address filter function 2012-08-29-->
  <IPAddrType opt="IPV4,IPV6"/><!--req, filter the IP address type-->
```

```
<filterType opt="forbid,permit"/><!--req,filter type: 0- forbid,1-permit-->
<IPTest><!--required, whether it supports IP address conflict test--></IPTest>
</IPAddrFilter>
<FTP><!--2012-08-29-->
    <AnonyFTP>
        <enabled>true</enabled><!--req,whether it supports anonymous FTP-->
    </AnonyFTP>
    <dirLevel opt="rootDir,topDir,subDir"/><!--req,the supported FTP directory levels-->
    <topDirMode opt="deviceName,deviceNO,deviceIP,monitor,time"/><!--req,the type of directory level no.1-->
    <subDirMode opt="chanName,chanNO,time,laneNO"/><!--req,the type of directory level no.2-->
    <serverTest><!--required, whether it supports FTP server test--></serverTest>
</FTP>
<SOCKS>
    <SOCKSVersion opt="SOCKS4,SOCKS5"/><!--req,SOCKS version-->
    <proxyIP opt="IPV4,IPV6,domain"/><!--req,the supported address types of proxy server-->
    <proxyPort min="" max="" /><!--req, port of proxy server -->
    <userNameLength min="" max="" /><!--req,the user name length of proxy server -->
    <passwordLength min="" max="" /><!--req,the password length of proxy server-->
</SOCKS>
<QoS>
    <manageDscp min="" max="" /><!--req,DSCP value of management data -->
    <alarmDscp min="" max="" /><!--req,DSCP value of alarm data -->
    <videoDscp min="" max="" /><!--req,DSCP value of video data-->
    <audioDscp min="" max="" /><!--req,DSCP value of audio data-->
    <flagType opt="videoAudio,videoAudioSeparate"/><!--req,flag types: videoAudio- video and audio
merged,videoAudioSeparate- video and audio separate-->
</QoS>
<HTTPS>
    <HTTPSPort min="" max="" /><!--req,HTTPS port-->
    <Certificate>
        <certtype opt="CA,Certificate,privateKey"/><!-- req,certificate type: 0-CA,1-Certificate,2-private key file-->
        <fileType opt="PEM,PFX"/><!-- req,certificate file type: 0-PEM,1-PFX-->
        <keyAlgorithm opt="RSA,DSA"/><!-- req,encryption algorithm type: 0-RSA, 1-DSA-->
        <keyLen opt="512,1024,2048" def=2048/><!-- req,key length-->
        <SignatureAlgorithm opt="MD5,RSA,DSA"/><!-- req,the signature algorithm types: MD5, RSA, DSA-->
    </Certificate>
</HTTPS>
<NAS>
    <enabled>true</enabled><!--opt,support NAS search-->
    <serverTest><!--required, whether it supports NAS server test--></serverTest>
</NAS>
<IPSAN>
    <enabled>true</enabled><!--opt,support IPSAN search-->
</IPSAN>
<NAT>
    <enabled>true</enabled><!--req,support NAT-->
</NAT>
<FuzzyUpgrade>
    <enabled>true</enabled>
    <!--req,support FuzzyUpgrade-->
</FuzzyUpgrade>
<AuxiliaryDevUpgrade>
```

```
<enabled>true</enabled>
<!--req,support AuxiliaryDevUpgrade (auxiliary device upgrade)(ENUM_UPGRADE_AUXILIARY_DEV)-->
<devType opt="keyborad,xxx,router,zone"/>
<!--req,upgraded device type: keyborad-Keyboard,xxx-Mechanical movement,router-Router,zone-Zone-->
</AuxiliaryDevUpgrade>
<EthernetChannel>
  <!--req,Port Trunking function,it will not display if the funciton doesn't be supported-->
  <ethernetPortNum>8</ethernetPortNum><!--req, port numbert-->
  <ethernetLinkNum>8</ethernetLinkNum><!--req,link number-->
</EthernetChannel>
<CMS>
<cmsNo min="" max="" />
<!--req, number of push mode device registration platforms-->
<CmsParam>
<enable opt="open,close"/>
<!--enable-->
<serverIpv4>true</serverIpv4>
<!--req, support IPV4-->
<serverIpv6>true</serverIpv6>
<!--req, support IPV6-->
<serverPort min="" max="" />
<!--req, platform port No.-->
<serverProtocolType opt="private"/>
<!--req, protocol supported by platform, private protocol-->
<deviceStatus opt="offline,online"/>
<!--req, device registration status-->
<deviceIdLength min="" max="" />
<!--req, device ID length-->
<netWork opt="auto,wiredNetWorkFirst,wiredNetWork,3GNetWork"/>
<!--req, auto, wired network priority, wired network, 3G network-->
<deviceID
opt="0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"
/>
  <!--req, characters supported by device ID-->
<addressType opt="IP/IPV6, domain"/>
<!--req, supported address type-->
<protocolVersion opt="v2.0, v4.0"/>
<!--opt,protocol version-->
</CmsParam>
</CMS>
<CloudStorage>
<enable opt="disable,enable"/>
<!--req, suppor cloud direct storage mode-->
<poolID min="" max="" />
<userNameLen min="" max="" />
<passwdLen min="" max="" />
<poolInfo opt="postVideoPool,illegalVideoPool,vehicleDetection"/>
<!--req, number array 0 means checkpoint recoding pool, number array 1 means violation recording pool, number
array 2 means vehicle detection data pool (IPC/D)-->
<V2.0>
  <enable opt="disable,enable"/>
  <!--req, support cloud storage V2.0 protocol-->
```

```

<isSupportAccessKey>
  <!--opt, xs:boolean, whether supports access key-->
</isSupportAccessKey>
<isSupportSecretKey>
  <!--opt, xs:boolean, whether supports encryption key-->
</isSupportSecretKey>
<postPoolID min="0" max="4294967295">
  <!--opt, xs:integer-->
</postPoolID>
<accessKeyLen min="1" max="64"/>
<!--dep, access key length, it is valid when isSupportAccessKey is true-->
<secretKeyLen min="1" max="64"/>
<!--dep, encryption key length, it is valid when isSupportSecretKey is true-->
<poolInfo opt="postVideoPool,illegalVideoPool,vehicleDetection"/>
<!--req array 0- video pool of monitoring point, array 1- video pool of traffic violation, array 2- data pool of vehicle
detection (IPC/IPD)-->
</V2.0>
<Test>
  <serverIpv4>true</serverIpv4>
  <!--support IPV4-->
  <serverIpv6>true</serverIpv6>
  <!--support IPV6-->
  <serverPort min="" max="" />
  <!--platform port No.-->
  <poolID min="" max="" />
  <userNameLen min="" max="" />
  <passwdLen min="" max="" />
  <poolInfo opt="postVideoPool,illegalVideoPool,vehicleDetection"/>
  <!--number array 0 means checkpoint recording pool, number array 1 means violation recording pool, number
array 2 means vehicle detection data pool (IPC/D)-->
  <isSupportAccessKey>
    <!--opt, xs:boolean, whether supports access key-->
  </isSupportAccessKey>
  <isSupportSecretKey>
    <!--opt, xs:boolean, whether supports encryption key-->
  </isSupportSecretKey>
  <accessKeyLen min="1" max="64"/>
  <!--dep, access key length, it is valid when isSupportAccessKey is true-->
  <secretKeyLen min="1" max="64"/>
  <!--dep, encryption key length, it is valid when isSupportSecretKey is true-->
</Test>
</CloudStorage>
<allBitRate opt="support"/>
<!--value of dwAllBitRate in NET_DVR_CHANNELSTATE_V30-->
<NetCfg>
  <Ethernet1>
    <IPAddrType opt="IPV4,IPV6"/>
    <!--req, IP address type-->
    <IPAddrMaskType opt="IPV4,IPV6"/>
    <!--req,IP address mask-->
    <netInterface opt="10MBase-T,10MBase-T_FullDuplex,100MBase-TX,100M_FullDuplex,Adaptive,
1000M_FullDuplex"/>

```

```
<port min="" max="" />
<mtu min="" max="" />
<macAddrLen min="" max="" />
</Ethernet1>
<Ethernet2>
<IPAddrType opt="IPV4,IPV6"/>
<!--req, IP address type-->
<IPAddrMaskType opt="IPV4,IPV6"/>
<!--req,IP address mask-->
<netInterface opt="10MBase-T,10MBase-T_FullDuplex,100MBase-TX,100M_FullDuplex,Adaptive,
1000M_FullDuplex"/>
<port min="" max="" />
<mtu min="" max="" />
<macAddrLen min="" max="" />
</Ethernet2>
<manageHost1IpAddrType opt="IPV4,IPV6"/>
<!--req,IP address type-->
<manageHost2IpAddrType opt="IPV4,IPV6"/>
<!--req,IP address type-->
<alarmHostIpAddrType opt="IPV4,IPV6"/>
<!--req,IP address type-->
<alarmHost2IpAddrType, opt="IPV4,IPV6"/>
<!--req,IP address type-->
<manageHost1Port min="" max="" />
<manageHost2Port min="" max="" />
<alarmHostIpPort min="" max="" />
<alarmHost2IpPort, min="", max="" />
<useDhcp opt="enable,disable"/>
<IPV6Mode opt="route,manual,dhcp"/>
<privateMulticastDiscovery opt="enable,disable"/>
<onvifMulticastDiscovery opt="enable,disable"/>
<dnsServer1IpAddrType opt="IPV4,IPV6"/>
<!--req,IP address type-->
<dnsServer2IpAddrType opt="IPV4,IPV6"/>
<!--req,IP address type-->
<IpResolverLen min="" max="" />
<ipResolverPort min="" max="" />
<httpPortNo min="" max="" />
<multicastIpAddrType opt="IPV4,IPV6"/>
<!--req,IP address type-->
<gatewayIpAddrType opt="IPV4,IPV6"/>
<!--req,IP address type-->
<PPPOE>
<pppoe opt="enable,disable"/>
<pppoeUserNameLen min="" max="" />
<pppoeUserPassWordLen min="" max="" />
<pppoeIPAddrType opt="IPV4,IPV6"/>
<!--req,IP address type-->
</PPPOE>
<RTMPCfg>
<isSupportRTMPCfg opt="true,false"/>
<!--req, whether to support RTMP-->
```

```
<streamType opt="mainStream,subStream"/>
<!--req, supported stream type-->
</RTMPCfg>
<enableDNS>
<enable DDNS, opt="enable,disable"/>
</NetCfg>
<BondCfg>
<bondNum min="" max="" /><!--opt,ro,number of NIC supported bonding-->
<bondMode opt="balance-rr, active-backup, balance-xor, broadcast, 802.3ad, balance-tlb, balance-alb"/>
<!--opt, bonding mode: round robin mode, active-backup mode, XOR mode, broadcast mode, IEEE 802.3ad, link
aggregation, adaptive transmit load balancing, adaptive load balancing (virtualization)-->
<isSupportBondExtension opt="true,false"/>
<!--opt,whether to support bond extension-->
<NetworkInterfaceBond>
<!--opt -->
<EthernetPortNo min="" max="" />
<!--opt,ro, network port No.-->
</NetworkInterfaceBond>
</BondCfg>
<SatelliteTiming>
<enabled opt="true,false" default="true"/>
<!--req, enable satellite time control-->
</SatelliteTiming>
</NetAppAbility>
```

Example

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

F.271 XML_NetworkCap

NetworkCap capability message in XML format

```
<NetworkCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<isSupportWireless>
<!--req, xs:boolean, whether it supports accessing via wireless network, "true"-yes, "false"-no-->
<isSupportWireless>
<isSupportWAN><!--optional, xs:boolean, whether it supports WAN--></isSupportWAN>
<isSupportPPPoE>
<!--req, xs:boolean, whether it supports PPPoE, "true"-yes, "false"-no-->
<isSupportPPPoE>
<isSupportBond>
<!--req, xs:boolean, whether it supports NIC bonding, "true"-yes, "false"-no-->
<isSupportBond>
<isSupport802_1x>
<!--req, xs:boolean, whether it supports 802_1x-->
</isSupport802_1x>
<isSupportNtp>
<!--optional, xs:boolean, whether it supports NTP-->
</isSupportNtp>
<isSupportFtp>
```

```
<!--optional, xs:boolean, whether it supports FTP-->
</isSupportFtp>
<isSupportUpnp>
  <!--optional, xs:boolean, whether it supports UPnP-->
</isSupportUpnp>
<isSupportPNP>
  <!--optional, xs:boolean, whether it supports PnP-->
</isSupportPNP>
<isSupportDdns>
  <!--optional, xs:boolean, whether it supports DNS-->
</isSupportDdns>
<isSupportHttps>
  <!--optional, xs:boolean, whether it supports HTTPS-->
</isSupportHttps>
<SnmpCap><!--optional, whether it supports SNMP-->
<isSupport><!--req, xs:boolean--></isSupport>
</SnmpCap>
<isSupportExtNetCfg>
  <!--optional, xs:boolean, whether it supports configuring extended network parameters-->
</isSupportExtNetCfg>
<isSupportIPFilter>
  <!--optional, xs:boolean, whether it supports filtering IP address-->
</isSupportIPFilter>
<isSupportSSH opt="true"><!--optional, xs:boolean, whether it supports SSH--></isSupportSSH>
<isSupportNetPreviewStrategy><!--optional, xs:boolean--></isSupportNetPreviewStrategy>
<isSupportEZVIZ>
  <!--optional, xs:boolean, whether it supports Guarding Vision-->
</isSupportEZVIZ>
<isSupportEhome>
  <!--optional, xs:boolean, whether it supports EHome (ISUP) protocol-->
</isSupportEhome>
<isSupportWirelessServer>
  <!--optional, xs:boolean, whether it supports Wi-Fi hotspot-->
</isSupportWirelessServer>
<isSupportWPS><!--optional, xs:boolean, whether it supports (WPS) Wi-Fi Protected Setup--></isSupportWPS>
<isWirelessMutexWithWirelessServer><!--optional, xs:boolean, whether it supports configuring the wireless access
to be mutually exclusive with the wireless service--></isWirelessMutexWithWirelessServer>
<isSupportWirelessDial>
  <!--optional, xs:boolean, whether it supports wireless dial configuration-->
</isSupportWirelessDial>
<WPS><!--optional, Wi-Fi Protected Setup configuration-->
<NetworkInterfaceList size="2">
  <NetworkInterface>
    <id><!--req, xs:string, NIC ID--></id>
    <enabled><!--req, xs:boolean, whether the NIC is enabled--></enabled>
    <isSupportAutoConnect><!--optional, xs:boolean--></isSupportAutoConnect>
    <isSupportDevicePinCode>
      <!--optional, xs:boolean, whether it supports device PIN code-->
    </isSupportDevicePinCode>
    <isSupportDevicePinCodeUpdate>
      <!--optional, xs:boolean, whether it supports updating device PIN code-->
    </isSupportDevicePinCodeUpdate>
```

```
<ApPinCode><!--opt-->
<ssid min="" max="">
  <!--optional, xs:string, maximum and minimum SSID length that can be returned by device-->
</ssid>
<pinCode min="" max="">
  <!--optional, xs:string, maximum and minimum PIN code length that can be returned by device-->
</pinCode>
</ApPinCode>
</NetworkInterface>
</NetworkInterfaceList>
</WPS>
<isSupportMACFilter>
  <!--optional, xs:boolean, whether it supports filtering MAC address-->
</isSupportMACFilter>
<verificationCode max="">
  <!--optional, xs:string, the maximum length of verificationCode that can be returned by device-->
</verificationCode>
<WPSCap><!--opt-->
<isSupport><!--req, xs: boolean--></isSupport>
<isSupportAutoConnect><!--req, xs: boolean--></isSupportAutoConnect>
</WPSCap>
<NetWorkMode>
<workMode><!--optional, xs:string, network mode: "close,wifi,wifiAp"--></workMode>
</NetWorkMode>
<VerificationCodeModification><!--optional, xs:string, whether the verification code can be edited by the admin user-->
<verificationCodeType opt="normal,empty"></verificationCodeType>
<isSupportDeclarationURL><!--optional, xs:boolean, whether it supports URL declared by the service--></isSupportDeclarationURL>
<isSupportPrivacyPolicyURL><!--optional, xs:boolean, whether it supports the privacy policy URL--></isSupportPrivacyPolicyURL>
<verificationCodeModify opt="true,false">
  <!--optional, whether the verification code is edited: "true"-yes, "false"-no, no return-not support-->
</verificationCodeModify>
<Hyperlinks><!--opt-->
<declarationURL><!--optional, xs:string--></declarationURL>
<privacyPolicyURL><!--optional, xs:string--></privacyPolicyURL>
</Hyperlinks>
<isSupportVerificationCodeCheck>
  <!--optional, xs: boolean, whether it supports verifying and configuring the verification code, true-yes, if this node is not returned or the value of the returned node is false, it indicates that not support-->
</isSupportVerificationCodeCheck>
<isSupportOldVerificationCode><!--optional, xs:boolean, whether it supports old EZVIZ password configuration. The old password contains six uppercase letters--></isSupportOldVerificationCode>
</VerificationCodeModification>
<EZVIZSecretKey>
  <!--optional, whether it supports capability of editing verification code for Guarding Vision-->
<offlineStatus>
  <!--ro,dep,xs:string; it is valid when registerStatus values "false", device offline status, opt="secretKeyInvalid"-->
  <!--invalid verification code-->
</offlineStatus>
<secretKey min="0" max="64"><!--optional, xs:string, verification code for Guarding Vision--></secretKey>
```

```
</EZVIZSecretKey>
<isSupportplatformAccess><!--optional, xs:boolean, capability of accessing the platform, whether it supports filtering IP addresses that access to the platform--></isSupportplatformAccess>
<isSupportIntegrate><!--optional, xs:boolean--></isSupportIntegrate>
<isSupportIntelligentBoost><!--optional, xs:boolean, whether it supports bandwidth adaption--></isSupportIntelligentBoost>
<isSupportWebSocket><!--optional, xs:boolean--></isSupportWebSocket>
<isSupportWebSocketS><!--optional, xs:boolean--></isSupportWebSocketS>
<isSupportHostName>
  <!--optional, xs:boolean, whether it supports host name, related URI: /ISAPI/System/Network/interfaces/<ID>/hostName/capabilities?format=json-->
</isSupportHostName>
<isSupportResourceStatistics><!--optional, xs:boolean, whether supports network resource information--></isSupportResourceStatistics>
<isSupportBandwidthLimit><!--optional, xs:boolean--></isSupportBandwidthLimit>
<isSupportPOEPortsDisableServer><!--optional, xs:boolean--></isSupportPOEPortsDisableServer>
<isSupportPOEConfiguration><!--optional, xs:boolean--></isSupportPOEConfiguration>
<Adaption>
  <streamType opt="0,1,2,3,4,5,7,8,9,10">
    <!--stream types that support network self-adaptive during live view: 0-main stream, 1-sub-stream, 2-third stream, 3-virtual stream, 4-stream 5, 5-stream 6, 7-stream 7, 8-stream 8, ..., and so on-->
  </streamType>
  <isSupportPlayback><!--optional, xs: boolean, whether it supports self-adaptive during playback--></isSupportPlayback>
</Adaption>
<isSupportVideoImgDB><!--optional, xs:boolean, whether it supports configuring the image and video library--></isSupportVideoImgDB>
<isSupportEventDataOverWebSocket opt="true,false"><!--optional, xs:boolean, whether it supports uploading events via WebSocket--></isSupportEventDataOverWebSocket>
<isSupportDynamicHostName><!--optional, xs:boolean, whether it supports configuring dynamic domain name--></isSupportDynamicHostName>
<isSupportRFIDData><!--optional, xs:boolean, whether it supports configuration of RFID data collection--></isSupportRFIDData>
<isSupportwifiProbeSSID><!--optional, xs:boolean, whether it supports SSID configuration of Wi-Fi probe--></isSupportwifiProbeSSID>
<isSupportPOEPortsDisableAdaptiveServer><!--optional, xs:boolean--></isSupportPOEPortsDisableAdaptiveServer>
<isSupportGetLinkSocketIP><!--optional, xs:boolean, whether it supports get the current linked SocketIP--></isSupportGetLinkSocketIP>
<isSupportTrafficMonitor><!--optional, xs:boolean, whether it supports traffic monitoring of 4G network card--></isSupportTrafficMonitor>
<isSupportDMSAuthInfo><!--optional, xs:boolean, whether it supports DMS authentication information management--></isSupportDMSAuthInfo>
<isSupportWiredandWirelessTrafficMonitor><!--optional, xs:boolean, whether it supports monitoring traffic of wired NIC and wireless NIC--></isSupportWiredandWirelessTrafficMonitor>
<isSupport4G><!--optional, xs:boolean, whether it supports 4G module--></isSupport4G>
<isSupport4GConfig><!--optional, xs:boolean, whether it supports 4G configuration--></isSupport4GConfig>
<isSupportSipTransmit><!--optional, xs:boolean, whether it supports configuring SIP transmission parameters--></isSupportSipTransmit>
<isSupportWifiCascade><!--optional, xs:boolean, whether it supports enabling automatic cascading of device Wi-Fi, related URI: /ISAPI/System/Network/wifiCascade?format=json --></isSupportWifiCascade>
<isSupportRelativeInfo><!--optional, xs:boolean, whether it supports getting information of connected devices, related URI: /ISAPI/System/Network/wifiCascade/relativeInfo?format=json--></isSupportRelativeInfo>
```

```
<isSupportEZVIZUnbind><!--optional, xs:boolean, whether it supports unbinding devices from the Guarding Vision account, related URI: /ISAPI/System/Network/EZVIZ/unbind?format=json--></isSupportEZVIZUnbind>
<isSupportEZVIZQRcode><!--optional, xs:boolean, whether it supports getting the Guarding Vision QR code, related URI: /ISAPI/System/Network/EZVIZ/QRCode?format=json--></isSupportEZVIZQRcode>
<isSupportEZVIZTiming><!--optional, xs:boolean, whether the device supports timing of Guarding Vision (related URI: /ISAPI/System/Network/EZVIZ)--></isSupportEZVIZTiming>
</NetworkCap>
```

F.272 XML_networkExtension

networkExtension message in XML format

```
<networkExtension version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <multicastAddress><!--opt-->
    <ipVersion><!--req, xs: string, "v4,v6,dual"--></ipVersion>
    <ipAddress><!--dep, xs: string--></ipAddress>
    <ipv6Address><!--dep, xs: string--></ipv6Address>
  </multicastAddress>
  <enVirtualHost><!--opt, xs: Boolean--></enVirtualHost>
</networkExtension>
```

F.273 XML_NetworkSetting

NetworkSetting message in XML format

```
<NetworkSetting version="2.0">
  /* Wi-Fi capability description*/
  <WirelessSetting>
    <InterfaceMode>
      /*0-auto-switch, 1-wired mode */
      <Range>0,1</Range>
    </InterfaceMode>
    <NetworkMode> //network model
  /*0 -managed; 1-ad-hoc*/
  <Range>0,1</Range>
  </NetworkMode>
  <NotSupportAutoDNS>true</NotSupportAutoDNS><!--req-->
  <SecurityMode> //security mode
    <Range>0,1,2,3,4,5</Range> /* 0-not encrypted; 1-wep; 2-wpa-psk; 3 wpa-enterprise; 4 wpa2-psk; 5 wpa2-enterprise*/
    <WEP> //WEP encrypt mode
    <AuthenticationType> //WEP authentication type
      //0-open, 1-share
      <Range>0,1</Range>
    </AuthenticationType>
    <WEPKeyLength> //WEP authentication key size
      //0-64 bit, 1-128 bit; 2-152 bit
      <Range>0,1,2</Range>
    </WEPKeyLength>
```

```
<WEPKeyType>// WEP private key type
//0-hexadecimal, 1-ASCII code
<Range>0,1</Range>
</WEPKeyType>
</WEP>
<WPA_PSK>//WAP_PSK authentication type
<WPAKeyLength>
<Min>8</Min>
<Max>63</Max>
</WPAKeyLength>
<WPAEncryptType>//WAP_PSK encrypt mode
//0-AES, 1-TKIP
<Range>0,1</Range>
</WPAEncryptType>
</WPA_PSK>
<WPA_enterprise>
<WPAEncryptType>
//0-AES, 1-TKIP
<Range>0,1</Range>
</WPAEncryptType>
</AuthType><!-- req, authentication type-->
<Range>0,1,2</Range><!-- req, authentication types: 0-EAP_TTLS, 1-EAP_PEAP, 2-EAP_TLS-->
</AuthType>
<EAP_TTLS>
<EapolVersion><!-- req, EAPOL version: 0-version 1, 1-version 2-->
<Range>0,1</Range>
</EapolVersion>
<InterAuthType><!-- req, internal authentication methods: 0-GTC, 1-MD5, 2-MSCHAPV2-->
<Range>0,1</Range>
</InterAuthType>
<AnonymousIdentityLength><!-- req, anonymous identity-->
<Min>0</Min>
<Max>32</Max>
</AnonymousIdentityLength>
<UserNameLength><!-- req, user name-->
<Min>1</Min>
<Max>32</Max>
</UserNameLength>
<PasswordLength><!-- req, password-->
<Min>1</Min>
<Max>32</Max>
</PasswordLength>
<Certificate>
<Certtype>
<Range>0,1,2</Range><!-- req, certificate file types: 0-CA, 1-Certificate, 2-key file--->
</Certtype>
<FileType>
<Range>0,1</Range><!-- req, certificate file types: 0-PEM,1-PFX--->
</FileType>
</Certificate>
</EAP_TTLS>
<EAP_PEAP>
```

```
<EapolVersion><!-- req,EAPOL version: 0-version 1, 1-version 2-->
<Range>0,1</Range>
</EapolVersion>
<InterAuthType><!-- req, internal authentication methods: 0-GTC, 1-MD5, 2-MSCHAPV2-->
<Range>0,1,2</Range>
</InterAuthType>
<PeapVersion><!-- req,PEAP version, 0-version 0, 1-version 1-->
<Range>0,1</Range>
</PeapVersion>
<PeapLabel><!-- req, PEAP tag: 0-old tag, 1-new tag-->
<Range>0,1</Range>
</PeapLabel>
<AnonymousIdentityLength><!-- req, anonymous identity-->
<Min>0</Min>
<Max>32</Max>
</AnonymousIdentityLength>
<UserNameLength><!-- req, user name-->
<Min>1</Min>
<Max>32</Max>
</UserNameLength>
<PasswordLength><!-- req, password-->
<Min>1</Min>
<Max>32</Max>
</PasswordLength>
<Certificate>
<Certtype>
<Range>0,1,2</Range><!-- req, certificate types: 0-CA, 1-Certificate, 2-key file-->
</Certtype>
<FileType>
<Range>0,1</Range><!-- req, certificate file types: 0-PEM,1-PFX-->
</FileType>
</Certificate>
</EAP_PEAP>
<EAP_TLS>
<EapolVersion><!-- req, EAPOL version: 0-version 1, 1-version 2-->
<Range>0,1</Range>
</EapolVersion>
<IdentityLength><!-- req, identity-->
<Min>1</Min>
<Max>32</Max>
</IdentityLength>
<PrivateKeyPswdLength><!-- req, private key password-->
<Min>1</Min>
<Max>32</Max>
</PrivateKeyPswdLength>
<Certificate>
<Certtype>
<Range>0,1,2</Range><!-- req, certificate types: 0-CA, 1-Certificate, 2-key file-->
</Certtype>
<FileType>
<Range>0,1</Range><!-- req, certificate file types: 0-PEM, 1-PFX-->
</FileType>
```

```

</Certificate>
</EAP_TLS>
</WPA_enterprise>
</SecurityMode>
<closeWifi opt="true,false"/><!--opt,whether to support disabling Wi-Fi: true-support, false-not support-->
<mutexAbility opt="wirelessServer"/><!-- opt, it is mutually exclusive with Wi-Fi hotspots-->
</WirelessSetting>
<WPS>
<Enabled>
<Range>0,1</Range><!-- req,whether to enable WPS: 0-disable, 1-enable-->
<Default>1</Default>
</Enabled>
<WPSConnect>
<Range>0,1,2,3</Range><!-- req,WPS connection methods: 0-PBC, 1-AP PIN, 2-device PIN code-->
</WPSConnect>
<isNotSupportUpdatePinCode><!--opt,xs:boolean,"true"-->"NET_DVR_GET_DEVICE_PIN" is invalid--></
isNotSupportUpdatePinCode>
</WPS>
<MessageConfig>
<enable opt="true,false"/>
<whiteListNum>8</whiteListNum>
<PhoneCfg>
<phoneNumLen>32</phoneNumLen>
<supportEntry opt="SMSAlarm, SMSCtrl, CallCtrl"/>
<SMSAlarmType opt="diskfull, diskerror, nicbroken, ipconflict, illaccess, AlarmInErr, tamper, vmd, wireless, pir,
callhelp, MOTION, hideAlarm, AudioDetection, scenechangeDetection, defocusDetection, facedetection,
LineDetection, FieldDetection, regionEntrance, regionExiting, loitering, group, rapidMove, parking,
unattendedBaggage, attendedBaggage, all"/><!-- req, xs:string, event type that triggers message alarms, "all"-->
<!-- determine the event type that triggers message alarms based on event linkage actions, related URL (/ISAPI/Event/
triggers/<ID>) -->
<SMSCtrlType opt="messageReboot,dial,getDialStatus,SMSAlarm "/><!-- opt, xs:string, message
operations:"messageReboot"-reboot message, "dial"-enable/disable dial-up, "getDialStatus"-get dial-up status,
"SMSAlarm"- receive/ reject message linkage-->
<name min="0" max="32"><!-- opt, xs:string, contact name, which is sensitive information that needs to be
encrypted--></name>
<maxListMemberNum><!-- opt, xs:integer, the maximum number of contacts, which is 8 by default, and the
node does not return--></maxListMemberNum>
</PhoneCfg>
</MessageConfig>
<SendSms>
<phoneNumLen>32</phoneNumLen>
<msgLen>140</msgLen>
</SendSms>
<WifiDHCPAddr>
<enable opt="true,false"/><!--req, whether to support configuring the DHCP range under Wi-Fi mode: true-support,
false-not support-->
</WifiDHCPAddr>
<WifiClientListInfo>
<enable opt="true,false"/><!--req, whether to support getting the client information connected by Wi-Fi: true-
support, false-not support-->
</WifiClientListInfo>
<support64bitKey opt="WPA-personal, WPA2-personal"/><!--opt, whether to support 64-bit hexadecimal passwords;

```

```
only WPA-personal and WPA2-personal support it-->
</NetworkSetting>
```

F.274 XML_OcrConfigCap

XML message about the capability for configuring the parameters of a scene

```
<?xml version="1.0" encoding="utf-8"?>
<OcrConfigCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <Scenes>
    <sceneId min="1" max="24"><!--req, xs:integer; scene No. (ranges from 1 to 24)--></sceneId>
    <videoResolution><!--req, resolution-->
      <width><!--req, xs:integer, resolution (width)--></width>
      <height><!--req, xs:integer, resolution (height)--></height>
    </videoResolution>
    <sceneParam><!--req-->
      <enabled opt="true,false"><!--req, xs:integer; whether to enable a scene: "true"-0 (yes), "false"-1 (no)--></enabled>
      <sceneName><!--req, xs:string; scene name--></sceneName>
      <sceneMode opt="none,preset,capture"><!--req, xs:string; scene mode: "none" (none), "preset" (preset scene), "capture" (capture scene)--></sceneMode>
        <bindPreset min="" max=""><!--dep, xs:integer; bind preset--></bindPreset>
        <sceneLocation min="" max=""><!--req, xs:integer; scene location: 0 (seaside left), 1 (seaside right), 2 (landsdie left), 3 (landsdie right), 4 (container side 1 composed of width and height), 5 (container side 2 composed of width and height), 6 (trailer No.1), 7 (trailer No.2), 8 (container top), 9 (container bottom), 10 (container top left divided by length), 11 (container top right divided by length), 12 (reserved 1), 13 (reserved 2), 14 (reserved 3), 15 (reserved 4)--></sceneLocation>
          <containerType min="" max=""><!--req, xs:integer; types of container: 0 (support 20 foot or 40 foot containers), 1 (only support 20 foot containers), 2 (only support 40 foot containers)--></containerType>
            <isCarMove opt="true,false"><!--req, xs:boolean; whether to recognize the moving truck--></isCarMove>
            <isCover opt="true,false"><!--req, xs:boolean; whether to recognize the flat deck--></isCover>
            <isRecognized opt="true,false"><!--req, xs:boolean, whether to recognize the container No.--></isRecognized>
            <isTrailerRecognized opt="true,false"><!--req, xs:boolean; whether to recognize the trailer No.--></isTrailerRecognized>
              <isLoadEnd opt="true,false"><!--req, xs:boolean, whether it is a loading end point--></isLoadEnd>
              <isUnLoadEnd opt="true,false"><!--req, xs:boolean, whether it is an unloading end point--></isUnLoadEnd>
            </sceneParam>
            <sceneCondition><!--scene condition parameters-->
              <loadCondition><!--loading condition parameters-->
                <rulesRelation opt="and,or"><!--req, xs:string; rule relations: "and" (and), "or" (or)--></rulesRelation>
                <Rules>
                  <signal opt="none,on2off,off2on"><!--req, xs:integer; signal: none (none), on2off (on to off), off2on (off to on)--></signal>
                    <laneNo min="" max=""><!--req, xs:integer, lane No.--></laneNo>
                    <centerPosX min="" max=""><!--req, xs:integer, X-axis central coordinate--></centerPosX>
                    <ruleX min="" max=""><!--req, xs:integer; types of X-axis rule: 0 (none), 1 (greater than), 2 (smaller than), 3 (range)--></ruleX>
                      <paramX min="" max=""><!--req, xs:short; X-axis range--></paramX>
                      <centerPosY min="" max=""><!--req, xs:short; Y-axis central coordinate--></centerPosY>
                      <ruleY min="" max=""><!--req, xs:integer; types of Y-axis rule: 0 (none), 1 (greater than), 2 (smaller than), 3 (range)--></ruleY>
                </Rules>
              </loadCondition>
            </sceneCondition>
          </bindPreset>
        </sceneLocation>
      </sceneParam>
    </videoResolution>
  </Scenes>
</OcrConfigCap>
```

```
<paramY min="" max=""><!--req, xs:short; Y-axis range--></paramY>
</Rules>
</loadCondition>
<unloadCondition><!--unload condition parameters-->
<rulesRelation opt="and,or"><!--req, xs:string; rule relations: "and" (and), "or" (or)--></rulesRelation>
<Rules>
    <signal opt="none,on2off,off2on"><!--req, xs:integer; signal: none (none), on2off (on to off), off2on (off to on)-->
</signal>
    <laneNo min="" max=""><!--req, xs:integer, lane No.--></laneNo>
    <centerPosX min="" max=""><!--req, xs:integer, X-axis central coordinate--></centerPosX>
    <ruleX min="" max=""><!--req, xs:integer; types of X-axis rule: 0 (none), 1 (greater than), 2 (smaller than), 3 (range)--></ruleX>
        <paramX min="" max=""><!--req, xs:short; X-axis range--></paramX>
        <centerPosY min="" max=""><!--req, xs:short; Y-axis central coordinate--></centerPosY>
        <ruleY min="" max=""><!--req, xs:integer; types of Y-axis rule: 0 (none), 1 (greater than), 2 (smaller than), 3 (range)--></ruleY>
            <paramY min="" max=""><!--req, xs:short; Y-axis range--></paramY>
            </Rules>
        </unloadCondition>
    </sceneCondition>
<ocrRegion>
    <Line>
        <RegionCoordinatesList size="2"><!--req, lane line coordinate-->
            <RegionCoordinates><!--opt-->
                <positionX><!--req, xs:integer; coordinate--></positionX>
                <positionY><!--req, xs:integer; coordinate--></positionY>
            </RegionCoordinates>
        </RegionCoordinatesList>
    </Line>
    <DomainList>
        <Domain>
            <RegionCoordinatesList size="2"><!--req, recognition area-->
                <RegionCoordinates><!--opt-->
                    <positionX><!--req, xs:integer; coordinate--></positionX>
                    <positionY><!--req, xs:integer; coordinate--></positionY>
                </RegionCoordinates>
            </RegionCoordinatesList>
        </Domain>
    </DomainList>
</ocrRegion>
</Scenes>
</OcrConfigCap>
```

F.275 XML_OnlineUpgradeCap

OnlineUpgradeCap message in XML format

```
<OnlineUpgradeCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <firmwareNum max="" />
        <!--req, the number of online upgrade packages, including full package and incremental package. Full package is
```

used for upgrading whole firmware, while incremental package is used for upgrade certain unit, such as openssl library-->

```
<firmwareCode max="" />
  <!--req, the maximum length of firmware code-->
<firmwareVersion max="" />
  <!--req, the maximum length of version-->
<firmwareCodeNumOnce max="" />
  <!--req, the maximum number of firmware codes can be obtained each time-->
<upgradePercent min="" max="" />
  <!--req-->
<Version>
  <!--req, upgrade package version information-->
  <newVersion max="" />
    <!--req-->
  <changeLog max="" />
    <!--req-->
</Version>
<DeviceParameter>
  <!--opt, online upgrade parameters-->
  <isSupportAutoDownloadPackage>
    <!--opt, xs:boolean, "true,false", whether supports automatic download of upgrade package-->
  </isSupportAutoDownloadPackage>
  <notSupportAutoUpgrade>
    <!--opt, xs:boolean, "true,false", whether not support automatic download of upgrade package and automatic upgrade-->
  </notSupportAutoUpgrade>
  <isSupportTimingUpgrade>
    <!--opt, xs:boolean, "true,false", whether supports scheduled upgrade-->
  </isSupportTimingUpgrade>
</DeviceParameter>
<ManualDownloadPackage>
  <!--opt, manually download upgrade package-->
  <supportOperation opt="start,cancel,pause,resume" />
    <!--opt, supported operations: "start,cancel,pause,resume"-->
</ManualDownloadPackage>
<isSupportIgnoreCurrentVersion>
  <!--opt, xs:boolean, "true,false", whether supports ignoring current version-->
</isSupportIgnoreCurrentVersion>
</OnlineUpgradeCap>
```

F.276 XML_OnlineUpgradeServer

OnlineUpgradeServer message in XML format

```
<OnlineUpgradeServer version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <connectStatus><!--ro, req xs:boolean, online upgrade server connection status--></connectStatus>
</OnlineUpgradeServer>
```

F.277 XML_OnlineUpgradeStatus

OnlineUpgradeStatus message in XML format

```
<OnlineUpgradeStatus version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <status>
    <!--ro, req, xs:string,
    "notUpgrade,upgrading,successful,languageMismatch,writeFlashError,packageTypeMismatch,packageVersionMismatch,netUnreachable,unknownError"-->
  </status>
  <percent><!-- ro, req, xs:integer "0-100" --></percent>
</OnlineUpgradeStatus>
```

F.278 XML_OnlineUpgradeVersion

OnlineUpgradeVersion message in XML format

```
<OnlineUpgradeVersion version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <newVersionAvailable><!--ro,req,xs:boolean, whether there is new verion of upgrade package--></newVersionAvailable>
  <newVersion><!--ro, dep,xs:string, new version No.--></newVersion>
  <changeLog><!--ro, dep,xs:string, update content of new version--></changeLog>
</OnlineUpgradeVersion>
```

F.279 XML_ParkAction

ParkAction message in XML format

```
<ParkAction version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled><!--required, xs: boolean, whether to enable parking--></enabled>
  <Parktime><!--required, xs:integer, parking time, unit: second--></Parktime>
  <Action> <!--parking actions-->
    <ActionType>
      <!--required, xs:string, action type:
      "atuoscan,framescan,randomscan,panoramascan,patrol,pattern,preset,areaScan,combinedPath,sceneTrace"-->
    </ActionType>
    <ActionNum>
      <!--required, xs:integer, parking No. This node will be used when <ActionType> is "patrol", "pattern", or "preset".
      For other parking action types, this node should be set to 0-->
    </ActionNum>
    <eventType>
      <!--optional, string, event type: "thermalVehicleDetection", "smokeAndFireDetection" (smoke and fire detection),
      "wasteGasDetection", "shipsDetection", "behavior" (behavior analysis)-->thermalVehicleDetection
    </eventType>
  </Action>
</ParkAction>
```

F.280 XML_POSAbility

XML message about POS capability

```
<?xml version="1.0" encoding="utf-8"?>
<POSAbility>
  <FilterRule>
    <maxFilterNum><!--required, maximum number of rules can be configured--></maxFilterNum>
    <enable opt ="true,false"/><!--required, whether to enable rule-->
    <characterSetType opt="0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18"/>
    <!--required, character set types supported by device-->
    <protocolType opt="Generic,AVE VSI-ADD,AVE Vnet,EPSON,NUCLEUS"/>
    <Generic><!--required, Generic protocol parameters-->
      <caseSensitive opt="true,false"/><!--required, whether it is case sensitive-->
      <GenericStartCond>
        <exclusive opt="true,false"/>
        <!--required, whether it contains the flag; currently, only the start flag is supported, it can control the start and end-->
        <maxSignalDataLen><!--required, maximum length of flag--></maxSignalDataLen>
        <supportAnyChar opt="true,false"/>
        <!--required, whether it supports any character: "true"-yes, "false"-no-->
        <supportHex opt="true,false"/>
        <!--required, whether it supports hexadecimal: "true"-yes, "false"-no-->
      </GenericStartCond>
      <GenericEndCond>
        <exclusive opt="true,false"/>
        <!--required, whether it contains the flag-->
        <maxSignalDataLen><!--required, maximum length of flag--></maxSignalDataLen>
        <moreLine min="0" max="4"/>
        <!--required, continue to search more lines after the transaction-->
        <supportHex opt="true,false"/>
        <!--required, whether it supports hexadecimal: "true"-yes, "false"-no-->
      </GenericEndCond>
      <GenericLineSpaceCond>
        <maxSignalDataLen><!--required, maximum length of flag--></maxSignalDataLen>
        <supportHex opt="true,false"/>
        <!--required, whether it supports hexadecimal: "true"-yes, "false"-no-->
      </GenericLineSpaceCond>
      <GenericIgnoreCond>
        <maxSignalDataLen><!--required, maximum length of flag--></maxSignalDataLen>
      </GenericIgnoreCond>
    </Generic>
    <AVE>
      <!--required, AVE protocol parameters-->
      <posAddr min="0" max="10"/>
      <!--required, range of POS address No.-->
    </AVE>
    <NUCLEUS>
      <!--optional, NUCLEUS protocol parameters-->
      <employeeNo min="" max="" />
      <!--optional, employee No.-->
    </NUCLEUS>
  </FilterRule>
</POSAbility>
```

```
<terminalNo min="" max="" />
<!--optional, POS No.-->
<shiftNo min="" max="" />
<!--optional, shift No.-->
</NUCLEUS>
</FilterRule>
<ConnectMode>
  <!--required, connection mode capability-->
  <enable opt="true,false"/><!--required, whether its supports configuring connection mode-->
  <connectMode opt="TCPMonitor,TCPRecv,Serial232,UDPMonitor,Sniff,MultiCast,USBToRS232"/>
    <!--required, "TCPMonitor"-listening mode (TCP), "TCPRecv"-receive via TCP network, "Serial"-connect with serial port, "UDPMonitor"-listening mode (UDP), "Sniff"-network sniffer, "MultiCast", "USBToRS232"-convert USB to RS-232-->
    <TCPMonitor>
      <!--required, network listening in TCP mode-->
      <port min="0" max="65535"/>
      <!--required, available port number range-->
      <restrictRemoteIPv4 opt="true,false"/>
      <!--optional, IP limit of remote access-->
    </TCPMonitor>
    <UDPMonitor>
      <!--required, network listening in UDP mode-->
      <port min="0" max="65535"/>
      <!--required, available port number range-->
      <restrictRemoteIPv4 opt="true,false"/>
      <!--optional, IP limit of remote access-->
    </UDPMonitor>
    <Sniff>
      <!--required, network sniffer-->
      <enableSourcePort opt="true,false"/>
      <!--required, whether it supports filtering data source port-->
      <enableDestAddr opt="true,false"/>
      <!--required, whether it supports filtering destination address-->
      <enableDestPort opt="true,false"/>
      <!--required, whether it supports filtering destination port-->
      <sourcePort min="0" max="65535"/>
      <!--required, range of data source port number-->
      <destPort min="0" max="65535"/>
      <!--required, range of destination address and port-->
    </Sniff>
    <multiCast>
      <!--required, multicast-->
      <port min="0" max="65535"/>
      <!--required, available port number range-->
    </multiCast>
    <USBToSerial232>
      <!--required, convert USB to RS-232 serial port-->
      <baudRate opt="50,75,110,150,300,600,1200,2400,4800,9600,19200,38400,57600,76800,115200"/>
      <!--required, available baud rate-->
      <dataBit opt="5,6,7,8"/>
      <!--required, number of data bits-->
      <stopBit opt="1,2"/>
```

```

<!--required, number of stop bits-->
<parity opt="none,odd,even"/>
<!--required, parity mode-->
<flowcontrol opt="none,hard,soft"/>
<!--required, flow control mode-->
<virtualSerialPort min="0" max="254"/>
<!--required, virtual serial port No.-->
</USBToSerial232>
</ConnectMode>
<ChanAssociationRule>
<!--required, linkage capability of channel and rule-->
<fontSize opt="8*16,16*32,32*64"/>
<!--required, available font size-->
<showPosInfo opt="true,false"/>
<!--required, whether the device supports overlay the POS information on the stream-->
<overlayMode opt="scroll,page"/>
<!--required, display type of character overlay-->
<delayTime min="5" max="3600"/>
<!--required, OSD delay time, unit: second-->
<supportOSDColor opt="true,false"/>
<!--required, whether the device supports setting OSD color-->
<timeOut min="5" max="3600"/>
<!--optional, timeout of receiving POS data, unit: second-->
</ChanAssociationRule>
<ExceptionHandle>
<alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,uploadcloud"/>
<!--required, exception handling types-->
</ExceptionHandle>
</POSAbility>

```

F.281 XML_PrivacyMaskRegion

PrivacyMaskRegion message in XML format

```

<PrivacyMaskRegion version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id><!--req, xs: integer--></id>
<enabled><!--req, xs: boolean--></enabled>
<RegionCoordinatesList><!--req-->
<RegionCoordinates><!--req-->
<positionX><!--req, xs: integer; coordinate--></positionX>
<positionY><!--req, xs: integer; coordinate--></positionY>
</RegionCoordinates>
</RegionCoordinatesList>
<privacymaskName><!--opt, xs: string--></privacymaskName>
<maskType>
<!--opt, xs:string "gray,red,yellow,blue,orange,green,transparent,half-transparent,mosaic,black"-->
</maskType>
<zoomdoorlimit><!--opt, xs: integer, the value is between 10 and 1000--></zoomdoorlimit>
</PrivacyMaskRegion>

```

F.282 XML_PTZAux

PTZAux message in XML format

```
<?xml version="1.0" encoding="utf-8"?>
<PTZAux version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id><!--req, xs:string, id--></id>
<type><!--req, ro, xs:string, "LIGHT,WIPER,FAN,HEATER", auxiliary type: light, wiper, fan, heater--></type>
<status><!--req, xs:string, "on,off", auxiliary status: turned on, turned off--></status>
</PTZAux>
```

F.283 XML_PTZChannel

PTZChannel message in XML format

```
<?xml version="1.0" encoding="utf-8"?>
<PTZChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id><!--req, xs:integer--></id>
<enabled><!--ro, req, xs:boolean--></enabled>
<serialNumber><!--req, xs:integer--></serialNumber>
<videoInputID><!--req, xs:integer--></videoInputID>
<panMaxSpeed><!--ro, opt, xs:integer, degrees/sec--></panMaxSpeed>
<tiltMaxSpeed><!--ro, opt, xs:integer, degrees/sec--></tiltMaxSpeed>
<presetSpeed><!--opt, xs:integer, 1..8--></presetSpeed>
<autoPatrolSpeed><!--opt, xs:integer, 0..100--></autoPatrolSpeed>
<keyBoardControlSpeed><!--opt, xs:integer, 0..100--></keyBoardControlSpeed>
<controlProtocol><!--opt, xs:string, "pelco-d,modbus-RTU,modbus-ASCII"--></controlProtocol>
<controlAddress><!--opt-->
<enabled><!--req, xs:boolean--></enabled>
<Address><!--opt, xs:string, 1-255--></Address>
</controlAddress>
<defaultPresetID><!--opt, xs:string, id--></defaultPresetID>
<PTZRs485Para><!--opt-->
<baudRate><!--req, xs:integer--></baudRate>
<dataBits><!--req, xs:integer--></dataBits>
<parityType><!--req, xs:string, "none, even, odd, mark, space"--></parityType>
<stopBits><!--req, xs:string, "1, 1.5, 2"--></stopBits>
<flowCtrl><!--req, xs:string, "none, software, hardware"--></flowCtrl>
</PTZRs485Para>
<manualControlSpeed>
<!--opt, xs:string, "pedestrian, nonMotorVehicle, motorVehicle, selfadaptive, compatible"-->
</manualControlSpeed>
<panSupport><!--optional, xs:boolean, whether it supports panning--></panSupport>
<tiltSupport><!--optional, xs:boolean, whether it supports tilting--></tiltSupport>
<zoomSupport><!--optional, xs:boolean, whether it supports zooming--></zoomSupport>
<PTPositiveDirection>
<!--optional, object, panning and tilting positive direction-->
<pan>
<!--optional, string, read-only, panning positive direction: left, right-->left
```

```
</pan>
<tile>
  <!--optional, string, read-only, tilting positive direction: up, down-->up
</tile>
</PTPositiveDirection>
</PTZChannel>
```

F.284 XML_PTZChanelCap

XML message about PTZ channel capability

```
<PTZChanelCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
<!--required, object, PTZ channel capability-->
<id min="1" max="16">
  <!--optional, int, index-->1
</id>
<enabled opt="true,false">
  <!--optional, boolean, whether to enable-->true
</enabled>
<AbsolutePanTiltPositionSpace><!--required, object, absolute PT range-->
  <XRange>
    <!--required, object, X-coordinate range-->
    <Min>
      <!--required, float, the minimum value-->0.000
    </Min>
    <Max>
      <!--required, float, the maximum value-->1.000
    </Max>
  </XRange>
  <YRange>
    <!--required, object, Y-coordinate range-->
    <Min>
      <!--required, float, the minimum value-->0.000
    </Min>
    <Max>
      <!--required, float, the maximum value-->1.000
    </Max>
  </YRange>
</AbsolutePanTiltPositionSpace>
<AbsoluteZoomPositionSpace><!--required, absolute zoom in/out range-->
  <ZRange>
    <!--required, object, zoom in/out range-->
    <Min>
      <!--required, float, the minimum value-->0.000
    </Min>
    <Max>
      <!--required, float, the maximum value-->1.000
    </Max>
  </ZRange>
</AbsoluteZoomPositionSpace>
```

```
<RelativePanTiltSpace><!--optional, object, relative PT range-->
<XRange><!--required, object, X-coordinate range-->
<Min>
  <!--required, float, the minimum value-->0.000
</Min>
<Max>
  <!--required, float, the maximum value-->1.000
</Max>
</XRange>
<YRange>
<!--required, object, Y-coordinate range-->
<Min>
  <!--required, float, the minimum value-->0.000
</Min>
<Max>
  <!--required, float, the maximum value-->1.000
</Max>
</YRange>
</RelativePanTiltSpace>
<RelativeZoomSpace><!--required, relative zoom in/out range-->
<ZRange>
<!--required, object, zooming in/out range-->
<Min>
  <!--required, float, the minimum value-->0.000
</Min>
<Max>
  <!--required, float, the maximum value-->1.000
</Max>
</ZRange>
</RelativeZoomSpace>
<ContinuousPanTiltSpace><!--required, object, continuous PT range-->
<XRange><!--required, object, X-coordinate range-->
<Min>
  <!--required, float, the minimum value-->0.000
</Min>
<Max>
  <!--required, float, the maximum value-->1.000
</Max>
</XRange>
<YRange>
<!--required, object, Y-coordinate range-->
<Min>
  <!--required, float, the minimum value-->0.000
</Min>
<Max>
  <!--required, float, the maximum value-->1.000
</Max>
</YRange>
</ContinuousPanTiltSpace>
<ContinuousZoomSpace><!--required, object, continuous zooming in/out range-->
<ZRange>
<!--required, object, zooming in/out range-->
```

```
<Min>
  <!--required, float, the minimum value-->0.000
</Min>
<Max>
  <!--required, float, the maximum value-->1.000
</Max>
</ZRange>
</ContinuousZoomSpace>
<MomentaryPanTiltSpace><!--required, object, instant PT range-->
  <XRange><!--required, object, X-coordinate range-->
    <Min>
      <!--required, float, the minimum value-->0.000
    </Min>
    <Max>
      <!--required, float, the maximum value-->1.000
    </Max>
  </XRange>
  <YRange>
    <!--required, object, Y-coordinate range-->
    <Min>
      <!--required, float, the minimum value-->0.000
    </Min>
    <Max>
      <!--required, float, the maximum value-->1.000
    </Max>
  </YRange>
</MomentaryPanTiltSpace>
<MomentaryZoomSpace><!--optional, object, instant zooming in/out range-->
  <ZRange>
    <!--required, object, zooming in/out range-->
    <Min>
      <!--required, float, the minimum value-->0.000
    </Min>
    <Max>
      <!--required, float, the maximum value-->1.000
    </Max>
  </ZRange>
</MomentaryZoomSpace>
<homePostionSupport><!--required, xs:boolean--></homePostionSupport>
<maxPresetNum>
  <!--required, xs:integer, max. supported preset number-->
</maxPresetNum>
<maxPatrolNum>
  <!--required,xs:integer, max. supported patrol number-->
</maxPatrolNum>
<maxPatternNum>
  <!--required,xs:integer, max. supported pattern number-->
</maxPatternNum>
<maxLimitesNum>
  <!--required,xs:integer, max. supported limit number-->
</maxLimitesNum>
<maxTimeTaskNum>
```

```
<!--required,xs:integer, max. supported timing task number-->
</maxTimeTaskNum>
<serialNumber min="1" max="4">
    <!--configuration capability of RS-485 serial port supported by current channel-->
</serialNumber>
<controlProtocol>
    <!--optional, xs:string, "pelco-d,modbus-RTU,modbus-ASCII", supported PTZ control protocol-->
</controlProtocol>
<controlAddress>
    <!--optional, xs:string, 0-255, address-->
</controlAddress>
<PTZRs485Para>
    <!--optional, PTZ RS485 parameters capability-->
    <baudRate>
        <!--required, xs:integer, baud rate-->
    </baudRate>
    <dataBits>
        <!--required, xs:integer, data bit-->
    </dataBits>
    <parityType>
        <!--required, xs:string, "none,even,odd,mark,space", verification type-->
    </parityType>
    <stopBits>
        <!--required, xs:string, "1,1.5,2" , stop bit-->
    </stopBits>
    <flowCtrl>
        <!--required, xs:string, "none, software, hardware", stream control type-->
    </flowCtrl>
</PTZRs485Para>
<PresetNameCap>
    <!--optional, preset name capability-->
    <presetNameSupport>
        <!--optional,xs:boolean, whether to support preset name?-->
    </presetNameSupport>
    <maxPresetNameLen>
        <!--dependent,xs:integer, preset name length-->
    </maxPresetNameLen>
    <specialNo/>
        <!--dependent, special preset-->
</PresetNameCap>
<isSupportPosition3D>
    <!--optional, xs:boolean, whether to support 3D position-->
</isSupportPosition3D>
<isSupportManualTrack>
    <!--optional,xs:boolean, whether to support manual tracking position(NET_DVR_PTZ_MANUALTRACE)?-->
</isSupportManualTrack>
<manualControlSpeed opt="compatible,pedestrian,nonMotorVehicle,motorVehicle,selfadaptive">
    <!--optional,xs:string, manual control speed: "compatible"-compatible mode, "pedestrian"-pedestrian,
    "nonMotorVehicle"-non-motor vehicle, "motorVehicle"-motor vehicle, "selfadaptive"-self adaptive-->
</manualControlSpeed>
<isSpprtPtzlimiteds>
    <!--optional,xs:boolean, whether to support PTZ limitation-->
```

```

</isSpprtPtzlimiteds>
<ParkAction>
  <enabled><!--required, xs: boolean, whether to enable parking--></enabled>
  <Parktime><!--required, xs:integer, parking time, unit: second--></Parktime>
  <Action> <!--parking actions-->
    <ActionType>
      <!--required, xs:string, action type:
"atuoscan,framescan,randomscan,panoramascan,patrol,pattern,preset,areaScan,combinedPath,sceneTrace"-->
    </ActionType>
    <ActionNum>
      <!--required, xs:integer, parking No. This node will be used when <ActionType> is "patrol", "pattern", or "preset". For other parking action types, this node should be set to 0-->
    </ActionNum>
    <eventType>
      <!--optional, string, event type: "thermalVehicleDetection", "smokeAndFireDetection" (smoke and fire detection), "wasteGasDetection", "shipsDetection", "behavior" (behavior analysis)-->thermalVehicleDetection
    </eventType>
  </Action>
</ParkAction>
<TimeTaskList>
  <enabled><!--req, xs: boolean--></enabled>
  <Parktime min="" max=""><!--req, xs:integer, seconds--></Parktime>
  <TimeTaskBlock>
    <dayOfWeek><!--req, xs: integer, day of the week based on ISO8601, "1"-Monday, ...--></dayOfWeek>
    <TimeTaskRange>
      <TaskID min="" max=""><!--req, xs: string; ID--></TaskID>
      <beginTime><!--req, xs: time, ISO8601 time--></beginTime>
      <endTime> <!--req, xs: time, ISO8601 time--></endTime>
      <Task>
        <TaskType
          <!--req, xs: strings-->
        </TaskType>
        <patrolTaskNum min="" max=""><!--dep, xs: integer, from 0 to 8--></patrolTaskNum>
        <patternTaskNum min="" max=""><!--dep, xs: integer, from 0 to 8--></patternTaskNum>
        <presetTaskNum min="" max=""><!--dep, xs: integer, from 0 to 8--></presetTaskNum>
        <auxoutputTaskNum min="" max=""><!--dep, xs: integer, from 0 to 8--></auxoutputTaskNum>
      </Task>
    </TimeTaskRange>
  </TimeTaskBlock>
  <isSupportTimeTaskCopy><!--req, xs: boolean--></isSupportTimeTaskCopy>
</TimeTaskList>
<Thermometry>
  <maxThermometryPresetNum>
    <!--optional, xs:integer-->
  </maxThermometryPresetNum>
</Thermometry>
<isSpprtPtzEagleFocusing>
  <!--optional, xs:boolean-->
</isSpprtPtzEagleFocusing>

```

```

<TrackingRatio>
    <!--optional, tracking zoom ratio-->
    <coefficient min="1" max="10" default="5">
        <!--optional, xs:integer, zoom ratio-->
    </coefficient>
</TrackingRatio>
<TrackInitPosition>
    <!--optional, tracking initial position-->
    <slaveCameralD>
        <!--optional, xs:integer, sub camera ID, 1..4 -->
    </slaveCameralD>
</TrackInitPosition>
<isSupportAbsoluteEx><!--optional,xs:boolean, whether to support extended capability for PTZ absolute position--></isSupportAbsoluteEx>
<isSupportCruise><!--optional,xs:boolean, whether to support auto-switch mode--></isSupportCruise>
<isSupportAreaScan><!--optional,xs:boolean, whether to support area scan--></isSupportAreaScan>
<isSupportFaceSnap3D><!--optional, xs: boolean--></isSupportFaceSnap3D>
<ManualPTZLockCap><!--optional, PTZ manual locking capability-->
    <lockStatus opt="lock,unlock"><!--optional, xs:string, PTZ locking status, the default status is locked--></lockStatus>
    <unLockTime><!--optional, xs:integer, remaining unlocking time duration (the automatic locking time duration is
300 s in the unlocked status)--></unLockTime>
</ManualPTZLockCap>
<isSupportOnepushSynchronizeFOV>
    <!--optional, xs:boolean, whether supports one-touch synchronize FOV, return true for supports, and no return
indicates not support-->
</isSupportOnepushSynchronizeFOV>
<isSupportLensCorrection>
    <!--optional, xs:boolean, whether supports lens correction, return true for supports, and no return indicates not
support-->
</isSupportLensCorrection>
<isSupportPTZTrackStatus>
    <!--optional, xs:boolean, whether to support getting PTZ tracking linkage status-->
</isSupportPTZTrackStatus>
<ptzctrlSupportSerialNumber opt="1,3"><!--optional, xs:string, RS-485 serial port range supported in PTZ control
mode--></ptzctrlSupportSerialNumber>
<LFPositionCap><!--optional, capability of calibrating position for smart linkage-->
    <elevation><!--optional, xs:integer, range: [-900,2700]--></elevation>
    <azimuth><!--optional, xs:integer, range: [0,3600]--></azimuth>
    <absoluteZoom><!--optional, xs:integer, range: [1,1000]--></absoluteZoom>
</LFPositionCap>
<pqrssZoom min="" max="" /><!--optional, xs: integer, zooming coordinates of Sony zoom camera module-->
<mnstFocus min="" max="" /><!--optional, xs: integer, focus coordinates of Sony zoom camera module-->
<isSupportPTZSave>
    <!--optional, xs: boolean, whether supports saving the current PTZ position information-->
</isSupportPTZSave>
<isSupportPTZSaveGet>
    <!--optional, xs:boolean, whether to support saving the PTZ position information of the current channel by GET
method-->
</isSupportPTZSaveGet>
<isSupportAutoGotoCfg>
    <!--optional, xs: boolean, whether supports automatically restoring to saved PTZ position: "true"-yes, this node is
not returned-no-->

```

```
</isSupportAutoGotoCfg>
<lockTime><!--optional, xs:integer, PTZ locking time, unit: second--></lockTime>
<isSupportPTZCtrlStatus><!--optional, xs:boolean, whether it supports getting the absolute PTZ position of the current channel: "true"-yes, "false"-no. If this node is not returned, it does not necessarily indicate that the device does not support this function--></isSupportPTZCtrlStatus>
<isSupportInitializationCtrl><!--optional, xs:boolean, whether it supports electric PTZ self-checking and control--></isSupportInitializationCtrl>
<isSupportComponentCtrl><!--optional, xs:boolean, whether it supports PTZ component control--></isSupportComponentCtrl>
<DoubleMirror><!--optional-->
<enabled opt="true,fasle" def="false"><!--required, xs:boolean--></enabled>
</DoubleMirror>
<MoveAutoTracking><!--optional, moving object tracking-->
<enabled opt="true,false"><!--required, xs:boolean, whether to enable this function--></enabled>
</MoveAutoTracking>
<isSupportTrackingRatioCfg><!--optional, xs:boolean, whether it supports configuring tracking parameters--></isSupportTrackingRatioCfg>
<isSupportGPSCtrl>
<!--optional, xs:boolean, whether current channel supports controlling pan and tilt by GPS, if supports, this node is returned and the value is true; if not, this node is not returned (related URI: /ISAPI/PTZCtrl/channels/<ID>/GPS?format=json)-->
</isSupportGPSCtrl>
<isSupportGyroVerify>
<!--optional, xs:boolean, whether device supports gyroscope attitude calibration (related URI: /ISAPI/PTZCtrl/channels/<ID>/gyroVerify?format=json)-->
</isSupportGyroVerify>
<isNotSupportPTZLimits>
<!--optional, boolean, read-only, whether the device supports PTZ limit: true (not support); if supports, this node will not be returned-->true
</isNotSupportPTZLimits>
<PTPositiveDirection>
<!--optional, object, read-only, panning and tilting positive direction-->
<pan opt="left,right">
<!--optional, string, read-only, panning positive direction: left, right-->left
</pan>
<tile opt="up,down">
<!--optional, string, read-only, tilting positive direction: up, down-->up
</tile>
</PTPositiveDirection>
<PTZOSDDisplay>
<!--optional, read-only, object, PTZ OSD parameters-->
<zoomlable opt="2sec, 5sec, 10sec, alwaysclose, alwaysopen">
<!--required, string, read-only, OSD duration of zoom status-->test
</zoomlable>
<azimuth opt="2sec, 5sec, 10sec, alwaysclose, alwaysopen">
<!--required, string, read-only, azimuth angle display duration while panning and tilting-->test
</azimuth>
<presetlable opt="2sec, 5sec, 10sec, alwaysclose, alwaysopen">
<!--required, string, read-only, preset name display duration when calling the preset-->test
</presetlable>
<positionDisplayFormat opt="PT,direction">
<!--optional, string, read-only, position display format: PT, direction-->PT
```

```
</positionDisplayFormat>
<actionStatusDisplayEnabled opt="true,fasle">
    <!--optional, boolean, read-only, whether to enable the PTZ status display-->true
</actionStatusDisplayEnabled>
</PTZOSDDisplay>
</PTZChanelCap>
```

F.285 XML_PTZOSDDisplay

PTZOSDDisplay message in XML format

```
<PTZOSDDisplay version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<zoomlable><!--req, xs:string, "2sec, 5sec, 10sec, alwaysclose, alwaysopen"--></zoomlable>
<azimuth><!--req, xs:string, "2sec, 5sec, 10sec, alwaysclose, alwaysopen"--></azimuth>
<presetlable><!--req, xs:string, "2sec, 5sec, 10sec, alwaysclose, alwaysopen"--></presetlable>
<actionStatusDisplayEnabled><!--opt, xs:boolean, display status or not--></actionStatusDisplayEnabled>
</PTZOSDDisplay>
```

F.286 XML_RacmCap

XML message about device storage capability

```
<?xml version="1.0" encoding="utf-8"?>
<RacmCap version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<isSupportZeroChan>
    <!--optional, xs:boolean, whether it supports channel-zero, "true"-yes, "false"-no-->
</isSupportZeroChan>
<inputProxyNums><!--optional, xs:integer, number of supported digital channels--></inputProxyNums>
<eSATAINums><!--optional, xs:integer, number of supported eSATAAs--></eSATAINums>
<miniSASINums><!--optional, xs:integer, number of supported miniSATAAs--></miniSASINums>
<nasNums><!--optional, xs:integer, number of supported NASs--></nasNums>
<ipSanNums><!--optional, xs:integer, number of supported IPSANs--></ipSanNums>
<isSupportRaid>
    <!--optional, xs:boolean, whether it supports RAID, "true"-yes, "false"-no-->
</isSupportRaid>
<isSupportExtHdCfg>
    <!--optional, xs:boolean, whether it supports HDD advanced management, "true"-yes, "false"-no-->
</isSupportExtHdCfg>
<isSupportTransCode><!--optional, xs:boolean, whether it supports auto-switch, "true"-yes, "false"-no--></isSupportTransCode>
<isSupportIpclImport>
    <!--optional, xs:boolean, whether it supports importing configuration files to network camera, "true"-yes, "false"-no-->
</isSupportIpclImport>
<NasMountType><!--optional-->
    <isNFSSupportAuthentication/><!--optional, xs:boolean, whether it supports NFS authentication, "true"-yes, "false"-no-->
    <isCIFSSupportAuthentication/><!--optional, xs:boolean, whether it supports CIFS authentication, "true"-yes, "false"-no-->
```

```

</NasMountType>
<isSupportIpcStreamType/><!--optional, xs:boolean-->
<isSupportIOInputProxy/><!--optional, xs:boolean, whether it supports dynamic alarm input channel: "true"-yes,
"false"-no-->
<isSupportIOOutputProxy/><!--optional, xs:boolean, whether it supports dynamic alarm output channel: "true"-yes,
"false"-no-->
<isSupportPTZRs485Proxy/><!--optional, xs:boolean, whether it supports dynamic PTZ485 channel, "true"-yes,
"false"-no-->
<isSupportSrcIDSearch/><!--optional, xs:boolean, whether it supports search by stream ID, "true"-yes, "false"-no-->
<isSupportReversePlayback/><!--optional, xs:boolean, whether it supports reverse playback, "true"-yes, "false"-no-->
<isSupportSMARTTest/><!--optional, xs:boolean, whether it supports HDD checking, "true"-yes, "false"-no-->
<isSupportDownloadByTime/><!--optional, xs:boolean, whether it supports downloading by time, "true"-yes, "false"-no-->
<pictureSearchType
opt="AllEvent,CMR,MOTION,ALARM,EDR,ALARMANDMOTION,Command,pir,wlsensor,callhelp,facedetection,FieldDetection,scenedetection,LineDetection,regionEntrance,regionExiting,loitering,group,rapidMove,parking,unattendedBaggage,attendedBaggage,vehicleDetection,manual,manualSnapShot,playSnapShot,allPic,evidence,illegalParking,crosslane,vehicleexist,lanechange,wrongdirection,congestion,turnround, parallelParking, pedestrian, construction, roadblock, abandonedObject, trafficAccident, fogDetection, personQueueCounting, personQueueTime, mixedTargetDetection, safetyHelmet, accessController, videoInterCom, GJD, Luminite, OPTEX, securityControlPanel, playCellphone, vibrationDetection, ATMPanel, ATMSurround, ATMFace, ATMSafetyCabin, temperatureIntervalMeasurement"/>
    <!--optional, xs:string, picture search conditions-->
<recordSearchType
opt="AllEvent,CMR,MOTION,ALARM,EDR,ALARMANDMOTION,Command,pir,wlsensor,callhelp,facedetection,FieldDetection,scenedetection,LineDetection,regionEntrance,regionExiting,loitering,group,rapidMove,parking,unattendedBaggage,attendedBaggage,vehicleDetection,manual,manualSnapShot,playSnapShot,AudioDetection,playCellphone,running,failDown,violentMotion,retention,allPerimeterEvent, allBehaviorEvent, vibrationDetection, ATMPanel, ATMSurround, ATMFace, ATMSafetyCabin, temperatureIntervalMeasurement"/>
    <!--optional, xs:string, video search conditions-->
<isSupportActivateIpc/><!--optional, xs:boolean, whether it supports activating network camera, "true"-yes, "false"-no-->
<isSupportCheckIpcSecurity><!--optional-->
    <isSupportCheckPassword/><!--optional, xs:boolean, whether it supports password verification, "true"-yes, "false"-no-->
</isSupportCheckIpcSecurity>
<isSupportMainAndSubRecord/>
    <!--optional, xs:boolean, whether it supports recording in main stream or sub-stream, "true"-yes, "false"-no-->
<isSupportSyncIPCPassword>
    <!--optional, xs:boolean, whether it supports synchronizing network camera's password, "true"-yes, "false"-no-->
</isSupportSyncIPCPassword>
<isSupportTransferIPC>
    <!--optional, xs:boolean, whether it supports network camera passthrough function, "true"-yes, "false"-no-->
</isSupportTransferIPC>
<isSupportPOS><!--optional, xs:boolean, whether it supports POS, "true"-yes, "false"-no--></isSupportPOS>
<isSupportPassBackBasicCfg>
    <!--optional, xs:boolean, whether it supports the ANR basic function of CVR, "true"-yes, "false"-no-->
</isSupportPassBackBasicCfg>
<PassBackTaskConfig><!--optional, task management function of ANR-->
    <isSupportAddTask><!--optional, xs:boolean, whether it supports adding task, "true"-yes, "false"-no--></isSupportAddTask>
        <isSupportSearchTask><!--optional, xs:boolean, whether it supports task search, "true"-yes, "false"-no--></isSupportSearchTask>

```

```
isSupportSearchTask>
  <isSupportControlTask><!--optional, xs:boolean, whether it supports task control, "true"-yes, "false"-no--></
isSupportControlTask>
  <isSupportDeleteTask><!--optional, xs:boolean, whether it supports deleting task, "true"-yes, "false"-no--></
isSupportDeleteTask>
</PassBackTaskConfig>
<PassBackPlanConfig><!--optional, task plan of ANR-->
  <isSupportAddPlan><!--optional, xs:boolean, whether it supports adding plan, "true"-yes, "false"-no--></
isSupportAddPlan>
  <isSupportSearchPlan><!--optional, xs:boolean, whether it supports plan search, "true"-yes, "false"-no--></
isSupportSearchPlan>
  <isSupportDeletePlan><!--optional, xs:boolean, whether it supports deleting plan, "true"-yes, "false"-no--></
isSupportDeletePlan>
</PassBackPlanConfig>
<IRaidCap/><!--optional, network RAID-->
<isSupportStorageExtraInfo>
  <!--optional, xs:boolean, whether it supports configuring storage additional information, "true"-yes, "false"-no-->
</isSupportStorageExtraInfo>
<isSupportRecordStatus><!--optional, xs:boolean--></isSupportRecordStatus>
<supportAIDTFSType opt="illegalParking,wrongdirection crosslane,vehicleexist,lanechange, turnround, evidence"/>
  <!--optional, xs:string, event picture search conditions supported by both AID and TFS-->
<isSupportRacmChannelsCap><!--optional, xs:boolean--></isSupportRacmChannelsCap>
<LockCap/><!--optional, lock or unlock video-->
<isSupportForamtAll><!--optional, xs:boolean, "true,false"--></isSupportForamtAll>
<isSupportExtendCabinetCfg>
  <!--optional, xs:boolean, whether it supports enclosure configuration-->
</isSupportExtendCabinetCfg>
<diskGroupNums><!--optional, xs:integer, number of supported HDD groups--></diskGroupNums>
<isSupportCountingSearchByUTC><!--optional, xs:boolean, whether it supports searching people counting results by
UTC time--></isSupportCountingSearchByUTC>
<isSupportPlaybackReverseByUTC><!--optional, xs:boolean, whether it supports reverse playback by UTC time--></
isSupportPlaybackReverseByUTC>
<isSupportWebPrivatePlaybackByUTC>
  <!--optional, xs: boolean, whether it supports playback based on Web private protocol-->
</isSupportWebPrivatePlaybackByUTC>
<isSupportFindCommonFileByUTC>
  <!--optional, xs: boolean, whether it supports extending the time zone for searching files-->
</isSupportFindCommonFileByUTC>
<isSupportFindEventFileByUTC><!--optional, xs: boolean, whether it supports extending the time zone for searching
files by event-->
</isSupportFindEventFileByUTC>
<isSupportSmartSearchRecordByUTC>
  <!--optional, xs: boolean, whether it supports extending the time zone for VCA search-->
</isSupportSmartSearchRecordByUTC>
<isSupportMRDSearchByTimeZone>
  <!--optional, xs: boolean, whether it supports extending the time zone for searching files by calendar-->
</isSupportMRDSearchByTimeZone>
<isSupportSearchRecordLabelByUTC>
  <!--optional, xs: boolean, whether it supports extending the time zone for searching video tags-->
</isSupportSearchRecordLabelByUTC>
<isSupportSearchPictureByUTC>
  <!--optional, xs: boolean, whether it supports extending the time zone for searching pictures-->
```

```
</isSupportSearchPictureByUTC>
<isSupportSmartSearchPictureByUTC>
  <!--optional, xs: boolean, whether it supports extending the time zone for searching pictures with smart
information-->
</isSupportSmartSearchPictureByUTC>
<isSupportFindLogByUTC>
  <!--optional, xs: boolean, whether it supports extending the time zone for searching log-->
</isSupportFindLogByUTC>
<isSupportUploadRecordByUTC>
  <!--optional, xs: boolean, whether it supports extending the time zone for uploading files to cloud storage-->
</isSupportUploadRecordByUTC>
<isSupportPlaybackByUTC>
  <!--optional, xs: boolean, whether it supports extending the time zone for playback by time and locating by time-->
</isSupportPlaybackByUTC>
<SecurityLog>
  <isSupportSecurityLog><!--optional, boolean, whether it supports security log--></isSupportSecurityLog>
  <isSupportLogServer><!--optional, boolean, whether it supports log server configuration--></isSupportLogServer>
  <isSupportLogServerTest><!--optional, xs: boolean, whether it supports log server test--></isSupportLogServerTest>
  <SecurityLogTypeList><!--required, xs: list, supported log type list-->
    <SecurityLogType>
      <primaryType><!--required, xs: string, major type--></primaryType>
      <secondaryType optional=""><!--required, xs: string, minor type--></secondaryType>
    </SecurityLogType>
  </SecurityLogTypeList>
</SecurityLog>
<iSptInputProxyChanCap>
  <!--optional, xs:boolean, whether it supports getting the capability of the digital channel-->
</iSptInputProxyChanCap>
<isSupportLogDataPackage>
  <!--optional, xs:boolean, whether it supports exporting logs: "true, false"-->
</isSupportLogDataPackage>
<logSearchTimeSpanNums>
  <!--optional, xs:integer, supported number of time periods for log search-->
</logSearchTimeSpanNums>
<isSupportManualRecord>
  <!--optional, xs:boolean, whether the device supports manual recording-->
</isSupportManualRecord>
<isSupportRemark><!--optional, xs:boolean, whether it supports file remarks--></isSupportRemark>
<FileUpload><!--optional, upload file-->
  <enabled><!--required, xs:boolean, whether it supports enabling file uploading--></enabled>
</FileUpload>
<PoliceInfoUpload><!--optional, upload police information-->
  <enabled><!--required, xs:boolean, whether it supports enabling police information uploading--></enabled>
</PoliceInfoUpload>
<PoliceInfo>
  <policeID><!--optional, xs:boolean, whether it supports police ID--></policeID>
  <policeCode><!--optional, xs:boolean, whether it supports police No.--></policeCode>
  <policeName><!--optional, xs:boolean, whether it supports police name--></policeName>
  <password><!--optional, xs:boolean, whether it supports police password--></password>
  <deviceID><!--optional, xs:boolean, whether it supports device ID--></deviceID>
</PoliceInfo>
<behaviorEventPicSearch
```

```

opt="allBehaviorEvent,running,group,violentMotion,failDown,playCellphone,peopleNumChange,leavePosition,retentio
n,sleepOnduty">
    <!--optional, xs:string, behavior analysis events supported by picture search, which is used for web display,
"allBehaviorEvent"-all events in behaviorEventPicSearch-->
</behaviorEventPicSearch>
<perimeterEventPicSearch
opt="allPerimeterEvent,linedetection,fielddetection,regionEntrance,regionExiting,loitering">
    <!--optional, xs:string, perimeter protection events that support picture search, which is used for web display,
"allPerimeterEvent"-all events in perimeterEventPicSearch-->
</perimeterEventPicSearch>
<isSupportAssignChannelID><!--optional, xs:boolean, whether it supports specify channel No. when adding IPC--></
isSupportAssignChannelID>
<isSupportAssignStreamID><!--optional, xs:boolean, whether it supports specify stream ID of the channel when
adding IPC--></isSupportAssignStreamID>
<isSupportTimeSearch><!--optional, xs:boolean, whether it supports searching for recording start and end time by
channel--></isSupportTimeSearch>
<CloudStorageServerCap><!--optional-->
    <isSupportCloudStorageParameter><!--optional, xs:boolean, whether it supports configuring cloud storage
parameters--></isSupportCloudStorageParameter>
    <isSupportCloudStoragePool><!--optional, xs:boolean, whether it supports configuring cloud storage pool
parameters--></isSupportCloudStoragePool>
</CloudStorageServerCap>
<CMSearchCount/><!--optional, xs:boolean, whether it supports searching for file quantity of dock station-->
<isSupportSSDSMARTTest>
    <!--optional, xs:boolean, whether the device supports SSD S.M.A.R.T detection (related URI: /ISAPI/ContentMgmt/
Storage/ssd/<ID>/SMARTTest/start)-->
</isSupportSSDSMARTTest>
<isSupportSpare><!--optional, boolean, whether the device supports hot spare configuration--></isSupportSpare>
<isSupportPTEventTableTemplate><!--optional, xs:boolean, whether the device supports importing/exporting Excel
templates of transparent transmission event type (related URI: /ISAPI/ContentMgmt/PTEventTableTemplate?
format=json)--></isSupportPTEventTableTemplate>
<isSupportPTEventTableFile><!--optional, xs:boolean, whether the device supports importing/exporting the Excel files
of transparent transmission event type (related URI: /ISAPI/ContentMgmt/PTEventTableFile?format=json)--></
isSupportPTEventTableFile>
</RacmCap>

```

F.287 XML_RecordAbility

RecordAbility message in XML format.

```

<xml version="1.0" encoding="utf-8">
<RecordAbility version="2.0"><!--req, video capability set description-->
    <Label><!--opt,whether support video tag--></Label>
    <CmdTriggerRecord><!--opt,whether support trigger video by command--></CmdTriggerRecord>
    <PackRecord><!--opt, whether support video package by time--></PackRecord >
    <RecordScheduleNum><!--req, number of video time quantum--></RecordScheduleNum>
    <recordPlanAbility opt="substream,passback,lockduration,recordbackup"/>
        <!--req, record plan ability, the corresponding structure is NET_DVR_RECORD_V30, just add the CVR, other support
by default-->
    <findBackupRecordAbility>

```

```
<!--req, whether supports searching video, this node does not appear if not support-->
</findBackupRecordAbility>
<MonthlyDistribution><!--opt, whether support monthly calendar searching--></MonthlyDistribution>
<DailyDistribution><!--opt, whether support calendar searching--></DailyDistribution>
<PlayBack>
    <controlSpeed min="" max="" /><!--req, whether support stream setting when playback-->
    <playBackReverseByName><!--req, whether support play reverse by file--></playBackReverseByName>
    <playBackReverseByTime><!--req, whether support play reverse by time--></playBackReverseByTime>
    <playbackBackupRecordAbility>
        <!--req, whether support searching playback video, non support non display-->
    </playbackBackupRecordAbility>
</PlayBack>
<MDRCondStreamType opt="main,sub">
    <!--opt, monthly query condition supported stream type-->
</MDRCondStreamType>
<MonthlyDistribution><!--opt, whether supports monthly calendar searching --></MonthlyDistribution>
<DailyDistribution><!--opt, whether supports calendar searching--></DailyDistribution>
<isSupportSearchIP><!--opt, xs:boolean, whether support returning IP parameter in search results-->
<sceneChangeDetectionRecord>
    <!--req, whether support scene change detection searching-->
</sceneChangeDetectionRecord>
<allAlarmTypeRecord>
    <!--opt, whether support "PIR|wireless alarm|distress call alarm"record, judging whether the function is supported,
you need judge whether the <PIRALarm> of alarm capability set exists -->
</allAlarmTypeRecord>
<CMDRecordParameter><!--opt, video parameter triggered by command-->
    <lockDuration min="0" max="0xffffffff"/><!--opt, lock time-->
    <backup opt="true,false"/><!--opt, whether to save-->
    <preRecord opt="true,false"/><!--opt, whether to prerecord-->
</CMDRecordParameter>
<supportGetCourseFile opt="true,false"/><!--opt, whether supports playback according to courses-->
<supportOptimalStreamType opt="true,false"/><!--opt, whether supports playback of optimal stream type-->
</RecordAbility>
```

F.288 XML_remotePermission

XML message about the configurations of remote permission

```
<remotePermission version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <record><!--opt, xs:boolean--></record>
    <booleant><!--opt, xs:boolean--></booleant>
    <preview><!--opt, xs:boolean--></preview>
    <videoChannelPermissionList><!--opt-->
        <videoChannelPermission><!--opt-->
            <id><!--req, corresponds to the video input channel ID--></id>
            <preview><!--opt, xs:boolean--></preview>
            <palyBack><!--opt, xs:boolean--></palyBack>
            <record><!--opt, xs:boolean--></record>
            <playBackDoubleVerification>
                <!--opt, xs:boolean, whether supports secondary authentication for playback and download-->
```

```

    </playBackDoubleVerification>
</videoChannelPermission>
</videoChannelPermissionList>
<ptzControl><!--opt, xs: boolean--></ptzControl>
<ptzChannelPermissionList><!--opt-->
<ptzChannelPermission><!--opt-->
  <id><!--req, corresponds to PTZ channel ID--></id>
  <ptzControl><!--opt, xs: boolean--></ptzControl>
</ptzChannelPermission>
</ptzChannelPermissionList>
<logOrStateCheck><!--opt, xs: boolean, permission to view log and status--></logOrStateCheck>
<parameterConfig><!--opt, xs: boolean, parameter configuration permission--></parameterConfig>
<restartOrShutdown><!--opt, xs: boolean, permission to reboot and shutdown--></restartOrShutdown>
<upgrade><!--opt, xs: boolean, permission to upgrade--></upgrade>
<voiceTalk><!--opt, xs: boolean--></voiceTalk>
<transParentChannel><!--opt, xs: boolean--></transParentChannel>
<contorlLocalOut><!--opt, xs: boolean--></contorlLocalOut>
<alarmOutOrUpload><!--opt, xs: boolean, permission to upload and output alarm--></alarmOutOrUpload>
<factoryReset><!--opt, xs:boolean, restore default parameters--></factoryReset>
<arm><!--opt, xs:boolean, arm--></arm>
<disarm><!--opt, xs:boolean, disarm--></disarm>
<accessControl><!--opt, xs:boolean, access control--></accessControl>
<subSysOrZoneArm><!--opt, xs:boolean, partition or zone arming permission--></subSysOrZoneArm>
<subSysOrZoneDisarm><!--opt, xs:boolean, partition or zone disarming permission--></subSysOrZoneDisarm>
<subSysOrZoneClearArm><!--opt, xs:boolean, permission to clear partition or zone alarms--></
subSysOrZoneClearArm>
<zoneBypass><!--opt, xs:boolean, permission to bypass zone--></zoneBypass>
<zoneBypassRecover><!--opt, xs:boolean, permission to recover bypassing zone--></zoneBypassRecover>
<IRaid>
  <rapidConfiguration><!--opt, xs:boolean--></rapidConfiguration>
  <reset><!--opt, xs:boolean--></reset>
</IRaid>
<subSystemList><!--opt, partitions linked to the operator when adding an operator or setting operator parameters-->
  <subSystem><!--opt, xs:integer, partition linked to the operator. When adding an operator or setting operator
parameters, one or more partitions can be linked to an operator. An operator should be linked to at least one partition
and up to four partitions--></subSystem>
</subSystemList>
<AIModelManagement><!--opt, xs:boolean, operation permission of AI model packages--></AIModelManagement>
<AITaskManagement><!--opt, xs:boolean, management and operation permission of AI tasks--></
AITaskManagement>
<subSysOrZoneArm>
  <!--read-only, optional, xs:boolean, arm permission of partition/alarm zone-->true
</subSysOrZoneArm>
<subSysOrZoneDisarm>
  <!--read-only, optional, xs:boolean, disarm permission of partition/alarm zone-->true
</subSysOrZoneDisarm>
<subSysOrZoneClearArm>
  <!--read-only, optional, xs:boolean, clear alarm permission of partition/alarm zone-->true
</subSysOrZoneClearArm>
<zoneBypass>
  <!--read-only, optional, xs:boolean, bypass permission in alarm zone-->true
</zoneBypass>

```

```
<zoneBypassRecover>
  <!--read-only, optional, xs:boolean, bypass recovery permission in alarm zone-->true
</zoneBypassRecover>
<operateOutput>
  <!--read-only, optional, xs:boolean, permission for operation on relay-->true
</operateOutput>
<supportLinkageSubSystemList>
  <!--read-only, optional, list of the partitions that support linkage-->
  <subSystem>
    <!--read-only, optional, xs:integer, the partition that supports linkage, value range:[1,64]-->1
  </subSystem>
</supportLinkageSubSystemList>
<subSystemList>
  <!--read-only, optional, xs:object, list of partitions-->
  <subSystem>
    <!--read-only, optional, xs:integer, partition, range:[1,64]-->1
  </subSystem>
</subSystemList>
<factoryReset>
  <!--read-only, optional, xs:boolean, recover default parameters-->true
</factoryReset>
<arm>
  <!--read-only, optional, xs:boolean, arm-->true
</arm>
<disarm>
  <!--read-only, optional, xs:boolean, disarm-->true
</disarm>
<accessControl>
  <!--read-only, optional, xs:boolean, access control-->true
</accessControl>
<restoreTamper opt="true,false">
  <!--optional, boolean, whether to restore tampering, this node is only valid for installer-->true
</restoreTamper>
<restoreConfirmedAlarm opt="true,false">
  <!--optional, boolean, whether to restore alarm acknowledgment, this node is for linked zones-->true
</restoreConfirmedAlarm>
<LPLListAudit>
  <!--optional, xs:boolean, allowlist/blocklist operation permission. Users with this permission can import, export, edit, delete, and search for license plate information in the allowlist or blocklist. See remarks for the URI for each operation-->true
</LPLListAudit>
</remotePermission>
```

Remarks

- Search for license plate information in the allowlist or blocklist: POST /ISAPI/Traffic/channels/<ID>/searchLPLListAudit;
import/export license plate information into/in allowlist or blocklist: GET/PUT /ISAPI/Traffic/channels/<ID>/licensePlateAuditData;
add/edit license plate information to/in the allowlist or blocklist: PUT /ISAPI/Traffic/channels/<channelID>/licensePlateAuditData/record?format=json;

delete license plate information in the allowlist or blocklist: PUT /ISAPI/Traffic/channels/<channelID>/DelLicensePlateAuditData?format=json.

F.289 XML_remotePermissionCap

XML message about the capability of managing the user permission (remote permission)

```
<remotePermissionCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<record><!--opt, xs: boolean--></record>
<booleant><!--opt, xs: boolean--></booleant>
<preview><!--opt, xs: boolean--></preview>
<videoChannelPermissionList><!--opt-->
<videoChannelPermission><!--opt-->
  <id><!--req, corresponds to the video input channel ID--></id>
  <preview><!--opt, xs: boolean--></preview>
  <palyBack><!--opt, xs: boolean--></palyBack>
  <record><!--opt, xs: boolean--></record>
  <playBackDoubleVerification>
    <!--opt, xs: boolean, whether supports secondary authentication for playback and download-->
  </playBackDoubleVerification>
</videoChannelPermission>
</videoChannelPermissionList>
<ptzControl><!--opt, xs: boolean--></ptzControl>
<ptzChannelPermissionList><!--opt-->
<ptzChannelPermission><!--opt-->
  <id><!--req, corresponds to PTZ channel ID--></id>
  <ptzControl><!--opt, xs: boolean--></ptzControl>
</ptzChannelPermission>
</ptzChannelPermissionList>
<logOrStateCheck><!--opt, xs: boolean, permission to view log and status--></logOrStateCheck>
<parameterConfig><!--opt, xs: boolean, parameter configuration permission--></parameterConfig>
<restartOrShutdown><!--opt, xs: boolean, permission to reboot and shutdown--></restartOrShutdown>
<upgrade><!--opt, xs: boolean, permission to upgrade--></upgrade>
<voiceTalk><!--opt, xs: boolean--></voiceTalk>
<transParentChannel><!--opt, xs: boolean--></transParentChannel>
<contorlLocalOut><!--opt, xs: boolean--></contorlLocalOut>
<alarmOutOrUpload><!--opt, xs: boolean, permission to upload and output alarm--></alarmOutOrUpload>
<factoryReset><!--opt, xs: boolean, restore default parameters--></factoryReset>
<arm><!--opt, xs: boolean, arm--></arm>
<disarm><!--opt, xs: boolean, disarm--></disarm>
<accessControl><!--opt, xs: boolean, access control--></accessControl>
<subSysOrZoneArm><!--opt, xs: boolean, partition or zone arming permission--></subSysOrZoneArm>
<subSysOrZoneDisarm><!--opt, xs: boolean, partition or zone disarming permission--></subSysOrZoneDisarm>
<subSysOrZoneClearArm><!--opt, xs: boolean, permission to clear partition or zone alarms--></subSysOrZoneClearArm>
<zoneBypass><!--opt, xs: boolean, permission to bypass zone--></zoneBypass>
<zoneBypassRecover><!--opt, xs: boolean, permission to recover bypassing zone--></zoneBypassRecover>
<IRAI>
<rapidConfiguration><!--opt, xs: boolean--></rapidConfiguration>
<reset><!--opt, xs: boolean--></reset>
```

```
</IRaid>
<subSystemNo min="" max=""><!--opt, xs:integer, range of partition No. When adding an operator or setting
operator parameters, one or more partitions can be linked to the operator--></subSystemNo>
<subSystem min="1" max="3"><!--opt, xs:integer, number of partitions that can be linked to the operator. The
attributes min="1" and max="3" are sample settings, which indicates that at least one partition and up to three
partitions can be linked. When adding an operator or setting operator parameters, one or more partitions can be
linked to an operator. Actually, an operator should be linked to at least one partition and up to four partitions--></
subSystemNo>
<LPListAudit>
  <!--optional, xs:boolean, allowlist/blocklist operation permission. Users with this permission can import, export,
edit, delete, and search for license plate information in the allowlist or blocklist. See remarks for the URI for each
operation-->true
</LPListAudit>
<restoreTamper opt="true,false">
  <!--optional, boolean, whether to restore tampering, this node is only valid for installer-->true
</restoreTamper>
<restoreConfirmedAlarm opt="true,false">
  <!--optional, boolean, whether to restore alarm acknowledgment, this node is for linked zones-->true
</restoreConfirmedAlarm>
<autoArmAndDisarm opt="true,false">
  <!--optional, boolean, whether it supports configuring time for arming/disarming partitions automatically (Admin)-->
</autoArmAndDisarm>
<remoteCtrlConfig opt="true,false">
  <!--optional, boolean, whether it supports remote control configuration (Installer)-->true
</remoteCtrlConfig>
<cardConfig opt="true,false">
  <!--optional, boolean, whether it supports card configuration (Installer)-->true
</cardConfig>
</remotePermissionCap>
```

Remarks

- Search for license plate information in the allowlist or blocklist: POST /ISAPI/Traffic/channels/<ID>/searchLPListAudit;
- import/export license plate information into/in allowlist or blocklist: GET/PUT /ISAPI/Traffic/channels/<ID>/licensePlateAuditData;
- add/edit license plate information to/in the allowlist or blocklist: PUT /ISAPI/Traffic/channels/<channelID>/licensePlateAuditData/record?format=json;
- delete license plate information in the allowlist or blocklist: PUT /ISAPI/Traffic/channels/<channelID>/DelLicensePlateAuditData?format=json.

F.290 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.291 XML_RTSPCertificate

XML message about parameters of RTSP authentication mode

```
<RTSPCertificate xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0" >
    <certificateType>
        <!--required, xs:string, authentication mode: digest (default), digest/basic-->
    </certificateType>
    <SecurityAlgorithm>
        <!--dependent, authentication algorithm, it is valid when certificateType is "digest" or "digest/basic"-->
        <algorithmType>
            <!--optional, xs:string, algorithm types: MD5, SHA256, MD5/SHA256-->
        </algorithmType>
    </SecurityAlgorithm>
</RTSPCertificate>
```

F.292 XML_RTSPCertificateCap

XML message about capability of RTSP authentication mode

```
<RTSPCertificateCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0" >
    <certificateType def="digest" opt="digest,digest/basic" >
        <!--required, xs:string-->
    </certificateType>
    <SecurityAlgorithm>
        <!--depend, authentication algorithm, it is valid when certificateType is "digest" or "digest/basic"-->
        <algorithmType opt="MD5,SHA256,MD5/SHA256" >
```

```

<!--optional, xs:string, algorithm type-->
</algorithmType>
</SecurityAlgorithm>
</RTSPCertificateCap>

```

F.293 XML_Schedule

Schedule message in XML format

```

<Schedule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:string, ID--></id>
  <eventType>
    <!--optional, xs:string, alarm/event types, see details in the "Remarks" below-->
  </eventType>
  <inputIOPortID><!--read-only, dependent, xs:string, alarm input No.--></inputIOPortID>
  <outputIOPortID><!--read-only, dependent, xs:string, alarm output No.--></outputIOPortID>
  <videoInputChannelID><!--read-only, dependent, xs:string, video input channel ID--></videoInputChannelID>
  <TimeBlockList size="8"><!--required-->
    <TimeBlock><!--list-->
      <dayOfWeek>
        <!--optional, xs:integer, day of the week based on ISO8601, "1"=Monday, .....>
      </dayOfWeek>
      <TimeRange><!--required-->
        <beginTime><!--required, xs:time, ISO 8601 time--></beginTime>
        <endTime><!--required, xs:time, ISO 8601 time--></endTime>
      </TimeRange>
      <CustomExtension>
        <vehicleDetectSceneID>
          <!--required, xs:integer-->
        </vehicleDetectSceneID>
      </CustomExtension>
    </TimeBlock>
  </TimeBlockList>
  <HolidayBlockList><!--optional-->
    <TimeBlock><!--list-->
      <TimeRange><!--required-->
        <beginTime><!--required, xs:time, ISO 8601 time--></beginTime>
        <endTime><!--required, xs:time, ISO 8601 time--></endTime>
      </TimeRange>
    </TimeBlock>
  </HolidayBlockList>
</Schedule>

```

Remarks

The node **<eventType>** can be set to the following values: IO, VMD, videoLoss, PIR, lineDetection, fieldDetection, audioException, faceDetection, regionEntrance, regionExiting, loitering, group, rapidMove, parking, unattendedBaggage, attendedBaggage, storageDetection, shipsDetection, HUMANATTRIBUTE, humanAttribute, faceContrast, faceSnap, faceLib, whiteListFaceContrast, personDensityDetection, personQueueDetection, mixedTargetDetection, fireDetection,

illegalParking, pedestrian, trafficAccident, construction, roadblock, abandonedObject, parallelParking, parkingState, congestion, intersectionAnalysis, heatMap, reverseEntrance, vehicleDetect, safetyHelmetDetection, vibrationDetection, TMPA, faceThermometry, HBDLib, detectorTemp, detectorSmoke, detectorTamper, smokeFireRecognizesmokeFireRecognize, indoorPasswayBlock, detectorShelter, detectorMotion, fireNoRegulation, peopleDetections.

F.294 XML_SecurityCap

SecurityCap capability message in XML format

```

<SecurityCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <supportUserNums><!--optional, xs:integer, number of supported users--></supportUserNums>
  <userBondIpNums><!--optional, xs:integer, number of bound IP addresses supported by the user--></userBondIpNums>
  <userBondMacNums><!--optional, xs:integer, number of bound MAC addresses supported by the user--></userBondMacNums>
  <isSupCertificate><!--optional, xs:boolean, whether it supports certification: "true", "false"--></isSupCertificate>
  <issupIllegalLoginLock><!--optional, xs: boolean, whether it supports locking illegal login: "true,false"--></issupIllegalLoginLock>
  <isSupportOnlineUser><!--optional, xs: boolean, "true,false"--></isSupportOnlineUser>
  <isSupportAnonymous><!--optional, xs: boolean, "true,false"--></isSupportAnonymous>
  <isSupportStreamEncryption><!--optional, xs:boolean, whether it supports stream encryption: "true", "false"--></isSupportStreamEncryption>
  <securityVersion opt="1,2"/><!--optional, xs:interger, encryption capability, each version contains encryption algorithm and node range to be encrypted. "1"-encrypt by AES128, "2"-encrypt by AES256-->
  <keyIterateNum>
    <!--dependent xs:integer, iteration times, this node depends on securityVersion, and the value is usually between 100 and 1000-->
  </keyIterateNum>
  <isSupportUserCheck>
    <!--dependent optional, xs:boolean, whether it supports verifying password when editing/adding/deleting user parameters, this node depends on securityVersion-->
  </isSupportUserCheck>
  <isSupportGUIDFileDataExport><!--optional, xs:boolean, "true,false"--></isSupportGUIDFileDataExport>
  <isSupportSecurityQuestionConfig><!--optional, xs:boolean, "true,false"--></isSupportSecurityQuestionConfig>
  <supportSecurityNode opt="wirelessServer,wirelessDial"><!--optional, xs:string, "true,false"--></supportSecurityNode>
  <isSupportGetOnlineUserListSC><!--optional, xs:boolean, "true,false"--></isSupportGetOnlineUserListSC>
  <SecurityLimits><!--optional-->
    <LoginPasswordLenLimit min="1" max="16"><!--opt--></LoginPasswordLenLimit>
    <SecurityAnswerLenLimit min="1" max="128"><!--opt--></SecurityAnswerLenLimit>
  </SecurityLimits>
  <RSAKeyLength opt="512,1024,2048" def=2048/><!--optional, RSA key length of the HTTPS certificate-->
  <isSupportONVIFUserManagement><!--optional, xs:boolean, whether it supports user management by ONVIF protocol: "true", "false"--></isSupportONVIFUserManagement>
  <WebCertificateCap><!--opt-->
    <CertificateType opt="basic, digest, digest/basic"><!--required, xs:string--></CertificateType>
    <SecurityAlgorithm><!--dependent, this node is valid when certificateType is "digest" or "digest/basic"-->
      <algorithmType opt="MD5,SHA256,MD5/SHA256"><!--optional, xs:string, algorithm type: "MD5", "SHA256", "MD5/SHA256"--></algorithmType>

```

```
</SecurityAlgorithm>
</WebCertificateCap>
<isSupportConfigFileImport>
  <!--optional, xs: boolean, whether it supports importing configuration files securely: "true"-yes. If it is not supported, this node will not be returned-->
</isSupportConfigFileImport>
<isSupportConfigFileExport>
  <!-- optional, xs:boolean, whether it supports exporting configuration files securely: "true"-yes. If it is not supported, this node will not be returned-->
</isSupportConfigFileExport>
<cfgFileSecretKeyLenLimit min="0" max="16">
  <!--optional, length limit of the configuration files' encryption key-->
</cfgFileSecretKeyLenLimit>
<isSupportDeviceCertificatesManagement>
  <!--optional, xs: boolean, whether it supports device certificate management: true-yes, if not support, this node will not be returned-->
</isSupportDeviceCertificatesManagement>
<supportIPCActivatePassword><!--optional, xs:boolean, whether it supports configuring password for activating the network camera--></supportIPCActivatePassword>
<isIrreversible>
  <!--dependent optional, xs:boolean, whether it supports irreversible password storage. If this node does not exist, irreversible password storage is not supported-->
</isIrreversible>
<salt><!--optional, xs:string, salt of user name--></salt>
<keypadPassword min="1" max="" /><!--optional, xs:string, keypad password length. If different types of users have different keypad password length, this node only indicates the administrator's keypad password length, otherwise this node indicates that all types of users have the same keypad password length-->
<installerKeypadPassword min="1" max="" /><!--optional, xs:string, installer's keypad password length-->
<operatorKeypadPassword min="1" max="" /><!--optional, xs:string, operator's keypad password length-->
<userOperateType opt="1,2,3">
  <!--optional, xs:string, user operation type: "1"-network user, "2"-keypad user, "3"-network user and keypad user-->
</userOperateType>
<isSupportOnvifInfo><!--optional, xs:boolean, whether it supports the API of getting ONVIF protocol information. If this node is returned and its value is "true", in indicates supporting this function; if this node is not returned, it indicates that this function is not supported--></isSupportOnvifInfo>
<isSupportPictureURICertificate opt="true,false"><!--optional, xs:boolean, whether it supports configuring picture URL authentication. If this node is not returned, it indicates that the default picture authentication method of the device is digest--></isSupportPictureURICertificate>
<isSupportUnloggedUserPermissionConfig opt="true,false"><!--optional, xs:boolean, whether it supports configuring permissions for users that have not logged in--></isSupportUnloggedUserPermissionConfig>
<isSupportUserNamePasswordCheckUpgrade><!--optional, xs:boolean, whether it supports upgrading user name and password verification. If this node is not returned, it indicates that this function is not supported--></isSupportUserNamePasswordCheckUpgrade>
<isSupportDeviceSelfSignCertExport><!--optional, xs:boolean--></isSupportDeviceSelfSignCertExport>
<isSupportSecurityEmail><!--optional, xs:boolean, whether it supports configuring security E-mail address. If this node is not returned, it indicates that this function is not supported--></isSupportSecurityEmail>
<isSupportRTSPCertificate opt="true,false"><!--optional, xs:boolean--></isSupportRTSPCertificate>
<isSptUserEnabled><!--optional, xs:boolean, whether it supports configuration of enabling user: "true"-yes, "false"-no--></isSptUserEnabled>
<isSptAdminCap><!--optional, xs:boolean, whether it supports getting administrator permission capability: "true"-yes, "false"-no--></isSptAdminCap>
<DoubleVerificationCap><!--optional, double verification capability-->
```

```

<isSupportUsersConfig>
    <!--optional, xs: boolean, "true,false", whether it supports configuring double verification user-->
</isSupportUsersConfig>
<isSupportUsersPermissionConfig>
    <!--optional, xs: boolean, "true,false", whether it supports configuring the permission of double verification user-->
</isSupportUsersPermissionConfig>
</DoubleVerificationCap>
<isSupportCertificateCustomID><!--optional, xs:Boolean, whether it supports certificate configuration with custom ID--></isSupportCertificateCustomID>
<maxIllegalLoginTimes min="3" max="20" def="5">
    <!--dependentxs:integer,"maximum illegal login attempts" -->
</maxIllegalLoginTimes>
<SecurityAdvanced>
    <securityEnhanced><!--optional, xs:boolean, security reinforcement--></securityEnhanced>
    <noOperationEnabled><!—required, xs:Boolean, whether to enable the timeout of no operation--></noOperationEnabled>
        <noOperationTime min="1" max="60" def="15"><!—required, xs:integer, timeout of no operation, unit: minute--></noOperationTime>
            <isSupportDigestStatus><!—optional, Boolean—digest status--></isSupportDigestStatus>
        </noOperationEnabled>
        <LoginLinkNum><maxLinkNum min="1" max="128" def="50"><!—required, xs:integer, maximum number of logged in accounts--></maxLinkNum><LoginLinkNum>
            <isSupportCCClientCertificate><!—optional, Boolean—whether to enable the CC client certificate--></isSupportCCClientCertificate>
            <passwordValidity min="0" max="365"><!--optional, xs:string, password validity period, unit: day. If this node is returned, it indicates supporting configuring password validity period. The password validity period can only be edited by the administrator user--></passwordValidity>
            <maxIllegalLoginLockTime min="1" max="120" def="30"><!—optional, xs:integer, the lock duration when maximum illegal login attempts reached the upper limit; unit: minute--></maxIllegalLoginLockTime>
        </LoginLinkNum>
    </SecurityAdvanced>

```

F.295 XML_SecurityQuestion

SecurityQuestion message in XML format

```

<SecurityQuestion version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <QuestionList>
        <Question><!—the number of security question must be 3-->
            <id><!—required, xs: integer, question ID, which corresponds to that of device--></id>
            <answer>
                <!—write-only, xs: string, answer of the security question, it will not be returned when getting question; when setting question, this node will be encrypted (BASE64-->AES128CBE) for transmission-->
            </answer>
            <mark>
                <!—required, read-only, xs: boolean, whether the configured security question is marked-->
            </mark>
        </Question>
    </QuestionList>
    <password>

```

```
<!--write-only, xs: string, password that encrypted by CBC, this node will not be returned when getting question; but  
it is required when setting question-->  
</password>  
</SecurityQuestion>
```

See Also

F.296 XML_SerialCap

Serial port capability message in XML format

```
<SerialCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  
  <rs485PortNums min="1" max="5" def="1">  
    <!--optional, xs: integer-->  
  </rs485PortNums>  
  <supportRS23Cconfig>  
    <!--optional, xs: boolean-->  
  </supportRS23Cconfig>  
  <rs422PortNums opt="0">  
    <!--optional, xs: integer-->  
  </rs422PortNums>  
  <rs232PortNums opt="1">  
    <!--optional, xs: integer, range of RS-232 serial port No. supported by the device-->  
  </rs232PortNums>  
  <rs485WorkMode opt="Led, CaptureTrigger">  
    <!--optional, xs: string, range of RS-485 serial port No. supported by the device-->  
  </rs485WorkMode>  
  <isSupportAuthenticationService>  
    <!--optional, xs:boolean, whether the device supports serial port authentication-->  
  </isSupportAuthenticationService>  
</SerialCap>
```

F.297 XML_Shield

Message about shielded area parameters in XML format

```
<Shield version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  
  <enabled><!--req, xs:string, whether enables the shield function: "true,false"--></enabled>  
  <RegionList>  
    <Region>  
      <id><!--req, xs:string, area ID--></id>  
      <enabled><!--req, xs:string, whether enables the shield function of this area: "true,false"--></enabled>  
      <RegionCoordinatesList>  
        <RegionCoordinates><!--req-->  
          <positionX><!--req, xs:integer, X-coordinate--></positionX>  
          <positionY><!--req, xs:integer, Y-coordinate--></positionY>  
        </RegionCoordinates>  
      </RegionCoordinatesList>  
    </Region>  
  </RegionList>  
</Shield>
```

```
</Region>
</RegionList>
</Shield>
```

F.298 XML_SlaveCamera

SlaveCamera message in XML format

```
<SlaveCameraInfo version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <id>
    <!--req, xs:string-->
  </id>
  <serverAddress>
    <addressingFormatType>
      <!--req, xs:string, "ipaddress,hostname"-->
    </addressingFormatType>
    <hostName>
      <!--dep, xs:string-->
    </hostName>
    <ipAddress>
      <!--dep, xs:string-->
    </ipAddress>
    <ipv6Address>
      <!--dep, xs:string-->
    </ipv6Address>
  </serverAddress>
  <userName>
    <!--req, xs:string-->
  </userName>
  <passWord>
    <!--wo, xs:string-->
  </passWord>
  <portNo>
    <!--req, xs:integer-->
  </portNo>
  <loginStatus>
    <!--req, xs:string, "login/logout"-->
  </loginStatus>
</SlaveCameraInfo>
```

F.299 XML_SlaveCameraStatus

SlaveCameraStatus message in XML format

```
<SlaveCameraStatus version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <SlaveCameraLinkStatusList>
    <!--req-->
    <SlaveCameraLinkStatus>
      <id>
```

```
<!--req, xs:string-->
</id>
<linkStatus/>
  <!--ro, xs:string, connection status: "online,offline"-->
</SlaveCameraLinkStatus>
</SlaveCameraLinkStatusList>
</SlaveCameraStatus>
```

F.300 XML_SNMP

XML message about SNMP configuration

```
<SNMP version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <SNMPv1c/><!--dependent -->
  <SNMPv2c/><!--dependent-->
  <SNMPAdvanced/><!--dependent, advanced parameters-->
  <listenPort> <!--optional, xs:integer, SNMP port--></listenPort>
</SNMP>
```

F.301 XML_SNMPAdvanced

XML message about SNMP advanced parameter configuration

```
<SNMPAdvanced xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0" >
  <localEngineID>
    <!--required, xs:hexBinary, local device engine-->
  </localEngineID>
  <authenticationNotificationEnabled>
    <!--optional, xs:boolean, notification of failure to enable authentication-->
  </authenticationNotificationEnabled>
  <SNMPUserList/>
    <!--optional -->
  <SNMPNotificationFilterList/>
    <!--optional, filtering table based on OIDS-->
  <notificationEnabled>
    <!--optional, xs:boolean -->
  </notificationEnabled>
  <SNMPNotificationReceiverList/>
    <!--optional-->
  <enabled>
    <!--required, xs:boolean -->
  </enabled>
  <SNMPTrapReceiverList/>
    <!--optional, all trap address information of SNMP in version 3-->
</SNMPAdvanced>
```

F.302 XML_SNMPTrapReceiver

XML message about parameters of single trap address of SNMP

```
<SNMPTrapReceiver xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0" >
<id>
  <!--required, xs:string, ID -->
</id>
<ReceiverAddress>
  <!--required -->
  <addressingFormatType>
    <!--required, xs:string, "ipaddress,hostname"-->
  </addressingFormatType>
  <hostName>
    <!--dependent, xs:string -->
  </hostName>
  <ipAddress>
    <!--dependent, xs:string -->
  </ipAddress>
  <ipv6Address>
    <!--dependent, xs:string -->
  </ipv6Address>
  <portNo>
    <!--optional, xs:integer -->
  </portNo>
</ReceiverAddress>
<notificationType>
  <!--required, xs:string, "trap,inform" -->
</notificationType>
<communityString>
  <!--optional, xs:string -->
</communityString>
</SNMPTrapReceiver>
```

F.303 XML_SNMPTrapReceiverList

XML message about trap address parameters of SNMP

```
<SNMPTrapReceiverList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <SNMPTrapReceiver/> <!--optional-->
</SNMPTrapReceiverList>
```

F.304 XML_ssd

JSON message about a SSD information

```
<?xml version="1.0" encoding="utf-8"?>
<ssd version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, read-only, xs:string, SSD ID--></id>
  <ssdName><!--required, read-only, xs:string, SSD name--></ssdName>
  <version><!--optional, xs:string, SSD firmware version--></version>
</ssd>
```

F.305 XML_ssdList

JSON message about information of all SSDs

```
<?xml version="1.0" encoding="utf-8"?>
<ssdList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ssd><!--optional, see XML-ssd for details--></ssd>
</ssdList>
```

See Also

[XML_ssd](#)

F.306 XML_SSDSMARTTest

XML message about SSD S.M.A.R.T detection parameters

```
<?xml version="1.0" encoding="utf-8"?>
<SSDSMARTTest version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <testType><!--optional, xs:string, self-test type: "short", "expanded", "conveyance"--></testType>
</SSDSMARTTest>
```

F.307 XML_SSD_SmartTestStatus

JSON message about SSD S.M.A.R.T detection status

```
<?xml version="1.0" encoding="utf-8"?>
<SmartTestStatus version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:string, SSD ID--></id>
  <temprature><!--required, xs:integer, temperature, unit: degree--></temprature>
  <powerOnDay><!--required, xs:integer, hour of use, unit: day(s)--></powerOnDay>
  <selfEvaluatingStatus><!--required, xs:string, self-evaluation status: "ok", "error"--></selfEvaluatingStatus>
  <allEvaluatingStatus><!--required, xs:string, overall evaluation: "functional"-in good condition, "badsectors"-bad
block, "fault"-error--></allEvaluatingStatus>
  <selfTestPercent><!--required, xs:integer, self-test progress--></selfTestPercent>
  <selfTestStatus><!--required, xs:string, self-test status: "ok", "aborted", "interrupted", "failed", "unkown",
"electronic_element_error", "servo_error", "read_failed", "progress"-in progress, "not_tested", "not_recognized"--></
selfTestStatus>
  <testType><!--required, xs:string, self-test type: "short", "expanded", "conveyance"--></testType>
  <TestResultList><!--required, "Size=30"-->
```

```
<TestResult><!--list-->
<attributeID><!--required, xs:string, attribute ID--></attributeID>
<attributeName><!--optional, xs:string, attribute name--></attributeName>
<status><!--required, xs:string, status: "ok"-normal, "illegal"-error--></status>
<flags><!--required, xs:integer, attribute operation mark--></flags>
<thresholds><!--required, xs:integer, threshold--></thresholds>
<value><!--required, xs:string, current value--></value>
<worst><!--required, xs:integer, the worst value--></worst>
<rawValue><!--required, xs:integer, raw value--></rawValue>
</TestResult>
</TestResultList>
</SmartTestStatus>
```

F.308 XML_StreamingChannel

StreamingChannel message in XML format

```
<StreamingChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id><!--req,xs:string;id--></id>
<channelName><!--req,xs:string--></channelName>
<enabled><!--req,xs:boolean--></enabled>
<Transport><!--req-->
<maxPacketSize><!--opt, xs: integer--></maxPacketSize>
<audioPacketLength><!--opt, xs: integer--></audioPacketLength>
<audioInboundPacketLength><!--opt, xs: integer--></audioInboundPacketLength>
<audioInboundPortNo><!--opt, xs: integer--></audioInboundPortNo>
<videoSourcePortNo><!--opt, xs: integer--></videoSourcePortNo>
<audioSourcePortNo><!--opt, xs: integer--></audioSourcePortNo>
<ControlProtocolList><!--req, protocol types for streaming-->
<ControlProtocol><!--req-->
<streamingTransport>
<!--req,xs:string,"HTTP,RTSP,SHTTP,SRTP"-->
</streamingTransport>
</ControlProtocol>
</ControlProtocolList>
<Unicast><!--opt-->
<enabled><!--req, xs: boolean--></enabled>
<interfaceID><!--opt, xs: string--></interfaceID>
<rtpTransportType
<!--opt, xs: string, "RTP/UDP, RTP/TCP"-->
</rtpTransportType>
</Unicast>
<Multicast><!--opt-->
<enabled><!--req, xs: boolean--></enabled>
<userTriggerThreshold><!--opt, xs: integer--></userTriggerThreshold>
<destIPAddress><!--dep, xs: string--></destIPAddress>
<videoDestPortNo><!--opt, xs: integer--></videoDestPortNo>
<audioDestPortNo><!--opt, xs: integer--></audioDestPortNo>
<destIPv6Address><!--dep, xs: string--></destIPv6Address>
<ttl><!--opt, xs:integer--></ttl>
```

```
<activeMulticastEnabled>
  <!--opt, xs: boolean, "true,false", whether to enable active multicast, which is mutual exclusion with passive
multicast-->
</activeMulticastEnabled>
<packagingFormat><!--opt, xs: string, container format--></packagingFormat>
<FecInfo><!--opt-->
<fecRatio>
  <!--req, read-only, xs: integer, extra bandwidth occupation ratio of forward error correction (FEC) data, the value
is between 0 and 100, the default value is 0-->
</fecRatio>
<fecDestPortNo>
  <!--opt, xs: integer, Port No. of FEC multicast, the default port No. is specified by device-->
</fecDestPortNo>
</FecInfo>
</Multicast>
<Security><!--opt-->
<enabled><!--req, xs: boolean--></enabled>
<certificateType><!--req, xs: string, opt="digest,digest/baisc" def="digest"--></certificateType>
<SecurityAlgorithm><!--dependency, this node is valid when certificateType is "digest" or "digest/basic"-->
  <algorithmType><!--optional, xs:string, algorithm type: "MD5", "SHA256", "MD5/SHA256"--></algorithmType>
</SecurityAlgorithm>
</Security>
<SRTPMulticast><!--opt-->
  <SRTPVideoDestPortNo><!--opt, xs:integer--></SRTPVideoDestPortNo>
  <SRTPAudioDestPortNo><!--opt, xs:integer--></SRTPAudioDestPortNo>
</SRTPMulticast>
</Transport>
<Video><!--opt-->
<enabled><!--req, xs: boolean--></enabled>
<videoInputChannelID>
  <!--req, xs: string, id-->
</videoInputChannelID>
<videoCodecType>
  <!--req, xs: string, "MPEG4,MJPEG,3GP,H.264, HK.264, MPNG, SVAC, H.265"-->
</videoCodecType>
<videoResolutionWidth>
  <!--req, xs: integer-->
</videoResolutionWidth>
<videoResolutionHeight>
  <!--req, xs: integer-->
</videoResolutionHeight>
<videoQualityControlType>
  <!--opt, xs: string, "CBR,VBR"-->
</videoQualityControlType>
<constantBitRate>
  <!--dep, xs: integer, constant bit rate, unit: kbps-->
</constantBitRate>
<vbrUpperCap>
  <!--dep, xs: integer, upper limit of variable bit rate, unit: kbps-->
</vbrUpperCap>
<vbrLowerCap>
  <!--dep, xs: integer, lower limit of variable bit rate, unit: kbps-->
```

```
</vbrLowerCap>
<maxFrameRate>
  <!--req, xs:integer, maximum frame rate, the value is multiplied by 100 to be returned-->
</maxFrameRate>
<keyFrameInterval><!--opt, xs:integer, milliseconds--></keyFrameInterval>
<rotationDegree><!--opt, xs:integer, degrees, 0..360--></rotationDegree>
<mirrorEnabled><!--opt, xs:boolean--></mirrorEnabled>
<snapShotImageType><!--opt, xs:string, "JPEG,GIF,PNG"--></snapShotImageType>
<Mpeg4Profile> <!--dep, xs:string, "SP,ASP"--></Mpeg4Profile>
<H264Profile><!--dep, xs:string, "Baseline,Main,High, Extended"--></H264Profile>
<SVACProfile><!--dep, xs:string, "Baseline,Main,High,Extended"--></SVACProfile>
<GovLength> <!--opt, xs:integer--></GovLength>
<SVC>
  <enabled><!--req, xs:boolean--></enabled>
  <SVCMode><!--dep, xs:string, "manual,auto"--></SVCMode>
</SVC>
<smoothing><!--opt, xs:integer--></smoothing>
<SmartCodec><!--dep, -->
  <enabled><!--req, xs:boolean--></enabled>
</SmartCodec>
<vbrAverageCap><!--dep, xs:integer, in kbps--></vbrAverageCap>
<IntelligentInfoDisplayMethod>
  <!--opt, intelligent information displaying method, xs:string, "player,non-player"-->
</IntelligentInfoDisplayMethod>
</Video>
<Audio><!--opt-->
  <enabled><!--req, xs:boolean--></enabled>
  <audioInputChannelID><!--req, xs:string,id--></audioInputChannelID>
  <audioCompressionType>
    <!--req, xs:string,"G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM, MP2L2"-->
  </audioCompressionType>
  <audioInboundCompressionType>
    <!--opt, xs:string,"G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM,MP2L2"-->
  </audioInboundCompressionType>
  <audioBitRate><!--opt, xs:integer, in kbps--></audioBitRate>
  <audioSamplingRate><!--opt, xs:float, in kHz--></audioSamplingRate>
  <audioResolution><!--opt, xs:integer, in bits--></audioResolution>
  <VoiceChanger><!--opt, xs:integer, -12..0..12-->
    <enabled><!--req, xs:boolean--></enabled>
    <level><!--req, xs:integer, "-12..12"--></level>
  </VoiceChanger>
</Audio>
<enableCABAC><!--opt, xs: boolean--></enableCABAC>
<subStreamRecStatus><!--opt, xs: boolean--></subStreamRecStatus>
<customStreamEnable><!--opt, xs: boolean, whether the stream is custom stream: "true"-yes, this node is not returned-no--></customStreamEnable>
</StreamingChannel>
```

F.309 XML_storage

XML message about storage working mode

```
<?xml version="1.0" encoding="utf-8"?>
<storage version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<hddList><!--optional, the contents are same to the message XML_hddList (related URI: /ISAPI/ContentMgmt/Storage/hdd)--></hddList>
<nasList><!--optional, the contents are same to the message XML_nasList (related URI: /ISAPI/ContentMgmt/Storage/nas)--></nasList>
<workMode><!--optional, xs:string, working mode: "group", "quota", "extract-></workMode>
<ssdList><!--optional, the contents are same to the message XML_ssdlst (related URI: /ISAPI/ContentMgmt/Storage/ssd)--></ssdList>
</storage>
```

F.310 XML_storageExtension

XML message about storage strategy.

```
<storageExtension xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
<LoopEnable>
<!--optional, xs:boolean, whether to enable overwriting-->
</LoopEnable>
<enableDormant>
<!--optional, xs:boolean-->
</enableDormant>
<packDuration>
<!--optional, xs:integer, unit: minute-->
</packDuration>
<logStorageMode>
<!--optional, xs:string, log storage mode: "system"-system default mode, "custom"-custom mode-->
</logStorageMode>
<logStorageId>
<!--dependent, xs:integer, disk ID (saving logs in a disk) or array number (saving logs in disk arrays), which are valid when logStorageMode is "custom"-->
</logStorageId>
<logStorageCycle>
<!--dependent, xs:integer, log storage period; it is valid when logStorageMode is "custom"-->
</logStorageCycle>
</storageExtension>
```

F.311 XML_SoftwareService

SoftwareService message in XML format

```
<SoftwareService version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<ThirdStream><!--opt, third stream configuration-->
```

```
<enabled><!--req, xs: boolean, "true,false", whether to enable third stream configuration--></enabled>
</ThirdStream>
<PanoramaDisplay><!--opt, display the image range on panorama view-->
<enabled>
<!--req, xs: boolean, "true,false", whether to enable image range display on panorama view-->
</enabled>
</PanoramaDisplay>
<MotionDetect>
<enabled><!--whether to enable motion detection--></enabled>
</MotionDetect>
</SoftwareService>
```

F.312 XML_SocketIP

SocketIP message in XML format

```
<SocketIP version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<ipAddress>
<!--opt, xs: string-->
</ipAddress>
<ipv6Address>
<!--opt, xs: string-->
</ipv6Address>
</SocketIP>
```

F.313 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 "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>

```

F.314 XML_SubscribeEventCap

SubscribeEventCap capability message in XML format

```

<SubscribeEventCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <heartbeat min="" max="" />
        <!--optional, heartbeat time interval, unit: second-->
    <format opt="xml,json"/><!--req, supported message format-->
    <channelMode opt="all,list" />
        <!--required, channel subscription mode: "all"-subscribe events/alarms of all channels, "list"-subscribe events/
alarms of specific channels-->
    <eventMode opt="all,list" />
        <!--required, event subscription mode: "all"-subscribe all event types (must be supported), "list"-subscribe specific
event types, if "list" is returned, "all" will also be returned-->
        <!--if both the channelMode and eventMode returns "all", it indicates that the device does not support subscribing
event/alarm by event type or channel-->
    <EventList><!--required, dependent, upload mode of specified alarms/events, it is valid only when eventMode is
"list"-->
        <Event><!--required-->
            <type><!--required, xs:string, refer to Supported Alarm/Event Types for details--></type>
            <minorAlarm opt="0x400,0x401,0x402,0x403">

```

```

<!--opt, xs:string, minor alarm type, see details in Access Control Event Type. This node is required when type is
"AccessControllerEvent"-->
</minorAlarm>
<minorException opt="0x400,0x401,0x402,0x403">
    <!--opt, xs:string, minor exception type, see details in Access Control Event Type. This node is required when type
is "AccessControllerEvent"-->
</minorException>
<minorOperation opt="0x400,0x401,0x402,0x403">
    <!--opt, xs:string, minor operation type, see details in Access Control Event Type. This node is required when type
is "AccessControllerEvent"-->
</minorOperation>
<minorEvent opt="0x01,0x02,0x03,0x04">
    <!--opt, xs:string, minor event type, see details in Access Control Event Type. This node is required when type is
"AccessControllerEvent"-->
</minorEvent>
<pictureURLType opt="binary,localURL,cloudStorageURL" def="" />
    <!--opt, xs:string, alarm picture format: "binary"-binary, "localURL"-device local URL, "cloudStorageURL"-cloud
storage URL, and the def is followed by the default format-->
</Event>
</EventList>
<pictureURLType opt="binary,localURL,cloudStorageURL" def="" />
    <!--opt, xs:string, alarm picture format: "binary"-binary picture, "localURL"-device local URL, "cloudStorageURL"-cloud
storage URL. This node is the method of uploading all pictures related to the event. If this node is applied,
<pictureURLType> in <Event> is invalid; otherwise, pictures will be uploaded using the default method returned by the
device capability. For front-end devices, the default method is uploading binary pictures; for back-end devices, the
default method is by device local URL-->
<ChangedUploadSub><!--message subscription-->
    <interval/><!--opt, xs:integer, lifecycle of arming GUID, the default value is 5 minutes, unit: second. The device will
generate new GUID for the arming connection after it is disconnected for the set lifecycle-->
    <StatusSub>
        <all/><!-- opt, xs:boolean, whether to subscribe all events-->
        <channel/><!--opt, xs:boolean, whether to subscribe channel status. This node is not required when <b><all></b> is
"true"-->
            <hd/><!--opt, xs:boolean, whether to subscribe disk status. This node is not required when <b><all></b> is "true"-->
            <capability/><!--opt, xs:boolean, whether to subscribe capability change status. This node is not required when
<b><all></b> is "true"-->
        </StatusSub>
    </ChangedUploadSub>
    <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-->
</SubscribeEventCap>

```

F.315 XML_SubscribeEventResponse

SubscribeEventResponse message in XML format

```

<SubscribeEventResponse>
    <id><!--req, xs:integer, subscription ID--></id>
    <FailedEventList>

```

```

<!--opt, list of subscription failed events. When subscription failed, it should be returned, and the upper layer can
check whether all event/alarm subscriptions are succeeded via the existence of node FailedEventList-->
<Event>
  <type>
    <!--req, xs:string, refer to Supported Alarm/Event Types for details-->
  </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 "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, opt="binary,localURL,cloudStorageURL", alarm picture transmission mode: "binary"-binary,
"localURL"-device local URL, "cloudStorageURL"-cloud storage URL-->
  </pictureURLType>
  <channels>
    <!--opt, xs:string, "1,2,3,4...", event related channel ID, supports multiple channels, and the channel ID is
separated by commas-->
  </channels>
  <subStatusCode>
    <!--req, string, subscription failure error code-->
  </subStatusCode>
</Event>
</FailedEventList>
</SubscribeEventResponse>

```

F.316 XML_SupplementLight

SupplementLight message in XML format

```

<SupplementLight><!--opt-->
  <mode><!--opt, xs: string, adjustment mode of supplement light, opt="schedule,off,on,auto"--></mode>
  <Schedule>
    <TimeRange><!--req-->
      <beginTime><!--req, xs: time, ISO8601 time--></beginTime>
      <endTime><!-- req, xs: time, ISO8601 time--></endTime>
    </TimeRange>
  </Schedule>
  <brightnessLimit>

```

```
<!--dep, xs: integer, brightness of supplement light, which is between 0 and 100; the brightness can be adjust when
<b>mode</b> is set to "on"-->
</brightnessLimit>
<supplementLightMode>
  <!--opt, xs: string, illumination mode: "mixed"-hybrid, "whitelight"-white light, "close"-disabled-->
</supplementLightMode>
<irLightBrightness>
  <!--dep, xs: integer, brightness of IR supplement light, which is between 0 and 100; this node is valid only when
<b>supplementLightMode</b> is set to "mixed" and <b>mixedLightBrightnessRegulatMode</b> is set to "manual"-->
</irLightBrightness>
<mixedLightBrightnessRegulatMode>
  <!--dep, xs: string, brightness adjustment mode of hybrid supplement light, opt="manual,auto"; this node is valid
only when <b>supplementLightMode</b> is set to "mixed"-->
</mixedLightBrightnessRegulatMode>
<highIrLightBrightness>
  <!--dep, xs: integer, brightness of far IR light, which is between 0 and 100; this node cannot be configured together
with node <b>irLightBrightness</b>-->
</highIrLightBrightness>
<highWhiteLightBrightness>
  <!--dep, xs: integer, brightness of far white light, which is between 0 and 100; this node cannot be configured
together with node <b>brightnessLimit</b>-->
</highWhiteLightBrightness>
<lowIrLightBrightness>
  <!--dep, xs: integer, brightness of near IR light, which is between 0 and 100; this node cannot be configured together
with node <b>irLightBrightness</b>-->
</lowIrLightBrightness>
<lowWhiteLightBrightness>
  <!--dep, xs: integer, brightness of near white light, which is between 0 and 100; this node cannot be configured
together with node <b>brightnessLimit</b>-->
</lowWhiteLightBrightness>
<whiteLightBrightness><!--dep, xs: integer, white light brightness, which is between 0 and 100--></
whiteLightBrightness>
<irLightbrightnessLimit><!--dep, xs:integer, IR light brightness limit, range: [0,100]; this node is valid when the value
of <b>mode "auto"</b>--></irLightbrightnessLimit>
<whiteLightbrightnessLimit><!--dep, xs:integer, white light brightness limit, range: [0,100]; this node is valid when
the value of <b>mode "auto"</b>--></whiteLightbrightnessLimit>
<unscheduledBrightness><!--optional, xs:integer, brightness in the unscheduled time, by default it is 0 (the
supplement light is disabled) and the value is between 0 and 100--></unscheduledBrightness>
</SupplementLight>
```

F.317 XML_TargetEnhancement

TargetEnhancement message in XML format

```
<TargetEnhancement version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled><!--req, xs:boolean--></enabled>
</TargetEnhancement>
```

F.318 XML_TargetEnhancementCap

TargetEnhancementCap message in XML format

```
<TargetEnhancementCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled opt="true,false"><!--req, xs:boolean--></enabled>
</TargetEnhancementCap>
```

F.319 XML_tempRange

tempRange message in XML format

```
<?xml version="1.0" encoding="utf-8"?>
<TempRange version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <mode>
    <!--req, xs:string, temperature range mode: "automatic"-automatic mode, "manual"-manual mode-->
  </mode>
  <temperatureUpperLimit>
    <!--dep, xs:float, maximum temperature, corrects to one decimal, the minimum value is "-20", the maximum value is "550", unit: Celsius degree; its value should be larger than temperatureLowerLimit; it is valid when the value of mode is "manual"-->
  </temperatureUpperLimit>
  <temperatureLowerLimit>
    <!--dep, xs:float, minimum temperature, corrects to one decimal, the minimum value is "-20", the maximum value is "550", unit: Celsius degree; its value should be smaller than temperatureUpperLimit; it is valid when the value of mode is "manual"-->
  </temperatureLowerLimit>
</TempRange>
```

F.320 XML_ThermalCap

XML message about thermal capability

```
<ThermalCap version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <isSupportFireDetection><!--optional, xs:boolean, whether supports fire detection--></isSupportFireDetection>
  <isSupportThermometry><!--optional, xs:boolean, whether supports temperature measurement--></isSupportThermometry>
  <isSupportRealtimeThermometry><!--optional, xs:boolean, whether supports uploading real-time temperature measurement data--></isSupportRealtimeThermometry>
  <isFireFocusZoomSupport><!--optional, xs:boolean, whether supports visible light lens zooming--></isFireFocusZoomSupport>
  <isSupportManualRanging>
    <!--optional, xs:boolean, this node will be returned if one or more channels of device supports this function, see detailed channel capability in the response information"-->
  </isSupportManualRanging>
  <isSupportPower><!--optional, xs:boolean, whether supports power on/off capability--></isSupportPower>
  <isSupportRealtimeTempHumi><!--optional, xs:boolean, whether supports real-time detection of temperature and
```

```
humidity--></isSupportRealtimeTempHumi>
<ManualThermCap>
<manualThermRuleNum>
    <!--optional, xs:integer, the max. number of supported rules for manual temperature measurement. If this node is not returned, it indicates manual temperature measurement is not supported-->
</manualThermRuleNum>
</ManualThermCap>
<isSupportManualThermBasic>
    <!--optional, xs:boolean, whether supports basic configuration of manual temperature measurement-->
</isSupportManualThermBasic>
<isSupportFireShieldMask><!--optional, xs:boolean--></isSupportFireShieldMask>
<isSupportSmokeShieldMask><!--optional, xs:boolean--></isSupportSmokeShieldMask>
<isSupportThermometryMode>
    <!--optional, xs:boolean, whether the device supports the configuration of temperature measurement mode-->
</isSupportThermometryMode>
<isSupportThermalPip>
    <!--optional, xs:boolean, whether the device supports the PIP configuration-->
</isSupportThermalPip>
<isSupportThermalIntelRuleDisplay><!--optional, xs:boolean, whether supports VCA rule configuration--></isSupportThermalIntelRuleDisplay>
<AlgVersionInfo><!--opt, whether supports getting the version information of thermal algorithms library-->
    <thermometryAlgName min = "1" max = "128">
        <!--read-only, xs:string, version information of temperature measurement algorithms library-->
    </thermometryAlgName>
    <shipsAlgName min = "1" max = "128"><!--read-only, xs:string, version name of ship detection algorithms library-->
</shipsAlgName>
</AlgVersionInfo>
<isSupportFaceThermometry><!--optional, xs:boolean, whether supports temperature screening configuration--></isSupportFaceThermometry>
<isSupportThermalBlackBody><!--optional, xs:boolean, whether supports black body configuration--></isSupportThermalBlackBody>
<isSupportThermalStreamParam><!--optional, xs:boolean, whether supports stream configuration--></isSupportThermalStreamParam>
<isSupportBodyTemperatureCompensation>
    <!--optional, xs:boolean, whether supports temperature compensation configuration-->
</isSupportBodyTemperatureCompensation>
<isSupportTemperatureCorrection><!--optional, xs:boolean, whether device supports temperature measurement correction--></isSupportTemperatureCorrection>
<isSupportClickToThermometry><!--optional, xs:boolean, whether device supports clicking to detect temperature--></isSupportClickToThermometry>
<isSupportThermometryHistorySearch><!--optional, xs:boolean--></isSupportThermometryHistorySearch>
<isSupportBurningPrevention><!--optional, xs:boolean, whether device supports burning prevention--></isSupportBurningPrevention>
<isSupportTemperatureCollection><!--optional, xs:boolean, whether device supports temperature ANR--></isSupportTemperatureCollection>
<isSupportJpegPicWithAppendData>
    <!--optional, xs:boolean, whether device supports getting JPEG picture with pixel-to-pixel temperature measurement data. If supports, it is returned and values true, if not support, it is not returned-->
</isSupportJpegPicWithAppendData>
<isSupportRealTimethermometryForHTTP>
    <!--optional, xs:boolean, whether device supports real-time temperature measurement. If supports, it is returned and its value is true, if not support, it is not returned-->
```

```
</isSupportRealTimethermometryForHTTP>
<isSupportShipsDetectionWithScene>
  <!--optional, xs:boolean, whether device supports ship detection by scene, this node and isSupportShipsDetection in XML_SmartCap are mutually exclusive-->
</isSupportShipsDetectionWithScene>
<isSupportthermometryOffLineCapture>
  <!--optional, xs:boolean, whether device supports offline capture. If supports, this node returned and its value is true; if not, it is not returned-->
</isSupportthermometryOffLineCapture>
<isSupportThermalTemperatureCorrect>
  <!--optional, xs:boolean, whether device supports temperature calibration (related URI: /ISAPI/Thermal/channels/>
<ID>/temperatureCorrect?format=json)-->
</isSupportThermalTemperatureCorrect>
<isSupportGreyScaleAlarm>
  <!--optional, xs:boolean, whether device supports grayscale alarm. If supports, this node returned and its value is true; if not, it is not returned-->
</isSupportGreyScaleAlarm>
<isSupportFaceSnapThermometry>
  <!--optional, xs:boolean, whether device supports uploading captured face picture with temperature information: true-support, no return-not support-->
</isSupportFaceSnapThermometry>
<isSupportTemperatureIntervalMeasurement>
  <!--optional, xs:boolean, whether device supports interval temperature measurement. If supports, this node returned and its value is true; if not, it is not returned-->
</isSupportTemperatureIntervalMeasurement>
<isSupportThermalVehicleDetection>
  <!--optional, xs:boolean, whether the device supports the thermal vehicle detection. If supports, this node will be returned and its value is true; if not, it is not returned-->
</isSupportThermalVehicleDetection>
<isSupportWasteGasDetection>
  <!--optionla, xs:boolean, whether the device supports the waste gas detection. If supports, this node will be returned and its value is true; if not, it is not returned-->
</isSupportWasteGasDetection>
</ThermalCap>
```

Remarks

When getting thermal product capabilities, **isSupportShipsDetectionWithScene** has a higher priority than **isSupportShipsDetection**. That is, firstly check if the node **isSupportShipsDetectionWithScene** exists and its value is "true", that indicates ship detection according to scene is supported, otherwise, check the if the node **isSupportShipsDetection** exists.

F.321 XML_TimeTaskList

XML message about all PTZ scheduled tasks parameters

```
<?xml version="1.0" encoding="utf-8"?>
<TimeTaskList size="">
  <enabled><!--req, xs: boolean--></enabled>
  <Parktime min="" max=""><!--req, xs:integer, seconds--></Parktime>
```

```

<TimeTaskBlock>
  <dayOfWeek><!--req, xs: integer, day of the week based on ISO8601, "1"-Monday, ...--></dayOfWeek>
  <TimeTaskRange>
    <TaskID min="" max=""><!--req, xs: string; ID--></TaskID>
    <beginTime><!--req, xs: time, ISO8601 time--></beginTime>
    <endTime> <!--req, xs: time, ISO8601 time--></endTime>
    <Task>
      <TaskType
        opt="disable,autoscan,framescan,randomscan,panoramascan,patrol,pattern,preset,tiltscan,period reboot,period adjust,
        auxoutput,combinedPath,sceneTrace,focus">
        <!--req, xs: strings-->
      </TaskType>
    <patrolTaskNum min="" max=""><!--dep, xs: integer, from 0 to 8--></patrolTaskNum>
    <patternTaskNum min="" max=""><!--dep, xs: integer, from 0 to 8--></patternTaskNum>
    <presetTaskNum min="" max=""><!--dep, xs: integer, from 0 to 8--></presetTaskNum>
    <auxoutputTaskNum min="" max=""><!--dep, xs: integer, from 0 to 8--></
auxoutputTaskNum>
    </Task>
  </TimeTaskRange>
</TimeTaskBlock>
<isSupportTimeTaskCopy><!--req, xs: boolean--></isSupportTimeTaskCopy>
</TimeTaskList>

```

F.322 XML_Track

XML message about recording schedule configuration parameters

```

<Track version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs: integer--></id>
  <Channel><!--required, xs: integer--></Channel>
  <Enable><!--required, xs: boolean--></Enable>
  <Description><!--required, xs: string--></Description>
  <TrackGUID><!--required, xs: string--></TrackGUID>
  <Size><!--optional, xs: integer--></Size>
  <Duration min="" max=""><!--optional, xs: string--></Duration>
  <DefaultRecordingMode><!--required, xs: string, default record type--></DefaultRecordingMode>
  <LoopEnable><!--optional, xs: string--></LoopEnable>
  <SrcDescriptor>
    <SrcGUID><!--required, xs: string--></SrcGUID>
    <SrcChannel><!--required, xs: integer--></SrcChannel>
    <StreamHint><!--required, xs: string--></StreamHint>
    <SrcDriver><!--required, xs: string--></SrcDriver>
    <SrcType><!--required, xs: string--></SrcType>
    <SrcUrl><!--required, xs: string--></SrcUrl>
    <SrcUrlMethods><!--required, xs: string--></SrcUrlMethods>
    <SrcLogin><!--required, xs: string--></SrcLogin>
  </SrcDescriptor>
  <TrackSchedule>
    <ScheduleBlockList>
      <ScheduleBlock>

```

```
<ScheduleBlockGUID><!--required, xs: string--></ScheduleBlockGUID>
<ScheduleBlockType><!--required, xs: string--></ScheduleBlockType>
<ScheduleAction>
  <id><!--required, xs: integer--></id>
  <ScheduleActionStartTime>
    <DayOfWeek><!--required, xs: string--></DayOfWeek>
    <TimeOfDay><!--required, xs: string--></TimeOfDay>
  </ScheduleActionStartTime>
  <ScheduleActionEndTime>
    <DayOfWeek><!--required, xs: string--></DayOfWeek>
    <TimeOfDay><!--required, xs: string--></TimeOfDay>
  </ScheduleActionEndTime>
  <ScheduleDSTEnable><!--required, xs: boolean--></ScheduleDSTEnable>
  <Description><!--required, xs: string--></Description>
  <Actions>
    <Record><!--required, xs: boolean--></Record>
    <Log><!--required, xs: boolean--></Log>
    <SaveImg><!--required, xs: boolean--></SaveImg>
    <ActionRecordingMode><!--required, xs: string, record type: temperatureIntervalMeasurement (interval
temperature measurement), CMR (continuously record according to schedule), MOTION (record triggered by motion
detection), ALARM (record triggered by alarm), EDR (record triggered by alarm/motion detection),
ALARMANDMOTION (record triggered by alarm and motion detection), Command (record by command), SMART
(smart record), AllEvent (all events)--></ActionRecordingMode>
  </Actions>
  </ScheduleAction>
</ScheduleBlock>
</ScheduleBlockList>
</TrackSchedule>
<CustomExtensionList>
  <CustomExtension>
    <CustomExtensionName><!--required, xs: string--></CustomExtensionName>
    <enableSchedule><!--required, xs: boolean--></enableSchedule>
    <SaveAudio><!--required, xs: boolean--></SaveAudio>
    <PreRecordTimeSeconds><!--required, xs: integer--></PreRecordTimeSeconds>
    <PostRecordTimeSeconds><!--required, xs: integer--></PostRecordTimeSeconds>
    <HolidaySchedule>
      <ScheduleBlock>
        <ScheduleBlockGUID><!--required, xs: string--></ScheduleBlockGUID>
        <ScheduleBlockType><!--required, xs: string--></ScheduleBlockType>
      </ScheduleBlock>
    </HolidaySchedule>
  </CustomExtension>
</CustomExtensionList>
<IntelligentRecord><!--optional, xs:boolean, whether to enable VCA recording function: 0-no, 1-yes--></
IntelligentRecord>
  <delayTime><!--optional, xs:integer, capture delay time, unit: second--></delayTime>
  <durationEnabled><!--optional, xs:boolean, whether to enable video expiry time. If this function is not supported,
this node will not be returned. If this function is supported, the video expiry date will be set by the node <Duration>--></durationEnabled>
</Track>
```

F.323 XML_Tracking

Tracking message in XML format

```
<Tracking version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <enabled>
    <!--req, xs:boolean-->
  </enabled>
  <mode>
    <!--dep, xs:string, "auto,manual"-->
  </mode>
  <trackingTime>
    <!--dep, xs:integer-->
  </trackingTime>
  <normalizedScreenSize>
    <!--req, ro-->
    <normalizedScreenWidth>
      <!--req, ro, xs:integer-->
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--req, ro, xs:integer-->
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <ManualRegionCoordinatesList min="" max="">
    <!--dep-->
    <RegionCoordinates>
      <!--opt-->
      <positionX>
        <!--req, xs:integer, X-coordinate-->
      </positionX>
      <positionY>
        <!--req, xs:integer, Y-coordinate-->
      </positionY>
    </RegionCoordinates>
  </ManualRegionCoordinatesList>
  <Schedule><!--opt, schedule, this node is valid only when enabled is "true". By default the schedule is from 00:00 to 24:00 if no schedule is configured-->
    <TimeRange>
      <beginTime>
        <!--UTC time, e.g.: "2018-03-13T19:42:27+08:00"-->
      </beginTime>
      <endTime>
        <!--UTC time, e.g.: "2018-03-13T19:42:27+08:00"-->
      </endTime>
    </TimeRange>
  </Schedule>
</Tracking>
```

F.324 XML_TrackList

TrackList message in XML format

```
<TrackList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <Track/><!--opt, recording schedule list, refer to the message XML_Track for details-->
</TrackList>
```

See Also

[XML_Track](#)

F.325 XML_TwoWayAudioChannel

TwoWayAudioChannel message in XML format

```
<TwoWayAudioChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>
    <!--req, xs:string, two-way audio channel ID-->
  </id>
  <enabled>
    <!--req, xs:boolean, whether to enable two-way audio: "true"-yes, "false"-no-->
  </enabled>
  <audioCompressionType>
    <!--req, xs:string, audio output encoding type: "G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.
729b,PCM,MP3,AC3,AAC,ADPCM,MP2L2"-->
  </audioCompressionType>
  <audioInboundCompressionType>
    <!--opt, xs:string, audio input encoding type: "G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.
729b,PCM,MP3,AC3,AAC,ADPCM"-->
  </audioInboundCompressionType>
  <speakerVolume>
    <!--opt, xs:integer, input volume-->
  </speakerVolume>
  <microphoneVolume>
    <!--opt, xs:integer, output volume-->
  </microphoneVolume>
  <noisereduce>
    <!--opt, xs:boolean, whether to enable noise reduction: "true, false"-->
  </noisereduce>
  <audioBitRate>
    <!--opt, xs:integer, audio frame rate, unit: kbs-->
  </audioBitRate>
  <audioInputType>
    <!--opt, xs:string, audio input type: "MicIn, LineIn"-->
  </audioInputType>
  <associateVideoInputs><!--opt-->
  <enabled>
    <!--req, xs:boolean-->
```

```
</enabled>
<videoInputChannelList>
  <!--req-->
  <videoInputChannelID>
    <!--opt, xs:string, ID-->
  </videoInputChannelID>
</videoInputChannelList>
</associateVideoInputs>
<lineOutForbidden>
  <!--read-only, xs:boolean, whether the audio output is not supported, if this node is not returned or the value is "false", it represents that audio output is supported; if the value is "true", it represents that audio output is not supported-->
</lineOutForbidden>
<micInForbidden>
  <!--read-only, xs: boolean, whether the audio input is not supported, if this node is not returned or the value is "false", it represents that audio input is supported; if the value is "true", it represents that audio input is not supported-->
</micInForbidden>
</TwoWayAudioChannel>
```

F.326 XML_TwoWayAudioChannelList

TwoWayAudioChannelList message in XML format

```
<TwoWayAudioChannelList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <TwoWayAudioChannel/><!--opt, see details in the message of XML_TwoWayAudioChannel-->
</TwoWayAudioChannelList>
```

See Also

[XML_TwoWayAudioChannel](#)

F.327 XML_User

XML message about a specific user's configuration

```
<User version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--req, xs:integer, the value is between 1 and 16, the ID of the administrator account is 1 and the administrator account cannot be deleted--></id>
  <enabled><!--opt, xs:boolean, whether to enable the user: "true"-yes, "false"-no--></enabled>
  <userName><!--req, xs:string, the sensitive information should be encrypted--></userName>
  <password><!--wo, req, xs:string, the sensitive information should be encrypted--></password>
  <keypadPassword><!--wo, opt, xs:string, keypad password, the sensitive information should be encrypted--></keypadPassword>
  <loginPassword><!--wo, dep, xs:string, this node depends on security in the URL and it is required when security exists. The sensitive information should be encrypted, and the password is the administrator password--></loginPassword>
  <userOperateType><!--opt, xs:integer, user operation type: "1"-network user, "2"-keypad user, "3"-network user and keypad user. When this node is set to 1 or NULL, password is valid, and password is required when the method is
```

POST, otherwise it is optional. When this node is set to 2, **keypadPassword** is valid, and **keypadPassword** is required when the method is POST, otherwise it is optional. When this node is set to 3, both **password** and **keypadPassword** are valid, and they are required when the method is POST, otherwise they are optional--></userOperateType>

```
<bondIpAddressList><!--opt-->
<bondIpAddress><!--opt-->
<id><!--req, xs:integer--></id>
<ipAddress><!--dep, xs:string--></ipAddress>
<ipv6Address><!--dep, xs:string--></ipv6Address>
</bondIpAddress>
</bondIpAddressList>
<bondMacAddressList><!--opt-->
<bondMacAddress><!--opt-->
<id><!--req, xs:integer--></id>
<macAddress><!--opt, xs:string--></macAddress>
</bondMacAddress>
</bondMacAddressList>
<userLevel><!--opt, xs:string, "Administrator,Operator,Viewer"--></userLevel>
<attribute><!--opt-->
<inherent><!--xs:boolean--></inherent>
</attribute>
<cardList>
<!--optional, object, list of cards that belong to the user-->
<card>
<!--optional, object, card-->
<id>
<!--optional, int, card ID, range:[1,16]-->1
</id>
<name>
<!--optional, string, card name, the maximum length is 32 bytes-->test
</name>
</card>
</cardList>
<remoteCtrlList>
<!--optional, object, list of keyfobs that belong to the user-->
<remoteCtrl>
<!--optional, object, keyfob-->
<id>
<!--optional, int, keyfob ID, range:[1,16]-->1
</id>
<name>
<!--optional, string, keyfob name, the maximum length is 32 bytes-->test
</name>
</remoteCtrl>
</remoteCtrlList>
<userNo>
<!--optional, int, displayed No. corresponding to the user ID-->501
</userNo>
<adminType>
<!--optional, string, read-only, admin type, "Cloud" (user account created in the cloud), "LAN" (user account created on LAN)-->Cloud
</adminType>
<installerType>
```

```
<!--optional, string, read-only, installer type, "Cloud" (user account created in the cloud), "LAN" (user account  
created on LAN)-->Cloud  
</installerType>  
</User>
```

F.328 XML_UserAbility

UserAbility message in XML format

```
<UserAbility version="2.0">  
  <userNum><!--req, number of users--></userNum>  
  <userPriority opt="admin,viewer,operator,none"/><!--req, user priority-->  
  <userNameLength min="1" max="32"/><!--req, the length of user name-->  
  <userPasswordLength min="1" max="16"/><!--req, the length of password-->  
  <LocalPermission>  
    <permissionType type="admin"  
opt="PTZControl,record,playback,parameterConfig,logOrStateCheck,restartOrShutdown,upgrade,backup,preview,playb  
ackDoubleVerification"/>  
    <!--req, local permission types of admin users, if the type is null, it indicates that the user has all permissions-->  
  </LocalPermission>  
  <LocalPermission>  
    <permissionType type="viewer"  
opt="PTZControl,record,playback,parameterConfig,logOrStateCheck,restartOrShutdown,upgrade,backup,preview,playb  
ackDoubleVerification"/>  
    <!--req, local permission types of visitors, if the type is null, it indicates that the user has all permissions-->  
  </LocalPermission>  
  <LocalPermission>  
    <permissionType type="operator"  
opt="PTZControl,record,playback,parameterConfig,logOrStateCheck,restartOrShutdown,upgrade,backup,preview,playb  
ackDoubleVerification"/>  
    <!--req, local permission types of operators, if the type is null, it indicates that the user has all permissions-->  
  </LocalPermission>  
  <RemotePermission>  
    <permissionType type="admin" opt="preview, voiceTalk, alarmOutOrUpload, transParentChannel,  
controlLocalOut,PTZControl,record,playback,parameterConfig,logOrStateCheck,restartOrShutdown,upgrade,backup,pl  
aybackDoubleVerification"/>  
    <!--req, remote permission types of admin users, if the type is null, it indicates that the user has all permissions-->  
  </RemotePermission>  
  <RemotePermission>  
    <permissionType type="viewer" opt="preview, voiceTalk, alarmOutOrUpload, transParentChannel,  
controlLocalOut,PTZControl,record,playback,parameterConfig,logOrStateCheck,restartOrShutdown,upgrade,backup,pl  
aybackDoubleVerification"/>  
    <!--req, remote permission types of visitors, if the type is null, it indicates that the user has all permissions-->  
  </RemotePermission>  
  <RemotePermission>  
    <permissionType type="operator" opt="preview, voiceTalk, alarmOutOrUpload, transParentChannel,  
controlLocalOut,PTZControl,record,playback,parameterConfig,logOrStateCheck,restartOrShutdown,upgrade,backup,pl  
aybackDoubleVerification"/>  
    <!--req, remote permission types of operators, if the type is null, it indicates that the user has all permissions-->  
  </RemotePermission>
```

```
<UserNet>
  <IPv4Address>
    <!--opt, whether it supports the user to assign the address of IPV4, if not supports, this node will not be returned-->
  </IPv4Address>
  <IPv6Address>
    <!--opt, whether it supports the user to assign the address of IPV6, if not supports, this node will not be returned-->
  </IPv6Address>
  <MACAddress>
    <!--opt, whether it supports the user to assign the MAC address, if not supports, this node will not be returned-->
  </MACAddress>
</UserNet>
<AlarmPermission><!--opt, permission of alarm inut-->
  <alarmOn><!--opt, arming permission of alarm input--></alarmOn>
  <alarmOff><!--opt, disarming permission of alarm input--></alarmOff>
  <alarmBypass><!--opt, bypass permission of alarm input--></alarmBypass>
</AlarmPermission>
<AlarmhostPermission><!--opt, permission of network security contorl panel-->
  <netUserNo min="" max="" /><!--req, network user No.-->
  <userPriority opt="admin,manage,operater" /><!--req, user type (user priority): supper admin, admin, operator-->
    <AdminPermissionType opt="arm,disarm,bypass,preview,record,playback,PTZControl,log,reboot,
setConfig,getConfig,resume,siren,upgrade,alarmout,serialControl,getwayControl,voiceTalk,
controlLocalOut,diskConfig,formatDisk,sensorControl" />
    <!--req, remote permission types of supper admin user-->
    <ManagerPermissionType opt="arm,disarm,bypass,preview,record,playback,PTZControl,log,reboot,
setConfig,getConfig,resume,siren,upgrade,alarmout,serialControl,getwayControl,voiceTalk,
controlLocalOut,diskConfig,formatDisk,sensorControl" />
    <!--req, remote permission types of admin user-->
    <OperatorPermissionType opt="arm,disarm,bypass,preview,record,playback,PTZControl,log,reboot,
setConfig,getConfig,resume,siren,upgrade,alarmout,serialControl,getwayControl,voiceTalk,
controlLocalOut,diskConfig,formatDisk,sensorControl" />
    <!--req, remote permission types of operator-->
</AlarmhostPermission>
<ViewerDefaultPermission>
  <LocalPermission>
    <permissionType
opt="PTZControl,record,playback,parameterConfig,logOrStateCheck,restartOrShutdown,upgrade,backup, preview" />
    <!--req, local permission types-->
  </LocalPermission>
  <RemotePermission>
    <permissionType opt="preview, voiceTalk, alarmOutOrUpload, transParentChannel, contorlLocalOut, PTZControl,
record, playback, parameterConfig, logOrStateCheck, restartOrShutdown, upgrade, backup" />
    <!--req, remote permission types-->
  </RemotePermission>
</ViewerDefaultPermission>
<OperatorDefaultPermission>
  <LocalPermission>
    <permissionType
opt="PTZControl,record,playback,parameterConfig,logOrStateCheck,restartOrShutdown,upgrade,backup, preview" />
    <!--req, local permission types-->
  </LocalPermission>
```

```
<RemotePermission>
  <permissionType opt="preview, voiceTalk, alarmOutOrUpload, transParentChannel, contorlLocalOut, PTZControl,
record, playback, parameterConfig, logOrStateCheck, restartOrShutdown, upgrade, backup"/>
  <!--req,remote permission types-->
</RemotePermission>
</OperatorDefaultPermission>
<VideoWallPermission><!--opt, permission of decoders and video wall controllers-->
<RemotePermission><!--req, remote permission types-->
  <permissionType opt="baseParameterConfig, screen, window, inputSignal, layout, plan, baseMap, OSD,
pictureView, arm, upgrade, restore, importFileOrExportFile, log, restart, manageScreenArea"/>
</RemotePermission>
</VideoWallPermission>
<loginPassword min="" max="">
  <!--opt, confirm password-->
</loginPassword>
</UserAbility>
```

F.329 XML_UserList

UserList message in XML format

```
<UserList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <User/><!--opt, see details in the message of XML_User-->
</UserList>
```

See Also

[XML_User](#)

F.330 XML_UserPermission

XML message about a specific user's permission

```
<UserPermission version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--req, xs:string--></id>
  <userID><!--req, xs:string, user ID--></userID>
  <userType>
    <!--req, xs:string, user type: "admin"-administrator, which has all permissions and can review and edit user's
permission, "operator"-operator, which has default permissions, "viewer"-viewer, which has default permissions,
"installer", "manufacturer"-->
  </userType>
  <localPermission/>
    <!--opt, local permission, see details in the message of XML_localPermission -->
  <remotePermission/>
    <!--opt, remote permission, see details in the message of XML_remotePermission -->
</UserPermission>
```

See Also

[XML_localPermission](#)

[XML_remotePermission](#)

F.331 XML_UserPermissionCap

XML message about the capability of managing the user permission

```
<UserPermissionCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <userType><!--req, xs:string, "admin"-administrator, "operator", "viewer", "installer", "manufacturer"--></userType>
  <localPermissionCap><!--opt, see details in the message of XML_localPermissionCap--></localPermissionCap>
  <remotePermissionCap><!--opt, see details in the message of XML_remotePermissionCap--></
remotePermissionCap>
</UserPermissionCap>
```

See Also

[XML_localPermissionCap](#)

[XML_remotePermissionCap](#)

F.332 XML_UserPermissionList

XML message about all users' permission

```
<UserPermissionList version="2.0" xmlns=" http://www.isapi.org/ver20/XMLSchema">
  <UserPermission/>
    <!--opt, user permission, see details in the message of XML_UserPermission-->
</UserPermissionList>
```

See Also

[XML_UserPermission](#)

F.333 XML_VCAResource

Message about intelligent resources switch in XML format

```
<VCAResource version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <type>
    <!--required, xs:string,
"basicBehavior,fullBehavior,facesnapBehavior,facesnap,TFS,smartVehicleDetection,smartHVTDetection,smart,judicial,s
mart264AndRoadDetection,smart264AndFaceDetection,smart264AndHeatMap,smartIntelligentMonitor,smartTrafficD
ataCollection,roadDetection,humanRecognition,perimeterCapture,vehicleDetection,HVTDetection,mixedTargetDetecti
on,trackingCaptureMode,nonTrackingCaptureMode,close,faceHumanModelingContrast,cityManagement,teacherBeha
vior,
12MPLiveView,personQueueDetection,verticalPeopleCounting,safetyHelmet,faceCounting,personArming,AIOpenPlatfo
```

```
rm"-->
</type>
</VCAResource>
```

F.334 XML_VideoCap

VideoCap message in XML format

```
<VideoCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<videoInputPortNums><!--opt, xs: integer--><videoInputPortNums>
<videoOutputPortNums><!--opt, xs: integer--><videoOutputPortNums>
<isSupportHeatmap><!--opt, xs: boolean, whether to support heat map function--></isSupportHeatmap>
<isSupportCounting><!--opt, xs: boolean--></isSupportCounting>
<countingType><!--dep, xs: string, "human,object"--></countingType>
<isSupportPreviewSwitch><!--opt, xs: boolean, whether to support live view switch--></isSupportPreviewSwitch>
<isSupportRecodStatus>
  <!--opt, xs: boolean, whether supports getting recording status-->
</isSupportRecodStatus>
<isSupportPrivacyMask>
  <!--opt, xs: boolean, whether supports priavte mask function-->
</isSupportPrivacyMask>
<isSupportBinocularPreviewSwitch>
  <!--opt, xs: boolean, whether supports auto-switch during the live view of dual lens camera-->
</isSupportBinocularPreviewSwitch>
<isSupportCalibCheck>
  <!--opt, xs: boolean, whether supports calibration verification-->
</isSupportCalibCheck>
<isSupportPIP><!--opt, xs: boolean, "true, false"--></isSupportPIP>
<channelFlexible opt ="name,enable,online,linknum">
  <!--capability of getting channel status by condition-->
</channelFlexible >
<isSupportFocusVideoMode>
  <!--opt, xs: boolean, video focus mode for installation and debug-->
</isSupportFocusVideoMode>
<isSupportExternalChannel>
  <!--opt, xs: boolean, whether supports extending analog channel-->
</isSupportExternalChannel>
<isSupportMultiChannelCounting>
  <!--opt, xs: boolean, whether supports people counting of multiple channels-->
</isSupportMultiChannelCounting>
<isSupportCountingCollection>
  <!--opt, xs:boolean, whether supports people counting data replenishment-->
</isSupportCountingCollection>
<isSupportHeatmapCollection>
  <!--opt, xs:boolean, whether supports heat map data replenishment-->
</isSupportHeatmapCollection>
<OSDLanguage opt="GBK,EUC-KR" def="GBK"/>
<isSupportInitLens><!--req, xs:boolean, whether to support initializing lens--><isSupportInitLens>
<isSupportOneFocus><!--req, xs:boolean, whether to support one-touch focusing--><isSupportOneFoucs>
<notSupportFocus><!--req, xs:boolean, the focus capability is not supported--><notSupportFoucs>
```

```

<notSupportIris><!--req, xs:boolean, the iris capability is not supported--><notSupportIris>
<isSupportCapturePicOverlays><!--opt, xs:boolean, whether to support text overlay on the captured picture--></
isSupportCapturePicOverlays>
<isSupportMergePicOverlays><!--opt, xs:boolean, whether to support text overlay on the composite picture--></
isSupportMergePicOverlays>
</VideoCap>

```

F.335 XML_VideoPic_CMSearchDescription

XML message about search conditions of video and picture

```

<CMSearchDescription version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<searchID>
  <!--required, xs:string, search ID, e.g., {812F04E0-4089-11A3-9A0C-0305E82C2906}-->
</searchID>
<trackIDList>
  <trackID>
    <!--required, xs:integer, ID, XX01-search for the main stream video of channel XX, XX03-search for the pictures of
channel XX-->
    </trackID>
  </trackIDList>
<timeSpanList>
  <timeSpan>
    <startTime><!--required, xs:time, ISO8601 time, start time of search, e.g.: 2017-08-02T00:00:00Z--></startTime>
    <endTime><!--required, xs:time, ISO8601 time, end time of search, e.g.: 2017-08-02T23:59:59Z--></endTime>
  </timeSpan>
</timeSpanList>
<searchTypeList>
  <searchKeyword><!--required, xs:string, keyword list for search, the maximum length is 32 and the minimum length
is 1--></searchKeyword>
</searchTypeList>
<contentTypeList>
  <contentType><!--required, xs:string, file type to be searched for: "video", "audio", "metadata"-picture, "text",
"mixed", "other"--></contentType>
  <pircamID><!--optional, list, xs:integer, pircam search ID. This node is valid only when the search type is pircam.
Types of pictures captured by pircams in the device may be different, and this node is used to distinguish between
different pircams--></pircamID>
</contentTypeList>
<uploadState><!--optional, xs:string, uploading status: all, uploaded, nonUploaded, uploadFailed--></uploadState>
<lockState><!--optional, xs:string, locking status: lock, unlock, all--></lockState>
<searchResultPostion><!--required, xs:integer, start position of search--></searchResultPostion>
<maxResults><!--optional, xs:integer, maximum number of returned results--></maxResults>
<metadataList>
  <metadataDescriptor>
    <!--required, xs:string, description of file to be searched for, format: "recordType.meta.xxx.com/<eventType>",
<eventType> is the event type to which related video should be searched for; for details, see the table below in
remarks. For example, to search for the video/picture of motion detection, the value of contentType is
"video"/"metadata" and the value of metadataDescriptor is "recordType.meta.xxx.com/MOTION"-->
    <!--to search for all videos, the value of contentType is "video" and value of metadataDescriptor is
"recordType.meta.xxx.com"; to search for all pictures, the value of contentType is "metadata" and value of

```

```

metadataDescriptor is "recordType.meta.xxx.com/allPic"-->
    <!--the supported video/picture type of device is returned in the recordSearchType/pictureSearchType of
XML_RacmCap (related URI: /ISAPI/ContentMgmt/capabilities)-->
</metadataDescriptor>
<SearchProperty>
    <plateSearchMask>
        <!--optional, xs:string, ranges from 1 to 31-->
    </plateSearchMask>
    <stateOrProvince>
        <!--optional, xs:integer-->
    </stateOrProvince>
    <country>
        <!--optional, xs:string, country or region ID: 0-the algorithm library does not support the country or region ID,1-
(CZ-Czech Republic),2-(FRA-France),3-(DE-Germany),4-(E-Spain),5-(IT-Italy),6-(NL-Netherlands),7-(PL-Poland),8-(SVK-
Slovakia), 9-(BY-Belorussia), 10-(MDA-Moldova),11-(RU-Russia),12-(UA-Ukraine), 0xff-(All)-->
    </country>
    <subType><!--optional, xs:string, sub type: all,motorVehicle,nonMotorVehicle,pedestrian--></subType>
    <capTypeDescriptor><!--optional, xs:string, capability type descriptor: anyType, event, evidence--></
capTypeDescriptor>
    <nation><!--optional, xs:string, region: "EU"-the Europe, "ER"-the Middle East, "AP"-the Asia Pacific,
"AfricaAndAmerica"-Africa and America-->EU</nation>
    <vehicleType><!--optional, xs:string, vehicle type: "bus", "truck", "vehicle", "van", "buggy", "SUVMPV"-SUV/MPV,
"pickupTruck"-pickup truck-->bus</vehicleType>
</SearchProperty>
</metadataList>
</CMSearchDescription>

```

Remarks

eventType	Description
AllEvent	All events
safetyHelmet	Hard hat detection
smokeDetection	Smoke detection
leavePosition	Absence detection
peopleNumChange	The number of people changed
linedetection	Line crossing detection
fielddetection	Intrusion detection
regionEntrance	Region entrance detection
regionExiting	Region exiting detection
loitering	Loitering detection
group	People gathering detection

eventType	Description
rapidMove	Fast moving detection
unattendedBaggage	Unattended baggage detection
attendedBaggage	Object removal detection
CMR	Scheduled recording or capture
ALARM	Alarm
EDR	Motion detection or alarm
ALARMANDMOTION	Motion detection and alarm
Command	Command is triggered
pir	PIR
wlsensor	Wireless alarm
callhelp	Calling alarm
facedetection	Face detection
parking	Parking
vehicleDetection	Vehicle detection
manual	Manual
manualSnapShot	Manual capture
playSnapShot	Capture of playback
dredgerDetection	Dredger detection
accessController	Access controller event
securityControlPanel	Security control panel event
violentMotion	Violent motion alarm
advReachHeight	Climbing detection
toiletTarry	In-Toilet overtime
audioAbnormal	Sudden change of sound intensity
standUp	Standing up
getUp	Getting up
vehicleMonitor	Vehicle arming
playCellphone	Playing cellphone

eventType	Description
retention	Overstay detection
failDown	People falling down alarm
sleepOnDuty	Sleeping on duty alarm
allPerimeterEvent	All perimeter events
allBehaviorEvent	All behavior analysis events
running	People running
spacingChange	Distance changing detection
faceSnapModeling	Face capture modeling
reverseEntrance	Reverse entering
studentsStoodUp	Student standing up detection
ATMPanel	ATM panel mode
ATMSurround	ATM environment mode
ATMFace	ATM human face mode
ATMSafetyCabin	ATM safety cabin mode
temperatureIntervalMeasurement	Interval temperature measurement
roadMaint	Road maintenance
pothole	Road pothole
crack	Road crack

F.336 XML_VideoPic_CMSearchResult

XML message about search results of videos and pictures

```
<CMSearchResult version="2.0" xmlns="http://www.isapi.com/ver20/XMLSchema">
<searchID><!--req, xs:string, search ID--></searchID>
<responseStatus><!--req, xs:string--></responseStatus>
<responseStatusStrg><!--req, xs:string--></responseStatusStrg>
<numOfMatches><!--opt, xs:integer--></numOfMatches>
<matchList>
<searchMatchItem>
<sourceID></sourceID>
<trackID></trackID>
<timeSpan>
```

```

<startTime></startTime>
<endTime></endTime>
</timeSpan>
<mediaSegmentDescriptor>
    <contentType><!--req, xs:string, file type, corresponding to search condition--></contentType>
    <codecType><!--req, xs:string, video encoding format--></codecType>
    <rateType><!--opt, xs:string, video bit rate,4Mbps--></rateType>
    <playbackURI><!--opt, xs:string, video file URL, "name=" and "size=" are file name and file size respectively, e.g., rtsp://10.65.130.168/Streaming/tracks/201/?starttime=20190213T091134Z&endtime=20190213T092116Z&name=01010000093000100&size=732335288--></playbackURI>
        <lockStatus><!--opt, xs:string, status of video file: lock-locked file, unlock-unlocked file--></lockStatus>
        <remainLockTime><!--dep, xs:integer, unit:s, remaining locking time, which is returned only when lockStatus is set to lock--></remainLockTime>
            <name><!--opt, xs:string, file name--></name>
            <size><!--opt, xs:integer, file size--></size>
            <mediaID><!--opt, xs:integer, file ID--></mediaID>
            <remark><!--opt, xs:string, file tag--></remark>
            <GPSInfo><!--opt, xs:string, file GPS information--></GPSInfo>
            <recorderCode><!--opt, xs:string, body camera No.--></recorderCode>
            <uploadState><!--opt, xs:string, status: uploaded, nonUploaded, uploadFailed, uploading--></uploadState>
            <uploadTime><!--dep, xs:datetime, time of completing video uploading, e.g., "2004-05-03T17:30:08+08:00"--></uploadTime>
                <policeCode><!--opt, xs:string, No. of the policeman who recorded the video--></policeCode>
                <policeName><!--opt, xs:string, name of the policeman who recorded the video--></policeName>
                <shootingTime><!--opt, xs:datetime, captured or recorded time, e.g., "2004-05-03T17:30:08+08:00"--></shootingTime>
                    <FileTime><!--opt, xs:integer, time duration of the video file, unit:min--></FileTime>
                    <remarks><!--opt, xs:string, file remarks--></remarks>
                    <cloudStorageType><!--opt, xs:string, storage server type: 0-cloud storage, 1-cvr, 2-pstor, which should be consistent with cloudStorageType in /ISAPI/ContentMgmt/channels/<ID>/cloudStorage/<ID>--></cloudStorageType>
                        <fileUrl><!--opt, xs:string, file URL--></fileUrl>
                        <thumbnailUrl><!--opt, xs:string, thumbnail URL of picture or video--></thumbnailUrl>
                        <encryption>
                            <enabled><!--opt, xs:boolean, whether to encrypt the file--></enabled>
                            <encryptionkey><!--opt, xs:string, encryption key, which is returned only when enabled is set to true--></encryptionkey>
                                <encryptionLevel><!--opt, xs:integer, encryption level: 1-AES128, 2-AES256, which is returned only when enabled is set to true--></encryptionLevel>
                                    </encryption>
                        </mediaSegmentDescriptor>
                        <metadataMatches>
                            <metadataDescriptor><!--searching conditions: "evidence,illegalParking,crosslane,vehicleexist,lanechange,wrongdirection,congestion,turnround,pedestrian,construction,roadBlock,abandonedObject,trafficAccident,fogDetection"--></metadataDescriptor>
                            </metadataMatches>
                        </searchMatchItem>
                    </matchList>
                </CMSearchResult>

```

F.337 XML_WebCertificate

WebCertificate message in XML format

```
<WebCertificate version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<CertificateType>
  <!--required, xs:string, authentication type: "basic", "digest", "digest/baisc"-->
</CertificateType>
<SecurityAlgorithm><!--dependent, this node is valid when certificateType is "digest" or "digest/basic"-->
<algorithmType>
  <!--optional, xs:string, algorithm type: "MD5", "SHA256", "MD5/SHA256"-->
</algorithmType>
</SecurityAlgorithm>
</WebCertificate>
```

F.338 XML_WirelessDialInterface

WirelessDialInterface message in XML format

```
<WirelessDialInterface version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id><!--req, xs:string, only one network card is supported--> </id>
<Dial/><!--opt,refer to the URI: /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dial-->
<Schedule/><!--opt ,refer to the URI: /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/schedule-->
<Dialstatus/><!--opt,refer to the URI: /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/dialstatus-->
<messageConfig/><!--opt,refer to the URI: /ISAPI/System/Network/WirelessDial/Interfaces/<ID>/messageConfig-->
<messageList/><!--opt-->
</WirelessDialInterface>
```

F.339 XML_WirelessDialInterfaceList

WirelessDialInterfaceList message in XML format

```
<WirelessDialInterfaceList version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<WirelessDialInterface/>
</WirelessDialInterfaceList>
```

F.340 XML_ZoomFocus

ZoomFocus message in XML format

```
<ZoomFocus version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<pqrssZoom/><!--opt, xs: integer, zoom coordinates of Sony zoom camera module-->
<mnstFocus/><!--opt, xs: integer, focus coordinates of Sony zoom camera module-->
</ZoomFocus>
```

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

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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 URLs 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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.
unitInfoEror	0x4000101C	Getting data analysis server information failed.
unitGetNodeInfoError	0x4000101D	Getting node information failed.
unitGetNetworkInfoErr or	0x4000101E	Getting the network information of data analysis server failed
unitSetNetworkInfoErr or	0x4000101F	Setting the network information of data analysis server failed
setSmartNodeInfoError	0x40001020	Setting node information failed.
setUnitNetworkInfoErr or	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.
uintVersionMismatche d	0x40001026	Data analysis server version mismatches.
deviceMOdelMismatch ed	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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.
numberReachedLlimit	0x40001119	Number reached the limit.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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.

Sub Status Code	Error Code	Description
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	Abnormal event detection 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"> • Deleting your own account is not allowed. • Editing your own level or permission is not allowed. • Getting information about users with higher permission is not allowed. • Elevating the user's level or permission is not allowed. <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">• The message contains only URL but no message body• The required node is not configured.• Node value exceeds the range limit (incorrect node value).	
badJsonContent	0x00f0000e	JSON message error: <ul style="list-style-type: none">• The message contains only URL but no message body• The required node is not configured.• Node value exceeds the range limit (incorrect node value).	
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"> • Check if the network connection is normal. • Check if the device is running normally. • Check the log print.
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.

