



# **Device Network SDK (ANPR)**

**Developer Guide**

## Legal Information

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

# Contents

<b>Chapter 1 Overview .....</b>	<b>1</b>
1.1 Introduction .....	1
1.2 Update History .....	1
<b>Chapter 2 Configure ANPR Alarm .....</b>	<b>2</b>
<b>Chapter 3 Configure Blocklist and Allowlist ANPR Alarm .....</b>	<b>6</b>
<b>Chapter 4 Control Barrier Gate Status .....</b>	<b>11</b>
<b>Chapter 5 Alarm/Event Receiving .....</b>	<b>13</b>
5.1 Receive Alarm/Event in Arming Mode .....	13
5.2 Receive Alarm/Event in Listening Mode .....	16
<b>Chapter 6 API Reference .....</b>	<b>20</b>
6.1 NET_DVR_GetDeviceAbility .....	20
6.2 NET_DVR_GetDeviceConfig .....	21
6.3 NET_DVR_GetDownloadState .....	22
6.4 NET_DVR_GetSTDAbility .....	22
6.5 NET_DVR_GetSTDConfig .....	23
6.6 NET_DVR_GetUploadState .....	24
6.7 NET_DVR_RemoteControl .....	26
6.8 NET_DVR_SetDeviceConfig .....	27
6.9 NET_DVR_SetSTDConfig .....	28
6.10 NET_DVR_STDXMLConfig .....	29
6.11 NET_DVR_StartDownload .....	30
6.12 NET_DVR_StopDownload .....	31
6.13 NET_DVR_UploadClose .....	31
6.14 NET_DVR_UploadFile_V40 .....	32
6.15 NET_DVR_Cleanup .....	33
6.16 NET_DVR_GetErrorMsg .....	33

6.17 NET_DVR_GetLastError .....	34
6.18 NET_DVR_Init .....	34
6.19 NET_DVR_Login_V40 .....	35
6.19.1 fLoginResultCallBack .....	35
6.20 NET_DVR_Logout .....	36
6.21 NET_DVR_SetSDKInitCfg .....	36
6.22 NET_DVR_CloseAlarmChan_V30 .....	38
6.23 NET_DVR_GetDVRConfig .....	38
6.24 NET_DVR_SetDVRConfig .....	39
6.25 NET_DVR_SetDVRMessageCallBack_V50 .....	40
6.25.1 MSGCallBack .....	48
6.26 NET_DVR_SetupAlarmChan_V50 .....	49
6.27 NET_DVR_StartListen_V30 .....	50
6.28 NET_DVR_StopListen_V30 .....	51
<b>Chapter 7 Structure and Enumeration .....</b>	<b>52</b>
7.1 Data Structure .....	52
7.1.1 ITC_LANE_DIRECTION_TYPE .....	52
7.1.2 ITC_LANE_USEAGE_TYPE .....	53
7.1.3 ITS_OVERLAP_ITEM_TYPE .....	54
7.1.4 ITC_RELA_LANE_DIRECTION_TYPE .....	58
7.1.5 ITC_TRIGGERMODE_TYPE .....	59
7.1.6 ITC_VIOLATION_DETECT_TYPE .....	60
7.1.7 NET_DVR_BARRIERGATE_CFG .....	62
7.1.8 NET_DVR_CRUISECHAN_INFO .....	63
7.1.9 NET_DVR_EVENT_SCHEDULE .....	63
7.1.10 NET_DVR_EVENT_TRIGGER .....	63
7.1.11 NET_DVR_GEOGLOCATION .....	64
7.1.12 NET_DVR_GUARD_CFG .....	65

7.1.13 NET_DVR_GUARD_COND .....	66
7.1.14 NET_DVR_HANDLEEXCEPTION_V41 .....	67
7.1.15 NET_DVR_INIT_CFG_ABILITY .....	67
7.1.16 NET_DVR_LLI_PARAM .....	68
7.1.17 NET_DVR_LLPOS_PARAM .....	69
7.1.18 NET_DVR_MIME_UNIT .....	69
7.1.19 NET_DVR_PLATE_INFO .....	70
7.1.20 NET_DVR_PLATE_RESULT .....	72
7.1.21 NET_DVR_PRESETCHAN_INFO .....	75
7.1.22 NET_DVR_PTZTRACKCHAN_INFO .....	75
7.1.23 NET_DVR_SCHEDTIME .....	75
7.1.24 NET_DVR_STD_ABILITY .....	76
7.1.25 NET_DVR_STD_CONFIG .....	76
7.1.26 NET_DVR_TIME_V30 .....	77
7.1.27 NET_DVR_TIME_V50 .....	78
7.1.28 NET_DVR_TIME_DETECTION .....	79
7.1.29 NET_DVR_TRIGGER_COND .....	80
7.1.30 NET_DVR_VEHICLE_ADDINFO .....	80
7.1.31 NET_DVR_VEHICLE_CONTROL_ALARM .....	81
7.1.32 NET_DVR_VEHICLE_INFO .....	83
7.1.33 NET_DVR_VIA_LANE_PARAM .....	85
7.1.34 NET_DVR_VIA_VTCOIL_PARAM .....	86
7.1.35 NET_DVR_XML_CONFIG_INPUT .....	86
7.1.36 NET_DVR_XML_CONFIG_OUTPUT .....	87
7.1.37 NET_IPC_LANE_HVT_PARAM .....	88
7.1.38 NET_IPC_POST_HVT_PARAM .....	89
7.1.39 NET_ITC_EPOLICE_IOTL_PARAM .....	90
7.1.40 NET_ITC_EPOLICE_LANE_PARAM .....	91

7.1.41 NET_ITC_EPOLICE_RS485_PARAM .....	93
7.1.42 NET_ITC_INTERVAL_PARAM .....	94
7.1.43 NET_ITC_IO_LIGHT_PARAM .....	95
7.1.44 NET_ITC_LANE_HVT_PARAM_V50 .....	95
7.1.45 NET_ITC_LANE_IMT_PARAM .....	98
7.1.46 NET_ITC_LANE_LOGIC_PARAM .....	99
7.1.47 NET_ITC_LANE_MPR_PARAM .....	100
7.1.48 NET_ITC_LANE_NOCOMITY_PEDESTRIAN_PARAM .....	102
7.1.49 NET_ITC_LANE_PARAM .....	103
7.1.50 NET_ITC_LANE_PRS_PARAM .....	105
7.1.51 NET_ITC_LANE_VIDEO_EPOLICE_PARAM .....	107
7.1.52 NET_ITC_LIGHT_ACCESSPARAM_UNION .....	109
7.1.53 NET_ITC_LINE .....	109
7.1.54 NET_ITC_NOCOMITY_PEDESTRIAN_PARAM .....	111
7.1.55 NET_ITC_PLATE_RECOG_PARAM .....	112
7.1.56 NET_ITC_PLATE_RECOG_REGION_PARAM .....	113
7.1.57 NET_ITC_POLYGON .....	114
7.1.58 NET_ITC_POST_HVT_PARAM_V50 .....	114
7.1.59 NET_ITC_POST_IMT_PARAM .....	116
7.1.60 NET_ITC_POST_IOSPEED_PARAM .....	117
7.1.61 NET_ITC_POST_MOBILE_PARAM .....	117
7.1.62 NET_ITC_POST_MPR_PARAM .....	119
7.1.63 NET_ITC_POST_PRS_PARAM .....	120
7.1.64 NET_ITC_POST_RS485_PARAM .....	121
7.1.65 NET_ITC_POST_RS485_RADAR_PARAM .....	122
7.1.66 NET_ITC_POST_SINGLEIO_PARAM .....	123
7.1.67 NET_ITC_POST_VTCOIL_PARAM .....	124
7.1.68 NET_ITC_POST_MPR_PARAM .....	125

7.1.69 NET_ITC_RADAR_PARAM .....	126
7.1.70 NET_ITC_REDLIGHT_PEDESTRIAN_PARAM .....	127
7.1.71 NET_ITC_RS485_LIGHT_PARAM .....	129
7.1.72 NET_ITC_SINGLE_IO_LIGHT_PARAM .....	129
7.1.73 NET_ITC_SINGLE_IOSPEED_PARAM .....	130
7.1.74 NET_ITC_SINGLE_IOTL_PARAM .....	133
7.1.75 NET_ITC_SINGLE_RS485_LIGHT_PARAM .....	135
7.1.76 NET_ITC_SINGLE_TRIGGERCFG .....	136
7.1.77 NET_ITC_SINGLE_VIDEO_DETECT_LIGHT_PARAM .....	137
7.1.78 NET_ITC_SINGLEIO_PARAM .....	138
7.1.79 NET_ITC_TRAFFIC_LIGHT_PARAM .....	139
7.1.80 NET_ITC_TRIGGER_PARAM_UNION .....	140
7.1.81 NET_ITC_TRIGGERCFG .....	142
7.1.82 NET_ITC_VIDEO_DETECT_LIGHT_PARAM .....	142
7.1.83 NET_ITC_VIOLATION_DETECT_LINE .....	143
7.1.84 NET_ITC_VIOLATION_DETECT_PARAM .....	144
7.1.85 NET_ITS_ILLEGAL_INFO .....	146
7.1.86 NET_ITS_OVERLAP_CFG_V50 .....	146
7.1.87 NET_ITS_OVERLAP_INFO_PARAM .....	147
7.1.88 NET_ITS_OVERLAP_ITEM_PARAM_V50 .....	148
7.1.89 NET_ITS_OVERLAP_SINGLE_ITEM_PARAM_V50 .....	149
7.1.90 NET_ITS_OVERLAPCFG_COND .....	151
7.1.91 NET_ITS_PLATE_RESULT .....	151
7.1.92 NET_ITS_PICTURE_INFO .....	155
7.1.93 NET_POS_PARAM .....	158
7.1.94 NET_VCA_LINE .....	158
7.1.95 NET_VCA_POINT .....	159
7.1.96 NET_VCA_RECT .....	159

7.1.97 VCA_PLATE_COLOR .....	160
7.1.98 VCA_PLATE_TYPE .....	161
7.1.99 DATE_TIME .....	163
7.1.100 NET_DVR_DEVICEINFO_V30 .....	164
7.1.101 NET_DVR_DEVICEINFO_V40 .....	168
7.1.102 NET_DVR_INIT_CFG_ABILITY .....	170
7.1.103 NET_DVR_LOCAL_SDK_PATH .....	171
7.1.104 NET_DVR_USER_LOGIN_INFO .....	172
7.1.105 NET_SDK_CALLBACK_STATUS_NORMAL .....	173
7.1.106 NET_VCA_RECT .....	173
7.1.107 NET_ALARM_CVR_SUBINFO_UNION .....	174
7.1.108 NET_ALARM_RECORD_EXCEPTION .....	174
7.1.109 NET_ALARM_RECORDFILE_LOSS .....	175
7.1.110 NET_ALARM_RESOURCE_USAGE .....	176
7.1.111 NET_ALARM_STREAM_EXCEPTION .....	176
7.1.112 NET_DVR_ALARMER .....	176
7.1.113 NET_DVR_ALARMINFO_DEV .....	177
7.1.114 NET_DVR_ALARMINFO_DEV_V40 .....	178
7.1.115 NET_DVR_ALARMINFO_V30 .....	179
7.1.116 NET_DVR_ALARMINFO_V40 .....	180
7.1.117 NET_DVR_ALARM_FIXED_HEADER .....	181
7.1.118 NET_DVR_ALARM_ISAPI_INFO .....	185
7.1.119 NET_DVR_ALARM_ISAPI_PICDATA .....	186
7.1.120 NET_DVR_ETHERNET_V30 .....	186
7.1.121 NET_DVR_IPADDR_UNION .....	187
7.1.122 NET_DVR_NETCFG_V50 .....	187
7.1.123 NET_DVR_PPPOECFG .....	188
7.1.124 NET_DVR_SETUPALARM_PARAM_V50 .....	189



7.1.125 NET_DVR_TIME .....	192
7.1.126 NET_DVR_TIME_EX .....	192
7.2 Enumeration .....	193
7.2.1 COUNTRY_INDEX .....	193
7.2.2 CR_INDEX .....	213
7.2.3 NET_SDK_DOWNLOAD_TYPE .....	233
7.2.4 NET_SDK_UPLOAD_TYPE .....	237
7.2.5 VLR_VEHICLE_CLASS .....	240
7.2.6 VTR_RESULT .....	245
<b>Appendix A. Request URIs .....</b>	<b>248</b>
A.1 /ISAPI/ITC/capability .....	250
A.2 /ISAPI/Traffic/channels/<ID>/capabilities .....	250
A.3 /ISAPI/Traffic/channels/<ID>/licensePlate/filtration?format=json .....	251
A.4 /ISAPI/Traffic/channels/<ID>/searchLPListAudit .....	252
<b>Appendix B. Request and Response Messages .....</b>	<b>253</b>
B.1 JSON_Filtration .....	253
B.2 JSON_ResponseStatus .....	253
B.3 XML_Desc_ITDeviceAbility .....	253
B.4 XML_EventNotificationAlert_ANPR .....	254
B.5 XML_EventTriggerCapType .....	268
B.6 XML_EventTriggersCap .....	269
B.7 XML_ITCCap .....	273
B.8 XML_ITDeviceAbility .....	275
B.9 XML_LPListAuditSearchDescription .....	289
B.10 XML_LPListAuditSearchResult .....	290
B.11 XML_ResponseStatus .....	290
B.12 XML_SubscribeEvent .....	291
B.13 XML_TrafficChannelCap .....	292

**Appendix C. Appendixes ..... 297**

    C.1 Device Network SDK Errors ..... 297

    C.2 Response Codes of Text Protocol ..... 340

    C.3 Error Codes Categorized by Functional Modules ..... 379

    C.4 Region Code ..... 396

    C.5 Country/Region Code ..... 396

# Chapter 1 Overview

This manual provides the integration methods and processes based on HCNetsDK for ANPR (Automatic Number Plate Recognition) applications.

## 1.1 Introduction

The ANPR (Automatic Number Plate Recognition) applications integrated by private protocol help to analyze and recognize the vehicle license plate, and support importing the blocklist or allowlist to trigger license plate recognition alarms. In addition, for some entrance and exit devices, you can also control the barrier gate remotely by calling APIs.

## 1.2 Update History

### Summary of Changes in Version 6.1.0.25\_Aug., 2019

1. Extended intelligent traffic capability message **XML\_ITCCap** (related URL: **/ISAPI/ITC/capability** ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a node **<isSupportVehicleDetection>** (whether to support vehicle detection).
2. Extended structure about the configuration parameters of ANPR arming schedule **NET\_DVR\_GUARD\_CFG** (related API: **NET\_DVR\_SetDeviceConfig** with "NET\_DVR\_SET\_GUARDCFG" (command No.: 3135)):  
added a member **byDirection** (triggered direction) by one byte.

### Summary of Changes in Version 6.0.2.30\_03/2019

1. Extended blocklist and allowlist ANPR alarm structure **NET\_DVR\_VEHICLE\_CONTROL\_ALARM** via one reserved byte:  
added one parameter **byPicTransType** (picture transmission method).
2. Edited some description mistakes in this document, and added some missing APIs or data structures.

### Summary of Changes in Version 5.2.5.5\_08/2018

New document.

## Chapter 2 Configure ANPR Alarm

If the vehicle appears in the monitoring image during a certain time period, and the recognition parameters are configured, the ANPR camera will capture the vehicle picture automatically. Then the camera analyzes the license plate and the ANPR alarm will be triggered.

### Before You Start

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

### Steps

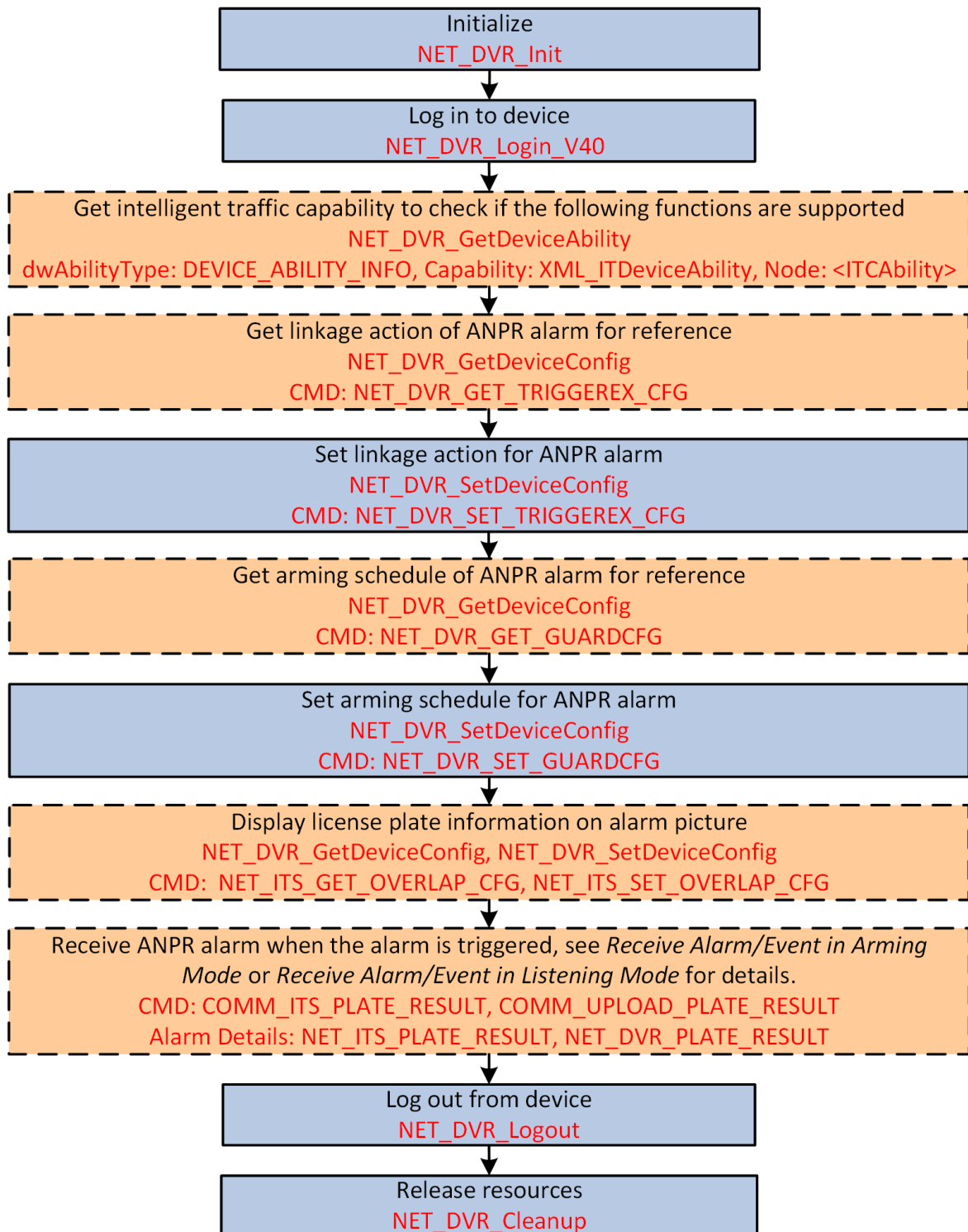


Figure 2-1 Programming Flow of Configuring ANPR Alarm

1. **Optional:** Call **NET\_DVR\_GetDeviceAbility** , set the capability type (**dwAbilityType**) to "DEVICE\_ABILITY\_INFO" (0x011), and set the input parameter pointer (**pInbuf**) to **XML\_Desc\_ITDeviceAbility** message for getting intelligent traffic capability to check if the following functions are supported.

You can also call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/ITC/capability** to get the intelligent traffic capability and check whether the ANPR function is supported.

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

The intelligent traffic capability is returned in the message of **XML\_ITDeviceAbility** , and the related node is <ITCAbility>.

2. **Optional:** Call **NET\_DVR\_GetDeviceConfig** with "NET\_DVR\_GET\_TRIGGEREX\_CFG" (command No.: 5074) and set the input buffer (**IpInBuffer**) to the structure **NET\_DVR\_TRIGGER\_COND** for getting the configured or existing triggering mode of ANPR alarm for reference.  
The triggering mode parameters are returned by the output buffer (**IpOutBuffer**) in the structure of **NET\_ITC\_TRIGGERCFG** .
3. Call **NET\_DVR\_SetDeviceConfig** with "NET\_DVR\_SET\_TRIGGEREX\_CFG" (command No.: 5075), set the input buffer (**IpInBuffer**) to the structure **NET\_DVR\_TRIGGER\_COND** , and set the input parameter (**IpInParamBuffer**) to the structure **NET\_ITC\_TRIGGERCFG** for setting the triggering mode.
4. **Optional:** Call **NET\_DVR\_GetDeviceConfig** with "NET\_DVR\_GET\_GUARDCFG" (command No.: 3134) and set the input buffer (**IpInBuffer**) to the structure **NET\_DVR\_GUARD\_COND** for getting the configured or existing arming schedule of ANPR alarm for reference.  
The arming schedule parameters are returned by the output buffer (**IpOutBuffer**) in the structure of **NET\_DVR\_GUARD\_CFG** .
5. Call **NET\_DVR\_SetDeviceConfig** with "NET\_DVR\_SET\_GUARDCFG" (command No.: 3135), set the input buffer (**IpInBuffer**) to the structure **NET\_DVR\_GUARD\_COND** , and set the input parameter (**IpInParamBuffer**) to the structure **NET\_DVR\_GUARD\_CFG** for setting arming schedule.



### Note

You can also configure the triggering mode and arming schedule for ANPR alarm by logging in to device via web browser.

- 
6. **Optional:** Configure parameters to display license plate information on alarm picture.
    - 1) Optional: Call **NET\_DVR\_GetDeviceConfig** with "NET\_ITS\_GET\_OVERLAP\_CFG\_V50" (command No.: 5055) and set the input buffer (**IpInBuffer**) to the structure **NET\_ITS\_OVERLAPCFG\_COND** for getting the configured or existing overlay parameters for reference.  
The overlay parameters are returned by the output buffer (**IpOutBuffer**) in the structure of **NET\_ITS\_OVERLAP\_CFG\_V50** .
    - 2) Call **NET\_DVR\_SetDeviceConfig** with "NET\_ITS\_SET\_OVERLAP\_CFG\_V50" (command No.: 5056), set the input buffer (**IpInBuffer**) to the structure **NET\_ITS\_OVERLAPCFG\_COND** , and set the input parameter (**IpInParamBuffer**) to the structure **NET\_ITS\_OVERLAP\_CFG\_V50** for setting the parameters to display license plate information on alarm picture.

7. **Optional:** Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Traffic/channels/<ID>/licensePlate/filtration?format=json** by PUT method and set **IpInputParam** to **JSON\_Filtration** to filter the duplicated license plates and receive the same alarm just for once.



### Note

To check whether the device supports filtering duplicated license plates, you can call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Traffic/channels/<ID>/capabilities** by GET method. The capability will be returned in the message **JSON\_Filtration** by **IpOutputParam**. If it supports, the node **<isSupportFiltration>** will be in the capability message and its value is "true".

8. **Optional:** Receive ANPR alarm in arming mode (see **Receive Alarm/Event in Arming Mode** ) or listening mode (see **Receive Alarm/Event in Listening Mode** ) when alarm is triggered.



### Note

The command (**ICommand**) to receive ANPR alarms should be set to "COMM\_ITS\_PLATE\_RESULT" (command No.: 0x3050) or "COMM\_UPLOAD\_PLATE\_RESULT" (command No.: 0x2800) in the alarm callback function **MSGCallBack** .

For alarm details, refer to **XML\_EventNotificationAlert\_ANPR** returned in the field **pXmlBuf** of **NET\_DVR\_PLATE\_INFO** in the structure **NET\_ITS\_PLATE\_RESULT** or **NET\_DVR\_PLATE\_RESULT** .

### What to do next

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

## Chapter 3 Configure Blocklist and Allowlist ANPR Alarm

After capturing the vehicle picture, you can control the entry of vehicles according to the ANPR results after configuring the alarm of license plate in blocklist or allowlist. The vehicles in blocklist are not allowed to enter, while the vehicles in the allowlist are allowed to enter.

### Before You Start

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



## Steps

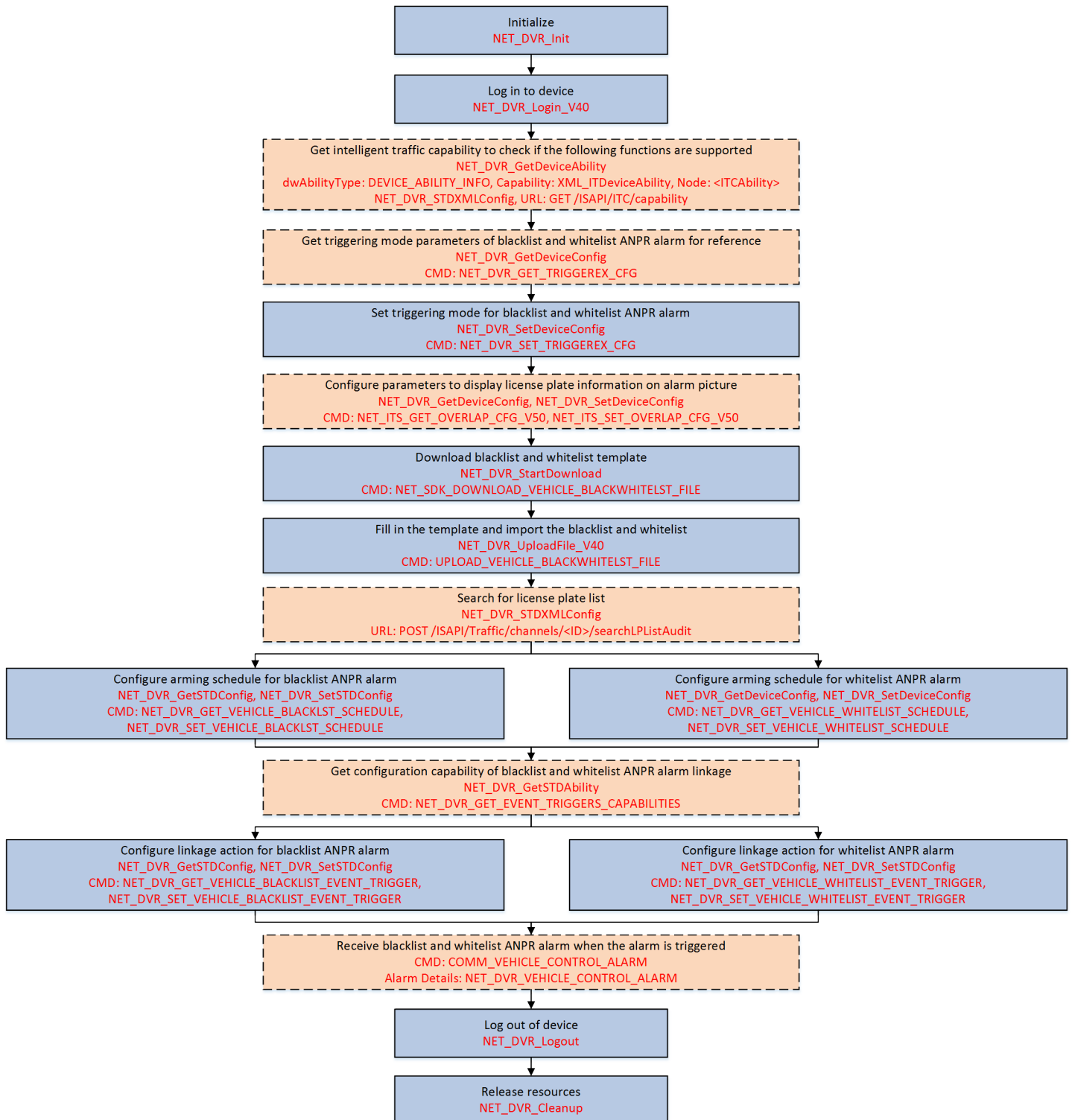


Figure 3-1 Programming Flow of Configuring Blocklist and Allowlist ANPR Alarm

1. **Optional:** Get intelligent traffic capability to check if the following functions are supported.
  - Call **NET\_DVR\_GetDeviceAbility** , set the capability type (**dwAbilityType**) to "DEVICE\_ABILITY\_INFO" (0x011), and set the input parameter pointer (**plnbuf**) to the message **XML\_Desc\_ITDeviceAbility** .

The intelligent traffic capability is returned in the message of **XML\_ITDeviceAbility** , and the related node is <ITCAbility>.
  - Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/ITC/capability** .

The intelligent traffic capability is returned in the message **XML\_ITCCap** by **lpOutBuffer** of **lpOutputParam**.
2. **Optional:** Call **NET\_DVR\_GetDeviceConfig** with "NET\_DVR\_GET\_TRIGGEREX\_CFG" (command No.: 5074) and set the input buffer (**lpInBuffer**) to the structure **NET\_DVR\_TRIGGER\_COND** for getting the configured or existing triggering mode of blacklist and allowlist ANPR alarm for reference.

The triggering mode parameters are returned by the output buffer (**lpOutBuffer**) in the structure of **NET\_ITC\_TRIGGERCFG** .
3. Call **NET\_DVR\_SetDeviceConfig** with "NET\_DVR\_SET\_TRIGGEREX\_CFG" (command No.: 5075), set the input buffer (**lpInBuffer**) to the structure **NET\_DVR\_TRIGGER\_COND** , and set the input parameter (**lpInParamBuffer**) to the structure **NET\_ITC\_TRIGGERCFG** for setting the triggering mode.
4. **Optional:** Configure parameters to display license plate information on alarm picture.
  - 1) Optional: Call **NET\_DVR\_GetDeviceConfig** with "NET\_ITS\_GET\_OVERLAP\_CFG\_V50" (command No.: 5055) and set the input buffer (**lpInBuffer**) to the structure **NET\_ITS\_OVERLAPCFG\_COND** for getting the configured or existing overlay parameters for reference.

The overlay parameters are returned by the output buffer (**lpOutBuffer**) in the structure of **NET\_ITS\_OVERLAP\_CFG\_V50** .
  - 2) Call **NET\_DVR\_SetDeviceConfig** with "NET\_ITS\_SET\_OVERLAP\_CFG\_V50" (command No.: 5056), set the input buffer (**lpInBuffer**) to the structure **NET\_ITS\_OVERLAPCFG\_COND** , and set the input parameter (**lpInParamBuffer**) to the structure **NET\_ITS\_OVERLAP\_CFG\_V50** for setting the parameters to display license plate information on alarm picture.
5. Call **NET\_DVR\_StartDownload** with "NET\_SDK\_DOWNLOAD\_VEHICLE\_BLOCKALLOWLIST\_FILE" (command No.: 7) to download the blacklist and allowlist template.
6. Call **NET\_DVR\_UploadFile\_V40** with "UPLOAD\_VEHICLE\_BLOCKALLOWLIST\_FILE" (command No.: 13) to import the blacklist and allowlist information filled in the template.
7. **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: POST **/ISAPI/Traffic/channels/<ID>/searchLPListAudit** and set **lpInBuffer** of **lpInputParam** to the message **XML\_LPListAuditSearchDescription** to search for the configured or existing blacklist or allowlist.
8. Configure arming schedule for blacklist ANPR alarm or allowlist ANPR alarm.
  - Configure arming schedule for blacklist ANPR alarm
    - a. Call **NET\_DVR\_GetSTDConfig** with "NET\_DVR\_GET\_VEHICLE\_BLOCKLIST\_SCHEDULE" (command No: 6622) to get the configured or existing arming schedule of blacklist ANPR alarm for reference.

---

### Note

The arming schedule parameters ( **NET\_DVR\_EVENT\_SCHEDULE** ) are returned by the output buffer (**lpOutBuffer**) of structure **NET\_DVR\_STD\_CONFIG** .

- b. Call **NET\_DVR\_SetSTDConfig** with "NET\_DVR\_SET\_VEHICLE\_BLOCKLIST\_SCHEDULE" (command No.: 6623) and set the input buffer (**lpInBuffer**) of structure **NET\_DVR\_STD\_CONFIG** to **NET\_DVR\_EVENT\_SCHEDULE** for setting arming schedule.
- Configure arming schedule for allowlist ANPR alarm
  - a. Call **NET\_DVR\_GetSTDConfig** with "NET\_DVR\_GET\_VEHICLE\_ALLOWLIST\_SCHEDULE" (command No: 6624) to get the configured or existing arming schedule of allowlist ANPR alarm for reference.

---

### Note

The arming schedule parameters ( **NET\_DVR\_EVENT\_SCHEDULE** ) are returned by the output buffer (**lpOutBuffer**) of structure **NET\_DVR\_STD\_CONFIG** .

- b. Call **NET\_DVR\_SetSTDConfig** with "NET\_DVR\_SET\_VEHICLE\_ALLOWLIST\_SCHEDULE" (command No.: 6625) and set the input buffer (**lpInBuffer**) of structure **NET\_DVR\_STD\_CONFIG** to **NET\_DVR\_EVENT\_SCHEDULE** for setting arming schedule.
9. Optional: Call **NET\_DVR\_GetSTDAbility** , set the **dwAbilityType** to "NET\_DVR\_GET\_EVENT\_TRIGGERS\_CAPABILITIES" (value: 3501), and set condition parameter **lpCondBuffer** in the structure of **NET\_DVR\_STD\_ABILITY** to "NULL" for getting the configuration capability of blocklist and allowlist ANPR alarm linkage.

The configuration capability is returned in the message **XML\_EventTriggersCap** by the output parameter **lpOutBuffer** in the structure of **NET\_DVR\_STD\_ABILITY** .

10. Configure linkage action for blocklist ANPR alarm or allowlist ANPR alarm.
- Configure linkage action for blocklist ANPR alarm
    - a. Call **NET\_DVR\_GetSTDConfig** with "NET\_DVR\_GET\_VEHICLE\_BLOCKLIST\_EVENT\_TRIGGER" (command No.: 6626) to get the configured or existing linkage action of blocklist ANPR alarm for reference.

---

### Note

The linkage action parameters ( **NET\_DVR\_EVENT\_TRIGGER** ) are returned by the output buffer (**lpOutBuffer**) of structure **NET\_DVR\_STD\_CONFIG** .

- b. Call **NET\_DVR\_SetSTDConfig** with "NET\_DVR\_SET\_VEHICLE\_BLOCKLIST\_EVENT\_TRIGGER" (command No.: 6627) and set the input buffer (**lpInBuffer**) of structure **NET\_DVR\_STD\_CONFIG** to **NET\_DVR\_EVENT\_TRIGGER** for setting linkage action.
- Configure linkage action for allowlist ANPR alarm
  - a. Call **NET\_DVR\_GetSTDConfig** with "NET\_DVR\_GET\_VEHICLE\_ALLOWLIST\_EVENT\_TRIGGER" (command No.: 6628) to get the configured or existing linkage action of allowlist ANPR alarm for reference.

---

### Note

The linkage action parameters ( [NET\\_DVR\\_EVENT\\_TRIGGER](#) ) are returned by the output buffer (**lpOutBuffer**) of structure [NET\\_DVR\\_STD\\_CONFIG](#) .

- b. Call [NET\\_DVR\\_SetSTDConfig](#) with "NET\_DVR\_SET\_VEHICLE\_ALLOWLIST\_EVENT\_TRIGGER" (command No.: 6629) and set the input buffer (**lpInBuffer**) of structure [NET\\_DVR\\_STD\\_CONFIG](#) to [NET\\_DVR\\_EVENT\\_TRIGGER](#) for setting linkage action.

**11. Optional:** Receive blocklist and allowlist ANPR alarm in arming mode (see [Receive Alarm/Event in Arming Mode](#) ) or listening mode (see [Receive Alarm/Event in Listening Mode](#) ) when alarm is triggered.

---

### Note

- The commands (**lCommand**) to receive blocklist and allowlist ANPR alarms should be set to "COMM\_VEHICLE\_CONTROL\_ALARM" (command No.: 0x3059) in [NET\\_DVR\\_SetDVRMessageCallBack\\_V50](#) and [NET\\_DVR\\_StartListen\\_V30](#) .
- For alarm details, refer to the structure of [NET\\_DVR\\_VEHICLE\\_CONTROL\\_ALARM](#) .

---

### What to do next

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

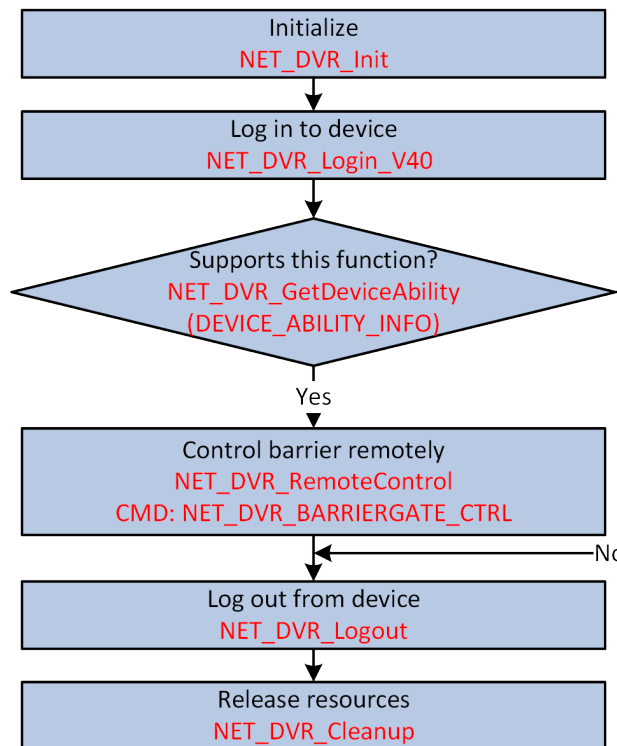
## Chapter 4 Control Barrier Gate Status

In the entrance and exit scene, you can remotely control the barrier gate status via device network SDK, such as falling the gate, rising the gate, stopping the gate at a certain position, and locking the gate.

### Before You Start

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

### Steps



**Figure 4-1 API Calling Flow of Controlling Barrier Gate Status**

1. Call **NET\_DVR\_GetDeviceAbility**, set the capability type (**dwAbilityType**) to "DEVICE\_ABILITY\_INFO" (0x011), and set **pInbuf** to **XML\_Desc\_ITDeviceAbility** for getting intelligent traffic capability to check whether the device supports this function.  
The intelligent traffic capability is returned in the message **XML\_ITDeviceAbility** by **pOutBuf**.  
If supports, the node <ITCAbility> is returned in the message, and then you can perform the following steps.  
Otherwise, this function is not supported, please end this task.
2. Call **NET\_DVR\_RemoteControl** with "NET\_DVR\_BARRIERGATE\_CTRL" (command No.: 3128) and set **lpInBuffer** to **NET\_DVR\_BARRIERGATE\_CFG** for remotely controlling the barrier.

### What to do next

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

## Chapter 5 Alarm/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.

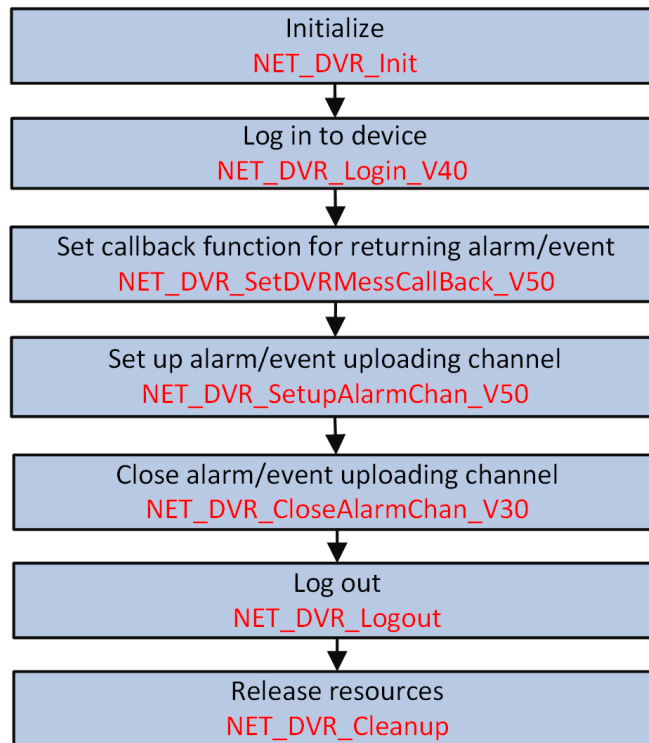
### 5.1 Receive Alarm/Event in Arming Mode

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 5-1 Programming Flow of Receiving Alarm/Event in Arming Mode**

1. Call **NET\_DVR\_SetDVRMessageCallBack\_V50** to set callback function for returning alarm/event information.



#### Note

- If the configured alarm is triggered or event occurred, the alarm/event information will be uploaded by device and returned in the callback function. You can view the alarm/event and do some processing operations.
- For the integration via device network SDK (HCNetSDK), to receive different types of alarm/event information, the parameter **lCommand** (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 **lCommand** 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
    lHandle = NET_DVR_SetupAlarmChan_V50(lUserID, &struSetupParamV50, NULL,
strlen(szSubscribe));
}
else
{
    //Subscribe
    LlHandle = NET_DVR_SetupAlarmChan_V50(lUserID, &struSetupParamV50,
szSubscribe, strlen(szSubscribe));
}

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

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

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

### What to do next

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

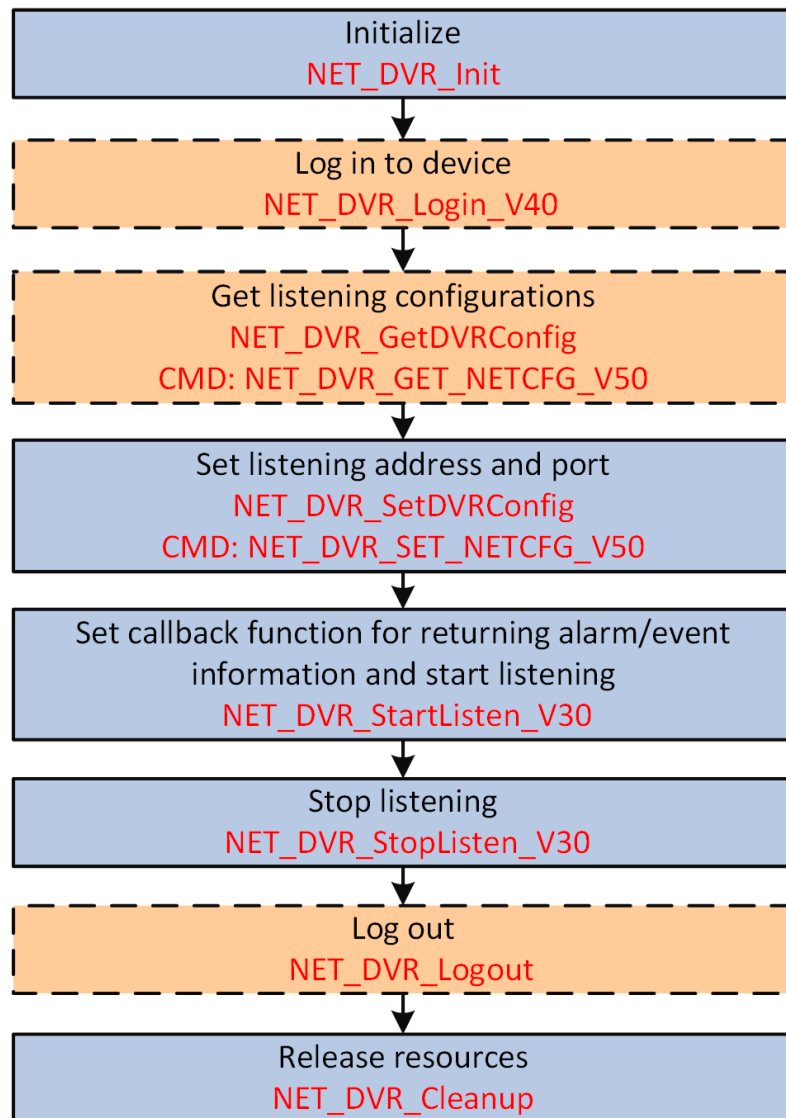
## 5.2 Receive Alarm/Event in Listening Mode

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 5-2 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 **lCommand** (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 **lCommand** 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.

## Chapter 6 API Reference

### 6.1 NET\_DVR\_GetDeviceAbility

Get the device capabilities.

#### API Definition

```
BOOL NET_DVR_GetDeviceAbility(  
    LONG        lUserID,  
    DWORD        dwAbilityType,  
    char         *pInBuf,  
    DWORD        dwInLength,  
    char         *pOutBuf,  
    DWORD        dwOutLength  
);
```

#### Parameters

##### lUserID

[IN] Value returned by NET\_DVR\_Login\_V40 .

##### dwAbilityType

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

##### pInBuf

[IN] Input parameter buffer pointer, which are different according to different devices and functions, and they are returned in the structure or messages.

##### dwInLength

[IN] Size of input buffer.

##### pOutBuf

[OUT] Output parameter buffer pointer, which are different according to different devices and functions, and they are returned in the structure or messages.

##### dwOutLength

[OUT] Size of buffer for receiving data.

#### Return Values

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

If *FALSE* is returned, you can call NET\_DVR\_GetLastError to get the error code.

## 6.2 NET\_DVR\_GetDeviceConfig

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

### API Definition

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

### Parameters

#### lUserID

[IN] Value returned by NET\_DVR\_Login\_V40 .

#### dwCommand

[IN] Device getting commands. The commands are different for different getting functions.

#### dwCount

[IN] Number of configurations (cameras) to get at a time. 0, 1-one camera, 2-two cameras, 3-three cameras, and so on. Up to 64 cameras' configuration information can be obtained at a time.

#### lpInBuffer

[IN] Pointer of configuration condition buffer, which specifies the number (**dwCount**) of configurations to get, and relates to the getting commands.

#### dwInBufferSize

[IN] Size of configuration condition buffer, which saves the obtained configuration information (the number is **dwCount**).

#### lpStatusList

[OUT] Error information list, and its memory is allocated by user, each error information contains 4 bytes (a unsigned 32-bit integer).

There is a one-to-one correspondence between the errors in the list and the cameras need to search, e.g., **lpStatusList[2]** corresponds to **lpInBuffer[2]**.

If the parameter value is 0 or 1, it refers to getting succeeded, otherwise, this parameter value is the error code.

#### lpOutBuffer

[OUT] Parameters returned by device, which relates to the getting commands. And there is a one-to-one correspondence between the parameters and the cameras need to search.

If the **lpStatusList** of one camera is larger than 1, the corresponding **lpOutBuffer** is invalid.

### **dwOutBufferSize**

[IN] Total size of returned results (the number is **dwCount**).

### **Return Values**

Returns *TRUE* for success, and returns *FALSE* for failure. If returns *TRUE*, it does not mean that all configurations are obtained, you can check the value of **lpStatusList[n]** to judge which one is succeeded.

If *FALSE* is returned, you can call **NET\_DVR\_GetLastError** to get the error code.

### **See Also**

**NET\_DVR\_SetDeviceConfig**

## **6.3 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.

## **6.4 NET\_DVR\_GetSTDAbility**

Get the device capabilities.



### API Definition

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

#### Parameters

##### lUserID

[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.

## 6.5 NET\_DVR\_GetSTDConfig

Get the device configuration information.

### API Definition

```
BOOL NET_DVR_GetSTDConfig(  
    LONG                lUserID,  
    DWORD               dwCommand,  
    NET_DVR_STD_CONFIG  lpConfigParam  
);
```

#### Parameters

##### lUserID

[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 **IpCondBuffer** and **IpOutBuffer** in the **IpConfigParam** are also different. See the structure **NET\_DVR\_STD\_CONFIG** for details.



## Note

When getting configuration parameters, the **IpInBuffer** 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\_SetSTDConfig**

## 6.6 NET\_DVR\_GetUploadState

Get the file uploading progress and status.

## API Definition

```
LONG NET_DVR_GetUploadState (
    LONG          lUploadHandle,
    DWORD         *pProgress
);
```

## Parameters

### lUploadHandle

[IN] Handling for uploading files, which is returned by **NET\_DVR\_UploadFile\_V40**.

### pProgress

[OUT] Returned progress value.

## Return Values

Return -1 for failure, and return other values as the uploading status codes, see details in the following table.

**Table 6-1 Uploading Status Code**

Return Value	Description
1	Uploaded successfully.
2	Uploading.

Return Value	Description
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.
37	Person ID error (The ID is saved in <b>customHumanID</b> of <b>FaceAppendData</b> ).
38	Modeling failed. Device inner error.

Return Value	Description
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.

## 6.7 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.

#### lpInBuffer

[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.

## 6.8 NET\_DVR\_SetDeviceConfig

Set device parameters in batch (sending data is supported).

### API Definition

```
BOOL NET_DVR_SetDeviceConfig(  
    LONG        lUserID,  
    DWORD       dwCommand,  
    DWORD       dwCount,  
    LPVOID      lpInBuffer,  
    DWORD       dwInBufferSize,  
    LPVOID      lpStatusList,  
    LPVOID      lpInParamBuffer,  
    DWORD       dwInParamBufferSize  
);
```

### Parameters

#### lUserID

[IN] Value returned by NET\_DVR\_Login\_V40 .

#### dwCommand

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

#### dwCount

[IN] Number of cameras to be set at a time. 0,1-one camera, 2-two cameras, 3-three cameras, and so on. Up to 256 cameras can be configured at a time.

#### lpInBuffer

[IN] Pointer of configuration condition buffer, e.g., stream ID, which specifies the number (**dwCount**) of cameras to set, and relates to the configuration commands.

#### dwInBufferSize

[IN] Size of configuration condition buffer, which saves the configured information of cameras with the number of **dwCount**.

#### lpStatusList

[OUT] Error information list, and its memory is allocated by user, each error information contains 4 bytes (a unsigned 32-bit integer).

There is a one-to-one correspondence between the errors in the list and the cameras that need to be searched, e.g., **lpStatusList[2]** corresponds to **lpInBuffer[2]**.

If the parameter value is 0, it refers to setting succeeded, otherwise, this parameter value is the error code.

#### lpInParamBuffer

[IN] Device parameters to set, which relates to the configuration commands. And there is a one-to-one correspondence between the parameters and the cameras that need to be searched.

### **dwInParamBufferSize**

[IN] Set the size of content buffer.

### **Return Values**

Returns *TRUE* for success, and returns *FALSE* for all failed. If returns *TRUE*, it does not indicate that all settings are succeeded, you can get the value of **lpStatusList[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**

## **6.9 NET\_DVR\_SetSTDConfig**

Set the device parameters.

### **API Definition**

```
BOOL NET_DVR_SetSTDConfig(  
    LONG                lUserID,  
    DWORD               dwCommand,  
    NET_DVR_STD_CONFIG  lpConfigParam  
);
```

### **Parameters**

#### **lUserID**

[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 **lpInBuffer** in the **lpConfigParam** are also different. See the structure **NET\_DVR\_STD\_CONFIG** for details.



#### **Note**

When getting configuration parameters, the **lpOutBuffer** 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\_GetSTDConfig**

## 6.10 NET\_DVR\_STDXMLConfig

Transmit request URL with XML or JSON format to implement some typical functions.

### API Definition

```
BOOL NET_DVR_STDXMLConfig(  
    LONG                                lUserID,  
    const NET_DVR_XML_CONFIG_INPUT      *lpInputParam,  
    NET_DVR_XML_CONFIG_OUTPUT           *lpOutputParam  
);
```

### Parameters

#### lUserID

[IN] Value returned by **NET\_DVR\_Login\_V40**.

#### lpInputParam

[IN] Input parameters, refer to the structure **NET\_DVR\_XML\_CONFIG\_INPUT** for details.

#### lpOutputParam

[IN][OUT] Output parameters, refer to the structure **NET\_DVR\_XML\_CONFIG\_OUTPUT** for details.

### Return Values

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

If *FALSE* is returned, you can call **NET\_DVR\_GetLastError** to get the error code.

### Remarks

The input parameter **lpInputParam** and output parameter **lpOutputParam** are different when transmitting text protocol for implementing different functions, and each parameter corresponds to a component of text protocol, see the relations below:

Parameter of NET_DVR_STDXMLConfig		Component of Text Protocol
IpInputParam	IpRequestUrl (see in structure <i><b>NET_DVR_XML_CONFIG_INPUT</b></i> )	Method+URL E.g., GET /ISAPI/System/capabilities
	IpInBuffer (see in structure <i><b>NET_DVR_XML_CONFIG_INPUT</b></i> )	Request Message
IpOutputParam	IpOutBuffer (see in structure <i><b>NET_DVR_XML_CONFIG_OUTPUT</b></i> )	Response Message
	IpStatusBuffer (see in structure <i><b>NET_DVR_XML_CONFIG_OUTPUT</b></i> )	Response Message

## 6.11 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.

## 6.12 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.

## 6.13 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.

## 6.14 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.

## 6.15 NET\_DVR\_Cleanup

Release the resources after the program is ended.

### API Definition

```
BOOL NET_DVR_Cleanup(  
);
```

### 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.

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.

## 6.16 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.

## 6.17 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.

## 6.18 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**

## 6.19 NET\_DVR\_Login\_V40

Log in to the device (supports asynchronous login).

### API Definition

```
LONG NET_DVR_Login_V40 (
    NET_DVR_USER_LOGIN_INFO  pLoginInfo,
    NET_DVR_DEVICEINFO_V40    lpDeviceInfo
);
```

### Parameters

#### pLoginInfo

[IN] Login parameters, including device address, user name, password, and so on. See details in the structure **NET\_DVR\_USER\_LOGIN\_INFO**.

#### lpDeviceInfo

[OUT] Device information. See details in the structure **NET\_DVR\_DEVICEINFO\_V40**.

### Return Values

- For asynchronous login, the callback function ( **fLoginResultCallBack** ) configured in the structure ( **NET\_DVR\_USER\_LOGIN\_INFO** ) returns the asynchronous login status, user ID and device information.
- For synchronous login, this API returns -1 for logging failed, and returns other values for the returned user IDs. The user ID is unique, and it helps to realize the further device operations.
- If -1 is returned, you can call **NET\_DVR\_GetLastError** to get the error code.

### Remarks

- When **bUseAsynLogin** in **pLoginInfo** is 0, it indicates that login is in synchronous mode; when **bUseAsynLogin** in **pLoginInfo** is 1, it indicates that login is in asynchronous mode.
- Up to 2048 users are allowed to log in to HCNetSDK at same time, and the values of returned **UserID** are ranging from 0 to 2047.

### See Also

**NET\_DVR\_Logout**

#### 6.19.1 fLoginResultCallBack

## Login Status Callback Function

Member	Data Type	Description
lUserID	LONG	User ID, which is returned by <b><u>NET_DVR_Login_V40</u></b> .
dwResult	DWORD	Login status: 0-asynchronously logging in failed, 1-asynchronously logged in.
lpDeviceInfo	<b><u>NET_DVR_DEVICEINFO_V40</u></b>	Device information, such as serial No., channel, capability, and so on.
pUser	void*	User data.

## 6.20 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**.

## 6.21 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 6-2 NET\_SDK\_INIT\_CFG\_TYPE**

enumType	Value	Description	lpInBuff
NET_SDK_INIT_CFG_ABILITY	1	Capability supported by SDK.	<b><u>NET_DVR_INIT_CFG_ABILITY</u></b>
NET_SDK_INIT_CFG_SDK_PATH	2	Set loading path for component libraries (supported by both Linux and Windows system).	<b><u>NET_DVR_LOCAL_SDK_PATH</u></b>
NET_SDK_INIT_CFG_LIBEAY_PATH	3	Set path (including library name) for libeay32.dll (Windows), libcrypto.so (Linux), and libcrypto.dylib (Mac) of OpenSSL in version 1.1.1 and 1.0.2.	Path in string format, e.g., <b><u>C:\libeay32.dll</u></b> .
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., <b><u>C:\ssleay32.dll</u></b> .

### lpInBuff

[IN] Input parameter. Different parameter types correspond to different structures, see details in the table above.

## Return Values

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

If *FALSE* is returned, you can call **NET\_DVR\_GetLastError** to get the error code.

## Remarks

This API should be called before calling **NET\_DVR\_Init** to initialize and check the dependent libraries or capabilities. This API only takes effect for POSIX. For Windows, it takes no effect but success will be returned.

## 6.22 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**.

## 6.23 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**

## 6.24 NET\_DVR\_SetDVRConfig

Set the device parameters.

### API Definition

```
BOOL NET_DVR_SetDVRConfig(  
    LONG        lUserID,  
    DWORD       dwCommand,  
    LONG        lChannel,  
    LPVOID      lpInBuffer,
```

```
DWORD    dwInBufferSize
);
```

### Parameters

#### IUserID

[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.

#### lpInBuffer

[IN] Pointer of input data buffer. For different configuration functions, the structures of this parameter are different.

#### dwInBufferSize

[IN] Size of input data buffer (unit: byte).

### Return Values

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

If *FALSE* is returned, you can call **NET\_DVR\_GetLastError** to get the error code.

The following error codes may be returned by this API: 0, 3, 6, 7, 8, 9, 10, 12, 17, 41, 43, 44, 47, 72, 73, and 76. See the corresponding error types and descriptions in the **Device Network SDK Errors**.

### See Also

**NET\_DVR\_GetDVRConfig**

## 6.25 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(lCommand)
    {
        case COMM_ALARM:
        {
            NET_DVR_ALARMINFO struAlarmInfo;
            memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_DVR_ALARMINFO));
            switch (struAlarmInfo.dwAlarmType)
```

```
{
    case 3: //Motion detection alarm
        for (i=0; i<16; i++)    //define MAX_CHANNUM    16    //The
maximum number of channels
        {
            if (struAlarmInfo.dwChannel[i] == 1)
            {
                printf("Channel Number with Motion Detection Alarm
%d\n", i+1);
            }
        }
        break;
    default:
        break;
}
break;
}
case COMM_UPLOAD_PLATE_RESULT:
{
    NET_DVR_PLATE_RESULT struPlateResult={0};
    memcpy(&struPlateResult, pAlarmInfo, sizeof(struPlateResult));
    printf("License Plate Number: %s\n",
struPlateResult.struPlateInfo.sLicense); //License plate number

    switch(struPlateResult.struPlateInfo.byColor) //License plate color
    {
    case VCA_BLUE_PLATE:
        printf("Vehicle Color: Blue\n");
        break;
    case VCA_YELLOW_PLATE:
        printf("Vehicle Color: Yellow\n");
        break;
    case VCA_WHITE_PLATE:
        printf("Vehicle Color: White\n");
        break;
    case VCA_BLACK_PLATE:
        printf("Vehicle Color: Black\n");
        break;
    default:
        break;
    }
    //Scene picture
    if (struPlateResult.dwPicLen != 0 && struPlateResult.byResultType
== 1 )
    {
        sprintf(filename, "testpic_%d.jpg", iNum);
        fSnapPic=fopen(filename, "wb");
        fwrite(struPlateResult.pBuffer1, struPlateResult.dwPicLen,
1, fSnapPic);
        iNum++;
        fclose(fSnapPic);
    }
}
```

```
//License plate picture
if (struPlateResult.dwPicPlateLen != 0 &&
struPlateResult.byResultType == 1)
{
    sprintf(filename,"testPicPlate_%d.jpg",iNum);
    fSnapPicPlate=fopen(filename,"wb");
    fwrite(struPlateResult.pBuffer1,struPlateResult.dwPicLen,
1,fSnapPicPlate);
    iNum++;
    fclose(fSnapPicPlate);
}
//Processing other data...
break;
}
case COMM_ITS_PLATE_RESULT:
{
    NET_ITS_PLATE_RESULT struITSPlateResult={0};
    memcpy(&struITSPlateResult, pAlarmInfo, sizeof(struITSPlateResult));

    for (i=0;i<struITSPlateResult.dwPicNum;i++)
    {
        printf("License Plate Number: %s\n",
struITSPlateResult.struPlateInfo.sLicense);//License plate number
        switch(struITSPlateResult.struPlateInfo.byColor)//License plate
color
        {
            case VCA_BLUE_PLATE:
                printf("Vehicle Color: Blue\n");
                break;
            case VCA_YELLOW_PLATE:
                printf("Vehicle Color: Yellow\n");
                break;
            case VCA_WHITE_PLATE:
                printf("Vehicle Color: White\n");
                break;
            case VCA_BLACK_PLATE:
                printf("Vehicle Color: Black\n");
                break;
            default:
                break;
        }
        //Save scene picture
        if ((struITSPlateResult.struPicInfo[i].dwDataLen !=
0)&&(struITSPlateResult.struPicInfo[i].byType== 1)||
(struITSPlateResult.struPicInfo[i].byType == 2))
        {
            sprintf(filename,"testITSpic%d_%d.jpg",iNum,i);
            fSnapPic=fopen(filename,"wb");
            fwrite(struITSPlateResult.struPicInfo[i].pBuffer,
struITSPlateResult.struPicInfo[i].dwDataLen,1,fSnapPic);
            iNum++;
            fclose(fSnapPic);
        }
    }
}
```

```
        }
        //License plate thumbnails
        if ((struITSPlateResult.struPicInfo[i].dwDataLen !=
0)&&(struITSPlateResult.struPicInfo[i].byType == 0))
        {
            sprintf(filename,"testPicPlate%d_%d.jpg",iNum,i);
            fSnapPicPlate=fopen(filename,"wb");
            fwrite(struITSPlateResult.struPicInfo[i].pBuffer,
struITSPlateResult.struPicInfo[i].dwDataLen, 1, \ fSnapPicPlate);
            iNum++;
            fclose(fSnapPicPlate);
        }
        //Processing other data...
    }
    break;
}
default:
    break;
}
}

void CALLBACK MessageCallbackNo2(LONG lCommand, NET_DVR_ALARMER *pAlarmer, char
*pAlarmInfo, DWORD dwBufLen, void* pUser)
{
    int i=0;
    char filename[100];
    FILE *fSnapPic=NULL;
    FILE *fSnapPicPlate=NULL;

    //This sample code is for reference only. Actually, it is not recommended
to process the data and save file in the callback function directly.
    //You'd better process the data in the message response function 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](#)**

### 6.25.1 MSGCallback

Alarm/event information callback function.

#### Callback Function Definition

```
typedef void(CALLBACK *MSGCallback) (
    LONG                lCommand,
    NET_DVR_ALARMER     *pAlarmer,
    char                *pAlarmInfo,
    DWORD               dwBufLen,
    void                *pUser
);
```

#### Parameters

##### **lCommand**

[OUT] Uploaded message type. You can distinguish the alarm/event information via the type.

##### **pAlarmer**

[OUT] Alarm device information, including serial No., IP address, login handle, and so on, see details in **[NET\\_DVR\\_ALARMER](#)**.

##### **pAlarmInfo**

[OUT] Alarm/event information, the details are returned in different structures according to **ICommand**.

### **dwBufLen**

[OUT] Size of alarm/event information buffer.

### **pUser**

[OUT] User data.

## 6.26 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**.

## 6.27 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**).

## 6.28 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](#) .

## Chapter 7 Structure and Enumeration

### 7.1 Data Structure

#### 7.1.1 ITC\_LANE\_DIRECTION\_TYPE

Enumerate the lane direction type.

##### Enumeration Definition

```
enum{
    ITC_LANE_DIRECTION_UNKNOWN    = 0,
    ITC_LANE_LEFT                 = 1,
    ITC_LANE_STRAIGHT             = 2,
    ITC_LANE_LEFT_STRAIGHT        = 3,
    ITC_LANE_RIGHT                = 4,
    ITC_LANE_LEFT_RIGHT           = 5,
    ITC_LANE_RIGHT_STRAIGHT       = 6,
    ITC_LANE_LEFT_RIGHT_STRAIGHT  = 7,
    ITC_LANE_LEFT_WAIT            = 9,
    ITC_LANE_STRAIGHT_WAIT        = 10,
    ITC_LANE_FORWARD              = 11,
    ITC_LANE_BACKWARD             = 12,
    ITC_LANE_BOTHWAY              = 13,
    ITC_LANE_STRAIGHT_WAIT_RIGHT  = 14
}ITC_LANE_DIRECTION_TYPE
```

##### Member

###### ITC\_LANE\_DIRECTION\_UNKNOWN

Unknown.

###### ITC\_LANE\_LEFT

Turn left.

###### ITC\_LANE\_STRAIGHT

Go straight.

###### ITC\_LANE\_LEFT\_STRAIGHT

Turn left and go straight.

###### ITC\_LANE\_RIGHT

Turn right.

###### ITC\_LANE\_LEFT\_RIGHT

Turn left and turn right.

### **ITC\_LANE\_RIGHT\_STRAIGHT**

Turn right and go straight.

### **ITC\_LANE\_LEFT\_RIGHT\_STRAIGHT**

Turn left, turn right and go straight.

### **ITC\_LANE\_LEFT\_WAIT**

Turn left and wait.

### **ITC\_LANE\_STRAIGHT\_WAIT**

Go straight and wait.

### **ITC\_LANE\_FORWARD**

Drive forward.

### **ITC\_LANE\_BACKWARD**

Drive backward.

### **ITC\_LANE\_BOTHWAY**

Drive bidirectionally.

### **ITC\_LANE\_STRAIGHT\_WAIT\_RIGHT**

Go straight and wait, and turn right.

## **7.1.2 ITC\_LANE\_USEAGE\_TYPE**

Enumerate the lane usage type.

### **Enumeration Definition**

```
enum {  
    ITC_LANE_USEAGE_UNKNOW = 0,  
    ITC_LANE_CARRIAGEWAY    = 1,  
    ITC_LANE_BUS             = 2,  
    ITC_LANE_FAST            = 3,  
    ITC_LANE_SLOW            = 4,  
    ITC_LANE_MOTOR           = 5,  
    ITC_LANE_NONMOTOR        = 6,  
    ITC_LANE_REVERSE_LANE    = 7,  
    ITC_LANE_BAN_TRUCKS      = 8,  
    ITC_LANE_MIX             = 9,  
    ITC_LANE_EMERGENCY       = 10,  
    ITC_LANE_BAN_LEFT        = 11,  
    ITC_LANE_BAN_RIGHT       = 12  
} ITC_LANE_USEAGE_TYPE
```

### **Member**

#### **ITC\_LANE\_USEAGE\_UNKNOW**

Unknown.

### **ITC\_LANE\_CARRIAGEWAY**

Normal lane.

### **ITC\_LANE\_BUS**

Bus lane.

### **ITC\_LANE\_FAST**

Fast lane.

### **ITC\_LANE\_SLOW**

Slow lane.

### **ITC\_LANE\_MOTOR**

Motor vehicle lane.

### **ITC\_LANE\_NONMOTOR**

Non-motor vehicle lane.

### **ITC\_LANE\_REVERSE\_LANE**

Opposite lane.

### **ITC\_LANE\_BAN\_TRUCKS**

Non-truck lane.

### **ITC\_LANE\_MIX**

All-purpose lane.

### **ITC\_LANE\_EMERGENCY**

Emergency lane.

### **ITC\_LANE\_BAN\_LEFT**

Lane banning turning left.

### **ITC\_LANE\_BAN\_RIGHT**

Lane banning turning right.

## **7.1.3 ITS\_OVERLAP\_ITEM\_TYPE**

Enumerate character overlay type.

### **Enumeration Definition**

```
enum{
    OVERLAP_ITEM_NULL                = 0,
    OVERLAP_ITEM_SITE                 = 1,
    OVERLAP_ITEM_ROADNUM,             = 2,
    OVERLAP_ITEM_INSTRUMENTNUM,       = 3,
    OVERLAP_ITEM_DIRECTION,           = 4,
```



```
OVERLAP_ITEM_DIRECTIONDESC,      = 5,
OVERLAP_ITEM_DIRECTIONDESC,      = 6,
OVERLAP_ITEM_LANEDES,            = 7,
OVERLAP_ITEM_CAPTIME,            = 8,
OVERLAP_ITEM_CAPTIME_MILLSECOND, = 9,
OVERLAP_ITEM_PLATENUM,           = 10,
OVERLAP_ITEM_CARCOLOR,           = 11,
OVERLAP_ITEM_CARTYPE,            = 12,
OVERLAP_ITEM_CARBRAND,           = 13,
OVERLAP_ITEM_CARSPEED,           = 14,
OVERLAP_ITEM_SPEEDLIMIT,         = 15,
OVERLAP_ITEM_CARLENGTH,          = 16,
OVERLAP_ITEM_ILLEGALNUM,         = 17,
OVERLAP_ITEM_MONITOR_INFO,       = 18,
OVERLAP_ITEM_ILLEGALDES,         = 19,
OVERLAP_ITEM_OVERSPEED_PERCENT,  = 20,
OVERLAP_ITEM_RED_STARTTIME,      = 21,
OVERLAP_ITEM_RED_STOPTIME,       = 22,
OVERLAP_ITEM_RED_DURATION,       = 23,
OVERLAP_ITEM_SECURITY_CODE,      = 24,
OVERLAP_ITEM_CAP_CODE,           = 25,
OVERLAP_ITEM_SEATBELT,           = 26,
OVERLAP_ITEM_MONITOR_ID,         = 27,
OVERLAP_ITEM_SUN_VISOR,          = 28,
OVERLAP_ITEM_LANE_DIRECTION,     = 29,
OVERLAP_ITEM_LICENSE_PLATE_COLOR, = 30,
OVERLAP_ITEM_SCENE_NUMBER,       = 31,
OVERLAP_ITEM_SCENE_NAME,         = 32,
OVERLAP_ITEM_YELLOW_SIGN_CAR,    = 33,
OVERLAP_ITEM_DANGEROUS_CAR,      = 34,
OVERLAP_ITEM_CAR_SUBBRAND,       = 35,
OVERLAP_ITEM_CAR_DIRECTION,      = 36,
OVERLAP_ITEM PENDANT,            = 37,
OVERLAP_ITEM_CALL,               = 38,
OVERLAP_ITEM_CAR_VALIDITY        = 39
} ITS_OVERLAP_ITEM_TYPE
```

### Member

#### OVERLAP\_ITEM\_NULL

Unknown.

#### OVERLAP\_ITEM\_SITE

Location.

#### OVERLAP\_ITEM\_ROADNUM

Intersection No.

#### OVERLAP\_ITEM\_INSTRUMENTNUM

Device No.

#### OVERLAP\_ITEM\_DIRECTION

Direction No.

### **OVERLAP\_ITEM\_DIRECTIONDESC**

Direction.

### **OVERLAP\_ITEM\_LANENUM**

Lane No.

### **OVERLAP\_ITEM\_LANEDES**

Lane.

### **OVERLAP\_ITEM\_CAPTIME**

Capture time (without milliseconds).

### **OVERLAP\_ITEM\_CAPTIME\_MILLISECOND**

Capture time (milliseconds).

### **OVERLAP\_ITEM\_PLATENUM**

License plate No.

### **OVERLAP\_ITEM\_CARCOLOR**

Vehicle color.

### **OVERLAP\_ITEM\_CARTYPE**

Vehicle type.

### **OVERLAP\_ITEM\_CARBRAND**

Vehicle brand.

### **OVERLAP\_ITEM\_CARSPEED**

Vehicle speed.

### **OVERLAP\_ITEM\_SPEEDLIMIT**

Speed limit sign.

### **OVERLAP\_ITEM\_CARLENGTH**

Vehicle length, ranges from 1 to 99, unit: m.

### **OVERLAP\_ITEM\_ILLEGALNUM**

Violation code, actually overlay illegal action information, such as low speed, overspeed, driving in the opposite direction, running the red light, occupying the road, driving over the yellow line, etc.

### **OVERLAP\_ITEM\_MONITOR\_INFO**

Camera information.

### **OVERLAP\_ITEM\_ILLEGALDES**

Illegal action.

### **OVERLAP\_ITEM\_OVERSPEED\_PERCENT**

Overspeed percentage.

**OVERLAP\_ITEM\_RED\_STARTTIME**

Red light start time.

**OVERLAP\_ITEM\_RED\_STOPTIME**

Red light end time.

**OVERLAP\_ITEM\_RED\_DURATION**

Duration of right light on.

**OVERLAP\_ITEM\_SECURITY\_CODE**

Security code.

**OVERLAP\_ITEM\_CAP\_CODE**

Capture No.

**OVERLAP\_ITEM\_SEATBELT**

Whether to fasten the seat belt.

**OVERLAP\_ITEM\_MONITOR\_ID**

Camera No.

**OVERLAP\_ITEM\_SUN\_VISOR**

Sun visor.

**OVERLAP\_ITEM\_LANE\_DIRECTION**

Driving direction of the lane.

**OVERLAP\_ITEM\_LICENSE\_PLATE\_COLOR**

License plate color.

**OVERLAP\_ITEM\_SCENE\_NUMBER**

Scene No.

**OVERLAP\_ITEM\_SCENE\_NAME**

Scene name.

**OVERLAP\_ITEM\_YELLOW\_SIGN\_CAR**

Yellow label vehicle.

**OVERLAP\_ITEM\_DANGEROUS\_CAR**

Hazardous material truck.

**OVERLAP\_ITEM\_CAR\_SUBBRAND**

Vehicle sub-brand.

**OVERLAP\_ITEM\_CAR\_DIRECTION**

Driving direction of the vehicle.

**OVERLAP\_ITEM PENDANT**

Pendant on the window.

### OVERLAP\_ITEM\_CALL

Talking on the phone.

### OVERLAP\_ITEM\_CAR\_VALIDITY

Confidence level.

### See Also

**NET\_ITS\_OVERLAP\_SINGLE\_ITEM\_PARAM\_V50**

## 7.1.4 ITC\_RELA\_LANE\_DIRECTION\_TYPE

Enumerate the lane directions.

### Enumeration Definition

```
enum{
    ITC_RELA_LANE_DIRECTION_UNKNOW      = 0,
    ITC_RELA_LANE_EAST_WEST              = 1,
    ITC_RELA_LANE_WEST_EAST              = 2,
    ITC_RELA_LANE_SOUTH_NORTH            = 3,
    ITC_RELA_LANE_NORTH_SOUTH            = 4,
    ITC_RELA_LANE_EASTSOUTH_WESTNORTH    = 5,
    ITC_RELA_LANE_WESTNORTH_EASTSOUTH    = 6,
    ITC_RELA_LANE_EASTNORTH_WESTSOUTH    = 7,
    ITC_RELA_LANE_WESTSOUTH_EASTNORTH    = 8
} ITC_RELA_LANE_DIRECTION_TYPE
```

### Member

#### ITC\_RELA\_LANE\_DIRECTION\_UNKNOW

Other

#### ITC\_RELA\_LANE\_EAST\_WEST

From east to west.

#### ITC\_RELA\_LANE\_WEST\_EAST

From west to east.

#### ITC\_RELA\_LANE\_SOUTH\_NORTH

From south to north.

#### ITC\_RELA\_LANE\_NORTH\_SOUTH

From north to south.

#### ITC\_RELA\_LANE\_EASTSOUTH\_WESTNORTH

From southeast to northwest.

#### ITC\_RELA\_LANE\_WESTNORTH\_EASTSOUTH

From northwest to southeast.

### **ITC\_RELA\_LANE\_EASTNORTH\_WESTSOUTH**

From northeast to southwest.

### **ITC\_RELA\_LANE\_WESTSOUTH\_EASTNORTH**

From southwest to northeast.

## **7.1.5 ITC\_TRIGGERMODE\_TYPE**

Enumerate the trigger modes.

### **Enumeration Definition**

```
enum{
    ITC_POST_IOSPEED_TYPE           = 0x1,
    ITC_POST_SINGLEIO_TYPE          = 0x2,
    ITC_POST_RS485_TYPE              = 0x4,
    ITC_POST_RS485_RADAR_TYPE        = 0x8,
    ITC_POST_VIRTUALCOIL_TYPE        = 0x10,
    ITC_POST_HVT_TYPE_V50            = 0x20,
    ITC_POST_MPR_TYPE                = 0x40,
    ITC_POST_PRS_TYPE                = 0x80,
    ITC_EPOLICE_IO_TRAFFICLIGHTS_TYPE = 0x100,
    ITC_EPOLICE_RS485_TYPE           = 0x200,
    ITC_PE_RS485_TYPE                = 0x10000,
    ITC_VIDEO_EPOLICE_TYPE           = 0x20000,
    ITC_VIA_VIRTUALCOIL_TYPE         = 0x40000,
    ITC_POST_IMT_TYPE                = 0x80000,
    IPC_POST_HVT_TYPE                = 0x100000,
    ITC_POST_MOBILE_TYPE             = 0x200000,
    ITC_REDLIGHT_PEDESTRIAN_TYPE     = 0x400000,
    ITC_NOCOMITY_PEDESTRIAN_TYPE     = 0x800000
} ITC_TRIGGERMODE_TYPE
```

### **Member**

#### **ITC\_POST\_IOSPEED\_TYPE**

Triggered by I/O speed detection (checkpoint)

#### **ITC\_POST\_SINGLEIO\_TYPE**

Triggered by single I/O (checkpoint)

#### **ITC\_POST\_RS485\_TYPE**

Triggered by RS-485 vehicle detector (checkpoint)

#### **ITC\_POST\_RS485\_RADAR\_TYPE**

Triggered by RS-485 radar (checkpoint)

#### **ITC\_POST\_VIRTUALCOIL\_TYPE**

Triggered by virtual coil (checkpoint)

### **ITC\_POST\_HVT\_TYPE\_V50**

Triggered by video (mixed checkpoint)

### **ITC\_POST\_MPR\_TYPE**

Triggered by multi-frame recognition (checkpoint)

### **ITC\_POST\_PRS\_TYPE**

Triggered by video detection

### **ITC\_EPOLICE\_IO\_TRAFFICLIGHTS\_TYPE**

Triggered by traffic light signal detector (intersection violation system)

### **ITC\_EPOLICE\_RS485\_TYPE**

Triggered by RS-485 vehicle detector (intersection violation system)

### **ITC\_PE\_RS485\_TYPE**

Triggered by RS-485 vehicle detector (checkpoint violation system)

### **ITC\_VIDEO\_EPOLICE\_TYPE**

Triggered by video (checkpoint violation system).

### **ITC\_VIA\_VIRTUALCOIL\_TYPE**

Triggered by VIA.

### **ITC\_POST\_IMT\_TYPE**

Triggered by smart monitoring.

### **IPC\_POST\_HVT\_TYPE**

Triggered by mixed checkpoint

### **ITC\_POST\_MOBILE\_TYPE**

Triggered by mobile device.

### **ITC\_REDLIGHT\_PEDESTRIAN\_TYPE**

Triggering by pedestrian red light running.

### **ITC\_NOCOMITY\_PEDESTRIAN\_TYPE**

Triggered by outing of comity to pedestrian.

## **7.1.6 ITC\_VIOLATION\_DETECT\_TYPE**

Enumerate violation detection types.

### **Structure Definition**

```
enum {  
    ITC_VIOLATION_POST                = 0x01,  
    ITC_VIOLATION_DRIVELINE           = 0x02,
```

```
ITC_VIOLATION_REVERSE           = 0x04,  
ITC_VIOLATION_REDLIGHT          = 0x08,  
ITC_VIOLATION_DIRECTION         = 0x10,  
ITC_VIOLATION_INTERSECTION_CONGEST = 0x20,  
ITC_VIOLATION_NONDRIVEWAY       = 0x40,  
ITC_VIOLATION_CHANGELANE        = 0x80,  
ITC_VIOLATION_BAN               = 0x100,  
ITC_VIOLATION_INTERSECTION_PARK  = 0x200,  
ITC_VIOLATION_GREEN_PARK        = 0x400,  
ITC_VIOLATION_BAN_DRIVE         = 0x800,  
ITC_VIOLATION_ACROSS_YELLOWLINE = 0x1000,  
ITC_VIOLATION_HIGH_SPEED        = 0x2000,  
ITC_VIOLATION_LOW_SPEED         = 0x4000,  
ITC_VIOLATION_TURN_AROUND       = 0x8000,  
ITC_VIOLATION_CONGESTION        = 0x10000  
} ITC_VIOLATION_DETECT_TYPE
```

### Member

#### **ITC\_VIOLATION\_POST**

Checkpoint capture.

#### **ITC\_VIOLATION\_DRIVELINE**

Driving on the lane line capture.

#### **ITC\_VIOLATION\_REVERSE**

Wrong-way driving capture.

#### **ITC\_VIOLATION\_REDLIGHT**

Red light running capture.

#### **ITC\_VIOLATION\_DIRECTION**

Driving against direction guidance capture.

#### **ITC\_VIOLATION\_INTERSECTION\_CONGEST**

Overstaying at intersection capture.

#### **ITC\_VIOLATION\_NONDRIVEWAY**

Motor vehicle on non-motor vehicle lane capture.

#### **ITC\_VIOLATION\_CHANGELANE**

Illegal lane change.

#### **ITC\_VIOLATION\_BAN**

Prohibition violation.

#### **ITC\_VIOLATION\_INTERSECTION\_PARK**

Stop vehicle over the stop line when the red light is on.

#### **ITC\_VIOLATION\_GREEN\_PARK**

Stop vehicle when the green light is on.

### ITC\_VIOLATION\_BAN\_DRIVE

Prohibition.

### ITC\_VIOLATION\_ACROSS\_YELLOWLINE

Cross the yellow line.

### ITC\_VIOLATION\_HIGH\_SPEED

Overspeed (only for checkpoint).

### ITC\_VIOLATION\_LOW\_SPEED

Low speed (only for checkpoint).

### ITC\_VIOLATION\_TURN\_AROUND

Illegal U-turning.

### ITC\_VIOLATION\_CONGESTION

Congestion.

### Remarks

- Crossing the yellow line means capturing lane change or U-turning of vehicles by cameras installed by the roadside, and the concept of illegal lane change is the same as that in the intersection violation system.
- Prohibition requires capturing all license plates of a specific area, and prohibition violation only captures a specific type of license plates in the lane.

## 7.1.7 NET\_DVR\_BARRIERGATE\_CFG

Barrier control parameter structure

### Structure Definition

```
struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    BYTE     byLaneNo;
    BYTE     byBarrierGateCtrl;
    BYTE     byEntranceNo;
    BYTE     byRes[13];
}NET_DVR_BARRIERGATE_CFG, *LPNET_DVR_BARRIERGATE_CFG;
```

### Members

#### dwSize

Structure size.

#### dwChannel

Channel No.



**byLaneNo**

Barrier No.: 0-invalid, 1-barrier No.1

**byBarrierGateCtrl**

Control parameters: 0-close barrier, 1-open barrier, 2-stop control, 3-lock barrier

**byEntranceNo**

Entrance/Exit No., which is between 1 and 8

**byRes**

Reserved, set to 0.

### 7.1.8 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.

### 7.1.9 NET\_DVR\_EVENT\_SCHEDULE

#### Arming Schedule Parameter Structure

Member	Data Type	Description
dwSize	DWORD	Structure size
struAlarmTime	Array of <b><u>NET_DVR_SCHEDULETIME</u></b>	Arming schedule, 7 days per week, 8 time periods per day
struHolidayAlarmTime	Array of <b><u>NET_DVR_SCHEDULETIME</u></b>	Holiday arming schedule, see details in the structure .
byRes	BYTE	Reserved.

### 7.1.10 NET\_DVR\_EVENT\_TRIGGER

## Structure About Event Linkage Configuration

Member	Data Type	Description
dwSize	DWORD	Structure size.
struHandleException	Array of <u><b>NET_DVR_HANDLEEXCEPTION_V41</b></u>	Exception handling mode
dwRelRecordChan	Array of DWORD	Actually triggered video channel, represented by value, read starts from 0, and it is invalid after the value of 0xffffffff being read.
struPresetChanInfo	Array of <u><b>NET_DVR_PRESETCHANNEL_INFO</b></u>	Information of channel that called preset
struCruiseChanInfo	Array of <u><b>NET_DVR_CRUISECHANNEL_INFO</b></u>	Information of channel that called patrol
struPtzTrackInfo	Array of <u><b>NET_DVR_PTZTRACKCHANNEL_INFO</b></u>	Information of channel that called pattern
byDirection	Array of BYTE	Triggering direction: 0-reserved, 1-all, 2-forward, 3-backward
szFDID	Char	Face picture library ID
byRes2	Array of BYTE	Reserved

### 7.1.11 NET\_DVR\_GEOGLOCATION

Address and location information structure.

#### Structure Definition

```
struct{
    int      iRes[2];
    DWORD    dwCity;
}NET_DVR_GEOGLOCATION, *LPNET_DVR_GEOGLOCATION;
```

#### Members

**iRes**

Reserved, set to 0.

## dwCity

Province and city.

## 7.1.12 NET\_DVR\_GUARD\_CFG

Structure about the configuration parameters of ANPR arming schedule.

### Structure Definition

```
struct{
    DWORD                                dwSize;
    NET_DVR_TIME_DETECTION              struAlarmSched[MAX_DAYS/*7*/];
    [MAX_TIMESEGMENT_V30/*8*/];
    NET_DVR_HANDLEEXCEPTION_V41        struHandleException;
    DWORD                                dwMaxRelRecordChanNum;
    DWORD                                dwRelRecordChanNum;
    DWORD                                dwRelRecordChan[MAX_CHANNUM_V30/*64*/];
    NET_DVR_TIME_DETECTION              struHolidayTime[MAX_TIMESEGMENT_V30/*8*/];
    BYTE                                byDirection;
    BYTE                                byRes[87];
}NET_DVR_GUARD_CFG, *LPNET_DVR_GUARD_CFG;
```

### Members

#### dwSize

Structure size

#### struAlarmSched

Arming schedule, 7 days for a week, maximum 8 time periods for a day, see details in the structure **NET\_DVR\_TIME\_DETECTION**.

#### struHandleException

Alarm linkage action, supports "center"-upload to center, see details in the structure **NET\_DVR\_HANDLEEXCEPTION\_V41**.

#### dwMaxRelRecordChanNum

Maximum number of triggered video channels (read only) that can be supported.

#### dwRelRecordChanNum

Actual number of triggered video channels that can be supported, that is, the number of channels in the current group.

#### dwRelRecordChan

Alarm triggered video channel No., E.g., if the value of **dwRecordChanNum** is 5, the values of **dwRelRecordChan** is between 0 and 4.

#### struHolidayTime

Holiday arming schedule, see details in the structure **NET\_DVR\_TIME\_DETECTION**.

### **byDirection**

Triggered direction: 0-reserved, 1-all, 2-forward, 3-backward.

### **byRes**

Reserved, set to 0.

### **Remarks**

- When the value of **dwMaxRelRecordChanNum** 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.

## **7.1.13 NET\_DVR\_GUARD\_COND**

Structure about the configuration conditions of ANPR arming schedule.

### **Structure Definition**

```
struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    BYTE     byRelateType;
    BYTE     byGroupNo;
    BYTE     byRes[62];
}NET_DVR_GUARD_COND, *LPNET_DVR_GUARD_COND;
```

### **Members**

#### **dwSize**

Structure size

#### **dwChannel**

Channel No.

#### **byRelateType**

Capture linkage mode: 0-invalid, 1-MPR mode (video triggered capture, for network camera only), 2-HVT mode

#### **byGroupNo**

Group No.

#### **byRes**

Reserved, set to 0.

### 7.1.14 NET\_DVR\_HANDLEEXCEPTION\_V41

#### Exception Information Structure

Member	Data Type	Description
dwHandleType	DWORD	Handling types, see details below: <ul style="list-style-type: none"><li>• 0x00: no response</li><li>• 0x01: display alarm on monitor screen</li><li>• 0x02: audio warning</li><li>• 0x04: upload to center</li><li>• 0x08: trigger alarm output</li><li>• 0x10: send picture with JPEG format by e-mail</li><li>• 0x20: trigger wireless sound and light alarm</li><li>• 0x40: trigger e-map (supported by PCNVR only)</li><li>• 0x200: capture picture and upload to FTP</li><li>• 0x400: focus mode linkage (for defocus detection)</li><li>• 0x800: PTZ linkage (speed dome tracks the target)</li><li>• 0x1000: capture picture and upload to cloud storage.</li><li>• 0x10000: message alarm</li></ul> E.g., if <b>dwHandleType</b> 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.
dwMaxAlarmOutChannelNum	DWORD	Maximum number of alarm outputs (read only) supported by the device.
dwRelAlarmOut	Array of DWORD	Alarm output No. triggered by alarm, which starts from 0, 0xffffffff-invalid. E.g. <b>byRelAlarmOut[i]==3</b> indicates that the alarm output No.4 is triggered.
byRes	Array of BYTE	Reserved, set to 0.

### 7.1.15 NET\_DVR\_INIT\_CFG\_ABILITY

## Initialization Capability Structure

Member	Data Type	Description
enumMaxLoginUsersNum	INIT_CFG_MAX_NUM	Maximum number of users can log in, see details below:  <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	Maximum number of alarm channels, see details below:  <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](#)

## 7.1.16 NET\_DVR\_LLI\_PARAM

### Longitude and Latitude Parameter Structure

Member	Data Type	Description
fSec	float	Second, range: [0.000000, 60.000000].
byDegree	BYTE	Degree, range of latitude: [0, 90], range of longitude: [0, 180].

Member	Data Type	Description
<b>byMinute</b>	BYTE	Minute, range: [0, 59].
<b>byRes</b>	BYTE[]	Reserved field whose size is 6 bytes.

### 7.1.17 NET\_DVR\_LLPOS\_PARAM

#### Position Information (Longitude and Latitude) Structure

Member	Data Type	Description
<b>byLatitudeType</b>	BYTE	Latitude type: 0-north, 1-south.
<b>byLongitudeType</b>	BYTE	Longitude type: 0-east, 1-west.
<b>byRes1</b>	BYTE[]	Reserved field whose size is 2 bytes.
<b>struLatitude</b>	<u><b>NET_DVR_LLI_PARAM</b></u>	Latitude information.
<b>struLongitude</b>	<u><b>NET_DVR_LLI_PARAM</b></u>	Longitude information.
<b>byRes</b>	BYTE[]	Reserved whose size is 16 bytes.

### 7.1.18 NET\_DVR\_MIME\_UNIT

#### Input Content Details Structure of Message Transmission API (NET\_DVR\_STDXMLConfig)

Member	Data Type	Description
<b>szContentType</b>	Array of char	Content type (corresponds to <b>Content-Type</b> field in the message), e.g., text/json. text/xml, and so on. The content format must be supported by HTTP.
<b>szName</b>	Array of char	Content name (corresponds to <b>name</b> field in the message), e.g., name="upload".
<b>szFilename</b>	Array of char	Content file name (corresponds to <b>filename</b> field in the message), e.g., filename="C:\Users\test\Desktop\11.txt".
<b>dwContentLen</b>	DWORD	Content size


Member	Data Type	Description
pContent	char*	Data point
bySelfRead	BYTE	0-External file, 1-Internal data, whose address is specified by <b>szFilename</b> .
byRes	Array of BYTE	Reserved. Set to 0. Maximum: 15 bytes.

## See Also


**NET\_DVR\_XML\_CONFIG\_INPUT**



### 7.1.19 NET\_DVR\_PLATE\_INFO

#### Structure About the Captured License Plate Information

Member	Data Type	Description
byPlateType	BYTE	License plate type.
byColor	BYTE	License plate color. For details, refer to <u><b>VCA_PLATE_COLOR</b></u> .
byBright	BYTE	License plate brightness.
byLicenseLen	BYTE	Number of characters on the license plate.
byEntireBelieve	BYTE	License plate confidence in percentage, which is between 0 and 100.
byRegion	BYTE	Region index: 0-reserved, 1-Europe, 2-Russian, 3-Europe and Russian (EU&CIS), 4-Middle East, 5-Asia-Pacific Region (APAC), 6-Africa and America, 0xff-all
byCountry	BYTE	Country/region index. For details, refer to <u><b>COUNTRY_INDEX</b></u> .  <div>  <b>Note</b>                      The value "COUNTRY_ALL" (0xff, it indicates all countries) is not supported.                 </div>
byArea	BYTE	Area in each country/region. The enumeration of areas in The United Arab Emirates is as follows:



Member	Data Type	Description
		<pre>enum EMI_AREA{     EMI_AREA_UNKNOWN = 0,          //     Unknown Area (the area recognition is     not supported)     EMI_AREA_AD,                    //Abu     Dhabi     EMI_AREA_FJR,                  //     Fujairah     EMI_AREA_DB,                   //Dubai     EMI_AREA_RAK,                  //Ras     Al Khaimah     EMI_AREA_AM,                   //Ajman     EMI_AREA_SJ,                   //     Sharjah     EMI_AREA_UMW,                  //Umm     Al Quwain     EMI_AREA_OTHER = 0xff,         //     Unrecognized (the device supports this     algorithm, but the area cannot be     recognized due to environment and so     on) }EMI_AREA;</pre>
<b>byPlateSize</b>	BYTE	License plate size: 0-unknown, 1-long, 2-short (for the Middle East).
<b>byAddInfoFlag</b>	BYTE	Additional information flag (whether the structure <b><u>NET_DVR_VEHICLE_ADDINFO</u></b> is valid): 0-no, 1-yes.
<b>wCRIndex</b>	WORD	Country/region index, which covers <b>byCountry</b> and can replace it. The <b>wCRIndex</b> is preferred over <b>byCountry</b> . If <b>wCRIndex</b> is larger than 256, you should set <b>byCountry</b> to "0xfd" (invalid).
<b>byRes</b>	BYTE[]	Reserved field whose size is 4 bytes.
<b>pAddInfoBuffer</b>	BYTE*	<p>Additional information pointer which points to the structure <b><u>NET_DVR_VEHICLE_ADDINFO</u></b> .</p> <p> <b>Note</b></p> <p>The size of this member is 8 bytes for 64-bit Windows Linux operation systems. For other operating systems, its size is 4 bytes.</p>
<b>byRes2</b>	BYTE[]	Reserved field whose size is 4 bytes.

Member	Data Type	Description
		 <b>Note</b> This member is valid for operation systems except 64-bit Windows and Linux.
<b>sPlateCategory</b>	char	Additional information on license plates in the Middle East, whose size 8 bytes.
<b>dwXmlLen</b>	DWORD	Length of the alarm information in XML format.
<b>pXmlBuf</b>	char*	Pointer of the alarm information in XML format, which points to the alarm details <b><u>XML_EventNotificationAlert_ANPR</u></b> .  <b>Note</b> This member is valid when <b>ICommand</b> in the callback function <b><u>MSGCallback</u></b> of <b><u>NET_DVR_SetDVRMessageCallBack_V50</u></b> is "COMM_ITS_PLATE_RESULT" (0x3050).
<b>struPlateRect</b>	<b><u>NET_VCA_RECT</u></b>	License plate position.
<b>sLicense</b>	char[]	License plate number whose size is 16 bytes.
<b>byBelieve</b>	BYTE[]	Confidence of each recognized character, whose size 16 bytes. For example, if the license plate number is "ZA12345" and the confidence of each character is 20, 30, 40, 50, 60, 60, and 70, it indicates that the correctness of character "Z" is 20%, the correctness of "A" is 30%, and so on.

## 7.1.20 NET\_DVR\_PLATE\_RESULT

ANPR result structure

### Structure Definition

```

struct{
    DWORD                dwSize;
    BYTE                 byResultType;
    BYTE                 byChanIndex;
    WORD                 wAlarmRecordID;
    DWORD                dwRelativeTime;
    BYTE                 byAbsTime[32];

```

```

DWORD      dwPicLen;
DWORD      dwPicPlateLen;
DWORD      dwVideoLen;
BYTE       byTrafficLight;
BYTE       byPicNum;
BYTE       byDriveChan;
BYTE       byVehicleType;
DWORD      dwBinPicLen;
DWORD      dwCarPicLen;
DWORD      dwFarCarPicLen;
BYTE       *pBuffer3;
BYTE       *pBuffer4;
BYTE       *pBuffer5;
BYTE       byRelaLaneDirectionType;
BYTE       byRes3[7];
NET_DVR_PLATE_INFO  struPlateInfo;
NET_DVR_VEHICLE_INFO struVehicleInfo;
BYTE       *pBuffer1;
BYTE       *pBuffer2;
}NET_DVR_PLATE_RESULT, *LPNET_DVR_PLATE_RESULT;

```

## Members

### dwSize

Structure size

### byResultType

Recognition type: 0-recognize via video, 1- recognize via picture, 2-continuous recorded video (support search)

### byChanIndex

Lane No.

### wAlarmRecordID

Alarm video ID (for video search only), and this parameter is valid only when **byResultType** is "2".

### dwRelativeTime

Time of UTC ± 00:00. (reserved)

### byAbsTime

Local time, accurate to millisecond, format: yyyyymmddhhmmssxxx, e.g. 20090810235959999.

### dwPicLen

Picture length (close-up picture).

### dwPicPlateLen

Length of license plate thumbnail (colorful picture).

### dwVideoLen

Video size.

**byTrafficLight**

0-capture without red or green light, 1-capture with green light, 2-capture with red light.

**byPicNum**

No. of continuously captured picture.

**byDriveChan**

Triggered lane No.

**byVehicleType**

Vehicle type, see details in **VTR\_RESULT** .

**dwBinPicLen**

Size of binary picture (for iDS-65 series only).

**dwCarPicLen**

Size of original vehicle picture (for iDS-65 series only).

**dwFarCarPicLen**

Size of long-shot picture (for iDS-65 series only).

**pBuffer3**

Binary picture of license plate (for iDS-65 series only).

**pBuffer4**

Original vehicle picture (for iDS-65 series only).

**pBuffer5**

Long-shot picture (for iDS-65 series only).

**byRelaLaneDirectionType**

Direction of linked lane, see details in **ITC\_RELA\_LANE\_DIRECTION\_TYPE** .

**byRes3**

Reserved.

**struPlateInfo**

License plate information.

**struVehicleInfo**

Vehicle information

**pBuffer1**

For close-up picture information, the size of this buffer equals to the value of **dwPicLen**; for video information, the size of this buffer equals to the value of **dwVideoLen**.

**pBuffer2**

For license plate thumbnail information, the size of this buffer equals to the value of **dwPicPlateLen**.

## Remarks

The uploaded picture or video information can be distinguished according to the information length (if the length is 0). The picture data includes scene picture and license plate thumbnail. If the video size is 0xffffffff, it indicates that the video is exception and only the alarm information (without video) will be uploaded, and the video pointer is NULL.

For iDS-65 series devices, the manually captured picture can only be uploaded to **pBuffer1** and **pBuffer2**.

If the **byResultType** is 2, the **wAlarmRecordID** can be set as the search conditions for alarm video search.

### 7.1.21 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.

### 7.1.22 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.

### 7.1.23 NET\_DVR\_SCHEDTIME

#### Structure About Start and End Time Parameters

Member	Data Type	Description
<b>byStartHour</b>	BYTE	Start time: hour.
<b>byStartMin</b>	BYTE	Start time: minute.

Member	Data Type	Description
<b>byStopHour</b>	BYTE	End time: hour.
<b>byStopMin</b>	BYTE	End time: minute.

#### 7.1.24 NET\_DVR\_STD\_ABILITY

##### Input and Output Parameter Structure for Getting Capabilities

Member	Data Type	Description
<b>lpCondBuffer</b>	LPVOID	Condition parameters (ASCII character format), e.g., the channel No., it can be set to "null".
<b>dwCondSize</b>	DWORD	Buffer size of condition parameters.
<b>lpOutBuffer</b>	LPVOID	Output parameters buffer (the parameter is returned in the message with XML format), it cannot be set to "null".
<b>dwOutSize</b>	DWORD	Output buffer size.
<b>lpStatusBuffer</b>	LPVOID	Get the returned status parameters ( <i><b><u>XML_ResponseStatus</u></b></i> ) when getting capabilities failed. It can be set to null.
<b>dwStatusSize</b>	DWORD	Status buffer size.
<b>dwRetSize</b>	DWORD	Obtained data size (if the capability is obtained, the value refers to the size of <b>lpOutBuffer</b> ; if getting failed, the value refers to the size of <b>lpStatusBuffer</b> ).
<b>byRes</b>	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 **lpCondBuffer** and output parameter **lpOutBuffer** are different. For details, refer to the typical applications.

#### 7.1.25 NET\_DVR\_STD\_CONFIG

## Structure About Configuring Input and Output Parameters

Member	Data Type	Description
<b>lpCondBuffer</b>	LPVOID	Condition parameters, e.g., channel No., it can be set to "NULL".
<b>dwCondSize</b>	DWORD	Size of buffer for storing condition parameters
<b>lpInBuffer</b>	LPVOID	Input parameters (a structure)
<b>dwInSize</b>	DWORD	Size of buffer for storing input parameters
<b>lpOutBuffer</b>	LPVOID	Output parameters (a structure)
<b>dwOutSize</b>	DWORD	Size of buffer for storing output parameters
<b>lpStatusBuffer</b>	LPVOID	Returned status parameters in XML format, it can be set to NULL.
<b>dwStatusSize</b>	DWORD	Size of buffer for storing status parameters
<b>lpXmlBuffer</b>	LPVOID	Request or response message in XML format, it is valid when <b>byDataType</b> is 1.
<b>dwXmlSize</b>	DWORD	Size of memory pointed by <b>lpXmlBuffer</b> .
<b>byDataType</b>	BYTE	Input or output parameter type: 0-valid when the input or output parameters is a structure; 1-valid when the input or output parameters is a XML message.
<b>byRes</b>	Array [BYTE]	Reserved, set to 0. The maximum size is 32 bytes.

### 7.1.26 NET\_DVR\_TIME\_V30

#### Time Parameter Structure

Member	Data Type	Description
<b>wYear</b>	WORD	Year.
<b>byMonth</b>	BYTE	Month.
<b>byDay</b>	BYTE	Day.
<b>byHour</b>	BYTE	Hour.

Member	Data Type	Description
<b>byMinute</b>	BYTE	Minute.
<b>bySecond</b>	BYTE	Second.
<b>byISO8601</b>	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.
<b>wMilliSec</b>	WORD	Millisecond.
<b>cTimeDifferenceH</b>	char	Time difference between time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when <b>byISO8601</b> is "1".
<b>cTimeDifferenceM</b>	char	Time difference between time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when <b>byISO8601</b> is "1".

### 7.1.27 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

### 7.1.28 NET\_DVR\_TIME\_DETECTION

ANPR arming schedule structure

#### Structure Definition

```
struct{
    NET_DVR_SCHEDULETIME    struSchedTime;
    BYTE                    byDetSceneID;
    BYTE                    byRes[15];
}NET_DVR_TIME_DETECTION, *LPNET_DVR_TIME_DETECTION;
```

### Members

#### **struSchedTime**

Arming schedule, start time and end time parameters, see details in the structure .

### **byDetSceneID**

Detection scene No., 0-invalid, other values: [1,4]. For network camera, the default value of this parameter is 0.

### **byRes**

Reserved, set to 0.

## **See Also**

### **7.1.29 NET\_DVR\_TRIGGER\_COND**

Structure about the configuration conditions of triggering mode.

#### **Structure Definition**

```
struct{
    DWORD      dwSize;
    DWORD      dwChannel;
    DWORD      dwtriggerMode;
    BYTE       byDetSceneID;
    BYTE       byRes[63];
}NET_DVR_TRIGGER_COND, *LPNET_DVR_TRIGGER_COND;
```

#### **Members**

##### **dwSize**

Structure size

##### **dwChannel**

Channel No.

##### **dwTriggerMode**

Triggering mode, see details in [ITC\\_TRIGGERMODE\\_TYPE](#)

##### **byDetSceneID**

Detection scene No.: 0-invalid, other values: [1,4].

##### **byRes**

Reserved, set to 0.

### **7.1.30 NET\_DVR\_VEHICLE\_ADDINFO**

## Structure about Additional Vehicle Information

Member	Data Type	Description
<b>struLLPos</b>	<b><u>NET_DVR_LLPOS_PARAM</u></b>	Longitude and latitude information of the vehicle.
<b>sVehicleNo</b>	char[]	Vehicle ID whose size is 64 bytes.
<b>byVehicleMonitorTaskID</b>	BYTE[]	Intelligent vehicle arming task ID whose size is 64 bytes. The ID is applied to the device by the upper layer when the task is created.
<b>byUUID</b>	BYTE[]	UUID whose size is 64 bytes. This member is used to link the same capture across multiple servers.
<b>byRes</b>	BYTE[]	Reserved field whose size is 832 bytes.

### 7.1.31 NET\_DVR\_VEHICLE\_CONTROL\_ALARM

Structure about the information of blacklist and allowlist ANPR alarm.

#### Structure Definition

```

struct{
    DWORD                dwSize;
    BYTE                 byListType;
    BYTE                 byPlateType;
    BYTE                 byPlateColor;
    BYTE                 byRes1;
    char                 sLicense[MAX_LICENSE_LEN/*16*/];
    char                 sCardNo[MAX_CARDNO_LEN/*48*/];
    NET_DVR_TIME_V30     struAlarmTime;
    DWORD                dwChannel;
    DWORD                dwPicDataLen;
    BYTE                 byPicType;
    BYTE                 byPicTransType
    BYTE                 byRes3[2];
    char                 *pPicData;
    BYTE                 byRes2[48];
}NET_DVR_VEHICLE_CONTROL_ALARM, *LPNET_DVR_VEHICLE_CONTROL_ALARM;

```

## Members

### **dwSize**

Structure size.

### **byListType**

List type: 0-allowlist, 1-blocklist, 2-temporary list.

### **byPlateType**

License plate type, see details in VCA\_PLATE\_TYPE.

### **byPlateColor**

License plate color, see details in VCA\_PLATE\_COLOR.

### **byRes1**

Reserved, set to 0.

### **sLicense**

License plate number

### **sCardNo**

Card No.

### **struAlarmTime**

Alarm time, see details in the structure NET\_DVR\_TIME\_V30.

### **dwChannel**

Device channel No.

### **dwPicDataLen**

Picture data size, 0-no picture, non-0-with picture data.

### **byPicType**

Picture format: 0-JPEG, 1-BMP, 2-PNG

### **byPicTransType**

Picture transmission method: 0-binary, 1-url

### **byRes3**

Reserved, set to 0.

### **pPicData**

Buffer of picture data in JPEG format.

### **byRes2**

Reserved, set to 0.

### 7.1.32 NET\_DVR\_VEHICLE\_INFO

Vehicle information structure

#### Structure Definition

```
struct{
    DWORD        dwIndex;
    BYTE          byVehicleType;
    BYTE          byColorDepth;
    BYTE          byColor;
    BYTE          byRaderState;
    WORD          wSpeed;
    WORD          wLength;
    BYTE          byIllegalType;
    BYTE          byVehicleLogoRecog;
    BYTE          byVehicleSubLogoRecog;
    BYTE          byVehicleModel;
    BYTE          byCustomInfo[16];
    WORD          wVehicleLogoRecog;
    BYTE          byIsParking;
    BYTE          byRes;
    DWORD        dwParkingTime;
    BYTE          byBelieve;
    BYTE          byRes3[7];
}NET_DVR_VEHICLE_INFO, *LPNET_DVR_VEHICLE_INFO;
```

#### Members

##### **dwIndex**

Vehicle No.

##### **byVehicleType**

Vehicle type: 0-others, 1-small-sized vehicle, 2-oversized vehicle, 3-pedestrian, 4-two-wheel vehicle, 5-three-wheel vehicle, 6-motor vehicle

##### **byColorDepth**

Vehicle color depth: 0-dark color, 1-light color

##### **byColor**

Vehicle color: 0-other, 1-white, 2-silver, 3-gray, 4-black, 5-red, 6-deep blue, 7-blue, 8-yellow, 9-green, 10-brown, 11-pink, 12-purple, 13-dark gray, 14-cyan, 0xff-unknown

##### **byRaderState**

Radar status: 0-normal, 1-fault, 2-keeps sending same speed, 3-no data sent, 4-the sent radar data is too large or too small.

##### **wSpeed**

Vehicle speed, unit: km/h

### **wLength**

Vehicle length

### **byIllegalType**

0-Normal, 1-Low Speed, 2-Overspeed, 3-Wrong-Way Driving, 4-Rad Light Running, 5-Driving on Lane Line, 6-Driving in Wrong Lane at Intersection, 7-Intersection Stranded, 8-Motor Vehicle Occupied Non-Motor Lane, 9-Illegal Lane Change, 10-Special Lane Occupancy, 11-Yellow Vehicle Restriction, 12-Intersection Parking, 13-Green Light Parking, 14-Out of Comity to Pedestrian, 15-Illegal Parking, 16-Illegal U-Turn, 17-Emergency Lane Occupancy, 18-No Right-Turn, 19-No Left-Turn, 20-Driving on Yellow Line, 21-Seatbelt Unfastened, 22-Pedestrian Red Light Running, 23-Vehicle Queue Jumping, 24-Illegal High Beam, 25-Driving with Making Call, 26-Left Turn not Yield to Straight, 27-Right Turn not Yield to Left Turn, 28-U-Turn not Yield to Straight, 29-Small Turn at Big Bend, 30-Running Green Light in Congestion, 31-Without Helmet, 32-Manned Non-Motor Vehicle, 33-Motor Vehicle on Non-Motor Vehicle Lane, 34-Non-Motor Vehicle Umbrella Tent, 35-Vehicle Discharging Black Smoke, 36-Honk, 37-Parking Over the Lane Line, 38-Occupying Two Parking Spaces, 39-Parking Over the Lane Line and Occupying Two Parking Spaces, 40-Not Yield to Vehicle from Right, 41-Not Yield to Vehicles in the Roundabout, 42-on Ramp Not Yield to Main Road, 43-Large-Sized Vehicle on the Lane, 44-Roadster Step on the Gas, 45-Smoke.

### **byVehicleLogoRecog**

Vehicle main brand. For details, refer to the enumeration [VLR\\_VEHICLE\\_CLASS](#) .

### **byVehicleSubLogoRecog**

Vehicle sub brand.

### **byVehicleModel**

Vehicle model of sub brand

### **byCustomInfo**

Custom information

### **wVehicleLogoRecog**

Vehicle main brands (it is compatible with **byVehicleLogoRecog**). For details, refer to the enumeration [VLR\\_VEHICLE\\_CLASS](#) .

### **byIsParking**

Whether the vehicle is parking: 0-invalid, 1-parking, 2-not parking.

### **byRes**

Reserved.

### **dwParkingTime**

Parking time, unit: second.

### **byBelieve**

Confidence of **byIllegalType**.

### byRes

Reserved.

### Remarks

For the vehicle main and sub brands, refer to the list provided by device.

## 7.1.33 NET\_DVR\_VIA\_LANE\_PARAM

Structure about the lane parameters for VIA video detection mode

### Structure Definition

```
struct{
    BYTE                byLaneNO;
    BYTE                byRes[63];
    NET_ITC_LANE_LOGIC_PARAM struLogicParam;
    NET_ITC_LINE        struLaneLine;
    NET_ITC_POLYGON     struPlateRecog;
    BYTE                byRes1[300];
}NET_DVR_VIA_LANE_PARAM, *LPNET_DVR_VIA_LANE_PARAM;
```

### Members

#### byLaneNO

Linked lane No.

#### byRes

Reserved.

#### struLogicParam

Lane attribute parameter, see details in the structure [NET\\_ITC\\_LANE\\_LOGIC\\_PARAM](#).

#### struLaneLine

Lane line, see details in the structure [NET\\_ITC\\_LINE](#).

#### struLaneLine

License plate recognition area parameter, see details in the structure [NET\\_ITC\\_POLYGON](#).

#### byRes1

Reserved.

### See Also

[NET\\_DVR\\_VIA\\_VTCOIL\\_PARAM](#)

### 7.1.34 NET\_DVR\_VIA\_VTCOIL\_PARAM

VIA video detection parameter structure.

#### Structure Definition

```
struct{
    BYTE                byEnable;
    BYTE                byLaneNum;
    BYTE                byRes[62];
    NET_ITC_LINE        struLaneBoundaryLine;
    NET_DVR_VIA_LANE_PARAM struLaneParam[MAX_ITC_LANE_NUM/*6*/];
    NET_ITC_PLATE_RECOG_PARAM struPlateRecog;
    BYTE                byRes1[624];
}NET_DVR_VIA_VTCOIL_PARAM, *LPNET_DVR_VIA_VTCOIL_PARAM;
```

#### Members

##### byEnable

Enable or not: 0-no, 1-yes.

##### byLaneNum

Number of recognized lanes.

##### byRes

Reserved.

##### struLaneBoundaryLine

Lane boundary line, which is the left boundary line of the leftmost lane, see details in the structure [NET\\_ITC\\_LINE](#).

##### struLaneParam

Lane parameter for VIA video detection, see details in the structure [NET\\_DVR\\_VIA\\_LANE\\_PARAM](#).

##### struPlateRecog

License plate recognition parameter, see details in the structure [NET\\_ITC\\_PLATE\\_RECOG\\_PARAM](#).

##### byRes1

Reserved.

#### See Also

[NET\\_ITC\\_TRIGGER\\_PARAM\\_UNION](#)

### 7.1.35 NET\_DVR\_XML\_CONFIG\_INPUT



**Input Parameter Structure of Message Transmission API (NET\_DVR\_STDXMLConfig)**

Member	Data Type	Description
dwSize	DWORD	Structure size.
IpRequestUrl	void*	Request URL (command) for implement different functions, and it is in string format.
dwRequestUrlLen	DWORD	Request URL size.
IpInBuffer	void*	Buffer for storing input parameters (request messages), see the input content details structure in <b><u>NET_DVR_MIME_UNIT</u></b> .
dwInBufferSize	DWORD	Input buffer size.
dwRecvTimeOut	DWORD	Receiving timeout, unit: ms, 0-5000ms (default).
byForceEncript	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 <b>IpInBuffer</b> in the structure <b><u>NET_DVR_MIME_UNIT</u></b> .
byRes	Array of BYTE	Reserved, set to 0.

**Related API****NET\_DVR\_STDXMLConfig****7.1.36 NET\_DVR\_XML\_CONFIG\_OUTPUT****Output Parameter Structure of Message Transmission API (NET\_DVR\_STDXMLConfig)**

Member	Data Type	Description
dwSize	DWORD	Structure size.
IpOutBuffer	void*	Buffer for storing output parameters (response messages), which is allocated when passing through URL by GET method.

Member	Data Type	Description
dwOutBufferSize	DWORD	Output buffer size.
dwReturnedXMLSize	DWORD	Actual size of response message.
lpStatusBuffer	void*	Response status (ResponseStatus message). This parameter will not be assigned if performing GET operation succeeded, and you can also set it to "NULL" if not required.
dwStatusSize	DWORD	Size of response status buffer.
lpDataBuffer	HPR_VOIDPTR	Buffer for transmitted data. This parameter is valid when the value of <b>byNumOfMultiPart</b> is larger than 0.
byNumOfMultiPart	HPR_UINT8	Number of parts that the message is divided into.
byRes [23]	BYTE	Reserved, set to 0.

## Related API

**NET\_DVR\_STDXMLConfig**

### 7.1.37 NET\_IPC\_LANE\_HVT\_PARAM

Mixed checkpoint lane parameter structure of network camera.

## Structure Definition

```

struct{
    BYTE                byLaneNO;
    BYTE                byCarDriveDirect;
    BYTE                byRes[62];
    NET_ITC_LINE        struLaneLine;
    NET_ITC_POLYGON     struPlateRecog;
    BYTE                byRes1[256];
}NET_IPC_LANE_HVT_PARAM, *LPNET_IPC_LANE_HVT_PARAM;

```

## Members

### byLaneNO

Lane No.

### byCarDriveDirect

Vehicle driving direction, which is enumerated below:

```
enum _ITC_LANE_CAR_DRIVE_DIRECT_{
    ITC_LANE_DRIVE_UNKNOW      = 0,
    ITC_LANE_DRIVE_UP_TO_DOWN  = 1,
    ITC_LANE_DRIVE_DOWN_TO_UP  = 2
}ITC_LANE_CAR_DRIVE_DIRECT
```

### **ITC\_LANE\_DRIVE\_UNKNOW**

Unknown

### **ITC\_LANE\_DRIVE\_UP\_TO\_DOWN**

Drive from up to bottom on image.

### **ITC\_LANE\_DRIVE\_DOWN\_TO\_UP**

Drive from bottom to up on image.

### **byRes**

Reserved, set to 0.

### **struLaneLine**

Lane line, refer to the structure **NET\_ITC\_LINE** for details.

### **struPlateRecog**

ANPR region parameters, refer to the structure **NET\_ITC\_POLYGON** for details.

### **byRes1**

Reserved, set to 0.

## **See Also**

**NET\_IPC\_POST\_HVT\_PARAM**

## **7.1.38 NET\_IPC\_POST\_HVT\_PARAM**

Structure of mixed checkpoint trigger parameters of network camera.

### **Structure Definition**

```
struct{
    BYTE                                byEnable;
    BYTE                                byLaneNum;
    BYTE                                byRes[62];
    NET_ITC_LINE                        struLaneBoundaryLine;
    NET_ITC_PLATE_RECOG_PARAM           struPlateRecog;
    NET_IPC_LANE_HVT_PARAM              struLaneParam[MAX_ITC_LANE_NUM/*6*/];
    char                                szSceneName[NAME_LEN/*32*/];
    NET_VCA_LINE                        struSnapLine;
    BYTE                                byRes1[392];
}NET_IPC_POST_HVT_PARAM, *LPNET_IPC_POST_HVT_PARAM;
```

### Members

#### **byEnable**

Whether to enable mixed checkpoint trigger mode of network camera: 0-no, 1-yes.

#### **byLaneNum**

Number of lanes to be recognized.

#### **byRes**

Reserved.

#### **struLaneBoundaryLine**

Left boundary line of left lane, refer to the structure [NET\\_ITC\\_LINE](#) for details.

#### **struPlateRecog**

ANPR parameters, refer to the structure [NET\\_ITC\\_LANE\\_MPR\\_PARAM](#) for details.

#### **struLaneParam**

Lane parameters, refer to the structure [NET\\_IPC\\_LANE\\_HVT\\_PARAM](#) for details.

#### **szSceneName**

Scene name.

#### **struSnapLine**

Capture line, it is valid only when the camera is mounted at road side, refer to the structure [NET\\_VCA\\_LINE](#) for details.

#### **byRes1**

Reserved.

### See Also

[NET\\_ITC\\_TRIGGER\\_PARAM\\_UNION](#)

## 7.1.39 NET\_ITC\_EPOLICE\_IOTL\_PARAM

Structure about IO traffic light parameters for the intersection violation system.

### Structure Definition

```
struct{
    NET_ITC_PLATE_RECOG_PARAM    struPlateRecog;
    NET_ITC_SINGLE_IOTL_PARAM    struSingleIOTL[MAX_IOSPEED_GROUP_NUM/*4*/];
    BYTE                          byRes[32];
}NET_ITC_EPOLICE_IOTL_PARAM, *LPNET_ITC_EPOLICE_IOTL_PARAM;
```

### Members

#### **struPlateRecog**

License plate recognition parameter, see details in the structure

**NET\_ITC\_PLATE\_RECOG\_PARAM**.

### **struSingleIOTL**

IO traffic light parameter of a single group, see details in the structure

**NET\_ITC\_SINGLE\_IOTL\_PARAM**.

### **byRes**

Reserved.

### **See Also**

**NET\_ITC\_TRIGGER\_PARAM\_UNION**

## **7.1.40 NET\_ITC\_EPOLICE\_LANE\_PARAM**

Structure about lane parameters of RS-485 vehicle detector trigger mode for the intersection violation system.

### **Structure Definition**

```
struct{
    BYTE                byEnable;
    BYTE                byRelatedDriveWay;
    WORD                wDistance;
    BYTE                byRecordEnable;
    BYTE                byRecordType;
    BYTE                byPreRecordTime;
    BYTE                byRecordDelayTime;
    BYTE                byRecordTimeOut;
    BYTE                bySignSpeed;
    BYTE                bySpeedLimit;
    BYTE                byOverlayDriveWay;
    NET_ITC_SERIAL_INFO struSerialInfo;
    BYTE                byRelatedIOOut[MAX_IOOUT_NUM/*4*/];
    BYTE                byFlashMode;
    BYTE                bySerialType;
    BYTE                byRelatedIOOutEx;
    BYTE                bySnapPicPreRecord;
    NET_ITC_PLATE_RECOG_REGION_PARAM struPlateRecog[MAX_LANEAREA_NUM/*2*/];
    BYTE                byBigCarSignSpeed;
    BYTE                byBigCarSpeedLimit;
    BYTE                byRedTrafficLightChan;
    BYTE                byYellowTrafficLightChan;
    BYTE                byRelaLaneDirectionType;
    BYTE                byRes3[11];
}NET_ITC_EPOLICE_LANE_PARAM, *LPNET_ITC_EPOLICE_LANE_PARAM;
```

### Members

#### **byEnable**

Whether to enable: 0-no, 1-yes.

#### **byRelatedDriveWay**

Linked lane No.

#### **wDistance**

Coil distance (reserved), unit: centimeter.

#### **byRecordEnable**

Whether to record by cycle for running red light event: 0-no, 1-yes.

#### **byRecordType**

Recording type for running red light event: 0-pre-record, 1-post-record.

#### **byPreRecordTime**

Pre-record time for running red light event, the default value is 0, unit: second.

#### **byRecordDelayTime**

Post-record time for running red light event, the default value is 0, unit: second.

#### **byRecordTimeOut**

Cycle recording timeout for running red light event, unit: second.

#### **bySignSpeed**

Marked speed limit (only valid for checkpoint intersection violation system), unit: km/h.

#### **bySpeedLimit**

Speed limit (only valid for checkpoint intersection violation system), unit: km/h.

#### **byOverlayDriveWay**

OSD overlayed lane No.

#### **struSerialInfo**

Vehicle detector parameters.

#### **byRelatedIOOut**

Linked IO output port. Multiple ports can be linked simultaneously.

#### **byFlashMode**

Flashing mode of the flash light: 0-simultaneous, 1-sequential.

#### **bySerialType**

Vehicle detector type: 0-private vehicle detector, 1-private OEM vehicle detector, 2-other vehicle detector.

#### **byRelatedIOOutEx**

Linked IO output port. The output port No. is represented by bit, e.g., bit 0 refers to IO output port 1, bit 1 refers to IO output port 2, and so on. For each bit, 0 means that the output port is not linked, and 1 means that the output port is linked. Up to 8 IO output ports can be linked. This member is compatible with **byRelatedIOOut**.

### **bySnapPicPreRecord**

Pre-record time for capturing pictures: 0-default (the second picture), 1-the first picture, 2-the second picture, 3-the third picture.

### **struPlateRecog**

License plate recognition area parameters, see details in the structure **NET\_ITC\_PLATE\_RECOG\_REGION\_PARAM**.

### **byBigCarSignSpeed**

Marked speed limit for large-sized vehicle, unit: km/h.

### **byBigCarSpeedLimit**

Speed limit for large-sized vehicle, unit: km/h.

### **byRedTrafficLightChan**

Red light channel No. of the traffic light detector, ranging from 1 to 16.

### **byYellowTrafficLightChan**

Yellow light channel No. of the traffic light detector, ranging from 1 to 16.

### **byRelaLaneDirectionType**

Linked lane direction type, see details in the structure **ITC\_RELA\_LANE\_DIRECTION\_TYPE**.

### **byRes3**

Reserved.

## **Remarks**

The linked lane No. configured by **byRelatedDriveWay** corresponds to the lane in the vehicle detector for capturing. The lane No. configured by **byOverlayDriveWay** is the overlay lane No. which is the actual lane No.

## **See Also**

**NET\_ITC\_EPOLICE\_RS485\_PARAM**

### **7.1.41 NET\_ITC\_EPOLICE\_RS485\_PARAM**

Structure about triggering parameters of RS-485 vehicle detector for intersection violation system and checkpoint intersection violation system.

## Structure Definition

```
struct{
    BYTE                byRelatedLaneNum;
    BYTE                byTrafficLightSignalSrc;
    BYTE                byRes1[2];
    NET_ITC_PLATE_RECOG_PARAM    struPlateRecog;
    NET_ITC_EPOLICE_LANE_PARAM    struLane[MAX_ITC_LANE_NUM/*6*/];
    BYTE                byRes[32];
}NET_ITC_EPOLICE_RS485_PARAM, *LPNET_ITC_EPOLICE_RS485_PARAM;
```

## Members

### byRelatedLaneNum

Number of linked lanes.

### byTrafficLightSignalSrc

Traffic light signal source: 0-vehicle detector, 1-traffic light detector.

### byRes1

Reserved.

### struPlateRecog

License plate recognition parameter, see details in the structure

**NET\_ITC\_PLATE\_RECOG\_PARAM**.

### struLane

Linked lane parameter, see details in the structure **NET\_ITC\_EPOLICE\_LANE\_PARAM**.

### byRes

Reserved.

## Remarks

The parameters **byRedTrafficLightChan** and **byYellowTrafficLightChan** in the structure **NET\_ITC\_EPOLICE\_LANE\_PARAM** are valid only when **byTrafficLightSignalSrc** is set to 1.

## See Also

**NET\_ITC\_TRIGGER\_PARAM\_UNION**

## 7.1.42 NET\_ITC\_INTERVAL\_PARAM

Structure about capture interval parameters.

## Structure Definition

```
struct{
    BYTE    byIntervalType;
    BYTE    byRes1[3];
```



```
WORD      wInterval [MAX_INTERVAL_NUM/*4*/];
BYTE      byRes [8];
}NET_ITC_INTERVAL_PARAM, *LPNET_ITC_INTERVAL_PARAM;
```

### Members

#### **byIntervalType**

Interval type: 0-time (default), 1-distance.

#### **byRes1**

Reserved.

#### **wInterval**

Burst interval (unit: millisecond) or burst distance (unit: decimeter). The burst interval type is determined by **byIntervalType**.

#### **byRes**

Reserved.

### 7.1.43 NET\_ITC\_IO\_LIGHT\_PARAM

Structure about IO access traffic light parameters.

### Structure Definition

```
struct{
    NET_ITC_SINGLE_IO_LIGHT_PARAM      struIOLight [MAX_LIGHT_NUM/*6*/];
    BYTE                                byRes [8];
}NET_ITC_IO_LIGHT_PARAM, *LPNET_ITC_IO_LIGHT_PARAM;
```

### Members

#### **struIOLight**

Single IO access traffic light parameters, see details in the structure

**NET\_ITC\_SINGLE\_IO\_LIGHT\_PARAM**.

#### **byRes**

Reserved.

### See Also

**NET\_ITC\_LIGHT\_ACCESSPARAM\_UNION**

### 7.1.44 NET\_ITC\_LANE\_HVT\_PARAM\_V50

Structure about the extended lane parameters of the mixed-traffic checkpoint.

## Structure Definition

```
struct{
    BYTE                byLaneNO;
    BYTE                byFlashMode;
    BYTE                bySignSpeed;
    BYTE                bySpeedLimit;
    BYTE                bySignLowSpeed;
    BYTE                byLowSpeedLimit;
    BYTE                byBigCarSignSpeed;
    BYTE                byBigCarSpeedLimit;
    BYTE                byBigCarSignLowSpeed;
    BYTE                byBigCarLowSpeedLimit;
    BYTE                bySnapTimes;
    BYTE                byDriveLineSnapTime;
    BYTE                byHighSpeedSnapTime;
    BYTE                byLowSpeedSnapTime;
    BYTE                byBanSnapTime;
    BYTE                byReverseSnapTime;
    BYTE                byRelatedDriveWay;
    BYTE                byLaneType;
    BYTE                byRelaLaneDirectionType;
    BYTE                byRes1[29];
    DWORD              dwVioDetectType;
    DWORD              dwRelatedIOOut;
    NET_ITC_LINE       struTrigLine;
    NET_ITC_LINE       struLineLeft;
    NET_ITC_POLYGON     struPlateRecog;
    NET_ITC_LANE_LOGIC_PARAM struLane;
    NET_ITC_INTERVAL_PARAM struInterval;
    BYTE                byRes2[280];
}NET_ITC_LANE_HVT_PARAM_V50, *LPNET_ITC_LANE_HVT_PARAM_V50;
```

## Members

### byLaneNO

Linked lane No. used for overlaying and uploading, ranging from 1 to 255.

### byFlashMode

Flashing mode of the flash light: 0-simultaneous, 1-sequential.

### bySignSpeed

Maximum marked speed limit for small-sized vehicle, unit: km/h.

### bySpeedLimit

Maximum speed limit for small-sized vehicle, unit: km/h.

### bySignLowSpeed

Minimum marked speed limit for small-sized vehicle, unit: km/h.

### byLowSpeedLimit

Minimum speed limit for small-sized vehicle, unit: km/h.

**byBigCarSignSpeed**

Maximum marked speed limit for large-sized vehicle, unit: km/h.

**byBigCarSpeedLimit**

Maximum speed limit for large-sized vehicle, unit: km/h.

**byBigCarSignLowSpeed**

Minimum marked speed limit for large-sized vehicle, unit: km/h.

**byBigCarLowSpeedLimit**

Minimum speed limit for large-sized vehicle, unit: km/h.

**bySnapTimes**

Number of captured pictures of checkpoint, ranging from 1 to 3.

**byDriveLineSnapTime**

Number of captured pictures of driving on the lane line, ranging from 1 to 3.

**byHighSpeedSnapTime**

Number of captured pictures of overspeed, ranging from 1 to 3.

**byLowSpeedSnapTime**

Number of captured pictures of low speed, ranging from 1 to 3.

**byBanSnapTime**

Number of captured pictures of prohibition violation, ranging from 1 to 3.

**byReverseSnapTime**

Number of captured pictures of wrong-way driving, ranging from 1 to 3.

**byRelatedDriveWay**

Linked lane No. used for matching vehicle detector.

**byLaneType**

Lane type: 0-unconfigured, 1-highway, 2-city express way, 0xff-other way.

**byRelaLaneDirectionType**

Linked lane direction type, see details in the structure *ITC\_RELA\_LANE\_DIRECTION\_TYPE* . It corresponds to **byRelatedDriveWay** to ensure that the lane is unique.

**byRes1**

Reserved, set to 0.

**dwVioDetectType**

Violation detection type, represented by bit. For each bit, 0 means that this type is disabled, 1 means that this type is enabled. See details in the structure *ITC\_VIOLATION\_DETECT\_TYPE* .

**dwRelatedIOOut**

Linked IO output port. Multiple ports can be linked simultaneously. The output port No. is represented by bit, e.g., bit 0 refers to IO output port 1, bit 1 refers to IO output port 2, and so on. For each bit, 0 means that the output port is not linked, and 1 means that the output port is linked.

### **struTrigLine**

Triggering line, see details in the structure **NET\_ITC\_LINE** . Currently only the triggering line of the first lane is used.

### **struLineLeft**

Left lane line, see details in the structure **NET\_ITC\_LINE** .

### **struPlateRecog**

License plate recognition area parameters, see details in the structure **NET\_ITC\_POLYGON** .

### **struLane**

Lane properties, see details in the structure **NET\_ITC\_LANE\_LOGIC\_PARAM** . In the structure **byUseageType** and **byCarDriveDirect** are valid.

### **struInterval**

Capture interval parameters, see details in the structure **NET\_ITC\_INTERVAL\_PARAM** .

### **byRes2**

Reserved, set to 0.

## **See Also**

**NET\_ITC\_POST\_HVT\_PARAM\_V50**

## **7.1.45 NET\_ITC\_LANE\_IMT\_PARAM**

Structure about lane parameters for smart monitoring.

### **Structure Definition**

```
struct{
    BYTE                byLaneNO;
    BYTE                byRelaLaneDirectionType;
    BYTE                byRes[146];
    NET_ITC_LINE        struLaneLine;
    BYTE                byRes1[256];
}NET_ITC_LANE_IMT_PARAM, *LPNET_ITC_LANE_IMT_PARAM;
```

### **Members**

#### **byLaneNO**

Overlay lane No.

#### **byRelaLaneDirectionType**

Linked lane direction type, see details in the structure **ITC\_RELA\_LANE\_DIRECTION\_TYPE** . It corresponds to the linked lane No. to ensure that the lane is unique.

### **byRes**

Reserved, set to 0.

### **struLaneLine**

Lane line, see details in the structure **NET\_ITC\_LINE** .

### **byRes1**

Reserved, set to 0.

## **See Also**

**NET\_ITC\_POST\_IMT\_PARAM**

## **7.1.46 NET\_ITC\_LANE\_LOGIC\_PARAM**

Structure about lane attribute parameters

### **Structure Definition**

```
struct{
    BYTE    byUseageType;
    BYTE    byDirectionType;
    BYTE    byCarDriveDirect;
    BYTE    byRes[33];
}NET_ITC_LANE_LOGIC_PARAM, *LPNET_ITC_LANE_LOGIC_PARAM;
```

### **Members**

#### **byUseageType**

Lane usage type, see details in the structure **ITC\_LANE\_USEAGE\_TYPE** .

#### **byDirectionType**

Lane direction type, see details in the structure **ITC\_LANE\_DIRECTION\_TYPE** .

#### **byCarDriveDirect**

Vehicle driving direction, see details below:

```
enum{
    ITC_LANE_DRIVE_UNKNOW          = 0,
    ITC_LANE_DRIVE_UP_TO_DOWN      = 1,
    ITC_LANE_DRIVE_DOWN_TO_UP      = 2
}ITC_LANE_CAR_DRIVE_DIRECT
```

#### **ITC\_LANE\_DRIVE\_UNKNOW**

Unknown.

### ITC\_LANE\_DRIVE\_UP\_TO\_DOWN

Driving from top to bottom on image (down direction).

### ITC\_LANE\_DRIVE\_DOWN\_TO\_UP

Driving from bottom to top on image (up direction).

### byRes

Reserved, set to 0.

### See Also

[NET\\_ITC\\_LANE\\_HVT\\_PARAM\\_V50](#)

[NET\\_ITC\\_LANE\\_VIDEO\\_EPOLICE\\_PARAM](#)

## 7.1.47 NET\_ITC\_LANE\_MPR\_PARAM

Structure about the lane parameters in video recognition mode.

### Structure Definition

```
struct{
    BYTE                byLaneNO;
    union{
        BYTE            uLen[4];
        struct{
            BYTE        byIONo;
            BYTE        byTriggerType;
            BYTE        byRes1[2];
        }struIO;
        struct{
            BYTE        byRelateChan;
            BYTE        byRes2[3];
        }struRS485;
    }uTssParamInfo;
    BYTE                byCarDriveDirect;
    BYTE                byRes[58];
    NET_ITC_LINE        struLaneLine;
    NET_ITC_POLYGON     struPlateRecog;
    BYTE                byRes1[256];
}NET_ITC_LANE_MPR_PARAM, *LPNET_ITC_LANE_MPR_PARAM;
```

### Members

#### byLaneNO

Vehicle lane No.

#### uTssParamInfo

Triggering parameter union, see details below.

**uLen**

Union size, which is 4 bytes.

**struIO**

Structure about the parameters of alarm input triggering mode, see details below.

**byIONo**

Linked alarm input No., which starts from 1.

**byTriggerType**

Triggering mode, 0-falling edge,1-rising edge

**byRes1**

Reserved, set as 0.

**struRS485**

Structure about the parameters of RS-485 triggering mode, see details below.

**byRelateChan**

Linked vehicle detector No., which is between 1 and 16.

**byRes2**

Reserved, set as 0.

**byCarDriveDirect**

Vehicle driving direction, see details below:

```
enum{
    ITC_LANE_DRIVE_UNKNOW      = 0,
    ITC_LANE_DRIVE_UP_TO_DOWN  = 1,
    ITC_LANE_DRIVE_DOWN_TO_UP  = 2
} ITC_LANE_CAR_DRIVE_DIRECT
```

**ITC\_LANE\_DRIVE\_UNKNOW**

Unknown.

**ITC\_LANE\_DRIVE\_UP\_TO\_DOWN**

Drive from top to bottom on image.

**ITC\_LANE\_DRIVE\_DOWN\_TO\_UP**

Drive from bottom to top on image.

**byRes**

Reserved

**struLaneLine**

Lane line, see details in the structure [NET\\_ITC\\_LINE](#) .

**struPlateRecog**

License plate region information, see details in the structure [NET\\_ITC\\_POLYGON](#) .

### byRes1

Reserved

### Remarks

The structure struIO in the union uTssParamInfo is valid when the value of parameter **bySourceType** (in structure **NET\_ITC\_POST\_MPR\_PARAM**) equal to 1; the structure struRS485 in the union uTssParamInfo is valid when the value of parameter **bySourceType** (in structure **NET\_ITC\_POST\_MPR\_PARAM**) equal to 2.

## 7.1.48 NET\_ITC\_LANE\_NOCOMITY\_PEDESTRIAN\_PARAM

Structure about lane parameters of triggering mode of not yielding to pedestrian.

### Structure Definition

```
struct{
    BYTE                byRelatedDriveWay;
    BYTE                byRelaLaneDirectionType;
    BYTE                byPedestriansNum;
    BYTE                byVehicleSpeed;
    DWORD               dwVehicleInterval;
    BYTE                byPedesDetRule;
    BYTE                byRes[3];
    NET_ITC_LINE        struLaneLine;
    NET_ITC_LINE        struStopLine;
    NET_ITC_POLYGON     struPlateRecog;
    BYTE                byRes1[280];
}NET_ITC_LANE_NOCOMITY_PEDESTRIAN_PARAM,
*LPNET_ITC_LANE_NOCOMITY_PEDESTRIAN_PARAM;
```

### Members

#### byRelatedDriveWay

Linked lane No.

#### byRelaLaneDirectionType

Linked lane direction type.

#### byPedestriansNum

Number of pedestrian threshold, ranging from 1 to 100, the default value is 1.

#### byVehicleSpeed

Vehicle speed threshold, ranging from 1 to 100, the default value is 0.

#### dwVehicleInterval

Vehicle following detection threshold, ranging from 0 to 65536, the default value is 0.

#### byPedesDetRule



Pedestrian detection rule, which indicates the walking direction of the pedestrian in the detection area: 0-from right to left, 1-from left to right.

**byRes**

Reserved, set to 0.

**struLaneLine**

Lane line, see details in the structure [NET\\_ITC\\_LINE](#).

**struStopLine**

Stop line, see details in the structure [NET\\_ITC\\_LINE](#).

**struPlateRecog**

License plate recognition area, see details in the structure [NET\\_ITC\\_POLYGON](#).

**byRes1**

Reserved, set to 0.

**See Also**

[NET\\_ITC\\_NOCOMITY\\_PEDESTRIAN\\_PARAM](#)

### 7.1.49 NET\_ITC\_LANE\_PARAM

Lane parameter structure

**Structure Definition**

```
struct{
    BYTE                byEnable;
    BYTE                byRelatedDriveWay;
    WORD                wDistance;
    WORD                wTrigDelayTime;
    BYTE                byTrigDelayDistance;
    BYTE                bySpeedCapEn;
    BYTE                bySignSpeed;
    BYTE                bySpeedLimit;
    BYTE                bySnapTimes;
    BYTE                byOverlayDriveWay;
    NET_ITC_INTERVAL_PARAM struInterval;
    BYTE                byRelatedIOOut[MAX_IOOUT_NUM];
    BYTE                byFlashMode;
    BYTE                byCartSignSpeed;
    BYTE                byCartSpeedLimit;
    BYTE                byRelatedIOOutEx;
    NET_ITC_PLATE_RECOG_REGION_PARAM struPlateRecog[MAX_LANEAREA_NUM];
    BYTE                byLaneType;
    BYTE                byUseageType;
    BYTE                byRelaLaneDirectionType;
    BYTE                byLowSpeedLimit;
```

```
BYTE                byBigCarLowSpeedLimit;  
BYTE                byLowSpeedCapEn;  
BYTE                byEmergencyCapEn;  
BYTE                byRes[9];  
}NET_ITC_LANE_PARAM, *LPNET_ITC_LANE_PARAM;
```

### Members

#### **byEnable**

Whether to enable this lane: 0-no, 1-yes.

#### **byRelatedDriveWay**

Linked lane No., which matches with the lane of vehicle detector, and it is used for capture.

#### **wDistance**

Distance between two coils, which is used for calculating speed.

#### **wTrigDelayTime**

Trigger delay time, unit: millisecond, the default value is 200.

#### **byTrigDelayDistance**

Trigger delay distance, unit: decimetre, the default value is 0.

#### **bySpeedCapEn**

Whether to trigger capture when overspeed is detected: 0-no, 1-yes.

#### **bySignSpeed**

Speed limit sign, unit: km/h

#### **bySpeedLimit**

Limit speed, unit: km/h.

#### **bySnapTimes**

Times of capture: 0-not capture, non-0-continuous capture, the maximum value is 5, the default value is 1.

#### **byOverlayDriveWay**

Lane No. to be displayed on video, which is the actual lane No.

#### **strInterval**

Time interval of capture, refer to the structure **NET\_ITC\_INTERVAL\_PARAM** for details.

#### **byRelatedIOOut**

Linked alarm output No., and multiple alarm outputs can be linked.

#### **byFlashMode**

Flash mode of flash light: 0-flash at same time, 1-switch to flash.

#### **byCartSignSpeed**

Speed limit sign of large-sized vehicle, unit: km/h.

**byCartSpeedLimit**

Limit speed of large-sized vehicle, unit: km/h.

**byRelatedIOOutEx**

Linked alarm output No., which is represented by bit: bit0-alarm output 1, bit1-alarm output 2, and so on. The value of bit: 0-not linked, 1-linked. Up to 8 alarm outputs can be linked. This parameter compatible with the parameter **byRelatedIOOut**.

**struPlateRecog**

ANPR region parameters, refer to the structure **NET\_ITC\_PLATE\_RECOG\_REGION\_PARAM** for details.

**byLaneType**

Lane type: 0-not configured, 1-highway, 1-urban expressway, 0xff-other road.

**byUsageType**

Lane usage, which is enumerated in **ITC\_LANE\_USEAGE\_TYPE**.

**byRelaLaneDirectionType**

Linked lane direction, which is enumerated in **ITC\_RELA\_LANE\_DIRECTION\_TYPE**.

**byLowSpeedLimit**

Low speed limit of small-sized vehicle, unit: km/h.

**byBigCarLowSpeedLimit**

Low speed limit of large-sized vehicle, unit: km/h.

**byLowSpeedCapEn**

Whether to trigger capture when low speed is detected: 0-no, 1-yes.

**byEmergencyCapEn**

Whether to trigger capture when emergency lane occupation is detected.

**byRes**

Reserved.

**Remarks**

### 7.1.50 NET\_ITC\_LANE\_PRS\_PARAM

Structure about lane parameters of video detection triggering mode.

**Structure Definition**

```
struct{
    BYTE                byLaneNO;
    union{
        BYTE            uLen[4];
        struct{
```

```
    BYTE    byIONo;
    BYTE    byTriggerType;
    BYTE    byRes1[2];
}struIO;
struct{
    BYTE    byRelateChan;
    BYTE    byRes2[3];
}struRS485;
}uTssParamInfo;
BYTE    byRes[59];
NET_ITC_LINE    struLaneLine;
NET_ITC_POLYGON    struPlateRecog;
BYTE    byRelaLaneDirectionType;
BYTE    byRes2[3];
NET_ITC_LINE    struTrigLine;
BYTE    byRes1[228];
}NET_ITC_LANE_PRS_PARAM, *LPNET_ITC_LANE_PRS_PARAM;
```

### Members

#### **byLaneNO**

Lane No.

#### **uTssParamInfo**

Triggering parameters information union, see its members below:

##### **uLen**

Union size (4 bytes).

##### **struIO**

Structure about triggering mode parameters, see its members below:

##### **byIONo**

Linked IO No., starting from 1.

##### **byTriggerType**

Triggering mode: 0-falling edge, 1-rising edge.

##### **byRes1**

Reserved, set to 0.

##### **struRS485**

Structure about RS-485 triggering mode parameters, see its members below:

##### **byRelateChan**

Linked vehicle detector channel No., ranging from 1 to 16.

##### **byRes2**

Reserved, set to 0.

#### **byRes**

Reserved.

#### **struLaneLine**

Lane line, see details in the structure **NET\_ITC\_LANE**.

#### **struPlateRecog**

License plate recognition area, see details in the structure **NET\_ITC\_POLYGON**.

#### **byRelaLaneDirectionType**

Linked lane direction type, see details in the structure **ITC\_RELA\_LANE\_DIRECTION\_TYPE**.

#### **byRes2**

Reserved.

#### **struTrigLine**

Triggering line, see details in the structure **NET\_ITC\_LANE**.

#### **byRes1**

Reserved.

### **See Also**

**NET\_ITC\_POST\_PRS\_PARAM**

## **7.1.51 NET\_ITC\_LANE\_VIDEO\_EPOLICE\_PARAM**

Structure about the lane parameters of video intersection violation system triggering mode.

### **Structure Definition**

```
struct{
    BYTE                                byLaneNO;
    BYTE                                bySensitivity;
    BYTE                                byEnableRadar;
    BYTE                                byRelaLaneDirectionType;
    NET_ITC_LANE_LOGIC_PARAM            struLane;
    NET_ITC_VIOLATION_DETECT_PARAM      struVioDetect;
    NET_ITC_VIOLATION_DETECT_LINE       struLine;
    NET_ITC_POLYGON                     struPlateRecog;
    BYTE                                byRecordEnable;
    BYTE                                byRecordType;
    BYTE                                byPreRecordTime;
    BYTE                                byRecordDelayTime;
    BYTE                                byRecordTimeOut;
    BYTE                                byCarSpeedLimit;
    BYTE                                byCarSignSpeed;
    BYTE                                bySnapPicPreRecord;
    NET_ITC_INTERVAL_PARAM              struInterval;
    BYTE                                byRes[36];
}NET_ITC_LANE_VIDEO_EPOLICE_PARAM, *LPNET_ITC_LANE_VIDEO_EPOLICE_PARAM;
```

### Members

#### **byLaneNO**

Linked lane No.

#### **bySensitivity**

Coil sensitivity, ranging from 1 to 100.

#### **byEnableRadar**

Whether to enable radar detection: 0-no, 1-yes.

#### **byRelaLaneDirectionType**

Linked lane direction type, see details in the structure *ITC\_RELA\_LANE\_DIRECTION\_TYPE* . It corresponds to the linked lane No. to ensure that the lane is unique.

#### **struLane**

Lane parameters, see details in the structure *NET\_ITC\_LANE\_LOGIC\_PARAM* .

#### **struVioDetect**

Violation detection parameters, see details in the structure *NET\_ITC\_VIOLATION\_DETECT\_PARAM* .

#### **struLine**

Violation detection line, see details in the structure *NET\_ITC\_VIOLATION\_DETECT\_LINE* .

#### **struPlateRecog**

License plate recognition area parameters, see details in the structure *NET\_ITC\_POLYGON* .

#### **byRecordEnable**

Whether to record by cycle for running red light event: 0-no, 1-yes.

#### **byRecordType**

Recording type for running red light event: 0-pre-record, 1-post-record.

#### **byPreRecordTime**

Pre-record time for running red light event, the default value is 0, unit: second.

#### **byRecordDelayTime**

Post-record time for running red light event, the default value is 0, unit: second.

#### **byRecordTimeOut**

Cycle recording timeout for running red light event, unit: second.

#### **byCarSpeedLimit**

Vehicle speed limit, unit: km/h.

#### **byCarSignSpeed**

Marked speed limit, unit: km/h.

#### **bySnapPicPreRecord**

Pre-record time for capturing pictures: 0-default (the second picture), 1-the first picture, 2-the second picture, 3-the third picture.

### **struInterval**

Capture interval parameters, see details in the structure [\*\*NET\\_ITC\\_INTERVAL\\_PARAM\*\*](#).

### **byRes**

Reserved, set to 0.

## **7.1.52 NET\_ITC\_LIGHT\_ACCESSPARAM\_UNION**

Traffic light connection parameters union.

### **Structure Definition**

```
union {  
    DWORD uLen[122];  
    NET_ITC_IO_LIGHT_PARAM struIOLight;  
    NET_ITC_RS485_LIGHT_PARAM struRS485Light;  
    NET_ITC_VIDEO_DETECT_LIGHT_PARAM struVideoDelectLight;  
}NET_ITC_LIGHT_ACCESSPARAM_UNION, *LPNET_ITC_LIGHT_ACCESSPARAM_UNION;
```

### **Members**

#### **uLen**

Union size.

#### **struIOLight**

IO access traffic light parameters, see details in the structure [\*\*NET\\_ITC\\_IO\\_LIGHT\\_PARAM\*\*](#).

#### **struRS485Light**

RS-485 access traffic light parameters, see details in the structure [\*\*NET\\_ITC\\_RS485\\_LIGHT\\_PARAM\*\*](#).

#### **struVideoDelectLight**

Traffic light parameters detected in the video, see details in the structure [\*\*NET\\_ITC\\_VIDEO\\_DETECT\\_LIGHT\\_PARAM\*\*](#).

### **See Also**

[\*\*NET\\_ITC\\_TRAFFIC\\_LIGHT\\_PARAM\*\*](#)

## **7.1.53 NET\_ITC\_LINE**

Traffic line information structure





Reserved, set to 0.

### 7.1.54 NET\_ITC\_NOCOMITY\_PEDESTRIAN\_PARAM

Structure about triggering parameters of not yielding to pedestrian.

#### Structure Definition

```
struct{
    BYTE                                byEnable;
    BYTE                                byLaneNum;
    BYTE                                byRes[74];
    NET_ITC_LINE                        struLaneBoundaryLine;
    NET_ITC_LINE                        struTriggerLine;
    NET_ITC_POLYGON                     struPedesDetRecog;
    NET_ITC_LANE_NOCOMITY_PEDESTRIAN_PARAM struLaneParam[MAX_ITC_LANE_NUM/
*6*/];
    NET_ITC_PLATE_RECOG_PARAM           struPlateRecog;
    BYTE                                byRes1[400];
}NET_ITC_NOCOMITY_PEDESTRIAN_PARAM, *LPNET_ITC_NOCOMITY_PEDESTRIAN_PARAM;
```

#### Members

##### byEnable

Whether to enable: 0-no, 1-yes.

##### byLaneNum

Number of recognized lanes, ranging from 1 to 3.

##### byRes

Reserved, set to 0.

##### struLaneBoundaryLine

Border line of the lane, which is the border line of the rightmost lane, see details in the structure [\*\*NET\\_ITC\\_LINE\*\*](#).

##### struTriggerLine

Triggering line of not yielding to pedestrian, see details in the structure [\*\*NET\\_ITC\\_LINE\*\*](#).

##### struPedesDetRecog

Pedestrian detection area, see details in the structure [\*\*NET\\_ITC\\_POLYGON\*\*](#).

##### struLaneParam

Lane parameters, each array represents a lane, see details in the structure [\*\*NET\\_ITC\\_LANE\\_NOCOMITY\\_PEDESTRIAN\\_PARAM\*\*](#).

##### struPlateRecog

License plate recognition parameters, see details in the structure [\*\*NET\\_ITC\\_PLATE\\_RECOG\\_PARAM\*\*](#).

**byRes1**

Reserved, set to 0.

**See Also**

**NET\_ITC\_TRIGGER\_PARAM\_UNION**

**7.1.55 NET\_ITC\_PLATE\_RECOG\_PARAM**

ANPR parameter structure.

**Structure Definition**

```
struct{
    BYTE    byDefaultCHN[MAX_CHJC_NUM/*3*/];
    BYTE    byEnable;
    DWORD   dwRecogMode;
    BYTE    byVehicleLogoRecog;
    BYTE    byProvince;
    BYTE    byRegion;
    BYTE    byRes1;
    WORD    wPlatePixelWidthMin;
    WORD    wPlatePixelWidthMax;
    BYTE    byRes[24];
}NET_ITC_PLATE_RECOG_PARAM,*LPNET_ITC_PLATE_RECOG_PARAM;
```

**Members****byDefaultCHN**

Province

**byEnable**

Whether to enable ANPR for this region: 0-no, 1-yes.

**dwRecogMode**

Recognition type:

- bit0-Recognition direction: 0-from front, 1-from back (back plate recognition);
- bit1-License plate size: 0-small,1-large;
- bit2-Vehicle color: 0-disable, 1-enable;
- bit3-Farm vehicle: 0-disable, 1-enable;
- bit4-Fuzzy recognition: 0-disable, 1-enable;
- bit5-Positioning mode: 0-positioning by frame, 1-positioning by scene;
- bit6-Recognition mode: 0-recognize by frame, 1-recognize by scene;
- bit7-Day or night: 0-day, 1-night
- bit8-Motorcycle: 0-disable, 1-enable;
- bit9-Scene mode: 0-intersection violation system/video, 1-checkpoint

- bit10-Mini license plate: 0-disable, 1-enable (60 to 80 pixels)
- bit12-Civil license plate: 0-disable, 1-enable;
- bit13-Overtilted license plate: 0-disable, 1-enable;
- bit14-Oversized license plate: 0-disable, 1-enable;
- bit15-Sun shield detection: 0-disable, 1-enable;
- bit16-Yellow-label vehicle detection: 0-disable, 1-enable;
- bit17-Dangerous goods vehicle detection: 0-disable, 1-enable;
- bit18-Embassy vehicle detection: 0-disable, 1-enable;
- bit19-Vehicle sub brand recognition: 0-disable, 1-enable;

### **byVehicleLogoRecog**

Whether to enable vehicle brand recognition: 0-no, 1-yes

### **byProvince**

Reserved

### **byRegion**

Region No.: 0-reserved, 1-Europe, 2-Russian, 3-Europe and Russian (EU&CIS)

### **byRes1**

Reserved

### **wPlatePixelWidthMin**

Minimum width of license plate that can be recognized, unit: pixel, range: [130,500]

### **wPlatePixelWidthMax**

Maximum width of license plate that can be recognized, unit: pixel, range: [130,500]

### **byRes**

Reserved.

## **7.1.56 NET\_ITC\_PLATE\_RECOG\_REGION\_PARAM**

Structure about license plate recognition area parameters.

### **Structure Definition**

```
struct{
    BYTE    byMode;
    BYTE    byRes1[3];
    union{
        NET_VCA_RECT            struRect;
        NET_ITC_POLYGON         struPolygon;
    }uRegion;
    BYTE    byRes[16];
}NET_ITC_PLATE_RECOG_REGION_PARAM, *LPNET_ITC_PLATE_RECOG_REGION_PARAM;
```

## Members

### byMode

Area type: 0-rectangle area, 1-polygon area.

### byRes1

Reserved.

### struRect

Rectangle area, see details in the structure [NET\\_VCA\\_RECT](#).

### struPolygon

Polygon area, see details in the structure [NET\\_ITC\\_POLYGON](#).

### byRes

Reserved.

## 7.1.57 NET\_ITC\_POLYGON

Polygon information structure.

### Structure Definition

```
struct{
    DWORD                dwPointNum;
    NET_VCA_POINT        struPos[ITC_MAX_POLYGON_POINT_NUM/*20*/];
}NET_ITC_POLYGON, *LPNET_ITC_POLYGON;
```

## Members

### dwPointNum

Number of valid points, which should be larger than and equal to 3, if three points are on a straight line, or the lines crossed, it indicates the invalid region.

### struPo

Polygon boundary points, the maximum number is 20, refer to the structure [NET\\_VCA\\_POINT](#) for details.

## 7.1.58 NET\_ITC\_POST\_HVT\_PARAM\_V50

Structure about the extended parameters of the mixed-traffic checkpoint.

### Structure Definition

```
struct{
    BYTE                byLaneNum;
    BYTE                byCapType;
```

```

BYTE          byCapMode;
BYTE          bySceneMode;
BYTE          bySpeedMode;
BYTE          byLineRuleEffect;
BYTE          byRes1[78];
NET_ITC_LINE  struLeftTrigLine;
NET_ITC_LINE  struRightTrigLine;
NET_ITC_LINE  struLaneBoundaryLine;
NET_ITC_POLYGON struDetectArea;
NET_DVR_GEOLOCATION struGeogLocation;
NET_ITC_LANE_HVT_PARAM_V50 struLaneParam[MAX_ITC_LANE_NUM/*6*/];
NET_ITC_PLATE_RECOG_PARAM struPlateRecog;
BYTE          byRes2[260];
}NET_ITC_POST_HVT_PARAM_V50, *LPNET_ITC_POST_HVT_PARAM_V50;

```

## Members

### byLaneNum

Number of recognized lanes, ranging from 1 to 6.

### byCapType

Capture type: 0-motor vehicle, non-motor vehicle and pedestrian (default), 1-motor vehicle.

### byCapMode

Capture mode: 0-video frame extracting, 1-interrupt and capture, 2-mixed mode.

### bySceneMode

Scene type: 0-city roads (default), 1-community entrance and exit, 2-highway.

### bySpeedMode

Speed detection mode: 0-no speed detection, 1-radar speed detection, 2-video speed detection.

### byLineRuleEffect

Validity of triggering rule line, each bit represents a triggering rule line, bit0-left triggering rule line, bit1-right triggering rule line, bit2-video detection area.

### byRes1

Reserved, set to 0.

### struLeftTrigLine

Left triggering line, which is a vertical line.

### struRightTrigLine

Right triggering line, which is a vertical line.

### struLaneBoundaryLine

Border line of the lane, which is the right border line of the rightmost lane.

### struDetectArea

Video detection area, see details in the structure **NET\_ITC\_POLYGON**.

**struGeogLocation**

Address and location, see details in the structure [NET\\_DVR\\_GEOGLOCATION](#).

**struLaneParam**

Properties of a single lane, each array indicates a kind of lane information, see details in the structure [NET\\_ITC\\_LANE\\_HVT\\_PARAM\\_V50](#).

**struPlateRecog**

License plate recognition parameters, see details in the structure [NET\\_ITC\\_PLATE\\_RECOG\\_PARAM](#).

**byRes2**

[NET\\_ITC\\_TRIGGER\\_PARAM\\_UNION](#)

## 7.1.59 NET\_ITC\_POST\_IMT\_PARAM

Structure about configuration parameters for smart monitoring.

### Structure Definition

```
struct{
    BYTE                byEnable;
    BYTE                byLaneNum;
    BYTE                bySnapMode;
    BYTE                byRes[61];
    NET_ITC_PLATE_RECOG_PARAM struPlateRecog;
    NET_ITC_LINE        struLaneBoundaryLine;
    NET_ITC_LANE_IMT_PARAM struLaneParam[MAX_ITC_LANE_NUM/*6*/];
    BYTE                byRes1[1584];
}NET_ITC_POST_IMT_PARAM, *LPNET_ITC_POST_IMT_PARAM;
```

### Members

**byEnable**

Whether to enable smart monitoring mode: 0-disable, 1-enable.

**byLaneNum**

Number of recognized lanes, ranging from 1 to 6.

**bySnapMode**

Capture type: 0-motor vehicle, 1-motor vehicle, non-motor vehicle, and pedestrian.

**byRes**

Reserved, set to 0.

**struPlateRecog**

License plate recognition parameters, see details in the structure [NET\\_ITC\\_PLATE\\_RECOG\\_PARAM](#).

### **struLaneBoundaryLine**

Lane boundary line, which is the left boundary line of the leftmost lane, see details in the structure [\*\*NET\\_ITC\\_LINE\*\*](#).

### **struLaneParam**

Properties of a single lane, see details in the structure [\*\*NET\\_ITC\\_LANE\\_IMT\\_PARAM\*\*](#).

### **byRes1**

Reserved, set to 0.

### **See Also**

[\*\*NET\\_ITC\\_TRIGGER\\_PARAM\\_UNION\*\*](#)

## **7.1.60 NET\_ITC\_POST\_IOSPEED\_PARAM**

Structure about checkpoint IO speed detection parameters.

### **Structure Definition**

```
struct{
    NET_ITC_PLATE_RECOG_PARAM      struPlateRecog;
    NET_ITC_SINGLE_IOSPEED_PARAM  struSingleIOSpeed[MAX_IOSPEED_GROUP_NUM/*4*/];
    BYTE                          byRes[32];
}NET_ITC_POST_IOSPEED_PARAM, *LPNET_ITC_POST_IOSPEED_PARAM;
```

### **Members**

#### **struPlateRecog**

License plate recognition parameters, see details in the structure [\*\*NET\\_ITC\\_PLATE\\_RECOG\\_PARAM\*\*](#).

#### **struSingleIOSpeed**

Parameters of a single IO speed detection group, see details in the structure [\*\*NET\\_ITC\\_SINGLE\\_IOSPEED\\_PARAM\*\*](#).

#### **byRes**

Reserved.

### **See Also**

[\*\*NET\\_ITC\\_TRIGGER\\_PARAM\\_UNION\*\*](#)

## **7.1.61 NET\_ITC\_POST\_MOBILE\_PARAM**

Structure about mobile traffic triggering parameters.

## Structure Definition

```
struct{  
    BYTE        byEnable;  
    BYTE        bySceneMode;  
    WORD        wExpressWayCapType;  
    WORD        wUrbanRoadCapType;  
    BYTE        byCapNum;  
    BYTE        byRecordEnable;  
    DWORD       dwPreRecordTime;  
    DWORD       dwOverRecordTime;  
    BYTE        byRes[256];  
}NET_ITC_POST_MOBILE_PARAM, *LPNET_ITC_POST_MOBILE_PARAM;
```

## Members

### **byEnable**

Whether to enable: 0-no, 1-yes.

### **bySceneMode**

Scene mode: 0-highway, 1-city road.

### **wExpressWayCapType**

Highway capture type, represented by bit: bit0-checkpoint, bit1-large-sized vehicle occupying lane, bit2-driving on hard shoulder.

### **wUrbanRoadCapType**

City road capture type, represented by bit: bit0-checkpoint, bit1-motor vehicle on non-motor vehicle lane, bit2-occupying dedicated lane.

### **byCapNum**

Number of captured pictures, ranging from 2 to 3.

### **byRecordEnable**

Whether to enable violation recording: 0-no, 1-yes.

### **dwPreRecordTime**

Pre-record time, unit: second.

### **dwOverRecordTime**

Recording timeout, unit: second.

### **byRes**

Reserved, set to 0.

## See Also

**NET\_ITC\_TRIGGER\_PARAM\_UNION**



### 7.1.62 NET\_ITC\_POST\_MPR\_PARAM

Structure about the trigger parameters of multi-frame detection.

#### Structure Definition

```
struct{
    BYTE                byEnable;
    BYTE                byLaneNum;
    BYTE                bySourceType;
    BYTE                byPicUploadType;
    BYTE                byRoadType;
    BYTE                byRes2;
    WORD                wCustomDelayTime;
    BYTE                byRes[56];
    NET_ITC_LINE        struLaneBoundaryLine;
    NET_ITC_PLATE_RECOG_PARAM struPlateRecog;
    NET_ITC_LANE_MPR_PARAM struLaneParam[MAX_ITC_LANE_NUM/*6*/];
    char                szSceneName[NAME_LEN/*32*/];
    NET_VCA_LINE        struSnapLine;
    BYTE                byRes1[392];
}NET_ITC_POST_MPR_PARAM, *LPNET_ITC_POST_MPR_PARAM;
```

#### Members

##### byEnable

Whether to enable: 0-no, 1-yes

##### byLaneNum

Number of recognized lanes

##### bySourceType

Signal source type, 0-triggered by MPR (triggered by video), 1-triggered by alarm input (inductive loop), 2-triggered by RS-485.

##### byPicUploadType

Picture uploading type: 0-upload all, 1-forward uploading, 2-backward uploading

##### byRoadType

Road Mode: 0-entrance/exit, 1-city road, 2-custom, 3-alarm input

##### byRes2

Reserved

##### wCustomDelayTime

Custom capture delay time (it is valid when **byRoadType** is "2"), value range: [0,15000], unit: ms

##### byRes

Reserved.

### **struLaneBoundaryLine**

Left boundary of left lane, see details in the structure [NET\\_ITC\\_LINE](#).

### **struPlateRecog**

ANPR parameters, see details in [NET\\_ITC\\_PLATE\\_RECOG\\_PARAM](#).

### **struLaneParam**

Lane multi-frame detection parameters, see details in [NET\\_ITC\\_LANE\\_MPR\\_PARAM](#).

### **szSceneName**

Scene name

### **struSnapLine**

Capture line, it valid when camera is mounted at road side, see details in [NET\\_VCA\\_LINE](#).

### **byRes1**

Reserved.

## 7.1.63 NET\_ITC\_POST\_PRS\_PARAM

Structure about parameters triggering video detection (PRS).

### Structure Definition

```
struct{
    BYTE                byEnable;
    BYTE                byLaneNum;
    BYTE                bySourceType;
    BYTE                bySnapMode;
    BYTE                byCapMode;
    BYTE                byNoPlatCarCap;
    BYTE                bySceneMode;
    BYTE                byRes[57];
    NET_ITC_LINE        struLaneBoundaryLine;
    NET_ITC_PLATE_RECOG_PARAM struPlateRecog;
    NET_ITC_LANE_PRS_PARAM struLaneParam[MAX_ITC_LANE_NUM/*6*/];
    BYTE                byRes1[440];
}NET_ITC_POST_PRS_PARAM, *LPNET_ITC_POST_PRS_PARAM;
```

### Members

#### **byEnable**

Whether to enable: 0-no, 1-yes.

#### **byLaneNum**

Number of recognized lanes.

#### **bySourceType**

Signal source type: 0-video detection, 1-linked IO trigger (inductive loops), 2-linked RS-485 triggering signal.

### **bySnapMode**

Capture mode: 0-panorama image, 1-panorama image+feature view.

### **byCapMode**

0-strobe light mode, 1-flash light mode. It is valid when **bySourceType** is set to 0.

### **byNoPlatCarCap**

Whether to capture vehicle without license plate: 0-no, 1-yes.

### **bySceneMode**

Scene mode: 0-normal entrance and exit, 1-toll station (vehicles will stay for longer time), 2-underground parking lot (there will be dark day and night).

### **byRes**

Reserved, set to 0.

### **struLaneBoundaryLine**

Lane boundary line, which is the left boundary line of the leftmost lane, see details in the structure **NET\_ITC\_LINE**.

### **struPlateRecog**

License plate recognition parameters, see details in the structure **NET\_ITC\_PLATE\_RECOG\_PARAM**.

### **struLaneParam**

Lane parameters, see details in the structure **NET\_ITC\_LANE\_PRS\_PARAM**.

### **byRes1**

Reserved, set to 0.

## **See Also**

**NET\_ITC\_TRIGGER\_PARAM\_UNION**

### **7.1.64 NET\_ITC\_POST\_RS485\_PARAM**

Trigger parameter structure of RS-485 vehicle detector in the checkpoint.

## **Structure Definition**

```
struct{
    BYTE                                byRelatedLaneNum;
    BYTE                                byTriggerSpareMode;
    BYTE                                byFaultToleranceTime;
    BYTE                                byRes1;
    NET_ITC_PLATE_RECOG_PARAM           struPlateRecog;
    NET_ITC_LANE_PARAM                  struLane[MAX_ITC_LANE_NUM/*6*/];
}
```

```
    BYTE                                     byRes[32];
}NET_ITC_POST_RS485_PARAM, *LPNET_ITC_POST_RS485_PARAM;
```

### Members

#### **byRelatedLaneNum**

Number of linked lanes.

#### **byTriggerSpareMode**

Trigger mode for spare (when the coil trigger is in fault): 0-none, 1-wireless virtual coil mode, 2-mixed checkpoint mode.

#### **byFaultToleranceTime**

Error tolerance time, unit: minute, which is used to check if the vehicle detector has normally worked for maximum time period.

#### **byRes1**

Reserved.

#### **struPlateRecog**

ANPR parameters, refer to the structure **NET\_ITC\_PLATE\_RECOG\_PARAM** for details.

#### **struLane**

Linked lane parameters, refer to the structure **NET\_ITC\_LANE\_PARAM** for details.

#### **byRes**

Reserved.

### See Also

**NET\_ITC\_TRIGGER\_PARAM\_UNION**

## 7.1.65 NET\_ITC\_POST\_RS485\_RADAR\_PARAM

Structure about checkpoint RS-485 radar triggering parameters.

### Structure Definition

```
struct{
    BYTE                                     byRelatedLaneNum;
    BYTE                                     byRes1[3];
    NET_ITC_PLATE_RECOG_PARAM               struPlateRecog;
    NET_ITC_LANE_PARAM                     struLane[MAX_ITC_LANE_NUM/*6*/];
    NET_ITC_RADAR_PARAM                   struRadar;
    BYTE                                     byRes[32];
}NET_ITC_POST_RS485_RADAR_PARAM, *LPNET_ITC_POST_RS485_RADAR_PARAM;
```

## Members

### byRelatedLaneNum

Number of linked lanes.

### byRes1

Reserved.

### struPlateRecog

License plate recognition parameters, see details in the structure

**NET\_ITC\_PLATE\_RECOG\_PARAM**.

### struLane

Linked lane parameters, see details in the structure **NET\_ITC\_LANE\_PARAM**.

### struRadar

Radar parameters, see details in the structure **NET\_ITC\_RADAR\_PARAM**.

### byRes

Reserved.

## See Also

**NET\_ITC\_TRIGGER\_PARAM\_UNION**

## 7.1.66 NET\_ITC\_POST\_SINGLEIO\_PARAM

Structure about single IO triggering parameters.

## Structure Definition

```
struct{
    NET_ITC_PLATE_RECOG_PARAM    struPlateRecog;
    NET_ITC_SINGLEIO_PARAM      struSingleIO[MAX_IOIN_NUMEX/*10*/];
}NET_ITC_POST_SINGLEIO_PARAM, *LPNET_ITC_POST_SINGLEIO_PARAM;
```

## Members

### struPlateRecog

License plate recognition parameters, see details in the structure

**NET\_ITC\_PLATE\_RECOG\_PARAM**.

### struSingleIO

Single IO triggering parameters, the array 0 refers to IO 1, the array 1 refers to IO 2, and so on, see details in the structure **NET\_ITC\_SINGLEIO\_PARAM**.

## See Also

**NET\_ITC\_TRIGGER\_PARAM\_UNION**

### 7.1.67 NET\_ITC\_POST\_VTCOIL\_PARAM

Structure of MPR trigger parameters in checkpoint.

#### Structure Definition

```
struct{
    BYTE                byEnable;
    BYTE                byLaneNum;
    BYTE                bySourceType;
    BYTE                byPicUploadType;
    BYTE                byRoadType;
    BYTE                byRes2;
    WORD                wCustomDelayTime;
    BYTE                byRes[56];
    NET_ITC_LINE        struLaneBoundaryLine;
    NET_ITC_PLATE_RECOG_PARAM struPlateRecog;
    NET_ITC_LANE_MPR_PARAM struLaneParam[MAX_ITC_LANE_NUM/*6*/];
    char                szSceneName[NAME_LEN/*32*/];
    NET_VCA_LINE        struSnapLine;
    BYTE                byRes1[392];
}NET_ITC_POST_MPR_PARAM, *LPNET_ITC_POST_MPR_PARAM;
```

#### Members

##### byEnable

Whether to enable checkpoint MPR trigger mode: 0-no, 1-yes.

##### byLaneNum

Number of lanes to be recognized.

##### bySourceType

Signal source type: 0-MPR triggered (video), 1-linked IO triggered (ground coil), 2-linked RS-485 triggered.

##### byPicUploadType

Picture type to be uploaded: 0-upload all, 1-, 2-

##### byRoadType

Lane type: 0-entrance and exit, 1-urban road, 2-custom, 3-alarm input.

##### byRes2

Reserved.

##### wCustomDelayTime

Custom capture delay time, it is valid when byRoadType is "2", unit: ms, value range: [0,15000].

##### byRes

Reserved.

### **struLaneBoundaryLine**

Left boundary line of left lane, refer to the structure [\*\*NET\\_ITC\\_LINE\*\*](#) for details.

### **truPlateRecog**

ANPR parameters, refer to the structure [\*\*NET\\_ITC\\_LANE\\_MPR\\_PARAM\*\*](#) for details.

### **struLaneParam**

MPR lane parameters, refer to the structure [\*\*NET\\_ITC\\_PLATE\\_RECOG\\_PARAM\*\*](#) for details.

### **szSceneName**

Scene name.

### **struSnapLine**

Capture line, it is valid only when the camera is mounted at road side, refer to the structure [\*\*NET\\_VCA\\_LINE\*\*](#) for details.

### **byRes1**

Reserved.

## **See Also**

[\*\*NET\\_ITC\\_TRIGGER\\_PARAM\\_UNION\*\*](#)

## **7.1.68 NET\_ITC\_POST\_MPR\_PARAM**

Structure about the trigger parameters of multi-frame detection.

### **Structure Definition**

```
struct{
    BYTE                byEnable;
    BYTE                byLaneNum;
    BYTE                bySourceType;
    BYTE                byPicUploadType;
    BYTE                byRoadType;
    BYTE                byRes2;
    WORD                wCustomDelayTime;
    BYTE                byRes[56];
    NET_ITC_LINE        struLaneBoundaryLine;
    NET_ITC_PLATE_RECOG_PARAM struPlateRecog;
    NET_ITC_LANE_MPR_PARAM struLaneParam[MAX_ITC_LANE_NUM/*6*/];
    char                szSceneName[NAME_LEN/*32*/];
    NET_VCA_LINE        struSnapLine;
    BYTE                byRes1[392];
}NET_ITC_POST_MPR_PARAM,*LPNET_ITC_POST_MPR_PARAM;
```

## **Members**

### **byEnable**

Whether to enable: 0-no, 1-yes

### **byLaneNum**

Number of recognized lanes

### **bySourceType**

Signal source type, 0-triggered by MPR (triggered by video), 1-triggered by alarm input (inductive loop), 2-triggered by RS-485.

### **byPicUploadType**

Picture uploading type: 0-upload all, 1-forward uploading, 2-backward uploading

### **byRoadType**

Road Mode: 0-entrance/exit, 1-city road, 2-custom, 3-alarm input

### **byRes2**

Reserved

### **wCustomDelayTime**

Custom capture delay time (it is valid when **byRoadType** is "2"), value range: [0,15000], unit: ms

### **byRes**

Reserved.

### **struLaneBoundaryLine**

Left boundary of left lane, see details in the structure [\*\*NET\\_ITC\\_LINE\*\*](#).

### **struPlateRecog**

ANPR parameters, see details in [\*\*NET\\_ITC\\_PLATE\\_RECOG\\_PARAM\*\*](#).

### **struLaneParam**

Lane multi-frame detection parameters, see details in [\*\*NET\\_ITC\\_LANE\\_MPR\\_PARAM\*\*](#).

### **szSceneName**

Scene name

### **struSnapLine**

Capture line, it valid when camera is mounted at road side, see details in [\*\*NET\\_VCA\\_LINE\*\*](#).

### **byRes1**

Reserved.

## **7.1.69 NET\_ITC\_RADAR\_PARAM**

Radar parameter structure.

### **Structure Definition**

```
struct{  
    BYTE    byRadarType;
```



```
BYTE    byLevelAngle;  
WORD    wRadarSensitivity;  
WORD    wRadarSpeedValidTime;  
BYTE    byRes1[2];  
float    fLineCorrectParam;  
int      iConstCorrectParam;  
BYTE    byRes2[8];  
}NET_ITC_RADAR_PARAM, *LPNET_ITC_RADAR_PARAM;
```

### Members

#### **byRadarType**

Radar type: 0-no radar, 1-Andoray radar, 2-Olvia, 3-TransMicrowave, 4-radar connecting I/O expansion box (this parameter is only used in checkpoint virtual coil interface and is not used for checkpoint RS-485 radar), 5-Andoray (without radar controller), 0xff-custom type.

#### **byLevelAngle**

Angle between the radar and the horizontal line, ranging from 0 degree to 90 degree, the default value is 25 degree.

#### **wRadarSensitivity**

Radar sensitivity.

#### **wRadarSpeedValidTime**

Radar speed valid time, ranging from 0 to 2000, 0 means it is not supported.

#### **byRes1**

Reserved.

#### **fLineCorrectParam**

Linear correction parameter (for multiplying operation), ranging from 0.0 to 2.0.

#### **iConstCorrectParam**

Constant correction parameter (for adding and subtracting operation), ranging from -100 to 100.

#### **byRes2**

Reserved.

### Remarks

In the checkpoint RS-485 radar triggering mode, the radar detection vehicle will be detected as soon as the vehicle passes by, which means that the speed and the capture signal are received almost simultaneously. Therefore, **wRadarSpeedValidTime** is invalid in this mode.

### 7.1.70 NET\_ITC\_REDLIGHT\_PEDESTRIAN\_PARAM

Structure about the parameters triggered by pedestrian running the red light.

## Structure Definition

```
struct{
    BYTE                byEnable;
    BYTE                bySnapNumTimes;
    BYTE                byPedesDir;
    BYTE                byDelayTime;
    BYTE                byStackTargetEnble;
    BYTE                byCalibRecogCtrl;
    BYTE                byRes1[2];
    NET_ITC_TRAFFIC_LIGHT_PARAM    struTrafficLight;
    NET_ITC_LINE                struStopLine;
    NET_ITC_POLYGON            struCalibRecog[MAX_CALIB_RECOG_NUM/*2*/];
    BYTE                byRes[440];
}NET_ITC_REDLIGHT_PEDESTRIAN_PARAM, *LPNET_ITC_REDLIGHT_PEDESTRIAN_PARAM;
```

## Members

### byEnable

Whether to enable: 0-disable, 1-enable.

### bySnapNumTimes

Number of captured pictures, ranging from 1 to 3, the default value is 3.

### byPedesDir

Pedestrian direction: 0-forward, 1-backward, 2-bidirectional.

### byDelayTime

Delay time, ranging from 1 to 5, unit: second.

### byStackTargetEnble

Whether to overlay the target frame on the captured picture, which means that the pedestrian running the red light on the first captured picture will be marked with a rectangle frame: 0-no, 1-yes.

### byCalibRecogCtrl

Manage calibration areas: 0-delete the calibration area, 1-add a calibration area.

### byRes1

Reserved, set to 0.

### struTrafficLight

Traffic light parameters, see details in the structure [NET\\_ITC\\_TRAFFIC\\_LIGHT\\_PARAM](#).

### struStopLine

Stop line, see details in the structure [NET\\_ITC\\_LINE](#).

### struCalibRecog

Calibration area, see details in the structure [NET\\_ITC\\_POLYGON](#).

### byRes

Reserved, set to 0.

### See Also

**NET\_ITC\_TRIGGER\_PARAM\_UNION**

### 7.1.71 NET\_ITC\_RS485\_LIGHT\_PARAM

Structure about RS-485 access traffic light parameters.

#### Structure Definition

```
struct{
    NET_ITC_SINGLE_RS485_LIGHT_PARAM    struRS485Light[MAX_LIGHT_NUM/*6*/];
    BYTE                                byRes[8];
}NET_ITC_RS485_LIGHT_PARAM, *LPNET_ITC_RS485_LIGHT_PARAM;
```

#### Members

##### **struRS485Light**

Single RS-485 access traffic light parameters, see details in the structure

**NET\_ITC\_SINGLE\_RS485\_LIGHT\_PARAM**.

##### **byRes**

Reserved, set to 0.

### See Also

**NET\_ITC\_LIGHT\_ACCESSPARAM\_UNION**

### 7.1.72 NET\_ITC\_SINGLE\_IO\_LIGHT\_PARAM

Structure about single IO access traffic light parameters.

#### Structure Definition

```
struct{
    BYTE    byLightType;
    BYTE    byRelatedIO;
    BYTE    byRedLightState;
    BYTE    byRes[17];
}NET_ITC_SINGLE_IO_LIGHT_PARAM, *LPNET_ITC_SINGLE_IO_LIGHT_PARAM;
```

#### Members

##### **byLightType**

Guiding direction type of the traffic light: 0-left turn signal, 1-straight signal, 2-right turn signal.

**byRelatedIO**

Linked IO port No., ranging from 1 to 6.

**byRedLightState**

Red light level status: 0-low-level red light, 1-high-level red light.

**byRes**

Reserved.

**See Also**

**NET\_ITC\_IO\_LIGHT\_PARAM**

### 7.1.73 NET\_ITC\_SINGLE\_IOSPEED\_PARAM

Structure about a single group of IO speed detection parameters.

**Structure Definition**

```
struct{
    BYTE                byEnable;
    BYTE                byTrigCoil1;
    BYTE                byCoil1IOStatus;
    BYTE                byTrigCoil2;
    BYTE                byCoil2IOStatus;
    BYTE                byRelatedDriveWay;
    BYTE                byTimeOut;
    BYTE                byRelatedIOOutEx;
    DWORD               dwDistance;
    BYTE                byCapSpeed;
    BYTE                bySpeedLimit;
    BYTE                bySpeedCapEn;
    BYTE                bySnapTimes1;
    BYTE                bySnapTimes2;
    BYTE                byBigCarSpeedLimit;
    BYTE                byBigCarSignSpeed;
    BYTE                byIntervalType;
    WORD                wInterval1[MAX_INTERVAL_NUM/*4*/];
    WORD                wInterval2[MAX_INTERVAL_NUM/*4*/];
    BYTE                byRelatedIOOut[MAX_IOOUT_NUM/*4*/];
    BYTE                byFlashMode;
    BYTE                byLaneType;
    BYTE                byCarSignSpeed;
    BYTE                byUseageType;
    NET_ITC_PLATE_RECOG_REGION_PARAM struPlateRecog[MAX_LANEAREA_NUM/*2*/];
    BYTE                byRelaLaneDirectionType;
    BYTE                byLowSpeedLimit;
    BYTE                byBigCarLowSpeedLimit;
    BYTE                byLowSpeedCapEn;
    BYTE                byEmergencyCapEn;
```

```
BYTE byRes[27];  
}NET_ITC_SINGLE_IOSPEED_PARAM, *LPNET_ITC_SINGLE_IOSPEED_PARAM;
```

### Members

#### **byEnable**

Whether to enable: 0-no, 1-yes.

#### **byTrigCoil1**

Coil 1 linked IO: 0-IO1, 1-IO2, 2-IO3, 3-IO4, 4-IO5, 5-IO6.

#### **byCoil1IOStatus**

Coil 1 IO input port status: 0-falling edge (default), 1-rising edge, 2-rising edge and falling edge, 3-high level, 4-low level.

#### **byTrigCoil2**

Cpil 2 linked IO: 0-IO1, 1-IO2, 2-IO3, 3-IO4, 4-IO5, 5-IO6.

#### **byCoil2IOStatus**

Coil 2 IO input port status: 0-falling edge (default), 1-rising edge, 2-rising edge and falling edge, 3-high level, 4-low level.

#### **byRelatedDriveWay**

Linked lane No.

#### **byTimeOut**

Timeout, the default value is 10, unit: second.

#### **byRelatedIOOutEx**

Linked IO output port. The output port No. is represented by bit, e.g., bit 0 refers to IO output port 1, bit 1 refers to IO output port 2, and so on. For each bit, 0 means that the output port is not linked, and 1 means that the output port is linked. Up to 8 IO output ports can be linked.

This member is compatible with **byRelatedIOOut**.

#### **dwDistance**

Coil distance, the default value is 1000, unit: centimeter.

#### **byCapSpeed**

Capture triggered speed, the default value is 30, unit: km/h.

#### **bySpeedLimit**

Speed limit, the default value is 60, unit: km/h.

#### **bySpeedCapEn**

Whether to enable overspeed capture: 0-no, 1-yes.

#### **bySnapTimes1**

Coil 1 capture times: 0-not capture (default), other value-number of pictures per burst capture (the maximum value is 5).

### **bySnapTimes2**

Coil 2 capture times (the default value is 1): 0-not capture, other value-number of pictures per burst capture (the maximum value is 5).

### **byBigCarSpeedLimit**

Speed limit for large-sized vehicle.

### **byBigCarSignSpeed**

Marked speed limit for large-sized vehicle, unit: km/h.

### **byIntervalType**

Interval type: 0-time (default), 1-distance.

### **wInterval1**

Coil 1 burst interval (unit: millisecond) or burst distance (unit: decimeter). The burst interval type is determined by **byIntervalType**.

### **wInterval2**

Coil 2 burst interval (unit: millisecond) or burst distance (unit: decimeter). The burst interval type is determined by **byIntervalType**.

### **byRelatedIOOut**

Linked IO output port. Multiple ports can be linked simultaneously. The array 0 refers to IO output port 1, the array 1 refers to IO output port 2, and so on. 0 means that the output port is not linked, and 1 means that the output port is linked.

### **byFlashMode**

Flashing mode of the flash light: 0-simultaneous, 1-sequential.

### **byLaneType**

Lane type: 0-unconfigured, 1-highway, 2-city express way, 0xff-other way.

### **byCarSignSpeed**

Marked speed limit for small-sized vehicle, unit: km/h.

### **byUseageType**

Lane usage type, see details in the structure **ITC\_LANE\_USEAGE\_TYPE**.

### **struPlateRecog**

License plate recognition parameters, see details in the structure **NET\_ITC\_PLATE\_RECOG\_REGION\_PARAM**. One license plate recognition area is available, and the other one is reserved.

### **byRelaLaneDirectionType**

Linked lane direction type, see details in the structure **ITC\_RELA\_LANE\_DIRECTION\_TYPE**.

### **byLowSpeedLimit**

Minimum speed limit for small-sized vehicle, unit: km/s.

### **byBigCarLowSpeedLimit**

Minimum speed limit for large-sized vehicle, unit: km/s.

### **byLowSpeedCapEn**

Whether to enable low speed capture: 0-no, 1-yes.

### **byEmergencyCapEn**

Whether to enable occupying emergency lane capture: 0-no, 1-yes.

### **byRes**

Reserved.

## **See Also**

**NET\_ITC\_POST\_IOSPEED\_PARAM**

### **7.1.74 NET\_ITC\_SINGLE\_IOTL\_PARAM**

Structure about a single group of IO traffic light parameters for the intersection violation system.

## **Structure Definition**

```
struct{
    BYTE                byEnable;
    BYTE                byLightIO;
    BYTE                byTrafficLight;
    BYTE                byTrigIO;
    BYTE                byTrigIOStatus;
    BYTE                byRelatedDriveWay;
    BYTE                byRecordEnable;
    BYTE                byRecordType;
    BYTE                byPreRecordTime;
    BYTE                byRecordDelayTime;
    BYTE                byRecordTimeOut;
    BYTE                byRedSnapTimes;
    BYTE                byGreenSnapTimes;
    BYTE                byRelatedIOOutEx;
    BYTE                byRes1;
    BYTE                byIntervalType;
    WORD                wRedInterval[MAX_INTERVAL_NUM/*4*/];
    WORD                wGreenInterval[MAX_INTERVAL_NUM/*4*/];
    BYTE                byRelatedIOOut[MAX_IOOUT_NUM/*4*/];
    BYTE                byFlashMode;
    BYTE                byRes2[3];
    NET_ITC_PLATE_RECOG_REGION_PARAM struPlateRecog[MAX_LANEAREA_NUM/*2*/];
    BYTE                byRes[32];
}NET_ITC_SINGLE_IOTL_PARAM, *LPNET_ITC_SINGLE_IOTL_PARAM;
```

## **Members**

### **byEnable**

Whether to enable: 0-no, 1-yes.

### **byLightIO**

Traffic light IO: 0-IO1, 1-IO2, 2-IO3, 3-IO4, 4-IO5, 5-IO6.

### **byTrafficLight**

Valid status of the traffic light: 0-high-level red light and low-level green light, 1-high-level green light and low-level red light.

### **byTrigIO**

Triggered IO port No.: 0-IO1, 1-IO2, 2-IO3, 3-IO4, 4-IO5, 5-IO6.

### **byTrigIOStatus**

Triggered IO port status: 0-falling edge (default), 1-rising edge, 2-rising edge and falling edge, 3-high level, 4-low level.

### **byRelatedDriveWay**

Linked lane No.

### **byRecordEnable**

Whether to record by cycle for running red light event: 0-no, 1-yes.

### **byRecordType**

Recording type for running red light event: 0-pre-record, 1-post-record.

### **byPreRecordTime**

Pre-record time for running red light event, the default value is 0, unit: second.

### **byRecordDelayTime**

Post-record time for running red light event, the default value is 0, unit: second.

### **byRecordTimeOut**

Cycle recording timeout for running red light event, unit: second.

### **byRedSnapTimes**

Red light capture times: 0-not capture, other value-number of pictures per burst capture (the maximum value is 5).

### **byGreenSnapTimes**

Green light capture times: 0-not capture, other value-number of pictures per burst capture (the maximum value is 5).

### **byRelatedIOOutEx**

Linked IO output port. The output port No. is represented by bit, e.g., bit 0 refers to IO output port 1, bit 1 refers to IO output port 2, and so on. For each bit, 0 means that the output port is not linked, and 1 means that the output port is linked. Up to 8 IO output ports can be linked.

This member is compatible with **byRelatedIOOut**.

### **byRes1**

Reserved.



### **byIntervalType**

Interval type: 0-time (default), 1-distance.

### **wRedInterval**

Burst time (unit: millisecond) or burst distance (unit: decimeter) of the red light. The burst interval type is determined by **byIntervalType**.

### **wGreenInterval**

Burst time (unit: millisecond) or burst distance (unit: decimeter) of the green light. The burst interval type is determined by **byIntervalType**.

### **byRelatedIOOut**

Linked IO output port. Multiple ports can be linked simultaneously.

### **byFlashMode**

Flashing mode of the flash light: 0-simultaneous, 1-sequential.

### **byRes2**

Reserved.

### **struPlateRecog**

License plate recognition area parameters, see details in the structure **NET\_ITC\_PLATE\_RECOG\_REGION\_PARAM**.

### **byRes**

Reserved.

## **See Also**

**NET\_ITC\_EPOLICE\_IOTL\_PARAM**

## **7.1.75 NET\_ITC\_SINGLE\_RS485\_LIGHT\_PARAM**

Structure about single RS-485 access traffic light parameters.

### **Structure Definition**

```
struct{
    BYTE    byLightType;
    BYTE    byRelatedLightChan;
    BYTE    byInputLight;
    BYTE    byRelatedYLightChan;
    BYTE    byRes[16];
}NET_ITC_SINGLE_RS485_LIGHT_PARAM, *LPNET_ITC_SINGLE_RS485_LIGHT_PARAM;
```

## **Members**

### **byLightType**

Guiding direction type of the traffic light: 0-left turn signal, 1-straight signal, 2-right turn signal.

### **byRelatedLightChan**

No. of the traffic light detector channel linked to the red light, ranging from 0 to 16, 0 indicates no red light.

### **byInputLight**

Accessed traffic light type: 0-access red light, 1-access green light. It is not supported by ITS cameras.

### **byRelatedYLightChan**

No. of the traffic light detector channel linked to the yellow light, ranging from 0 to 16, 0 indicates no yellow light.

### **byRes**

Reserved, set to 0.

## **Remarks**

For example, if the traffic light types in the intersection includes left turn+straight and right turn, you need to configure three arrays. The first array is for left turn signal and linked to detector channel 1, the second array is for straight signal and linked to detector channel 1, and the third array is for right turn signal and linked to detector channel 2.

## **See Also**

**NET\_ITC\_RS485\_LIGHT\_PARAM**

## **7.1.76 NET\_ITC\_SINGLE\_TRIGGERCFG**

A single triggering parameter structure

### **Structure Definition**

```
struct{
    BYTE                byEnable;
    BYTE                byRes1[3];
    DWORD               dwTriggerType;
    NET_ITC_TRIGGER_PARAM_UNION uTriggerParam;
    BYTE                byRes[64];
}NET_ITC_SINGLE_TRIGGERCFG, *LPNET_ITC_SINGLE_TRIGGERCFG;
```

## **Member**

### **byEnable**

Whether to enable: 0-disable, 1-enable

### **byRes1**

Reserved, set to 0

### **dwTriggerType**

Trigger mode, see details in [ITC\\_TRIGGERMODE\\_TYPE](#).

### **uTriggerParam**

Trigger parameters, see details in the structure [NET\\_ITC\\_TRIGGER\\_PARAM\\_UNION](#).

### **See Also**

[NET\\_ITC\\_TRIGGERCFG](#)

## **7.1.77 NET\_ITC\_SINGLE\_VIDEO\_DETECT\_LIGHT\_PARAM**

Structure about the traffic light parameters detected in a single video.

### **Structure Definition**

```
struct{
    BYTE                byLightNum;
    BYTE                byStraightLight;
    BYTE                byLeftLight;
    BYTE                byRightLight;
    BYTE                byRedLight;
    BYTE                byGreenLight;
    BYTE                byYellowLight;
    BYTE                byYellowLightTime;
    NET_POS_PARAM       struLightRect;
    BYTE                byRes[24];
}NET_ITC_SINGLE_VIDEO_DETECT_LIGHT_PARAM,
*LPNET_ITC_SINGLE_VIDEO_DETECT_LIGHT_PARAM;
```

### **Members**

#### **byLightNum**

Number of traffic lights.

#### **byStraightLight**

Whether the straight signal is on: 0-no, 1-yes.

#### **byLeftLight**

Whether the left turn signal is on: 0-no, 1-yes.

#### **byRightLight**

Whether the right turn signal is on: 0-no, 1-yes.

#### **byRedLight**

Whether the red light is on: 0-no, 1-yes.

#### **byGreenLight**

Whether the green light is on: 0-no, 1-yes.

### **byYellowLight**

Whether the yellow light is on: 0-no, 1-yes.

### **byYellowLightTime**

Yellow light duration used to correct recognition deviation of red light and green light.

### **struLightRect**

Traffic light area, see details in the structure **NET\_POS\_PARAM**.

### **byRes**

Reserved, set to 0.

## **See Also**

**NET\_ITC\_VIDEO\_DETECT\_LIGHT\_PARAM**

## **7.1.78 NET\_ITC\_SINGLEIO\_PARAM**

Structure about single IO parameters.

### **Structure Definition**

```
struct{
    BYTE                byDefaultStatus;
    BYTE                byRelatedDriveWay;
    BYTE                bySnapTimes;
    BYTE                byRelatedIOOutEx;
    NET_ITC_INTERVAL_PARAM struInterval;
    BYTE                byRelatedIOOut[MAX_IOOUT_NUM/*4*/];
    BYTE                byFlashMode;
    BYTE                byEnable;
    BYTE                byUseageType;
    BYTE                byEmergencyCapEn;
    NET_ITC_PLATE_RECOG_REGION_PARAM struPlateRecog[MAX_LANEAREA_NUM/*2*/];
    BYTE                byRes[32];
}NET_ITC_SINGLEIO_PARAM, *LPNET_ITC_SINGLEIO_PARAM;
```

### **Members**

#### **byDefaultStatus**

Default IO triggering status: 0-low level, 1-high level.

#### **byRelatedDriveWay**

Linked lane No.

#### **bySnapTimes**

Capture times (the default value is 1): 0-not capture, other value-number of pictures per burst capture (the maximum value is 5).

**byRelatedIOOutEx**

Linked IO output port. The output port No. is represented by bit, e.g., bit 0 refers to IO output port 1, bit 1 refers to IO output port 2, and so on. For each bit, 0 means that the output port is not linked, and 1 means that the output port is linked. Up to 8 IO output ports can be linked.

This member is compatible with **byRelatedIOOut**.

**struInterval**

Capture interval parameters, see details in the structure **NET\_ITC\_INTERVAL\_PARAM**.

**byRelatedIOOut**

Linked IO output port. Multiple ports can be linked simultaneously.

**byFlashMode**

Flashing mode of the flash light: 0-simultaneous, 1-sequential.

**byEnable**

Whether to enable single IO: 0-no, 1-yes.

**byUsageType**

Lane usage type, see details in the structure **ITC\_LANE\_USEAGE\_TYPE**.

**byEmergencyCapEn**

Whether to enable occupying emergency lane capture: 0-no, 1-yes.

**struPlateRecog**

License plate recognition area parameters, see details in the structure **NET\_ITC\_PLATE\_RECOG\_REGION\_PARAM**.

**byRes**

Reserved.

**See Also**

**NET\_ITC\_POST\_SINGLEIO\_PARAM**

## 7.1.79 NET\_ITC\_TRAFFIC\_LIGHT\_PARAM

Traffic light parameter structure.

**Structure Definition**

```
struct{
    BYTE                                bySource;
    BYTE                                byRes1[3];
    NET_ITC_LIGHT_ACCESSPARAM_UNION    struLightAccess;
    BYTE                                byRes[32];
}NET_ITC_TRAFFIC_LIGHT_PARAM, *LPNET_ITC_TRAFFIC_LIGHT_PARAM;
```

## Members

### bySource

Traffic light accessing source: 0-IO access, 1-RS-485 access, 2-video detection.

### byRes1

Reserved, set to 0.

### struLightAccess

Traffic light accessing parameters, see details in the structure

**NET\_ITC\_LIGHT\_ACCESSPARAM\_UNION**.

### byRes

Reserved, set to 0.

## 7.1.80 NET\_ITC\_TRIGGER\_PARAM\_UNION

Trigger parameter union

## Structure Definition

```
union{
    DWORD
    NET_ITC_POST_IOSPEED_PARAM          uLen[1070];
    NET_ITC_POST_SINGLEIO_PARAM         struIOSpeed;
    NET_ITC_POST_RS485_PARAM            struSingleIO;
    NET_ITC_POST_RS485_RADAR_PARAM      struPostRs485;
    NET_ITC_POST_VTCOIL_PARAM           struPostRadar;
    NET_ITC_POST_HVT_PARAM_V50          struVtCoil;
    NET_ITC_POST_MPR_PARAM              struHvtV50;
    NET_ITC_POST_PRS_PARAM              struPostMpr;
    NET_ITC_EPOLICE_IOTL_PARAM           struPostPrs;
    NET_ITC_EPOLICE_RS485_PARAM          struIOTL;
    NET_ITC_EPOLICE_RS485_PARAM          struEpoliceRs485;
    NET_DVR_VIA_VTCOIL_PARAM            struPERs485;
    NET_ITC_POST_IMT_PARAM              struViaVtCoil;
    NET_IPC_POST_HVT_PARAM              struPostImt;
    NET_ITC_POST_MOBILE_PARAM           struIpcHvt;
    NET_ITC_REDLIGHT_PEDESTRIAN_PARAM   struPostMobile;
    NET_ITC_NOCOMITY_PEDESTRIAN_PARAM   struRedLightPed;
    NET_ITC_TRIGGER_PARAM_UNION, *LPNET_ITC_TRIGGER_PARAM_UNION;
    struNoComityPed;
```

## Members

### uLen

Union size, total 4280 bytes (1070\*4).

### struIOSpeed

(Checkpoint) I/O speed detection parameters, see details in the structure **NET\_ITC\_POST\_IOSPEED\_PARAM** .

### **struSingleIO**

(Checkpoint) Single I/O parameters, see details in the structure **NET\_ITC\_POST\_SINGLEIO\_PARAM** .

### **struPostRs485**

(Checkpoint) RS-485 vehicle detector parameters, see details in the structure **NET\_ITC\_POST\_RS485\_PARAM** .

### **struPostRadar**

(Checkpoint) RS485 radar parameters, see details in the structure **NET\_ITC\_POST\_RS485\_RADAR\_PARAM** .

### **struVtCoil**

(Checkpoint) Virtual coil parameters, see details in the structure **NET\_ITC\_POST\_VTCOIL\_PARAM** .

### **struHvtV50**

(Checkpoint) Trigger parameters of mixed checkpoint's video, see details in the structure **NET\_ITC\_POST\_HVT\_PARAM\_V50** .

### **struPostMpr**

(Checkpoint) Trigger parameters of multi-frame detection (MPR), see details in the structure **NET\_ITC\_POST\_MPR\_PARAM** .

### **struPostPrs**

(PRS) Video detection trigger parameters, see details in the structure **NET\_ITC\_POST\_PRS\_PARAM** .

### **struOTL**

(Intersection violation system) Traffic light signal detector parameters, see details in the structure **NET\_ITC\_EPOLICE\_IOTL\_PARAM** .

### **struEpolicRs485**

(Intersection violation system) RS485 vehicle detector parameter, see details in the structure **NET\_ITC\_EPOLICE\_RS485\_PARAM** .

### **struPERs485**

RS485 vehicle detector parameters, see details in the structure **NET\_ITC\_EPOLICE\_RS485\_PARAM** .

### **struViaVtCoil**

(VIA) Video detection parameters, see details in the structure **NET\_DVR\_VIA\_VTCOIL\_PARAM** .

### **struPostImt**

Smart monitoring configuration parameters, see details in the structure **NET\_ITC\_POST\_IMT\_PARAM** .

### **strulpcHvt**

(Network camera) Mixed checkpoint parameters, see details in the structure [NET\\_IPC\\_POST\\_HVT\\_PARAM](#).

### **struPostMobile**

Trigger mode of mobile device, see details in the structure [NET\\_ITC\\_POST\\_MOBILE\\_PARAM](#).

### **struRedLightPed**

Trigger parameters of pedestrian red light running, see details in the structure [NET\\_ITC\\_REDLIGHT\\_PEDESTRIAN\\_PARAM](#).

### **struNoComityPed**

Trigger parameters of outing of comity to pedestrian, see details in the structure [NET\\_ITC\\_NOCOMITY\\_PEDESTRIAN\\_PARAM](#).

## **See Also**

[NET\\_ITC\\_SINGLE\\_TRIGGERCFG](#)

## **7.1.81 NET\_ITC\_TRIGGERCFG**

Trigger parameter structure.

### **Structure Definition**

```
struct{
    DWORD                dwSize;
    NET_ITC_SINGLE_TRIGGERCFG  struTriggerParam;
    BYTE                 byRes[32];
}NET_ITC_TRIGGERCFG,*LPNET_ITC_TRIGGERCFG;
```

## **Members**

### **dwSize**

Structure size

### **struTriggerParam**

Single trigger parameter, see details in the structure [NET\\_ITC\\_SINGLE\\_TRIGGERCFG](#).

### **byRes**

Reserved, set to 0.

## **7.1.82 NET\_ITC\_VIDEO\_DETECT\_LIGHT\_PARAM**

Structure about the traffic light parameters detected in the video.



## Structure Definition

```
struct{
    NET_ITC_SINGLE_VIDEO_DETECT_LIGHT_PARAM
struTrafficLight[MAX_VIDEO_DETECT_LIGHT_NUM/*12*/];
    BYTE byRes[8];
}NET_ITC_VIDEO_DETECT_LIGHT_PARAM, *LPNET_ITC_VIDEO_DETECT_LIGHT_PARAM;
```

## Members

### **struTrafficLight**

Traffic light parameters detected in a single video, see details in the structure

**NET\_ITC\_SINGLE\_VIDEO\_DETECT\_LIGHT\_PARAM**.

### **byRes**

Reserved, set to 0.

## See Also

**NET\_ITC\_LIGHT\_ACCESSPARAM\_UNION**

## 7.1.83 NET\_ITC\_VIOLATION\_DETECT\_LINE

Structure about violation detection line parameters.

## Structure Definition

```
struct{
    NET_ITC_LINE struLaneLine;
    NET_ITC_LINE struStopLine;
    NET_ITC_LINE struRedLightLine;
    NET_ITC_LINE struCancelLine;
    NET_ITC_LINE struWaitLine;
    NET_ITC_LINE struRes[8];
}NET_ITC_VIOLATION_DETECT_LINE, *LPNET_ITC_VIOLATION_DETECT_LINE;
```

## Members

### **struLaneLine**

Lane line parameters, see details in the structure **NET\_ITC\_LINE**.

### **struStopLine**

Stop line parameters, see details in the structure **NET\_ITC\_LINE**.

### **struRedLightLine**

Triggering line parameters of red light running, see details in the structure **NET\_ITC\_LINE**.

### **struCancelLine**

Canceling line parameters of triggering location of going straight, see details in the structure **NET\_ITC\_LINE**.

### **struWaitLine**

Stop line parameters of waiting area, see details in the structure **NET\_ITC\_LINE**.

### **struRes**

Reserved, set to 0.

### **Remarks**

- If the vehicle runs over the triggering line of red light running when the red light is on, the vehicle will be judged to be red light running violation. Generally, the triggering line of red light running is below the triggering line of going straight.
- If the vehicle runs over the canceling line of triggering location of going straight, the vehicle will be judged to go straight instead of turning left or right. It is used to judge violation of not driving according to the lane guidance.

### **See Also**

**NET\_ITC\_LANE\_VIDEO\_EPOLICE\_PARAM**

## **7.1.84 NET\_ITC\_VIOLATION\_DETECT\_PARAM**

Violation detection parameter structure.

### **Structure Definition**

```
struct{
    DWORD    dwVioDetectType;
    BYTE     byDriveLineSnapTimes;
    BYTE     byReverseSnapTimes;
    WORD     wStayTime;
    BYTE     byNonDriveSnapTimes;
    BYTE     byChangeLaneTimes;
    BYTE     bybanTimes;
    BYTE     byDriveLineSnapSen;
    WORD     wSnapPosFixPixel;
    BYTE     bySpeedTimes;
    BYTE     byTurnAroundEnable;
    BYTE     byThirdPlateRecogTime;
    BYTE     byPostSnapTimes;
    BYTE     byRes1[18];
    WORD     wStopLineDis;
    BYTE     byRes[14];
}NET_ITC_VIOLATION_DETECT_PARAM, *LPNET_ITC_VIOLATION_DETECT_PARAM;
```

### Members

#### **dwVioDetectType**

Violation detection type represented by bit. For each bit, 0 refers to disabling, 1 refers to enabling, see details in the structure ***ITC\_VIOLATION\_DETECT\_TYPE***.

#### **byDriveLineSnapTimes**

Number of captured pictures of driving on the lane line, ranging from 2 to 3.

#### **byReverseSnapTimes**

Number of captured pictures of wrong-way driving, ranging from 2 to 3.

#### **wStayTime**

Duration of motor vehicle on non-motor vehicle lane, unit: second. The camera will start capturing after this period.

#### **byNonDriveSnapTimes**

Number of captured pictures of motor vehicle on non-motor vehicle lane, ranging from 2 to 3.

#### **byChangeLaneTimes**

Number of captured pictures of illegal lane change, ranging from 2 to 3.

#### **bybanTimes**

Number of captured pictures of prohibition violation, ranging from 2 to 3.

#### **byDriveLineSnapSen**

Sensitivity of driving on the lane line, ranging from 0 to 100.

#### **wSnapPosFixPixel**

The minimum offset of the location between the second captured picture and the third captured picture, unit: pixel. It is valid when driving against the traffic light.

#### **bySpeedTimes**

Number of captured pictures of illegal overspeed, ranging from 2 to 3.

#### **byTurnAroundEnable**

Whether to enable illegal U-turning: 0-no, 1-yes.

#### **byThirdPlateRecogTime**

Time of recognizing the third license plate, ranging from 0 to 180, unit: second.

#### **byPostSnapTimes**

Number of captured pictures of checkpoint, ranging from 1 to 2.

#### **byRes1**

Reserved, set to 0.

#### **wStopLineDis**

The minimum distance between the vehicle and the stop line on the second violation picture of the intersection violation system, ranging from 0 to 300, unit: pixel.

**byRes**

Reserved, set to 0.

**See Also**

**NET\_ITC\_LANE\_VIDEO\_EPOLICE\_PARAM**

**7.1.85 NET\_ITS\_ILLEGAL\_INFO**

Traffic violation code structure.

**Structure Definition**

```
struct{  
    BYTE    byIllegalInfo[MAX_ILLEGAL_LEN/*64*/];  
    BYTE    byRes[256];  
}NET_ITS_ILLEGAL_INFO,*LPNET_ITS_ILLEGAL_INFO;
```

**Members****byIllegalInfo**

Traffic violation code (in character string format)

**byRes**

Reserved, set to 0.

**See Also**

**NET\_ITS\_PLATE\_RESULT**

**7.1.86 NET\_ITS\_OVERLAP\_CFG\_V50**

Extended text overlay parameter structure (V50)

**Structure Definition**

```
struct{  
    DWORD    dwSize;  
    BYTE    byEnable;  
    BYTE    byRes1[3];  
    NET_ITS_OVERLAP_ITEM_PARAM_V50    struOverLapItemV50;  
    NET_ITS_OVERLAP_INFO_PARAM    struOverLapInfo;  
    BYTE    byRes[120];  
}NET_ITS_OVERLAP_CFG_V50,*LPNET_ITS_OVERLAP_CFG_V50;
```

### Members

#### **dwSize**

Structure size.

#### **byEnable**

Whether to enable: 0-no, 1-yes

#### **byRes1**

Reserved.

#### **struOverLapItemV50**

Character string parameter, refer to the structure [NET\\_ITS\\_OVERLAP\\_ITEM\\_PARAM\\_V50](#) for details.

#### **struOverLapInfo**

Character string details, refer to the structure [NET\\_ITS\\_OVERLAP\\_INFO\\_PARAM](#) for details.

#### **byRes**

Reserved.

### 7.1.87 NET\_ITS\_OVERLAP\_INFO\_PARAM

Structure about the character string to be displayed.

### Structure Definition

```
struct{
    BYTE    bySite[128];
    BYTE    byRoadNum[32];
    BYTE    byInstrumentNum[32];
    BYTE    byDirection[32];
    BYTE    byDirectionDesc[32];
    BYTE    byLaneDes[32];
    BYTE    byRes1[32];
    BYTE    byMonitoringSite1[44];
    BYTE    byMonitoringSite2[32];
    BYTE    byRes[64];
}NET_ITS_OVERLAP_INFO_PARAM, *LPNET_ITS_OVERLAP_INFO_PARAM;
```

### Members

#### **bySite**

Location description.

#### **byRoadNum**

Intersection No.

#### **byInstrumentNum**

Device No.

**byDirection**

Direction No.

**byDirectionDesc**

Direction description.

**byLaneDes**

Lane description

**byRes1**

Reserved, set to 0.

**byMonitoringSite1**

Information of camera No.1

**byMonitoringSite2**

Information of camera No.2

**byRes**

Reserved, set to 0.

**See Also**

**NET\_ITS\_OVERLAP\_CFG\_V50**

### 7.1.88 NET\_ITS\_OVERLAP\_ITEM\_PARAM\_V50

Structure about configuration parameters for overlaying characters

#### Structure Definition

```
struct{
    NET_ITS_OVERLAP_SINGLE_ITEM_PARAM_V50    struSingleItem[MAX_OVERLAP_ITEM_NUM/
*50*[];
    DWORD                                     dwLinePercent;
    DWORD                                     dwItemsStlye;
    WORD                                      wStartPosTop;
    WORD                                      wStartPosLeft;
    WORD                                      wCharStyle;
    WORD                                      wCharSize;
    WORD                                      wCharInterval;
    BYTE                                      byRes1[2];
    DWORD                                     dwForeClorRGB;
    DWORD                                     dwBackClorRGB;
    BYTE                                      byColorAdapt;
    BYTE                                      byParamFillZeroEnble;
    BYTE                                      byPlateLeftCornerEnable;
    BYTE                                      byRes2;
```

```
WORD        wStartSPicPosTop;
WORD        wStartSPicPosLeft;
BYTE        byOsdLocate;
BYTE        byRes[63];
}NET_ITS_OVERLAP_ITEM_PARAM_V50, *LPNET_ITS_OVERLAP_ITEM_PARAM_V50;
```

### Members

#### **struSingleItem**

Single character parameter.

#### **dwLinePercent**

Percentage of overlayed lines, ranges from 0 to 100, the default value is 100.

#### **dwItemsStlye**

Overlaying type: 0-horizontal (default), 1-vertical.

#### **wStartPosTop**

The coordinate of top starting point, it is only valid for overlaying within the picture, ranges from 0 to 2,448, the default value is 0.

#### **wStartPosLeft**

The coordinate of left starting point, it is only valid for overlaying within the picture, ranges from 0 to 2,448, the default value is 0.

#### **wCharStyle**

Font type: 0-SimSun, 1-STXinwei (default).

#### **wCharSize**

Character size: 0-16\*16, 1-32\*32, 2-48\*48, 3-64\*64 (default), 4-128\*128.

#### **wCharInterval**

Space distance between two characters, ranges from 0 to 16, available unit: pixel (default).

#### **byRes1**

Reserved, set to 0.

#### **dwForeClorRGB**

RGB value of foreground color:

### See Also

## 7.1.89 NET\_ITS\_OVERLAP\_SINGLE\_ITEM\_PARAM\_V50

Structure about overlaying single character information.

## Structure Definition

```
struct{
    BYTE    byRes1[2];
    BYTE    byItemType;
    BYTE    byChangeLineNum;
    BYTE    bySpaceNum;
    BYTE    byRes2[2];
    BYTE    byEnablePos;
    WORD    wStartPosTop;
    WORD    wStartPosLeft;
    BYTE    byItemTypeCustom[32];
    BYTE    byRes[8];
}NET_ITS_OVERLAP_SINGLE_ITEM_PARAM_V50,
*LPNET_ITS_OVERLAP_SINGLE_ITEM_PARAM_V50;
```

## Members

### byRes1

Reserved.

### byItemType

Type, see details in the structure [ITS\\_OVERLAP\\_ITEM\\_TYPE](#).

### byChangeLineNum

Number of line feeds after overlaying items, ranges from 0 to 10, the default value is 0.

### bySpaceNum

Number of spaces after overlaying items, ranges from 0 to 255, the default value is 0.

### byRes2

Reserved.

### byEnablePos

Whether to enable displaying coordinates: 0-no, 1-yes.

### wStartPosTop

The coordinate of top starting point, it is only valid for overlaying within the picture, ranges from 0 to 2,448, the default value is 0.

### wStartPosLeft

The coordinate of left starting point, it is only valid for overlaying within the picture, ranges from 0 to 2,448, the default value is 0.

### byItemTypeCustom

Custom type, corresponds to the parameter **byItemType**, customize type name for **byItemType** parameter. If **byItemTypeCustom** is "NULL", the type name of **byItemType** parameter will be used by default.

### byRes



Reserved.

## See Also

**NET\_ITS\_OVERLAP\_ITEM\_PARAM\_V50**

### 7.1.90 NET\_ITS\_OVERLAPCFG\_COND

Structure about the OSD configuration conditions.

#### Structure Definition

```
struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    DWORD    dwConfigMode;
    BYTE     byPicModeType;
    BYTE     byRes[15];
}NET_ITS_OVERLAPCFG_COND, *LPNET_ITS_OVERLAPCFG_COND;
```

#### Members

##### **dwSize**

Structure size

##### **dwChannel**

Channel No.

##### **dwConfigMode**

Configuration mode: 0-via client software, 1-access device via web browser to configure

##### **byPicModeType**

Picture mode: 0-thumbnail, 1-large picture (composite picture)

##### **byRes**

Reserved, set to 0

### 7.1.91 NET\_ITS\_PLATE\_RESULT

ANPR result structure

#### Structure Definition

```
struct{
    DWORD    dwSize;
    DWORD    dwMatchNo;
    BYTE     byGroupNum;
    BYTE     byPicNo;
```

```

BYTE                bySecondCam;
BYTE                byFeaturePicNo;
BYTE                byDriveChan;
BYTE                byVehicleType;
BYTE                byDetSceneID;
BYTE                byVehicleAttribute;
WORD                wIllegalType;
BYTE                byIllegalSubType[8];
BYTE                byPostPicNo;
BYTE                byChanIndex;
WORD                wSpeedLimit;
BYTE                byChanIndexEx;
BYTE                byRes2;
NET_DVR_PLATE_INFO  struPlateInfo;
NET_DVR_VEHICLE_INFO struVehicleInfo;
BYTE                byMonitoringSiteID[48];
BYTE                byDeviceID[48];
BYTE                byDir;
BYTE                byDetectType;
BYTE                byRelaLaneDirectionType;
BYTE                byCarDirectionType;
DWORD               dwCustomIllegalType;
BYTE                *pIllegalInfoBuf;
BYTE                byIllegalFromatType;
BYTE                byPendant;
BYTE                byDataAnalysis;
BYTE                byYellowLabelCar;
BYTE                byDangerousVehicles;
BYTE                byPilotSafebelt;
BYTE                byCopilotSafebelt;
BYTE                byPilotSunVisor;
BYTE                byCopilotSunVisor;
BYTE                byPilotCall;
BYTE                byBarrierGateCtrlType;
BYTE                byAlarmDataType;
NET_DVR_TIME_V50    struSnapFirstPicTime;
DWORD               dwIllegalTime;
DWORD               dwPicNum;
NET_ITS_PICTURE_INFO struPicInfo[6];
}NET_ITS_PLATE_RESULT, *LPNET_ITS_PLATE_RESULT;

```

## Members

### dwSize

Structure Size

### dwMatchNo

Match ID, which consists of vehicle No., data type, and lane No.

### byGroupNum

Number of picture groups (total number of one vehicle picture groups captured by multiple cameras), the default value is 1.

### **byPicNo**

No. of continuously captured picture (if **byPicNo** is **byGroupNum**, it indicates that the last picture is received).

### **bySecondCam**

Whether the picture is captured by the second camera.

### **byFeaturePicNo**

Intersection violation system for red light running, the No. of picture to be set as close-up picture, 0xff-not set.

### **byDriveChan**

Triggered lane No.

### **byVehicleType**

Vehicle type: 0-unknown, 1-bus, 2-truck, 3-car, 4-mini bus, 5-van, 6-pedestrian, 7-two-wheel vehicle, 8-three-wheel vehicle, 9-SUV/MPV, 10-medium bus, 11-motor vehicle, 12-non-motor vehicle, 13-mini car, 14-micro car, 15-pickup truck. Generally, the vehicle type is determined by this member. If this member is set to 0, the vehicle type is determined by the member

**byVehicleType** in the structure **NET\_DVR\_VEHICLE\_INFO** .

### **byDetSceneID**

Detection scene No., 0-invalid, other values: [1,4], for network camera, its value is 0 (not support)

### **byVehicleAttribute**

Vehicle properties, 0-no additional properties, other values: bit1-yellow label vehicle, bit2-dangouous goods vehicle, bit value: 0-no, 1-yes.

### **willlegalType**

Traffic violation type. When the value of this parameter is 0, see the traffic violation type in **dwCustomIllegalType**.

### **byIllegalSubType**

Sub type of traffic violation.

### **byPostPicNo**

The No. of picture to be set as the checkpoint picture, 0xff-not set.

### **byChanIndex**

Channel No.

### **wSpeedLimit**

Upper limit of speed (it is valid only when overspeed).

### **byChanIndexEx**

**byChanIndexEx\*256+byChanIndex** is the actual channel No.

**byRes2**

Reserved

**struPlateInfo**

License plate information, see details in the structure **NET\_DVR\_PLATE\_INFO** .

**struVehicleInfo**

Vehicle information , see details in the structure **NET\_DVR\_VEHICLE\_INFO** .

**byMonitoringSiteID**

Monitoring point ID

**byDeviceID**

Device ID

**byDir**

Detection direction: 1-backward, 2-forward, 3-bidirection, 4-from east to west, 5from south to north, 6-from west to east, 7from north to south, 8-other

**byDetectType**

Detection type: 0-vehicle detection, 1-triggered by inductive loop, 2- triggered by video detection, 3-multi-frame recognition, 4- triggered by radar, 5-mixed-traffic detection.

**byRelaLaneDirectionType**

Linked lane direction, see details in **ITC\_RELA\_LANE\_DIRECTION\_TYPE** , which corresponds to the linked lane No.

**byCarDirectionType**

Vehicle driving direction: 0-from top to bottom in image, 1- from bottom to top in image.

**dwCustomIllegalType**

Traffic violation type (custom), this parameter is valid only when **wIllegalType** is 0.

**plIllegalInfoBuf**

Violation code, it is valid when **byIllegalFromatType** is 1. The code pointer points to the structure **NET\_ITS\_ILLEGAL\_INFO** .

**byIllegalFromatType**

Traffic violation information format: 0-digital format, 1-character format

**byPendant**

Objects hanging on window?: 0-unknown, 1-yes, 2-no

**byDataAnalysis**

Data analyzed?: 0-no, 1-yes

**byYellowLabelCar**

Yellow label vehicle? 0-unknown, 1-yes, 2-no

**byDangerousVehicles**

Dangerous goods vehicle? 0-unknown, 1-yes, 2-no

**byPilotSafebelt**

Driver wearing safety belt? 0-unknown, 1-yes, 2-no

**byCopilotSafebelt**

Co-driver wearing safety belt?: 0-unknown, 1-yes, 2-no

**byPilotSunVisor**

Sun shield of co-driver room opened?: 0-unknown, 1-yes, 2-no

**byCopilotSunVisor**

Sun shield of co-driver room opened?: 0-unknown, 1-yes, 2-no

**byPilotCall**

Driver making call? 0-unknown, 1-yes, 2-no

**byBarrierGateCtrlType**

Barrier control type: 0-open, 1-close.

**byAlarmDataType**

Alarm data type: 0-real-time data, 1-history data.

**struSnapFirstPicTime**

Time of first captured picture, unit: ms, see details in the structure **NET\_DVR\_TIME\_V50** .

**dwIllegalTime**

Violation duration (unit: ms)=time of last captured picture-time of first captured picture

**dwPicNum**

Number of pictures.

**struPicInfo**

Picture information, call back one by one, and up to 6 pictures are allowed, see details in the structure **NET\_ITS\_PICTURE\_INFO** .

**Remarks**

Generally, refer to the parameter **byVehicleType** of this structure for the vehicle type. If **byVehicleType** is 0, refer to the parameter **byVehicleType** of structure **NET\_DVR\_VEHICLE\_INFO** .

## 7.1.92 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          byCompatibleAblity;
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

**byCompatibleAblity**

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&ptime=PicTime. E.g., http://10.192.65.140:8009/pic?did=35b9cbd0-8ffa-1031-87e6-0025903c6a50&bid=387&pid=2952790009&ptime=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.

## **7.1.93 NET\_POS\_PARAM**

Region parameter structure.

### **Structure Definition**

```
struct{  
    WORD    wLeft;  
    WORD    wTop;  
    WORD    wRight;  
    WORD    wBottom;  
}NET_POS_PARAM, *LPNET_POS_PARAM;
```

### **Members**

#### **wLeft**

X-coordinate of the upper-left corner of the boundary frame, value range: [1, 1000].

#### **wTop**

Y-coordinate of the upper-left corner of the boundary frame, value range: [1, 1000].

#### **wRight**

X-coordinate of the lower-right corner of the boundary frame, value range: [1, 1000].

#### **wBottom**

Y-coordinate of the lower-right corner of the boundary frame, value range: [1, 1000].

### **See Also**

**NET\_ITC\_SINGLE\_VIDEO\_DETECT\_LIGHT\_PARAM**

## **7.1.94 NET\_VCA\_LINE**

Structure about line parameters.

### **Structure Definition**

```
struct{  
    NET_VCA_POINT    struStart;  
    NET_VCA_POINT    struEnd;  
}NET_VCA_LINE, *LPNET_VCA_LINE;
```



## Members

### struStart

Start point, see details in the structure [NET\\_VCA\\_POINT](#)

### struEnd

End point, see details in the structure [NET\\_VCA\\_POINT](#) .

## 7.1.95 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.

## 7.1.96 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.

### 7.1.97 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

### 7.1.98 VCA\_PLATE\_TYPE

Enumerate the license plate types.

#### Enumeration Definition

```
enum{
    VCA_STANDARD92_PLATE    =0,
    VCA_STANDARD02_PLATE    =1,
    VCA_WJPOLICE_PLATE      =2,
    VCA_JINGCHE_PLATE        =3,
    STANDARD92_BACK_PLATE   =4,
    VCA_SHIGUAN_PLATE        =5,
    VCA_NONGYONG_PLATE       =6,
    VCA_MOTO_PLATE           =7,
    NEW_ENERGY_PLATE         =8,
    VCA_CONSULATE_PLATE      =9,
    VCA_EMERGENCY_PLATE      = 10
    TRANSPORT_PLATE          = 0x20,
    COMMERCIAL_PLATE,
    PRIVATE_PLATE,
    LEARNING_PLATE,
    CD_PLATE,
    CC_PLATE,
    ARMY_PLATE,
    PROTOCOL_PLATE,
    GOVERNMENT_PLATE,
    EXPORT_PLATE,
    TAXI_PLATE,
    TESTING_PLATE,
    TRANSFER_PLATE,
    TRUCK_PLATE,
    BUS_PLATE,
    PUBLIC_PLATE,
    PUB_TRANS_PLATE,
    PRI_TRANS_PLATE,
    UNKNOWN_PLATE           = 0xff
}VCA_PLATE_TYPE
```

#### Member

##### VCA\_STANDARD92\_PLATE

License plate of civil and military vehicle.

##### VCA\_STANDARD02\_PLATE

License plate of civil vehicle

##### VCA\_WJPOLICE\_PLATE

License plate of armed police vehicle.

**VCA\_JINGCHE\_PLATE**

License plate of police vehicle.

**STANDARD92\_BACK\_PLATE**

Tail board of civil vehicle.

**VCA\_SHIGUAN\_PLATE**

License plate of embassy vehicle.

**VCA\_NONGYONG\_PLATE**

License plate of farm vehicle.

**VCA\_MOTO\_PLATE**

License plate of motor vehicle.

**NEW\_ENERGY\_PLATE**

License plate of new energy vehicle.

**VCA\_CONSULATE\_PLATE**

License plate of consulate.

**VCA\_EMERGENCY\_PLATE**

License plate of emergent vehicle.

**TRANSPORT\_PLATE**

License plate of carrier vehicle.

**COMMERCIAL\_PLATE**

License plate of commercial vehicle.

**PRIVATE\_PLATE**

License plate of private car.

**LEARNING\_PLATE**

License plate of driver-training vehicle.

**CD\_PLATE**

License plate of embassy vehicle.

**CC\_PLATE**

License plate of embassy vehicle.

**ARMY\_PLATE**

License plate of military vehicle.

**PROTOCOL\_PLATE**

Protocol license plate.

**GOVERNMENT\_PLATE**

License plate of government vehicle.

**EXPORT\_PLATE**

Export license plate.

**TAXI\_PLATE,**

Taxi license plate.

**TESTING\_PLATE**

Test license plate.

**TRANSFER\_PLATE**

Transfer license plate.

**TRUCK\_PLATE**

Truck license plate.

**BUS\_PLATE**

Bus license plate.

**PUBLIC\_PLATE**

Public license plate.

**PUB\_TRANS\_PLATE**

Public transfer license plate.

**PRI\_TRANS\_PLATE**

Private transfer license plate.

**UNKNOWN\_PLATE**

Unknown (unrecognized).

### 7.1.99 DATE\_TIME

#### Date and Time Structure

Member	Data Type	Description
<b>year</b>	short	Year.
<b>month</b>	short	Month.
<b>dayOfWeek</b>	short	Days of the week: 0-Sunday, 1-Monday, 2-Tuesday, 3-Wednesday, 4-Thursday, 5-Friday, 6-Saturday.
<b>day</b>	short	Day.
<b>hour</b>	short	Hour.

Member	Data Type	Description
<b>minute</b>	short	Minute.
<b>second</b>	short	Second.
<b>milliSecond</b>	short	Millisecond.

### 7.1.100 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.

Member	Data Type	Description
		<ul style="list-style-type: none"> <li>bySupport&amp;0x1: whether supports VCA search.</li> <li>bySupport&amp;0x2: whether supports backup.</li> <li>bySupport&amp;0x4: whether supports getting encoding parameters.</li> <li>bySupport&amp;0x8: whether supports dual-NIC.</li> <li>bySupport&amp;0x10: whether supports remote SADP.</li> <li>bySupport&amp;0x20: whether supports RAID card.</li> <li>bySupport&amp;0x40: whether supports searching in IPSAN directory.</li> <li>bySupport&amp;0x80: whether supports RTP over RTSP.</li> </ul>
bySupport1	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> <li>bySupport1&amp;0x1: whether supports SNMP with version 30.</li> <li>bySupport1&amp;0x2: whether supports playback and downloading video files.</li> <li>bySupport1&amp;0x4: whether supports setting the arming priority.</li> <li>bySupport1&amp;0x8: whether supports extending the arming time period.</li> <li>bySupport1&amp;0x10: whether supports multiple HDDs (more than 33).</li> <li>bySupport1&amp;0x20: whether supports RTP over RTSP.</li> <li>bySupport1&amp;0x80: whether supports license plate recognition alarm.</li> </ul>
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>

Member	Data Type	Description
		<ul style="list-style-type: none"> <li>bySupport2&amp;0x1: whether supports getting stream via URL.</li> <li>bySupport2&amp;0x2: whether supports FTP with version 40.</li> <li>bySupport2&amp;0x4: whether supports ANR.</li> <li>bySupport2&amp;0x20: whether supports getting device status.</li> <li>bySupport2&amp;0x40: whether supports encrypting stream.</li> </ul>
wDevType	WORD	Device model
bySupport3	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, while, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> <li>bySupport3&amp;0x1: whether supports multi-stream.</li> <li>bySupport3&amp;0x4: whether supports configuring by group (e.g., image, alarm input, alarm output, user, device status, JPEG picture capture, continuous and scheduled capture, .HDD group management, and so on).</li> <li>bySupport3&amp;0x20: whether supports getting stream via DDNS.</li> </ul>
byMultiStreamProto	BYTE	<p>Whether supports multi-stream, if the result of bitwise operation is 0, it refers to not support, if the result is 1, it refers to support.</p> <ul style="list-style-type: none"> <li>byMultiStreamProto&amp;0x1: whether supports third-stream.</li> <li>byMultiStreamProto&amp;0x2: whether supports fourth-stream.</li> <li>byMultiStreamProto&amp;0x40: whether supports main stream.</li> <li>byMultiStreamProto&amp;0x80: whether supports sub-stream.</li> </ul>
byStartDChan	BYTE	Start No. of digital channel, 0-no digital channel (e.g., DVR, network camera).



Member	Data Type	Description
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 supported, if the result is 1, it indicates that the capability is supported. <ul style="list-style-type: none"> <li>bySupport4&amp;0x01: whether all stream types support RTSP and private protocol.</li> <li>bySupport4&amp;0x02: whether the device supports transmitting form format data via API (NET_DVR_STDXMLConfig).</li> <li>bySupport4&amp;0x10: whether supports loading network disk by domain name.</li> </ul>
byLanguageType	BYTE	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. <ul style="list-style-type: none"> <li>byLanguageType ==0: this field is not supported by device.</li> <li>byLanguageType&amp;0x1: whether supports Chinese.</li> <li>byLanguageType&amp;0x2: whether supports English.</li> </ul>
byVoiceInChanNum	BYTE	Number of audio input channels
byStartVoiceInChanNo	BYTE	Start No. of audio input channel, 0-invalid.
byRes3	Array of BYTE	Reserved, set to 0.
byMirrorChanNum	BYTE	Number of mirror channels
wStartMirrorChanNo	WORD	Start No. of mirror channel
byRes2	Array of BYTE	Reserved, set to 0.

## Remarks

- The maximum number of digital channels equal to  $\text{byIPChanNum} + \text{byHighDChanNum} * 256$ .
- 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****7.1.101 NET\_DVR\_DEVICEINFO\_V40****Device Parameter Structure (V40)**

Member	Data Type	Description
struDeviceV30	<u><b>NET_DVR_DEVICEINFO_V30</b></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 <b>bySupportLock</b> is 1.
byPasswordLevel	BYTE	Password strength: 0-invalid, 1-default password, 2-valid password, 3-risky password. For default password or risky password, the users are reminded to change password.
byProxyType	BYTE	Proxy type: 0-no proxy, 1-standard proxy, 2-EHome proxy.
dwSurplusLockTime	DWORD	Remaining locking time, unit: second. It is valid only when <b>bySupportLock</b> is 1. During the locking time, if the user try to log in to again, the remaining locking time will resume to 30 minutes.
byCharEncodeType	BYTE	Character encodings. 0-no decoding information, 1-GB2312 (Simplified Chinese), 2-GBK, 3-BIG5 (Traditional Chinese), 4-Shift_JIS (Japanese), 5-EUC-KR (Korean), 6-UTF-8, 7-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.

Member	Data Type	Description
bySupport	BYTE	Whether it supports uploading changes, it depends on the result of bitwise AND (&) operation: 0-not support, 1-support. The result of <b>bySupport&amp;0x1</b> indicates that this member is reserved; the result of <b>bySupport&amp;0x2</b> indicates that whether it supports uploading changes: 0-not support, 1-support. This member is the capability set extension.
byLoginMode	BYTE	Login mode: 0-login via private protocol, 1-login via text protocol. For private protocol, the default login port number is 8000, and for text protocol, the default login port number is 80 or 443.
dwOEMCode	DWORD	OEM code.
iResidualValidity	int	Remaining valid days of the user's password, unit: day. If the negative number is returned, it indicates that the password being used has expired. For example, if -3 is returned, it indicates that the password being used has expired for three days.
byResidualValidity	BYTE	Whether the member <b>iResidualValidity</b> is valid: 0-invalid, 1-valid.
bySingleStartDTalkChan	BYTE	Start channel No. for connecting independent audio tracks to the device. The value 0 is reserved and invalid. The channel No. of audio tracks cannot start from 0.
bySingleDTalkChanNums	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

Member	Data Type	Description
		has been changed by the administrator, the non-admin user will be prompted "Please set a new password" each time he/she logs in to the device until he/she changes the password).
bySupportStreamEncrypt	BYTE	Whether it supports stream encryption, it depends on the result of bitwise AND (&) operation: 0-no, 1-yes. The result of <b>bySupportStreamEncrypt&amp;0x1</b> indicates whether to support RTP/TLS streaming, the result of <b>bySupportStreamEncrypt&amp;0x2</b> indicates whether to support SRTP/UDP streaming, and the result of <b>bySupportStreamEncrypt&amp;0x4</b> indicates whether to support SRTP/MULTICAST streaming.
byRes2	Array of BYTE	Reserved, set to 0.

### Remarks

- Four character types are allowed in the password, including digits, lowercase letters, uppercase letters and symbols. The maximum password length is 16 bits, and there are four password strength levels, see details below:
  - Level 0 (Risky Password): The password length is less than 8 bits, or only contains one kind of the character types. Or the password is the same with the user name, or is the mirror writing of the user name.
  - Level 1 (Weak Password): The password length is more than or equal to 8 bits, and contains two kinds of the character types. Meanwhile, the combination should be (digits + lowercase letters) or (digits + uppercase letters).
  - Level 2 (Medium Password): The password length is more than or equal to 8 bits, and contains two kinds of the character types. Meanwhile, the combination cannot be (digits + lowercase letters) and (digits + uppercase letters).
  - Level 3 (Strong Password): The password length is more than or equal to 8 bits, and at least contains three kinds of the character types.
- For login via text protocol, the following parameters are not supported: **bySupportLock**, **byRetryLoginTime**, **byPasswordLevel**, **byProxyType**, **dwSurplusLockTime**, **byCharEncodeType**, and **bySupportDev5**.

### 7.1.102 NET\_DVR\_INIT\_CFG\_ABILITY

## Initialization Capability Structure

Member	Data Type	Description
enumMaxLoginUsersNum	INIT_CFG_MAX_NUM	Maximum number of users can log in, see details below:  <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	Maximum number of alarm channels, see details below:  <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](#)

## 7.1.103 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.

### 7.1.104 NET\_DVR\_USER\_LOGIN\_INFO

#### Structure About Login Parameters

Member	Data Type	Description
sDeviceAddress	char	Device IP address, or domain name.
byUseTransport	BYTE	Enable capability transmission or not: 0-no (default), 1-yes.
wPort	WORD	Device port number, e.g., 8000 (when login by private protocol), 80 (when login by text protocol).
sUserName	char	User name for logging in to device.
sPassword	char	Login password.
cbLoginResult	<b><i><u>fLoginResultCallback</u></i></b>	Callback function used to return login status, it is valid only when <b>bUseAsynLogin</b> is "1".
pUser	void*	User data.
bUseAsynLogin	BOOL	Whether to enable asynchronous login: 0-no, 1-yes.
byProxyType	BYTE	Proxy server type: 0-no proxy, 1-standard proxy, 2-EHome proxy.
byUseUTCTime	BYTE	0-not convert (default), 1-input or output UTC time, 2-input or output local time.
byLoginMode	BYTE	Login mode: 0-login by private protocol, 1-login by text protocol, 2-self-adaptive (it is available when the protocol type supported by device is 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

Member	Data Type	Description
		protocol type supported by device is unknown. Both HTTP and HTTPS requests will be sent).
iProxyID	LONG	Proxy server No.
byVerifyMode	BYTE	Whether to enable verification mode: 0-no, 1-bidirectional verification (currently not available), 2-unidirectional verification (it is valid when <b>byLoginMode</b> is 0 and <b>byHttps</b> is 1); when <b>byVerifyMode</b> is 0, CA certificate is not required, when <b>byVerifyMode</b> is 2, you should call NET_DVR_SetSDKLocalCfg to load CA certificate, and the enumeration value is "NET_SDK_LOCAL_CFG_CERTIFICATION".
byRes3	BYTE[]	Reserved, the maximum length is 119 bytes.

#### 7.1.105 NET\_SDK\_CALLBACK\_STATUS\_NORMAL

##### Enumeration About Persistent Connection Status

Enumeration Type	Marco Definition Value	Description
NET_SDK_CALLBACK_STATUS_SUCCESS	1000	Succeeded.
NET_SDK_CALLBACK_STATUS_PROCESSING	1001	Connecting. The <b>lpBuffer</b> is 4-byte status.
NET_SDK_CALLBACK_STATUS_FAILED	1002	Failed. The <b>lpBuffer</b> is the value of 4-byte status and 4-byte error code.

#### 7.1.106 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.

### 7.1.107 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	<b><u>NET_ALARM_RECORD_FILE_LOSS</u></b>	Video loss alarm information, the value of <b>dwAlarmType</b> in <b><u>NET_DVR_ALARMINFO_DEV_V40</u></b> is 8.
struStreamException	<b><u>NET_ALARM_STREAM_EXCEPTION</u></b>	Streaming exception alarm information, the value of <b>dwAlarmType</b> in <b><u>NET_DVR_ALARMINFO_DEV_V40</u></b> is 9.
struResourceUsage	<b><u>NET_ALARM_RESOURCE_USAGE</u></b>	Resource usage alarm information, the value of <b>dwAlarmType</b> in <b><u>NET_DVR_ALARMINFO_DEV_V40</u></b> is 10.
struRecordException	<b><u>NET_ALARM_RECORD_EXCEPTION</u></b>	Recording exception alarm information, the value of <b>dwAlarmType</b> in <b><u>NET_DVR_ALARMINFO_DEV_V40</u></b> is 12.

### 7.1.108 NET\_ALARM\_RECORD\_EXCEPTION



**Structure about Recording Exception Alarm Information**

Member	Data Type	Description
<b>byReason</b>	BYTE	Exception reason: 0-video volume full, 1-video volume exception, 2-no available video volume.
<b>byRes1</b>	BYTE[]	Reserved, set to 0. The maximum array length is 3 bytes.
<b>sVolumeName</b>	BYTE[]	Video volume name, the maximum array length is "MAX_VOLUMENAME_LEN" (32 bytes).
<b>dwVolumeID</b>	DWORD	Video volume ID, or HDD No.
<b>byRes</b>	BYTE[]	Reserved, set to 0. The maximum array length is 452 bytes.

**7.1.109 NET\_ALARM\_RECORDFILE\_LOSS****Structure about Video Loss Alarm Information**

Member	Data Type	Description
<b>struInspectStart</b>	<b><u>NET_DVR_TIME_EX</u></b>	Start time of video loss check.
<b>struInspectEnd</b>	<b><u>NET_DVR_TIME_EX</u></b>	End time of video loss check.
<b>struIP</b>	<b><u>NET_DVR_IPADDR_UNION</u></b>	IP address of video loss channel.
<b>dwChanNo</b>	DWORD	Channel No.
<b>dwIDIndex</b>	DWORD	Encoder ID.
<b>sName</b>	BYTE[]	Encoder name, the maximum array length is "STREAM_ID_LEN" (32 bytes).
<b>struLossStartTime</b>	<b><u>NET_DVR_TIME_EX</u></b>	Start time of video loss.
<b>struLossEndTime</b>	<b><u>NET_DVR_TIME_EX</u></b>	End time of video loss.
<b>dwLostNum</b>	DWORD	Number of lost video files, 0xffffffff-all video files are lost.
<b>byRes</b>	BYTE[]	Reserved, set to 0. The maximum array length is 240 bytes.

### 7.1.110 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.

### 7.1.111 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).
byExceptionCase	BYTE	Exception reason: 0-data writing exception, 1-network exception.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 307 bytes.

### 7.1.112 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

Member	Data Type	Description
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 <b><u>NET DVR Login V40</u></b> , 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.

### 7.1.113 NET\_DVR\_ALARMINFO\_DEV

#### Device Alarm Information Structure

Memeber	Data Type	Description
<b>dwAlarmType</b>	DWORD	Alarm types: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection

Member	Data Type	Description
		alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel.
<b>struTime</b>		Alarm time
<b>byRes</b>	Array of BYTE	Reserved, set to 0.
<b>dwNumber</b>	DWORD	Number of alarm triggered channels.
<b>pNO</b>	WORD*	Channel No. or disk No., which ranges from 0 to 65535.

### Remarks

For **pNO**: if **dwAlarmType** is 0, 3, 6, or 7, it may be channel No.; if **dwAlarmType** is 5, it may be disk No.

## 7.1.114 NET\_DVR\_ALARMINFO\_DEV\_V40

### Structure about CVR Alarm Information

Member	Data Type	Description
<b>dwAlarmType</b>	DWORD	Alarm categories: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel, 8-video loss alarm, 9-real-time health monitoring alarm, 10-usage alarm, 11-CVR exception recovered, 12-recording exception.
<b>struTime</b>	<u>NET_DVR_TIME</u>	Alarm time
<b>uSubAlarmInfo</b>	<u>NET_ALARM_CVR_SUBINFO_UNION</u>	CVR alarm information structure, and it is valid when the alarm type is 8, 9, 10, and 12.
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 256 bytes.

Member	Data Type	Description
<b>dwNumber</b>	DWORD	Number of alarm triggered channels.
<b>pNO</b>	WORD*	Channel No. or disk No., which ranges from 0 to 65535.

### Remarks

For **pNO**: if **dwAlarmType** is 0, 3, 6, or 7, it may be channel No.; if **dwAlarmType** is 5, it may be disk No.

## 7.1.115 NET\_DVR\_ALARMINFO\_V30

### Structure About Uploaded Alarm Information

Member	Data Type	Description
<b>dwAlarmType</b>	DWORD	Alarm types: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel, 8-video loss alarm, 9-real-time health monitoring alarm, 10-usage alarm, 11-CVR exception recovered, 12-recording exception.
<b>dwAlarmInputNumber</b>	DWORD	Alarm input No., it is valid when alarm type is 0 or 23
<b>byAlarmOutputNumber</b>	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.
<b>byAlarmRelateChannel</b>	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.

Member	Data Type	Description
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.

## 7.1.116 NET\_DVR\_ALARMINFO\_V40

### Structure About Uploaded Alarm Information

Member	Data Type	Description
struAlarmFixedHeader	<b><u>NET_DVR_ALARM_FIXED_HEADER</u></b>	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\_ALARM\_FIXED\_HEADER** , see details in the table below:

**Table 7-1 Relations Between pAlarmData and dwAlarmType**

dwAlarmType	Description	pAlarmData
0, 23	Alarm input alarm, pulse alarm	dwTrigerAlarmOutNum*(DWORD) Alarm output No.,

dwAlarmType	Description	pAlarmData
		+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

### 7.1.117 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

Member	Data Type	Description
		exception, 30-no SD card, 31-supply voltage exception, 32-PTZ locked
struAlarmTime	<b><u>NET DVR TIME EX</u></b>	Alarm time
uStruAlarm	Union ( <b><u>Table 7-2</u></b> )	Alarm information union
pRes	DWORD*	Reserved.
byTimeDiffFlag	BYTE	Whether the time difference parameter is valid: 0-invalid, 1-valid.
cTimeDifferenceH	char	Time difference between time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when <b>byISO8601</b> is "1".
cTimeDifferenceM	char	Time difference between time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when <b>byISO8601</b> is "1".
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 5 bytes.

**Table 7-2 Union about Alarm Information Structures (uStruAlarm)**

Member	Data Type	Description
byUnionLen	Array of BYTE	Union size, which is 116 bytes.
struIOAlarm	Struct ( <b><u>Table 7-3</u></b> )	Structure about alarm input parameters
struAlarmChannel	Struct ( <b><u>Table 7-4</u></b> )	Structure about alarm channel parameters
struAlarmHardDisk	Struct ( <b><u>Table 7-5</u></b> )	Structure about HDD alarm parameters
struRecordingHost	Struct ( <b><u>Table 7-6</u></b> )	Structure about alarm parameters of education sharing system
struVoltageInstable	Struct ( <b><u>Table 7-7</u></b> )	Structure about alarm parameters of supply voltage exception
struPTLocking	Struct ( <b><u>Table 7-8</u></b> )	Structure about parameters of PTZ locked alarm



**Table 7-3 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 <b>pAlarmData</b> in <b><u>NET_DVR_ALARMINFO_V40</u></b> , each alarm output is represented by 4 bytes.
dwTrigerRecordChanNum	DWORD	The number of triggered recording channels. It is used for calculating the number of all triggered recording channels by <b>pAlarmData</b> of <b><u>NET_DVR_ALARMINFO_V40</u></b> , each channel is represented by 4 bytes.

**Table 7-4 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 <b>pAlarmData</b> of <b><u>NET_DVR_ALARMINFO_V40</u></b> , each alarm channel is represented by 4 bytes.
dwPicLen	DWORD	Size of JPEG picture.
byPicURL	BYTE	Picture data format: 0-binary data, 1-URL.
byTarget	BYTE	Detection target type: 0-not supported, 1-person, 2-vehicle.
byRes1	Array of BYTE	Reserved, the maximum size is 2 bytes.
pDataBuff	char*	Alarm picture data or URL. The pointer size is 8 bytes.
byRes3	Array of BYTE	Reserved, the maximum size is 4 bytes. This member is only available for 64-bit Window operating system and 64-bit Linux operating system.

**Table 7-5 Structure about HDD Alarm Parameters (struAlarmHardDisk)**

Member	Data Type	Description
dwAlarmHardDiskNum	DWORD	The number of alarm HDD. It is used for calculating the number of all alarm HDDs by <b>pAlarmData</b> of <b>NET DVR ALARMINFO V40</b> , each alarm HDD is represented by 4 bytes.

**Table 7-6 Structure about Alarm Parameters of Education Sharing System (struRecordingHost)**

Member	Data Type	Description
bySubAlarmType	BYTE	Alarm minor type: 1-one-touch post-record
byRes1	Array of BYTE	Reserved, set to 0. The maximum size is 3 bytes.
struRecordEndTime	<b>NET DVR TIME EX</b>	Recording end time.

**Table 7-7 Structure about Alarm Parameters of Supply Voltage Exception (struVoltageInstable)**

Member	Data Type	Description
fVoltageValue	float	Supply voltage, unit: V, corrects to one decimal place.
byVoltageAlarmType	BYTE	Supply voltage exception type: 0-high supply voltage, 1-low supply voltage
byRes1	Array of BYTE	Reserved, set to 0. The maximum size is 3 bytes.

**Table 7-8 Structure about Parameters of PTZ Locked Alarm (struPTLocking)**

Member	Data Type	Description
fTemperature	float	Sensor temperature, which is accurate to one decimal place.
dwCustomInfoLength	DWORD	Custom information length.
pCustomInfo	BYTE*	Custom information.
byType	BYTE	PTZ locked direction: 1-panning is locked, 2-tilting is locked.
byDeicingEnabled	BYTE	Whether to enable heat for PTZ: 0-no, 1-yes.

## Remarks

**dwAlarmType**==0, 23 corresponds to the structure **struIOAlarm**; **dwAlarmType**==2/3/6/9/10/11/13/15/16/28 corresponds to the structure **struAlarmChannel**; **dwAlarmType**==

1/4/5 corresponds to the structure `struAlarmHardDisk`; `dwAlarmType== 17` corresponds to the structure `struRecordingHost`; `dwAlarmType== 31` corresponds to the structure `struVoltageInstable`; for other value, the union is not available.

### 7.1.118 NET\_DVR\_ALARM\_ISAPI\_INFO

#### Structure about Alarm Information Transmitted Based on Text Protocol

Member	Data Type	Description
<b>pAlarmData</b>	char*	Alarm information based on text protocol (XML or JSON message without binary data).
<b>dwAlarmDataLen</b>	DWORD	Alarm data length.
<b>byDataType</b>	BYTE	Alarm data type: 0-invalid, 1-XML, 2-JSON.
<b>byPicturesNumber</b>	BYTE	The number of pictures (number of <b>pPicPackData</b> returned). When this member is 1, only one structure of <b><u>NET_DVR_ALARM_ISAPI_PICDATA</u></b> will be returned by <b>pPicPackData</b> . When this member is larger than 1, multiple structures of <b><u>NET_DVR_ALARM_ISAPI_PICDATA</u></b> will be returned by <b>pPicPackData</b> .
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 2 bytes.
<b>pPicPackData</b>	void*	Alarm picture structure, see <b><u>NET_DVR_ALARM_ISAPI_PICDATA</u></b> for details.
<b>byRes</b>	Array of BYTE	Reserved. The maximum size is 32 bytes.

#### Remarks

When enabling the listening mode, you should call the network configuration API based on text protocol to set the IP address for the listening service.

### 7.1.119 NET\_DVR\_ALARM\_ISAPI\_PICDATA

#### Structure about Alarm Picture Data Transmitted Based on Text Protocol

Member	Data Type	Description
<b>dwPicLen</b>	DWORD	Alarm picture data length.
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 4 bytes.
<b>szFilename</b>	Array of char	Picture file saving path, including file name. The maximum size is 256 bytes.
<b>pPicData</b>	BYTE*	Pointer that pointing to the uploaded image data.

### 7.1.120 NET\_DVR\_ETHERNET\_V30

#### Ethernet Configuration Structure

Member	Data Type	Description
struDVRIP	<u><b>NET_DVR_IPADDR_UNION</b></u>	Device IP address
struDVRIPMask	<u><b>NET_DVR_IPADDR_UNION</b></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.

### 7.1.121 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.

### 7.1.122 NET\_DVR\_NETCFG\_V50

#### Network Configuration Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
struEtherNet	Array of <u><b>NET_DVR_ETHERNET_V30</b></u>	Ethernet interface
struRes1	Array of	Reserved, set to 0.
struAlarmHostIpAddr	<u><b>NET_DVR_IPADDR_UNION</b></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><b>NET_DVR_IPADDR_UNION</b></u>	IP address of domain name server 1
struDnsServer2IpAddr	<u><b>NET_DVR_IPADDR_UNION</b></u>	IP address of domain name server 2

Member	Data Type	Description
byIpResolver	Array of BYTE	IP resolver domain name or IP address (if the port No. of device is 8000, the domain name is not supported).
wIpResolverPort	WORD	IP resolver port No.
wHttpPortNo	WORD	HTTP port No.
struMulticastIpAddr	<b><u>NET_DVR_IPADDR_UNION</u></b>	Multicast group address
struGatewayIpAddr	<b><u>NET_DVR_IPADDR_UNION</u></b>	Gateway address
struPPPoE	<b><u>NET_DVR_PPPOECFG</u></b>	PPPoE parameters
byEnablePrivateMulticastDiscovery	BYTE	Private multicast search (SADP): 0-default, 1-enable, 2-disable
byEnableOnvifMulticastDiscovery	BYTE	Onvif multicast search (SADP): 0-default, 1-enable, 2-disable
wAlarmHost2IpPort	WORD	Port No. of listening host 2.
struAlarmHost2IpAddr	<b><u>NET_DVR_IPADDR_UNION</u></b>	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**.

### 7.1.123 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	<b><u>NET_DVR_IPADDR_UNION</u></b>	PPPoE IP address

### 7.1.124 NET\_DVR\_SETUPALARM\_PARAM\_V50

#### Arming Parameter Structure

Member	Data Type	Description
<b>dwSize</b>	DWORD	Structure size.
<b>byLevel</b>	BYTE	Arming priority: 0-high, 1-medium, 2-low.
<b>byAlarmInfoType</b>	BYTE	Intelligent traffic alarm information type: 0-old (NET_DVR_PLATE_RESULT),1-new (NET_ITS_PLATE_RESULT).
<b>byRetAlarmTypeV40</b>	BYTE	0-the motion detection, video loss, video tampering, and alarm input alarm information is uploaded in normal mode (alarm type: COMM_ALARM_V30, alarm information structure: <b><u>NET_DVR_ALARMINFO_V30</u></b> ); 1-alarm information is uploaded in variable size (alarm type: COMM_ALARM_V40, alarm information structure: <b><u>NET_DVR_ALARMINFO_V40</u></b> ).
<b>byRetDevInfoVersion</b>	BYTE	Alarm types of CVR: 0-COMM_ALARM_DEVICE (alarm information structure: <b><u>NET_DVR_ALARMINFO_DEV</u></b> ), 1-COMM_ALARM_DEVICE_V40 (alarm information structure: <b><u>NET_DVR_ALARMINFO_DEV_V40</u></b> ).
<b>byRetVQDAlarmType</b>	BYTE	VQD alarm types: 0-COMM_ALARM_VQD (alarm information structure: NET_DVR_VQD_

Member	Data Type	Description
		DIAGNOSE_INFO), 1-COMM_ALARM_VQD_EX (alarm information structure: NET_DVR_VQD_ALARM, including camera information and captured pictures)
<b>byFaceAlarmDetection</b>	BYTE	Face detection alarm types: 1-face detection alarm (alarm type: COMM_ALARM_FACE_DETECTION, alarm information structure: NET_DVR_FACE_DETECTION), 0-face capture alarm (alarm type: COMM_UPLOAD_FACESNAP_RESULT, alarm information structure: NET_VCA_FACESNAP_RESULT).
<b>bySupport</b>	BYTE	Capabilities, which is represented by bit: <ul style="list-style-type: none"> <li>• bit0-whether to upload picture: 0=yes, 1=no</li> <li>• bit1-whether to enable ANR: 0=no, 1=yes</li> <li>• bit4-whether to upload behavior analysis events of all detection targets: 0=no, 1=yes. It is used to enable the NVR to get events of all targets detected by network cameras.</li> <li>• bit5-whether to enable all-day event or alarm uploading: 0=no, 1=yes. It is used to enable the NVR to receive all alarms from network cameras.</li> </ul>
<b>byBrokenNetHttp</b>	BYTE	ANR type, which is represented by bit and should be supported by device: <ul style="list-style-type: none"> <li>• bit0-whether to enable ANR for ANPR: 0=no, 1=yes.</li> <li>• bit1-whether to enable ANR for people counting: 0=no, 1=yes.</li> <li>• bit2-whether to enable ANR for heat map: 0=no, 1=yes.</li> <li>• bit3-whether to enable ANR for face capture: 0=no, 1=yes.</li> <li>• bit4-whether to enable ANR for face picture comparison: 0=no, 1=yes.</li> <li>• bit5-whether to enable ANR for JSON message transmission: 0=no, 1=yes.</li> </ul>



Member	Data Type	Description
		<ul style="list-style-type: none"> <li>bit6: whether to enable ANR for uploading heat map data by dwell time duration and by people quantity: 0-no, 1-yes.</li> <li>bit7: whether to enable ANR for uploading intersection analysis result: 0-no, 1-yes.</li> </ul>
<b>wTaskNo</b>	BYTE	Task No.
<b>byDeployType</b>	BYTE	Arming type: 0-arm via client software, 1-real-time arming.
<b>bySubScripton</b>	BYTE	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.
<b>byRes1</b>	Array [BYTE]	Reserved, set to 0. The maximum size is 2 bytes.
<b>byAlarmTypeURL</b>	BYTE	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. <ul style="list-style-type: none"> <li>bit0-type of captured face pictures: 0-binary data, 1-URL</li> <li>bit1-type of picture uploaded in message: 0-binary, 1-URL</li> <li>bit2-type of picture uploaded for face picture comparison: 0-binary, 1-URL</li> </ul>
<b>byCustomCtrl</b>	BYTE	Custom control type, which is represented by bit, bit0-whether to upload the face thumbnail of the front passenger: 0-no, 1-yes
<b>byRes4</b>	Array [BYTE]	Reserved, set to 0. The maximum size is 128 bytes.

## Remarks

- The parameters **byLevel** and **byAlarmInfoType** are available for traffic cameras. Up to 1 cameras can be armed in the priority of level 0, up to 3 cameras can be armed in the priority of level 1, and up to 5 cameras can be armed in the priority of level 3, the alarm/event information from the camera in highest priority will be uploaded first.
- For arming via client software, only supports arming one channel, and supports uploading the alarm/event when device is offline; for real-time arming, up to four channels can be armed at same time, but uploading alarm/event when device is offline is not supported.
- The parameter **wTaskNo** is used to distinguish different arming connections. If the value of this parameter in different arming connections is same, error will be returned.

### 7.1.125 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

### 7.1.126 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.

## 7.2 Enumeration

### 7.2.1 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_SRB           = 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_LAO           = 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\_NEP**

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\_LAO**

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.

## **7.2.2 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_SRB	= 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_NEP	= 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_LAO      = 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_SAH          = 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\_NEP**

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\_LAO**

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\_AZO**

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)

## **7.2.3 NET\_SDK\_DOWNLOAD\_TYPE**

Enumerate file types to be downloaded.

### **Enumeration Definition**

```
typedef enum {  
    NET_SDK_DOWNLOAD_CERT                = 0,  
    NET_SDK_DOWNLOAD_IPC_CFG_FILE        = 1,  
    NET_SDK_DOWNLOAD_BASELINE_SCENE_PIC  = 2,  
    NET_SDK_DOWNLOAD_VQD_ALARM_PIC       = 3,  
    NET_SDK_DOWNLOAD_CONFIGURATION_FILE   = 4,  
    NET_SDK_DOWNLOAD_SCENE_CONFIGURATION_FILE = 5,  
    NET_SDK_DOWNLOAD_FILE_FORM_DB        = 6,  
    NET_SDK_DOWNLOAD_TME_FILE            = 7,  
}
```

```
NET_SDK_DOWNLOAD_VEHICLE_BLOCKALLOWLIST_FILE      = 8,
NET_SDK_DOWNLOAD_GUID_FILE                          = 9,
NET_SDK_DOWNLOAD_FILE_FORM_CLOUD                   = 10,
NET_SDK_DOWNLOAD_PICTURE                           = 11,
NET_SDK_DOWNLOAD_VIDEO                             = 12,
NET_DVR_DOWNLOAD_SCREEN_FILE                       = 13,
NET_SDK_DOWNLOAD_PUBLISH_MATERIAL                  = 14,
NET_SDK_DOWNLOAD_THERMOMETRIC_FILE                 = 15,
NET_SDK_DOWNLOAD_LED_CHECK_FILE                    = 16,
NET_SDK_DOWNLOAD_VEHICLE_INFORMATION               = 17,
NET_SDK_DOWNLOAD_CERTIFICATE_BLOCKLIST_TEMPLATE    = 18,
NET_SDK_DOWNLOAD_LOG_FILE                          = 19,
NET_SDK_DOWNLOAD_FILEVOLUME_DATA                   = 20,
NET_SDK_DOWNLOAD_FD_DATA                           = 21,
NET_SDK_DOWNLOAD_SECURITY_CFG_FILE                 = 22,
NET_SDK_DOWNLOAD_PUBLISH_SCHEDULE                  = 23,
NET_SDK_DOWNLOAD_RIGHT_CONTROLLER_AUDIO            = 24,
NET_SDK_DOWNLOAD_MODBUS_CFG_FILE                   = 25,
NET_SDK_DOWNLOAD_RS485_PROTOCOL_DLL_FILE           = 26,
NET_SDK_DOWNLOAD_CLUSTER_MAINTENANCE_LOG           = 27,
NET_SDK_DOWNLOAD_SQL_ARCHIVE_FILE                  = 28,
NET_SDK_DOWNLOAD_SUBWIND_STREAM                    = 29,
NET_SDK_DOWNLOAD_DEVTYPE_CALIBFILE                 = 30,
NET_SDK_DOWNLOAD_HD_CAMERA_CORRECT_TABLE           = 31,
NET_SDK_DOWNLOAD_CLIENT_CALIBFILE                  = 32,
NET_SDK_DOWNLOAD_FOUE_CAMERAS_PICTURES             = 33,
NET_SDK_DOWNLOAD_DOOR_CONTENT                      = 34,
NET_SDK_DOWNLOAD_PUBLISH_MATERIAL_THUMBNAIL        = 35,
NET_SDK_DOWNLOAD_PUBLISH_PROGRAM_THUMBNAIL         = 36,
NET_SDK_DOWNLOAD_PUBLISH_TEMPLATE_THUMBNAIL        = 37,
NET_SDK_DOWNLOAD_DARK_FIGHTER_X_CORRECT_TABLE_MAIN = 38,
NET_SDK_DOWNLOAD_DARK_FIGHTER_X_CORRECT_TABLE_BACKUP = 39,
NET_SDK_DOWNLOAD_OFFLINE_CAPTURE_INFO_TEMPLATE     = 40,
NET_SDK_DOWNLOAD_CAPTURE_DATA                      = 41,
NET_SDK_DOWNLOAD_HD_CAMERA_CORRECT_TABLE_FILE      = 42,
NET_SDK_DOWNLOAD_CLIENT_CALIBFILE_FILE             = 43,
NET_SDK_DOWNLOAD_FOUR_CAMERAS_PICTURES_FILE        = 44,
NET_SDK_DOWNLOAD_SCENE_FILE                        = 45,
NET_SDK_DOWNLOAD_OPEN_SOURCE_CERT                  = 46,
NET_SDK_DOWNLOAD_RATIOSTITCHING_FILE               = 47,
NET_SDK_DOWNLOAD_LENS_PARAM_FILE                   = 48,
NET_SDK_DOWNLOAD_SELECT_DEVTYPE_CALIBFILE          = 49
} NET_SDK_DOWNLOAD_TYPE;
```

### Enumeration Type

#### **NET\_SDK\_DOWNLOAD\_CERT**

Certificate.

#### **NET\_SDK\_DOWNLOAD\_IPC\_CFG\_FILE**

Network camera configuration file.

**NET\_SDK\_DOWNLOAD\_BASELINE\_SCENE\_PIC**

Base scene picture.

**NET\_SDK\_DOWNLOAD\_VQD\_ALARM\_PIC**

VQD (video quality diagnosis) alarm picture.

**NET\_SDK\_DOWNLOAD\_CONFIGURATION\_FILE**

Configuration file.

**NET\_SDK\_DOWNLOAD\_SCENE\_CONFIGURATION\_FILE**

Scene configuration file.

**NET\_SDK\_DOWNLOAD\_FILE\_FORM\_DB**

File in the image and video library.

**NET\_SDK\_DOWNLOAD\_TME\_FILE**

Entrance and exit management file.

**NET\_SDK\_DOWNLOAD\_VEHICLE\_BLOCKALLOWLIST\_FILE**

Blocklist and allowlist configuration file.

**NET\_SDK\_DOWNLOAD\_GUID\_FILE**

GUID file.

**NET\_SDK\_DOWNLOAD\_FILE\_FORM\_CLOUD**

Picture in the cloud storage.

**NET\_SDK\_DOWNLOAD\_PICTURE**

Picture.

**NET\_SDK\_DOWNLOAD\_VIDEO**

Video.

**NET\_DVR\_DOWNLOAD\_SCREEN\_FILE**

Screen server file.

**NET\_SDK\_DOWNLOAD\_PUBLISH\_MATERIAL**

Local material file of information release.

**NET\_SDK\_DOWNLOAD\_THERMOMETRIC\_FILE**

Thermometry calibration file.

**NET\_SDK\_DOWNLOAD\_LED\_CHECK\_FILE**

LED correction file.

**NET\_SDK\_DOWNLOAD\_VEHICLE\_INFORMATION**

Vehicle information to be exported.

**NET\_SDK\_DOWNLOAD\_CERTIFICATE\_BLOCKLIST\_TEMPLAT**

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.

## **7.2.4 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.
UPLOAD_VIOLATION_FILE	7	Violation dictionary file.
UPLOAD_TG_FILE	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_BLOCKALLOWLIST_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.



Enumeration Type	Macro Definition Value	Description
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.
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.

Enumeration Type	Macro Definition Value	Description
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).
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.

### 7.2.5 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_ERA           = 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_HMZHENGZHOU   = 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 Wenzhi 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_YOYI	= 177, //Youyi Bus
VLR_SHANGRAO	= 178, //Shangrao Bus
VLR_JIJIANG	= 179, //Jijiang
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_HENGtian	= 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_DAF	= 206, //Daihatsu Motor
VLR_SHENYE	= 207, //Shenye 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_WUZHOU LONG      = 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_GZ YUNBAO        = 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 Wenzi
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;
```

### 7.2.6 VTR\_RESULT

Enumerate the vehicle type recognition results.

#### Enumeration Definition

```
typedef enum _VTR_RESULT{
    VTR_RESULT_OTHER           = 0,
    VTR_RESULT_BUS             = 1,
    VTR_RESULT_TRUCK           = 2,
    VTR_RESULT_CAR             = 3,
    VTR_RESULT_MINIBUS         = 4,
    VTR_RESULT_SMALLTRUCK      = 5,
    VTR_RESULT_HUMAN           = 6,
    VTR_RESULT_TUMBREL         = 7,
    VTR_RESULT_TRIKE           = 8,
    VTR_RESULT_SUV_MPV         = 9,
    VTR_RESULT_MEDIUM_BUS      = 10,
    VTR_RESULT_MOTOR_VEHICLE   = 11,
```

```
VTR_RESULT_NON_MOTOR_VEHICLE = 12,  
VTR_RESULT_SMALLCAR           = 13,  
VTR_RESULT_MICROCAR           = 14,  
VTR_RESULT_PICKUP             = 15,  
VTR_RESULT_CONTAINER_TRUCK    = 16,  
VTR_RESULT_MINI_TRUCK         = 17,  
VTR_RESULT_SLAG_CAR           = 18,  
VTR_RESULT_CRANE              = 19,  
VTR_RESULT_OIL_TANK_TRUCK     = 20,  
VTR_RESULT_CONCRETE_MIXER     = 21,  
VTR_RESULT_PLATFORM_TRAILER  = 22,  
VTR_RESULT_HATCHBACK          = 23,  
VTR_RESULT_SALOON             = 24,  
VTR_RESULT_SPORT_SEDAN        = 25  
}VTR_RESULT;
```

### Members

#### **VTR\_RESULT\_OTHER**

Unknown.

#### **VTR\_RESULT\_BUS**

Bus.

#### **VTR\_RESULT\_TRUCK**

Truck.

#### **VTR\_RESULT\_CAR**

Car.

#### **VTR\_RESULT\_MINIBUS**

Minivan.

#### **VTR\_RESULT\_SMALLTRUCK**

Light truck.

#### **VTR\_RESULT\_HUMAN**

Pedestrian.

#### **VTR\_RESULT\_TUMBREL**

Two wheeler.

#### **VTR\_RESULT\_TRIKE**

Tricycle.

#### **VTR\_RESULT\_SUV\_MPV**

SUV/MPV.

#### **VTR\_RESULT\_MEDIUM\_BUS**

Middle-sized bus.

#### **VTR\_RESULT\_MOTOR\_VEHICLE**



Motor vehicle.

### **VTR\_RESULT\_NON\_MOTOR\_VEHICLE**

Non-motor vehicle.

### **VTR\_RESULT\_SMALLCAR**

Small sedan.

### **VTR\_RESULT\_MICROCAR**

Mini sedan.

### **VTR\_RESULT\_PICKUP**

Pick-up truck.

### **VTR\_RESULT\_CONTAINER\_TRUCK**

Container truck.

### **VTR\_RESULT\_MINI\_TRUCK**

Mini cargo truck.

### **VTR\_RESULT\_SLAG\_CAR**

Dump truck.

### **VTR\_RESULT\_CRANE**

Construction vehicle.

### **VTR\_RESULT\_OIL\_TANK\_TRUCK**

Oil tank truck.

### **VTR\_RESULT\_CONCRETE\_MIXER**

Concrete mixer.

### **VTR\_RESULT\_PLATFORM\_TRAILER**

Flatbed trailer.

### **VTR\_RESULT\_HATCHBACK**

Hatchback.

### **VTR\_RESULT\_SALOON**

Saloon.

### **VTR\_RESULT\_SPORT\_SEDAN**

Sports sedan.

## Appendix A. 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_IOOutputPortList 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

## A.1 /ISAPI/ITC/capability

Get intelligent traffic capability.

### Request URI Definition

**Table A-1 GET /ISAPI/ITC/capability**

<b>Method</b>	GET
<b>Description</b>	Get intelligent traffic capability
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u>XML_ITCCap</u> Failed: <u>XML_ResponseStatus</u>

## A.2 /ISAPI/Traffic/channels/<ID>/capabilities

Get traffic channel capability.

### Request URI Definition

**Table A-2 GET /ISAPI/Traffic/channels/<ID>/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get traffic channel capability.

Query	None
Request	None
Response	Succeeded: <u><i>XML_TrafficChannelCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

**Remarks**

The <ID> in the request URI refers to the traffic channel ID.

**A.3 /ISAPI/Traffic/channels/<ID>/licensePlate/filtration?format=json**

Get or set the parameters of filtering duplicated license plate.

**Request URI Definition****Table A-3 GET /ISAPI/Traffic/channels/<ID>/licensePlate/filtration?format=json**

Method	GET
Description	Get the parameters of filtering duplicated license plate.
Query	<b>format:</b> determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_Filtration</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

**Table A-4 PUT /ISAPI/Traffic/channels/<ID>/licensePlate/filtration?format=json**

Method	PUT
Description	Set the parameters of filtering duplicated license plate.
Query	<b>format:</b> determine the format of request or response message.
Request	<u><i>JSON_Filtration</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

**Remarks**

The <ID> in the request URI refers to channel ID.

## A.4 /ISAPI/Traffic/channels/<ID>/searchLPListAudit

Search for license plate list by channel.

### Request URI Definition

Table A-5 POST /ISAPI/Traffic/channels/<ID>/searchLPListAudit

Method	POST
Description	Search for license plate list by channel.
Query	None
Request	<u><i>XML_LPListAuditSearchDescription</i></u>
Response	<u><i>XML_LPListAuditSearchResult</i></u>

### Remarks

The <ID> in the request URI refers to the channel ID.

## Appendix B. Request and Response Messages

### B.1 JSON\_Filtration

Message about license plate filtration in JSON format

```
{
  "Filtration": {
    "enabled":
/*optional, boolean, whether to enable filtering duplicated license plate, by
default, it is "false"*/
  }
}
```

### B.2 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*/
}
```

### B.3 XML\_Desc\_ITDeviceAbility

Input description message for getting intelligent traffic capability.

```
<?xml version="1.0" encoding="utf-8"?>
<!--req, description of input parameter pInBuf for getting intelligent traffic
capability-->
<ITDeviceAbility version="2.0">
  <channelNO><!--req, xs:integer, channel No.--></channelNO>
  <ITCAbility/><!--opt, intelligent traffic capability-->
</ITDeviceAbility>
```

### Remarks

Refer to the message [XML\\_ITDeviceAbility](#) of for the intelligent traffic capability details.

## B.4 XML\_EventNotificationAlert\_ANPR

XML message about ANPR results

```
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema"><!--required, xs:object, ANPR result-->
  <ipAddress><!--required, xs:string, IPv4 address of the device that triggers
the alarm--></ipAddress>
  <ipv6Address><!--optional, xs:string, IPv6 address of the device that
triggers the alarm--></ipv6Address>
  <portNo><!--optional, xs:integer, port No. of the device that triggers the
alarm--></portNo>
  <protocol><!--required, xs:string, transmission communication protocol type:
HTTP, HTTPS, EHome. The value should be HTTP when ISAPI protocol is transmitted
via EZVIZ protocol. The value should be EHome when ISAPI protocol is
transmitted via ISUP--></protocol>
  <macAddress><!--optional, xs:string, MAC address--></macAddress>
  <dynChannelID><!--optional, xs:string, digital channel No.--></dynChannelID>
  <channelID><!--optional, xs:string, channel (video channel) number of the
device that triggers the alarm--></channelID>
  <relatedChannelList><!--optional, xs:string, list of channels linked with
the alarm (the same alarm source). These channels, which are used to display
live view or playback when the platform receives alarms, and that whose number
is the value of <channelID> are in the same camera. Multiple channel No. should
be separated by commas--></relatedChannelList>
  <dateTime>
    <!--required, xs:datetime, alarm triggering time, e.g.,
2017-04-22T15:39:01+08:00-->
  </dateTime>
  <activePostCount><!--required, xs:integer, times that the same alarm has been
uploaded--></activePostCount>
  <eventType><!--required, xs:string, event types, here it should be "ANPR"--></
eventType>
  <eventState><!--required, xs:string, continuous event's status: active (valid
event), inactive (invalid event)--></eventState>
  <eventDescription><!--required, xs:string, event description--></
eventDescription>
  <channelName><!--optional, xs:string, channel name--></channelName>
```



```
<deviceID><!--optional, xs:string, device ID, which should be returned when
the alarm is uploaded via ISUP--></deviceID>
<ANPR><!--optional, xs:object, ANPR alarm information. This node is valid
only when the value of <eventType> is ANPR-->
  <region><!--optional, xs:string, region. For details, refer to Region Code--
--></region>
  <country><!--optional, xs:string, country/region. For details, refer to
Country/Region Code--></country>
  <area><!--optional, xs:string, regions in United Arab Emirates: FJR (Al
Fujayrah), AD (Abu Dhabi), unknown, UMW (Umm Al Qaiwain), other, AM (Ajman),
RAK (Ras Al Khaimah), DB (Dubai), SJ (Sharjah)--></area>
  <licensePlate><!--required, xs:string, license plate number: "noPlate"
(vehicle without license plate), "unknown" (no license plate recognized),
"XXXXXX" (recognized license plate number). The maximum string size is 32
bytes--></licensePlate>
  <line><!--required, xs:integer, recognized lane number--></line>
  <direction><!--optional, xs:string, license plate recognition direction:
"reverse", "forward", "unknown"--></direction>
  <confidenceLevel><!--required, xs:integer, confidence level, which ranges
from 0 to 100--></confidenceLevel>
  <plateType><!--optional, xs:string, license plate type: "unknown",
"92TypeCivil"-92 civil vehicle, "92FarmVehicle"-civil vehicle two-line license
plate, "arm"-police vehicle, "upDownMilitay"-military vehicle (up/down format),
"92TypeArm"-92 armed police vehicle, "leftRightMilitay"-military vehicle (left/
right format), "02TypePersonalized"-02 personalized vehicle, "yellowTwoLine"-
yellow two-line license plate, "04NewMilitay"-04 new military vehicle,
"embassy"-embassy car, "oneLineArm"-new armed police vehicle with one-line
structure, "twoLineArm"-new armed police vehicle with two-line structure,
"yellow1225FarmVehicle"-yellow agricultural vehicle with 1225 structure,
"green1325FarmVehicle"-green agricultural vehicle with 1325 structure,
"yellow1325FarmVehicle"-yellow agricultural vehicle with 1325 structure,
"motorola"-motorcycle, "newEnergy"-new energy vehicle license plate,
"civilAviation"-civil aviation license plate, "coach"-driver-training car,
"tempTravl"-temporary license plate car, "trailer", "consulate"-consulate car,
"hongKongMacao"-vehicle entering and exiting Hong Kong and Macao, "tempEntry"-
temporary entry car, "emergency"-emergency license plate,
"oneLineArmHeadquarters"-armed police headquarter license plate (one-line),
"twoLineArmHeadquarters"-armed police headquarter license plate (two-line)--></
plateType>
  <plateColor>
    <!--optional, xs:string, license plate color: "black", "blue", "golden",
"orange", "red", "yellow", "white", "unknow", "other", "newEnergyYellowGreen"
(new energy green and yellow), "civilAviationBlack" (civil aviation black),
"civilAviationGreen" (civil aviation green), "green", "mixedColor" (mixed
color), "newEnergyGreen" (new energy green)-->
  </plateColor>
  <licenseBright>
    <!--optional, xs:integer, license plate brightness, which ranges from 0
to 255-->
  </licenseBright>
  <Rect><!--optional, coordinates of the license plate thumbnail in the
matched picture. The origin is the upper-left corner of the screen-->
```

```
<height><!--required, xs:float, height, value range: [0.000,1.000]--></height>
<width><!--required, xs:float, width, value range: [0.000,1.000]--></width>
<x><!--required, xs:float, X-coordinate of the upper-left point, value range: [0.000,1.000]--></x>
<y><!--required, xs:float, Y-coordinate of the upper-left point, value range: [0.000,1.000]--></y>
</Rect>
<pilotsafebelt>
  <!--optional, xs:string, whether the driver is wearing safety belt: "unknown, yes, no"-->
</pilotsafebelt>
<vicepilotsafebelt>
  <!--optional, xs:string, whether the co-driver is wearing safety belt: "unknown, yes, no"-->
</vicepilotsafebelt>
<pilotsunvisor>
  <!--optional, xs:string, whether the driver room's sun visor is open: "unknown, yes, no"-->
</pilotsunvisor>
<vicepilotsunvisor>
  <!--required, xs:string, whether the co-driver room's sun visor is open: "unknown, yes, no"-->
</vicepilotsunvisor>
<envprosign><!--optional, xs:string, whether it is a yellow-label vehicle: "unknown, yes, no"--></envprosign>
<dangmark>
  <!--optional, xs:string, whether it is dangerous goods vehicle: "unknown, yes, no"-->
</dangmark>
<uphone>
  <!--optional, xs:string, whether the driver is making call: "unknown, yes, no"-->
</uphone>
<pendant>
  <!--optional, xs:string, whether there is window hangings detected: "unknown, yes, no"-->
</pendant>
<tissueBox>
  <!--optional, xs:string, whether there is tissue box detected: "unknown, yes, no"-->
</tissueBox>
<frontChild>
  <!--optional, xs:string, whether the co-driver is with baby in arm: "unknown, yes, no"-->
</frontChild>
<label>
  <!--optional, xs:string, whether there is label detected: "unknown, yes, no"-->
</label>
<decoration>
```

```
    <!--optional, xs:string, whether there is decoration detected: "unknown,
yes, no"-->
  </decoration>
  <smoking>
    <!--optional, xs:string, whether there is smoking detected: "unknown,
yes, no"-->
  </smoking>
  <perfumeBox>
    <!--optional, xs:string, whether there is perfume box detected: "unknown,
yes, no"-->
  </perfumeBox>
  <pdvs>
    <!--optional, xs:string, whether there is a person sticking out of
sunroof: "unknown, yes, no"-->
  </pdvs>
  <helmet>
    <!--optional, xs:string, whether there is helmet detected: "unknown, yes,
no"-->
  </helmet>
  <twoWheelVehicle>
    <!--optional, xs:string, whether there is two-wheel detected: "unknown,
yes, no"-->
  </twoWheelVehicle>
  <threeWheelVehicle>
    <!--optional, xs:string, whether there is three-wheel detected: "unknown,
yes, no"-->
  </threeWheelVehicle>
  <blackness>
    <!--optional, xs:integer, Ringelmann emittance, which is used for smoke
detection-->
  </blackness>
  <plateCharBelieve>
    <!--optional, xs:string, confidence of the license plate's character,
value range: [0,100.00]. The value is accurate to two decimal places. For
example, if the confidence of characters in the license plate "ZA12345" is 20,
30, 40, 50, 60, and 70, it indicates that the possibility that the first
character recognized is "Z" is 20%, the possibility that the second character
recognized is "A" is 30%, and so forth-->
  </plateCharBelieve>
  <speedLimit>
    <!--optional, xs:integer, maximum speed limit, this node is valid only
when overspeeding occurred-->
  </speedLimit>
  <illegalInfo>
    <!--optional, traffic violation information of the vehicle-->
    <illegalCode>
      <!--required, xs:string, violation code-->
    </illegalCode>
    <illegalName>
      <!--required, xs:string, violation name-->
    </illegalName>
    <illegalDescription>
```

```

        <!--optional, xs:string, violation description-->
    </illegalDescription>
</illegalInfo>
<vehicleType>
    <!--optional, xs:string, vehicle type: "SUVMPV" (SUV/MPV), "buggy" (small-
sized truck), "bus", "concreteMixer" (concrete mixer), "containerTruck"
(container truck), "coupe", "crane", "hatchback", "largeBus" (large-sized bus),
"lightTruck" (light truck), "mediumBus" (middle-sized bus), "mediumHeavyTruck"
(medium and heavy truck), "miniCar" (mini sedan (transformed to "vehicle")),
"minibus", "minitruck", "motorVehicle" (motor vehicle (transformed to
"vehicle")), "nonmotorVehicle" (non-motor vehicle (transformed to
"threeWheelVehicle")), "oilTankTruck" (oil tank truck), "pedestrian",
"pickupTruck" (pickup truck (transformed to "buggy")), "platformTrailer"
(platform trailer), "saloon", "slagTruck" (dump truck), "smallCar" (small sedan
(transformed to "vehicle")), "threeWheelVehicle" (tricycle), "truck",
"twoWheelVehicle" (two wheeler), "unknown", "van", "vehicle" (sedan)-->
    </vehicleType>
    <postPicFileName>
        <!--optional, xs:string, name of the picture selected as the checkpoint
picture when illegal action occurs, "none" refers to not selecting any picture--
>
    </postPicFileName>
    <featurePicFileName>
        <!--optional, xs:string, name of the picture selected as the close-up
picture when running the red light in the intersection violation system is
detected, "none" refers to not selecting any picture-->
    </featurePicFileName>
    <detectDir>
        <!--optional, xs:integer, detection direction: 1-upward, 2-downward, 3-
bidirectional, 4-westward, 5-northward, 6-eastward, 7-southward, 8-other-->
    </detectDir>
    <detectType>
        <!--optional, xs:integer, detection type: 1-inductive loop trigger, 2-
video trigger, 3-multiple-frame recognition, 4-radar trigger-->
    </detectType>
    <barrierGateCtrlType>
        <!--optional, xs:integer, whether the barrier gate is opened: 0-opened, 1-
not opened-->
    </barrierGateCtrlType>
    <alarmDataType>
        <!--optional, xs:integer, 0-real-time data, 1-history data-->
    </alarmDataType>
    <dwIllegalTime>
        <!--optional, xs:integer, violation duration, which is the time
difference between the capture time of the last picture and that of the first
picture, unit: millisecond-->
    </dwIllegalTime>
    <vehicleInfo><!--optional, xs:object, vehicle information-->
        <index>
            <!--required, xs:integer, vehicle No.-->
        </index>
    </vehicleInfo>
    </vehicleType>

```

```
<!--optional, xs:integer, vehicle type: 0-other vehicle, 1-small-sized
vehicle, 2-large-sized vehicle, 3-pedestrian trigger, 4-two wheeler trigger, 5-
tricycle trigger, 6-motor vehicle trigger-->
</vehicleType>
<colorDepth>
  <!--required, xs:integer, shade of the vehicle color: 0-deep color, 1-
light color-->
</colorDepth>
<color>
  <!--required, xs:string, vehicle color: "unknown", "white", "silver"-
silvery, "gray", "blacks"-black, "red", "deepBlue"-dark blue, "blue", "yellow",
"green", "brown", "pink", "purple", "deepGray"-dark gray, "cyan", "orange"-->
</color>
<speed>
  <!--required, xs:integer, vehicle speed, unit: km/h-->
</speed>
<length>
  <!--required, xs:integer, length of the former vehicle, unit:
decimeter-->
</length>
<vehicleLogoRecog>
  <!--required, xs:integer, vehicle parent brand-->
</vehicleLogoRecog>
<vehileSubLogoRecog>
  <!--optional, xs:integer, vehicle sub-brand-->
</vehileSubLogoRecog>
<vehileModel>
  <!--optional, xs:integer, time to market of the vehicle sub-brand-->
</vehileModel>
<vehicleTypeByWeight>
  <!--optional, xs:integer, 1-class one vehicle (buses with seven or less
seats, trucks with capacity of 2 tons or less), 2-class two vehicle (buses with
8 to 19 seats, trucks with capacity of 2 to 5 (included) tons), 3-class three
vehicle (buses with 20 to 39 seats, trucks with capacity of 5 to 10 (included)
tons), 4-class four vehicle (buses with 40 or more seats, trucks with capacity
of 10 to 15 (included) tons), 5-class five vehicle (trucks with capacity of
more than 15 tons), 6-class six vehicle (trucks with capacity of more than 15
tons)-->
</vehicleTypeByWeight>
<CarWindowFeature><!--optional, xs:object, window feature, which is
configured by the node <CarWindowFeature> in the message of /ISAPI/ITC/
carFeatureParam-->
  <tempPlate><!--optional, xs:string, whether there is a temporary
license plate: unknown, yes, no-->unknown</tempPlate>
  <passCard><!--optional, xs:string, whether there is a vehicle pass:
unknown, yes, no-->unknown</passCard>
  <carCard><!--optional, xs:string, whether there is a card (business
card, leaflet, etc.)-->unknown</carCard>
</CarWindowFeature>
<CarBodyFeature><!--optional, xs:object, vehicle body attribute, which is
configured by the node <CarBodyFeature> in the message of /ISAPI/ITC/
carFeatureParam-->
```

```
<sparetire><!--optional, xs:string, whether there is a spare tire:
unknown, yes, no-->unknown</sparetire>
<rack><!--optional, xs:string, whether there is a roof rack: unknown,
yes, no-->unknown</rack>
<sunRoof><!--optional, xs:string, whether there is a sunroof: unknown,
yes, no-->unknown</sunRoof>
<words><!--optional, xs:string, whether there are characters painted on
the vehicle: unknown, yes, no-->unknown</words>
<slagTruckCoverPlate><!--optional, xs:string, whether there is a cover
on the dump truck: unknown, yes, no-->unknown</slagTruckCoverPlate>
</CarBodyFeature>
<vehicleUseType><!--optional, xs:string, vehicle type: taxi, ambulance,
bus, schoolBus, coach, unknown. The vehicle type can be configured by the node
<b>vehicleUseEnable</b> in the message of /ISAPI/ITC/carFeatureParam-->taxi</
vehicleUseType>
</vehicleInfo>
<EntranceInfo><!--optional, xs:object, entrance and exit information-->
<parkingID>
<!--optional, xs:string, parking space No.-->
</parkingID>
<gateID>
<!--optional, xs:string, entrance and exit No.-->
</gateID>
<direction>
<!--optional, xs:string, entering and exiting direction-->
</direction>
<cardNo>
<!--optional, xs:string, card No.-->
</cardNo>
<parkType>
<!--optional, xs:string, parking type: "permanent", "temporary"-->
</parkType>
</EntranceInfo>
<pictureInfoList><!--required, xs:object, picture list. Up to 8 pictures
can be supported-->
<pictureInfo><!--required, xs:object, picture information-->
<fileName>
<!--required, xs:string, picture name, which must correspond to the
picture name transmitted with the alarm message-->
</fileName>
<type>
<!--required, xs:string, picture type: "detectionPicture,
licensePlatePicture, pilotPicture, copilotPicture, compositePicture,
plateBinaryPicture, nonMotorPicture, pedestrianDetectionPicture,
pedestrianPicture"-->
</type>
<dataType>
<!--required, xs:integer, data type: 0-upload data, 1-upload URL-->
</dataType>
<picRecogMode>
<!--optional, xs:integer, 0-front license plate recognition, 1-rear
license plate recognition-->
```

```
</picRecogMode>
<redLightTime>
  <!--optional, xs:integer, red light time elapsed, unit: second-->
</redLightTime>
<vehicleHead>
  <!--optional, xs:integer, "unknown", "forward"-front license plate
recognition, "back"-rear license plate recognition-->
</vehicleHead>
<absTime>
  <!--optional, xs:time, absolute time, format: yyyyMMddHHmmssxxx,
e.g.: 20090810235959999, the last three number is time in millisecond-->
</absTime>
<plateRect>
  <!--dependent, the normalized value is the current image size in
percentage multiplying 1000 and it is accurate to three decimal places. This
node is valid only when <type> is "detectionPicture"-->
  <X>
    <!--required, xs:integer, X-coordinate of the upper-left corner of
the boundary frame-->
  </X>
  <Y>
    <!--required, xs:integer, Y-coordinate of the upper-left corner of
the boundary frame-->
  </Y>
  <width>
    <!--required, xs:integer, width of the boundary frame-->
  </width>
  <height>
    <!--required, xs:integer, height of the boundary frame-->
  </height>
</plateRect>
<vehicelRect>
  <!--dependent, the normalized value is the current image size in
percentage multiplying 1000. This node is valid only when <type> is
"detectionPicture"-->
  <X><!--required, xs:integer, X-coordinate of the upper-left point of
the boundary frame-->
  </X>
  <Y><!--required, xs:integer, Y-coordinate of the upper-left point of
the boundary frame-->
  </Y>
  <width>
    <!--required, xs:integer, width of the boundary frame-->
  </width>
  <height>
    <!--required, xs:integer, height of the boundary frame-->
  </height>
</vehicelRect>
<pictureURL>
  <!--dependent, xs:string, picture URL, which is valid only when
<data>Type is "URL"-->
</pictureURL>
```

```

        <pId><!--optional, xs:string, the maximum string size is 32 bytes.
Recommended generation rule: device serial number+time since the device started
+random number-->null</pId>
    </pictureInfo>
</pictureInfoList>
<hasMoreData>
    <!--optional, xs:boolean, whether there is more data. This node is used
to report the license plate information first, and then report XML message with
picture data; the XM message with picture data and license plate information
are linked by UUID-->
</hasMoreData>
<listType><!--optional, xs:string, list type: white (allowlist), black
(blocklist), temporary (temporary list)-->white</listType>
<originalLicensePlate>
    <!--optional, xs:string, original license plate number, When the license
plate number is a minor language, return the original license plate number-->
</originalLicensePlate>
<CRIndex>
    <!--optional, xs:integer, country/region index. For details, refer to
Country/Region Code-->
</CRIndex>
<VehicleGPSInfo><!--optional, GPS information of the vehicle-->
    <longitudeType><!--required, xs:string, longitude, "E,W"--></
longitudeType>
    <latitudeType><!--required, xs:string, latitude, "S,N"--></latitudeType>
    <Longitude><!--required, longitude information-->
        <degree><!--required, xs:integer--></degree>
    <minute><!--required, xs:integer--></minute>
    <sec><!--required, xs:float, accurate to 6 decimal places--></sec>
    </Longitude>
    <Latitude><!--required, latitude information-->
    <degree><!--required, xs:integer--></degree>
    <minute><!--required, xs:integer,--></minute>
    <sec><!--required, xs:float, accurate to 6 decimal places--></sec>
    </Latitude>
</VehicleGPSInfo>
<vehiclePositionControl><!--optional, xs:string, arming type:
"vehicleMonitor"-intelligent arming of vehicle (PUT /ISAPI/Traffic/channels/
<ID>/vehicleMonitor/<taskID>/startTask), "manualVehicleMonitor"-manual arming
of vehicle (PUT /ISAPI/Traffic/channels/<ID>/manualVehicleMonitor),
"dailyVehicleMonitor"-daily arming of vehicle (you can check whether this
arming type is supported via the node isSupportDailyVehicleMonitor in the
capability message returned by /ISAPI/Traffic/channels/<ID>/vehicleDetect/
capabilities; when daily arming of vehicle is enabled, both alarm of ANPR and
intelligent arming of vehicle will be uploaded; if this node is not returned,
it is normal vehicle detection--></vehiclePositionControl>
    <vehicleMonitorTaskID><!--optional, xs:string, task ID of intelligent
arming of vehicle, the maximum size is 64 bytes, this node is returned when the
value of vehiclePositionControl is "vehicleMonitor"--></vehicleMonitorTaskID>
    <vehicleListName><!--optional, xs:string, name of the list that the vehicle
belongs to, the maximum size is 128 bytes--></vehicleListName>
    <vehicleThermometryEnabled><!--optional, xs:boolean, whether to enable

```



```
vehicle temperature measurement-->true</vehicleThermometryEnabled>
  <currTemperature><!--optional, xs:float, temperature-->36.5</
currTemperature>
  <thermometryUnit><!--optional, xs:string, temperature unit: celsius,
fahrenheit, kelvin-->celsius</thermometryUnit>
  <plateCategory><!--optional, xs:string, additional license plate
information, the maximum string size is 8 bytes. This node is only used for
license plates of the Middle East-->test</plateCategory>
  <plateSize><!--optional, xs:int, license plate size: 0 (unknown), 1 (long),
2 (short, which is used for license plates of the Middle East)-->1</plateSize>
</ANPR>
<UUID>
  <!--optional, xs:string, common ID, which is used to link the same capture
across multiple servers-->
</UUID>
<picNum>
  <!--optional, xs:integer, number of pictures-->
</picNum>
<monitoringSiteID>
  <!--optional, xs:string, camera No.-->
</monitoringSiteID>
<ePlateUUID>
  <!--optional, xs:string, electronic license plate ID. If this node is
configured with a value, it indicates that an electronic license plate is
linked-->
</ePlateUUID>
<isDataRetransmission><!--optional, xs:boolean, data retransmission mark--></
isDataRetransmission>
  <SceneInfo><!--optional, scene information-->
    <scenesID><!--optional, xs:string, scene ID, value range: [1,16]--></
scenesID>
    <sceneName><!--optional, xs:string, scene name, the maximum size is 32
bytes--></sceneName>
    <PTZPos><!--optional, PTZ information-->
      <elevation><!--optional, xs:integer, value range: [-900,2700]--></
elevation>
      <azimuth><!--optional, xs:integer, value range: [0,3600]--></azimuth>
      <absoluteZoom><!--optional, xs:integer, value range: [0,1000]--></
absoluteZoom>
    </PTZPos>
  </SceneInfo>
  <monitorDescription><!--optional, xs:string, camera information--></
monitorDescription>
  <DeviceGPSInfo>
    <!--optional, xs:object, GPS location information of the device-->
    <longitudeType>
      <!--required, xs:string, longitude type: E, W-->E
    </longitudeType>
    <latitudeType>
      <!--required, xs:string, latitude type: S, N-->S
    </latitudeType>
    <Longitude>
```

```
<!--required, object, longitude-->
<degree>
  <!--required, int, degree-->60
</degree>
<minute>
  <!--required, int, minute, value range: [0,59]-->59
</minute>
<sec>
  <!--required, float, second, value range: [0,59.999999]-->59.000000
</sec>
</Longitude>
<Latitude>
  <!--required, object, latitude-->
  <degree>
    <!--required, int, degree-->60
  </degree>
  <minute>
    <!--required, int, minute, value range: [0,59]-->59
  </minute>
  <sec>
    <!--required, float, second, value range: [0,59.999999], the value is
accurate to six decimal places-->59.000000
  </sec>
</Latitude>
</DeviceGPSInfo>
<pilotStandardSafebelt>
  <!--optional, enum, whether the driver is buckled well: unknown, yes, no--
>yes
</pilotStandardSafebelt>
<vicepilotStandardSafebelt>
  <!--optional, enum, whether the front passenger is buckled well: unknown,
yes, no-->yes
</vicepilotStandardSafebelt>
<trafficLightSnap>
  <!--optional, enum, whether the picture is captured at the traffic light:
yes, no-->yes
</trafficLightSnap>
<sequence>
  <!--optional, int, vehicle capture number by the burst triggered by the
network, value range: [1,4294967295]. The burst control command will be applied
after triggered by the network. Related URI: /ISAPI/Traffic/startRecognition.
Related API of Device Network SDK: NET_DVR_ContinuousShoot-->0
</sequence>
<relaLaneDirectionType>
  <!--optional, int, linked lane direction: 0 (other), 1 (from east to west),
2 (from west to east), 3 (from south to north), 4 (from north to south) ,5
(from southeast to northwest), 6 (from northwest to southeast), 7 (from
northeast to southwest), 8 (from southwest to northeast)-->1
</relaLaneDirectionType>
<carDirectionType>
  <!--optional, int, vehicle moving direction on the lane: 0 (downward), 1
(upward)-->1
```

```
</carDirectionType>
<targetID>
  <!--optional, string, vehicle target ID, value range:[1,64]. The device
should ensure that it is unique. It corresponds to dwMatchNo in the Device
Network SDK-->test
</targetID>
<isSecondCamera>
  <!--optional, bool, whether the picture is captured by the second camera--
>false
</isSecondCamera>
<dataAnalysisType>
  <!--optional, int, data analysis type: 0 (data not analyzed), 1 (data
already analyzed)-->1
</dataAnalysisType>
<RecordInfo>
  <!--optional, object, vehicle video information-->
  <fileName>
    <!--required, string, video file name: record.mp4 (vehicle video). When
the video is transmitted in binary format, the value of this node must be the
same as that of Content-ID-->record.mp4
  </fileName>
  <dataType>
    <!--required, int, data type: 0 (binary data), 1 (URL)-->0
  </dataType>
  <URL>
    <!--optional, string, URL. This node is valid when the value of
<b>data</b>Type is URL-->null
  </URL>
</RecordInfo>
<VehicleWeightInfo>
  <!--optional, object, vehicle weight information-->
  <isOverWeight>
    <!--optional, bool, whether the vehicle is overweight-->false
  </isOverWeight>
  <axleNum>
    <!--optional, int, number of axles, value range:[1,10]-->4
  </axleNum>
  <overWeight>
    <!--optional, float, weight over the limit, value range: [0.000,100.000],
unit: ton. The value is accurate to three decimal places-->4.502
  </overWeight>
  <weight>
    <!--optional, float, vehicle weight, value range: [0.000,100.000], unit:
ton. The value is accurate to three decimal places-->4.502
  </weight>
  <limitWeight>
    <!--optional, float, weight limit, value range: [0.000,100.000], unit:
ton. The value is accurate to three decimal places-->4.502
  </limitWeight>
  <axleLen>
    <!--optional, float, wheelbase, value range:[0.00,100.00], unit: ton. The
value is accurate to two decimal places-->4.502
```

```
</axleLen>
<devDescInfo>
  <!--optional, string, device description, the maximum string size is 64
bytes-->null
</devDescInfo>
<AxleInfoList>
  <!--optional, object, list of the axle information. The number of
elements in the list is the same as the value of <axleNum-->
  <AxleInfo>
    <!--optional, object, information of an axle-->
    <axleWeight>
      <!--optional, float, axle weight, value range:[0.00,10000.00], unit:
kg-->100.00
    </axleWeight>
    <axleDistance>
      <!--optional, int, axle distance, value range:[0,100000], unit: mm.
It is the distance between the current axle and the next axle-->30000
    </axleDistance>
  </AxleInfo>
</AxleInfoList>
<length>
  <!--optional, int, vehicle length, value range:[1,1000000], unit: cm--
>4000
</length>
<width>
  <!--optional, int, vehicle width, value range:[1,1000000], unit: cm-->4000
</width>
<height>
  <!--optional, int, vehicle height, value range:[1,1000000], unit: cm--
>4000
</height>
<tiresNum>
  <!--optional, int, number of tires, value range:[1,20]-->4
</tiresNum>
<approvedPassengers>
  <!--optional, int, maximum number of passengers allowed, value range:
[1,100]-->7
</approvedPassengers>
</VehicleWeightInfo>
<isNotSlowZebraCrossing>
  <!--optional, bool, whether the vehicle did not slow down at zebra
crossing-->false
</isNotSlowZebraCrossing>
<isTurnRightStop>
  <!--optional, bool, whether the vehicle did not stop before turning right--
>false
</isTurnRightStop>
<PlateInfoList>
  <!--optional, object, license plate information (only used for Hong Kong/
Macau license plates)-->
  <PlateInfo>
    <!--optional, object, information of a license plate-->
```

```
<plateRect>
  <!--optional, object, coordinate of the license plate area. This node
is valid when the value of <type> is detectionPicture. The value is normalized
and equal to the size in percentage of the current image multiplying 1000. The
origin is the upper-left corner of the screen-->
    <X>
      <!--required, int, X-coordinate of the upper-left corner of the
boundary frame, value range: [0,1000]-->1000
    </X>
    <Y>
      <!--required, int, Y-coordinate of the upper-left corner of the
boundary frame, value range: [0,1000]-->1000
    </Y>
    <width>
      <!--required, int, width of the boundary frame, value range:[0,1000]--
>1000
    </width>
    <height>
      <!--required, int, height of the boundary frame, value range:
[0,1000]-->1000
    </height>
  </plateRect>
  <plateColor>
    <!--required, enum, license plate color: black, blue, golden, orange,
red, yellow, white, unknow, other, newEnergyYellowGreen (new energy green and
yellow), civilAviationBlack (civil aviation black), civilAviationGreen (civil
aviation green), green, mixedColor (mixed color), newEnergyGreen (new energy
green)-->black
  </plateColor>
  <licensePlate>
    <!--required, string, license plate number: noPlate (vehicle without
license plate), unknown (no license plate recognized), XXXXXX (recognized
license plate number)-->A283KY77
  </licensePlate>
  <confidenceLevel>
    <!--required, int, confidence level, value range:[0,100]-->50
  </confidenceLevel>
</PlateInfo>
</PlateInfoList>
<deviceUUID>
  <!--optional, string, device number, the maximum string size is 32 bytes.
It is the device's serial number by default and can be edited by the node
<deviceID> in the message of /ISAPI/System/deviceInfo-->12345
</deviceUUID>
</EventNotificationAlert>
```

### See Also

#### Region Code

#### Country/Region Code

## B.5 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--></isSupEmail>
  <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 linage 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>

```

## B.6 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>
  <NichbrokenTriggerCap><!--optional, xs: EventTriggerCapType--></
NichbrokenTriggerCap>
  <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>
<HighHDDTemperatureTriggerCap><!--optional, xs:EventTriggerCapType, HDD high
temperature detection--></HighHDDTemperatureTriggerCap>
<LowHDDTemperatureTriggerCap><!--optional, xs:EventTriggerCapType, HDD low
temperature detection--></LowHDDTemperatureTriggerCap>
<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 blacklist 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](#)

## B.7 XML\_ITCCap

XML message about intelligent traffic capability

```
<ITCCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<isSupportITC><!--dependent, xs:boolean--></isSupportITC>
<isSupportITCStatus>
<!--req, xs:boolean, whether it supports device status-->
</isSupportITCStatus>
<isSupportIllegalDictionary>
<!--req, xs:boolean, whether it supports traffic violation dictionary-->
</isSupportIllegalDictionary>
<isSupportVehicleDetection>
<!--dep, xs: boolean, whether it supports vehicle detection-->
</isSupportVehicleDetection>
<isSupportHVTVehicleDetection><!--dependent, xs:boolean--></
isSupportHVTVehicleDetection>
<isSupportlicencePlateAuditData><!--optional, xs:boolean--></
```

```
isSupportLicencePlateAuditData>
  <isSupportSearchLPListAudit><!--optional, xs:boolean--></
isSupportSearchLPListAudit>
  <isSupportEvidenceDictionary>
    <!--req, xs:boolean, whether it supports traffic law enforcement
dictionary-->
  </isSupportEvidenceDictionary>
  <isSupportITCSetUp>
    <!--req, xs:boolean, whether it supports installation parameters-->
  </isSupportITCSetUp>
  <isSupportTrafficParam>
    <!--req, xs:boolean, whether it supports vehicle counting statistics-->
  </isSupportTrafficParam>
  <isSupportManualCap>
    <!--req, xs:boolean, whether it supports manual capture-->
  </isSupportManualCap>
  <isSupportIllegalUploadPic>
    <!--req, xs:boolean, whether it supports uploading pictures-->
  </isSupportIllegalUploadPic>
  <isSupportContinueCap>
    <!--req, xs:boolean, whether it supports continuous capture-->
  </isSupportContinueCap>
  <isSupportWiper>
    <!--req, xs:boolean, whether it supports wiper-->
  </isSupportWiper>
  <isSupportEntranceCap>
    <!--optional, xs:boolean, whether it supports entrance and exit functions-->
  </isSupportEntranceCap>
  <isSupportPlateRecognitionParam>
    <!--req, xs:boolean, whether it supports license plate recognition
parameters-->
  </isSupportPlateRecognitionParam>
  <isSupportSyncSignalOutput>
    <!--req, xs:boolean, whether it supports output parameters-->
  </isSupportSyncSignalOutput>
  <isSupportSyncPower>
    <!--req, xs:boolean, whether it supports signal light synchronization-->
  </isSupportSyncPower>
  <isSupportImageMerge>
    <!--req, xs:boolean, whether it supports picture composition-->
  </isSupportImageMerge>
  <isSupportCabinetParam>
    <!--req, xs:boolean, whether it supports device cabinet alarm-->
  </isSupportCabinetParam>
  <isSupportCarFeatureParam>
    <!--req, xs:boolean, whether it supports vehicle features-->
  </isSupportCarFeatureParam>
  <isSupportLightCorrect>
    <!--req, xs:boolean, whether it supports exporting illegal action codes-->
  </isSupportLightCorrect>
  <isSupportSnapshot>
    <!--req, xs:boolean, whether it supports image capture resolution-->
```

```
</isSupportSnapshot>
<isSupportIllegalCodeData>
  <!--req, xs:boolean, whether it supports signal light correction-->
</isSupportIllegalCodeData>
<isSupportNetStorage>
  <!--req, xs:boolean, whether it supports network storage-->
</isSupportNetStorage>
<isSupportAlgorithmsVersion>
  <!--req, xs:boolean, whether it supports getting algorithm library status-->
</isSupportAlgorithmsVersion>
<isSupportAlgorithmsState>
  <!--req, xs:boolean, whether it supports getting algorithm library version--
>
</isSupportAlgorithmsState>
<isSupportPlateCorrection>
  <!--opt, xs:boolean, whether it supports manual license plate correction-->
</isSupportPlateCorrection>
<isSupportRadarSetUp>
  <!--req, xs:boolean, whether it supports radar construction parameters-->
</isSupportRadarSetUp>
<isSupportRadarMeasurement>
  <!--opt, xs:boolean, whether it supports radar measurement configuration-->
</isSupportRadarMeasurement>
</ITCCap>
```

## B.8 XML\_ITDeviceAbility

ITDeviceAbility message in XML format.

```
<ITDeviceAbility version="2.0"><!--capabilities of traffic camera and traffic
terminal server-->
  <channelNO><!--req, xs:integer, channel No.--></channelNO>
  <ITCAbility><!--req, traffic camera capability-->
    <IOInNo min="" max=""/>
    <!--req, xs:integer, max and min number of IO input ports-->
    <IOOutNo min="" max=""/>
    <!--req, xs:integer, max and min number of IO output ports-->
    <singleIOTriggerNum>4</singleIOTriggerNum>
    <!--req, xs:integer, group number of single IO trigger-->
    <lightArrayNum>8</lightArrayNum>
    <!--req, xs:integer, group number of traffic lights-->
    <measureArrayNum>4</measureArrayNum>
    <!--req, xs:integer, group number of velocimetry modes-->
    <lensMode opt="CCD,CMOS"/>
    <!-- Lens mode-->
    <PreTrigger>
      <enabled></enabled>
      <!--req, whether it is the original trigger mode, no this node if not
support-->
    </PreTrigger>
```

```
<triggerMode
opt="postIOSpeed,postSingleIO,postRS485,postRS485Radar,postVirtualcoil,epoliceIo
TrafficLights,epoliceRS485,peRS485,postNoComityPed, postRedLightPed,
videoEpolice"/>
  <!--req, trigger mode, the cameras of the version smaller than V3.1 are not
support it-->
  <support
opt="imageOverlayString,calibrateTime,multiNIC,NICBonding,voiceIntercom,IOEnable
Cfg,FTPExpand"/>
  <!--req, bySupport??
  bySupport&0x1??indicates whether support characters overlay configuration
extension
  bySupport&0x2??indicates whether support extend time correction
configuration
  bySupport&0x4, indicates whether support multi-NIC(multi-NIC segregation
  bySupport&0x8, indicates whether support NIC bonding function(network
failover)
  bySupport&0x10, indicates whether support voice talk
  bySupport&0x20??indicates whether support single IO trigger interface
configuration
  bySupport&0x40??indicates whether support FTP interface extension(support
in v3.6)-->
<supportMultiRadar opt="postRS485Radar,postVirtualcoil,videoEpolice,postHVT"/>
  <!--req, wSupportMultiRadar??
  wSupportMultiRadar&0x1??indicates whether bayonet RS485 radar support lane
associated with radar
wSupportMultiRadar&0x2??indicates whether bayonet virtual coilsupport lane
associated with radar
wSupportMultiRadar&0x4??indicates whether hybrid bayonet support lane
associated with radar
wSupportMultiRadar&0x8??indicates whether video detection support lane
associated with radar-->
<ICRPresetNum min="0" max="8"/><!--req, ICR preset number (optical filter
offset point) byICRPresetNum-->
<ICRTimeSlot min="0" max="8"/><!--req, ICR time slot number??1??8??
byICRTimeSlot-->
<expandRs485SupportSensor opt="peRS485,epoliceRS485"/>
  <!--req, byExpandRs485SupportSensor??
  byExpandRs485SupportSensor &0x1??indicates e-police vehicle inspection
support vehicle detector
byExpandRs485SupportSensor &0x2??indicates bayonet e-police inspection support
detector -->
  <expandRs485SupportSignalLampDet opt="videoEpolice,epoliceRS485,peRS485"/>
  <!--req, byExpandRs485SupportSignalLampDet??
  byExpandRs485SupportSignalLampDet &0x1??indicates e-police inspection
support signal detector
byExpandRs485SupportSignalLampDet &0x2??indicates bayonet e-police inspection
support external signal detector
byExpandRs485SupportSignalLampDet &0x4??indicates video e-police support
external signal detector-->

<ImageCombine>
```

```
<enabled>true</enabled>
<!--req, it supports the picture mergence, no this node if not support-->
</ImageCombine>
<TrafficDataUpload>
  <UploadType1>
    <enabled>
      true</enabled>
      <!--req, it supports the uploading mode no.1 (that is, the type
supported by intelligent traffic camera,
      COMM_UPLOAD_PLATE_RESULT of arming mode), no this node if not
support-->
    </UploadType1>
    <UploadType2>
      <enabled>
        true</enabled>
        <!--req, it supports the uploading mode no.2 (that is, the type
supported by ITS,
        COMM_ITS_PLATE_RESULT of listening mode), no this node if not
support-->
      </UploadType2>
      <UploadType3>
        <enabled>
          true</enabled>
          <!--req, it supports the uploading mode no.2 (that is, the type
supported by intelligent traffic camera,
          COMM_ITS_PLATE_RESULT of arming mode), no this node if not support--
>
        </UploadType3>
      </TrafficDataUpload>

  <VideoLocalRecord>
    <enabled>true</enabled>
    <!--req, whether it supports local recording, no this node if not
support-->
  </VideoLocalRecord>
  <PicLocalStore>
    <enabled>true</enabled>
    <!--req, whether it supports local storage of pictures, no this node if
not support-->
  </PicLocalStore>
  <PlateRecognise>
    <enabled>true</enabled>
    <!--req, whether it supports license plate recognition, no this node if
not support-->
    <regionType opt="rect,polygon"/>
    <!--the supported types of license plate recognition area-->
    <plateTypeopt
opt="standard92,standard02,armedPolice,police,standard92Back,embassy,agricultura
l,moto"/>
    <plateColor opt="blue,yellow,white,black,green"/>
    <vehicleType opt="small,big"/>
    <bodyworkColor
```

```
opt="white,silver,gray,black,red,darkblue,blue,yellow,green,brown,pink,violet"/>

</PlateRecognise>
<imageOverlayString>
  <model>
    <!--opt, the character overlay on the picture of original intelligent
traffic camera, no this node if not support-->
    <overlayStringType
opt="monitor,time,speed,speedRatio,speedLimit,plate,carColor,carLength,carType,l
aneNum,milliSecond,illegalInfo,redOnTime"/>
    <!--opt, monitoring site, time, speed, overspeed ratio, speed limit
sign, license plate, color of vehicle, length of vehicle,
    vehicle type, traffic lane ID, millisecond, violation information, the
time that the red light has been lighted-->
  </model>
  <mode2>
    <!--opt, the character overlay on the picture of ITS or camera of ITS,
no this node if not support-->
    <overlapType
opt="monitor,site,roadNum,instrumentNum,directionNum,directionDes,laneNum,laneDe
s,capTime,capTimeMilli,plate,carColor,carLength,carType,carBrand,speed,speedRati
o,speedLimit,illegalDes,redStart,redStop,redOnTime,securityCode,capCode"/>
    <!--opt, monitoring site, location, intersection ID, device ID,
direction ID, direction description, lane ID, lane description,
    the time of capturing, the time of capturing(millisecond), license
plate number, color of vehicle, length of vehicle, vehicle type,
    vehicle brands, speed, overspeed ratio, speed limit sign, violation
information, the starting time of red light, the ending time of
    red light, the time that the red light has been lighted, security code,
capturing ID-->
    <itemLength min="0" max=""/>
    <!--opt, length of character overlay-->
    <changeLineNum min="0" max="10"/>
    <!--opt, the number of line breaks-->
    <spaceNum min="0" max="255"/>
    <!--opt, the number of spaces-->
    <linePercent min="0" max="100"/>
    <!--opt, percentage of overlay line-->
    <itemsStlye opt="horizontal,vertical"/>
    <!--opt, overlay style: horizontal, vertical -->
    <startPosTop min="0" max="2448"/>
    <!--opt, the upper coordinate of starting point-->
    <startPosLeft min="0" max="2448"/>
    <!--opt, the left coordinate of starting point-->
    <charStyle opt="SongTi,WeiTi"/>
    <!--opt, font type: standard typeface of Chinese, typeface of Wei-->
    <charSize opt="16*16,32*32,48*48,64*64"/>
    <!--req, font size: 0-32*32(Chinese)/16*32(English), 1-64*64(Chinese)/
32*64(English), 2-48*48 -->
    <charInterval min="0" max="16"/>
    <!--opt, character spacing-->
    <ForeClorRGB>
```



```
<enabled>
  true</enabled>
  <!--opt, whether it supports to adjust RGB value of foreground
color, no this node if not support-->
</ForeClorRGB>
<BackClorRGB>
  <enabled>
    true</enabled>
    <!--opt, whether it supports to adjust RGB value of background
color, no this node if not support-->
  </BackClorRGB>
  <ColorSelfAdapt>
    <enabled>
      true</enabled>
      <!--opt, whether the color is adaptive, no this node if not
support-->
    </ColorSelfAdapt>
    <itemTypeCustomLength min="0" max="32"/><!--opt, Custom definition type
length(added in NET_ITS_OVERLAP_CFG_V50), it will not be displayed if not
supported.-->
  </mode2>
</imageOverlayString>
<TPSAlarm>
  <!--req, whether it supports uploading information of traffic counting,
no this node if not support-->
  <UploadType1>
    <enabled>
      true</enabled>
      <!--req, it supports the uploading mode no.2(that is, the type
supported by ITS, COMM_ITS_TRAFFIC_COLLECT of listening mode), no this node if
not support-->
    </UploadType1>
    <UploadType2>
      <enabled>
        true</enabled>
        <!--req, it supports the uploading mode no.2(that is, the type
supported by intelligent traffic camera, COMM_ITS_PLATE_RESULT of arming
mode(COMM_ITS_TRAFFIC_COLLECT), no this node if not support-->
      </UploadType2>
    </TPSAlarm>
  <CameraSetup>
    <!--req, it supports camera mounting configuration, no this node if not
support-->
    <enabled>
      true</enabled>
  </CameraSetup>
  <AlarmRecordDown>
    <!--req, it supports to download the alarm record, no this node if not
support-->
    <enabled>
      true</enabled>
  </AlarmRecordDown>
```

```
<supportMilliCheckTime>
  <!--req, support millisecond time correction -->
  <enabled>true</enabled>
</supportMilliCheckTime>

<SnapStatusDetectCFG>
  <!--req, enabled parameter capability of status detection, the
corresponding stucture is (NET_DVR_STATUS_DETECTCFG), the node does not
display when there is no support-->
  <triggerIODetect opt="true,false"/><!--req, IO status detection triggered
by ITC, true-enable??false-disable-->
  <flashOutDetect opt="true,false"/><!--req, ITC sync-output status
detection, true-enable??false-disable-->
  <RS485Detect opt="true,false"/><!--req, ITC RS485 receive status
detection, true-enable??false-disable-->
  <trafficLightDetect opt="true,false"/><!--req, ITC traffic light
detection, true-enable??false-disable-->
</SnapStatusDetectCFG>

<SnapRadarParam>
  <!--req, radar configuration parameter capability, the corresponding
stucture is (NET_ITC_RADAR_PARAM), the node does not display when there is no
support-->
  <radarType opt="noRadar,Andaolei,Olivia,microwave,IOExpanBox,other"/><!--
req, radar type, 0-noRadar,1-Andaolei,2-Olivia,3-microwave, 4-IOExpanBox,0xff-
other -->
  <levelAngle min="0" max="90"/><!--req, the angle with horizontal line,
0~90 -->
  <radarSensitivity min="0" max="65535"/><!--req, radar sensitivity,
0~65535 -->
  <radarSpeedValidTime min="0" max="2000"/><!--req, valid time of radar
speed [0~2000] -->
  <lineCorrectParam min="0.0" max="2.0"/>
  <!--req, linear correction parameters -->
  <constCorrectParam min="-100" max="100"/>
  <!--req, constant correction parameters -->
</SnapRadarParam>

<SnapEnableCfgr>
  <!--req, the corresponding stucture is (NET_DVR_SNAPENABLECFG) -->
  <plateEnable opt="true,false"/>
  <!--req, whether support plate recognition??0-false??1-true -->
  <jpegPICSsize min="64" max="8196"/>
  <!--req Jpeg size[64-8196]-->
  <frameFlip opt="noturn,turn"/>
  <!--req 0-noturn(no flip) 1-turn(flip)-->
  <flipAngle opt="0,90,180,270"/>
  <!--req angle of image: 0,90,180,270-->
  <lightPhase min="0" max="360"/>
  <!--req phase, data range[0, 360]-->
  <lightSyncPower opt="noSynchronized,Synchronized"/>
  <!--req whether the signal lamp is synchronized with power
```

```
supply??0-noSynchronized??1-Synchronized-->
    <frequency min="0" max="255"/>
        <!--req signal frequency-->
    <uploadSDEnable opt="false,true"/>
        <!--req whether auto upload SD picture: 0-false, 1-true-->
>
    <plateMode opt="videoTrigger,externalTrigger"/>
        <!--req recognition mode parameter: 0-videoTrigger, 1-
externalTrigger-->
    <uploadInfoFTP opt="false,true"/>
        <!--req whether upload the additional info of capture
to FTP: 0-false, 1-true-->
    <autoFormatSD opt="false,true"/>
        <!--req whether auto format SD card: 0-false, 1-
true-->
    </SnapEnableCfg>

    <ITCIOoutParam>
        <defaultStatus opt="lowlevel,highlevel"/>
            <!--req IO default status: 0 - lowlevel??1 - highlevel-->
    <IOoutStatus opt="lowlevel,highlevel,impulse"/>
        <!--req IO valid status: 0 - lowlevel,1 - highlevel,2 - impulse-->
    <flashMode opt="video,post,illegal"/>
        <!--req flash work mode,described by bit: 0-work??1-not work, bit0 -
video, bit1 - post,bit2 - illegal-->
    <timeDelay min="" max=""/>
        <!--req IO valid contious time, unit: us-->
    <aheadTime min="" max=""/>
        <!--req output IO ahead time,unit: us-->
    <dutyRate min="0" max="40"/>
        <!--req rate??[0,40%]-->
    <detectBrightness opt="false,true"/>
        <!--req auto detect the brightness of enabled flash: 0-false,
1-true-->
    <brightnessThreld min="0" max="100"/>
        <!--req brightness threld of flash, range:[0,100], flash when higher than
threld-->
    <flashLightEnable opt="false,true"/><!--req set flash time enable: 0-false, 1-
true-->
    <autoPlateBrightness opt="false,true"/>
        <ioWorkMode opt="flashlight,polarizer"/>
            <!--IO output port working mode: 0-Flash light, 1- Polarizer-->
    </ITCIOoutParam>

    <CameraSetUpCfg>
        <setupHeight min="0" max="65535"/><!--req setup height/-->
        <lensType opt="unKnown,8mm,12mm,16mm,25mm,35mm,50mm"/><!--req lens type/-->
>
        <setupHeightUnit opt="M,CM"/>
            <!--req 0~M,1~CM-->
        <sceneDis min="0" max="65535"/><!--req the horizontal distance between the
bottom of image and camera, unit: cm-->
```

```
<videoDetCoefficient min="" max=""/><!--req video test
coefficient[0,300]-->
<erectMethod opt="normalinstallation,sideinstallation"/><!--req, erect
mode: 0- normal ??1- side -->
<horFieldAngle min="" max=""/>
<!-- The horizontal filed angle range [0??~360??]-->
<verFieldAngle min="" max=""/>
<!-- The vertical filed angle range [0??~360??]-->
</CameraSetUpCfg>

<PlcCfg>
  <PLCEnable opt="false,true"/>
  <!--req plate brightness compensation??default to enable????0-close??1-
enable-->
  <plateExpectBright min="0" max="100"/>
  <!--req expectation brightness of plate,?? default to 50??, data
range: [0, 100]-->
  <brightChangeEnable opt="false,true"/>
  <!--req illumination variation??default to enable????0-close??1-enable
-->
  <brightChangeThreld min="0" max="100"/>
  <!--req brightness change threld ?? default to 50????data range: [0,
100]-->
  <tradeOffFlash opt="false,true"/>
  <!--req whether consider influence of flash: 0 - no; 1 -
yes(default), when use the flash compensation, 1- weaken the enhancement effect
of flash brightness, 0 - no -->
  <correctFactor min="0" max="100"/>
  <!--req correction factor, data range: [0,100], default it to 50
(restore the default value when swicth tradeoff_flash?? -->
  <loopStatsEn opt="false,true"/>
  <!--req brightness of coil, described by bit: 0- No statistical??1-
statistical-->
  <PLCBrightOffSet min="0" max="100"/>
  <!--req sensitivity of plate brightness compensation(vitual coil mode
valid only)??data range: 1- 100-->
</PlcCfg>

<CabinetAlarmParamCfg>
  <supportCabinetNum min="0" max="8"/>
  <cabinetNameLen min="0" max="32"/>
  <associateIO opt="1,2,3,4,0xff"/>
  <!--req 1-4??0xff?? 1-4: IO channel, 0xff- disable-->
  <cabinetState opt="lowlevel,hightlevel"/>
  <!--req 0- lowlevel, 1- highlevel-->
  <alarmIntervalTime min="1" max="60"/>
  <!--req data range: 1-60??uint: s -->
</CabinetAlarmParamCfg>

<ExceptionAlarmITC>
  <exceptionType
opt="diskError,nicBroken,ipConflict,sensors,signalDetector"/>
```

```

        <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/><!--req,
handle ,picture: capture and send mail-->
        </ExceptionAlarmITC>
        <ICRCFG>
        <autoCtrlTime>4</autoCtrlTime>
        <ICRPreset>2</ICRPreset>
        <switchType opt="no,auto,manual,time,algorithmAuto"/>
        <!--0 ~ Not switch, 1- Auto switch, 2~Manually switch,3~Scheduling
switch??4~Auto-arithmetic-->
        <AlgorithmAutoSwitch>
        <detThreshold min="" max="" def=""/><!--Check the threshold,
range[0~100], default:58-->
        <unBrightnessThreshold min="" max="" def=""/><!--Abnormal
brightness, range[0~255],default:12 -->
        </AlgorithmAutoSwitch>
        </ICRCFG>

        <ITCFTP>
        <ItcFtpTypeCond>
        <workMode opt="ftp1,ftp2"/><!--req 0-FTP1(main FTP)??1-FTP2( backupFTP)--
>
        </ItcFtpTypeCond>
        <ItcFtpCfg>
        <enableFtp opt="true,false"/><!--req 0-false,1-true-->
        <addressType opt="ipaddress,domainname"/><!--req 0- actual ipv4 ipv6
address, 1-domain-->
        <ftpPort min="" max=""/><!--req ftp port range-->
        <domainLen min="" max=""/><!--req domain length-->
        <userNameLen min="" max=""/><!--req user name length-->
        <passwordLen min="" max=""/><!--req password length-->
        <dirLevel
opt="rootdirectory,parentdirectory,childdirectory,threedirectory,fourdirectory"/
>
        <!--req 0 = don't use directory??save in the root directory,1 = use
level 1 directory, 2= use level 2 directory, 3 = use level 3 directory, 4=use
level 4 directory-->
        <uploadDataType opt="all,post,illegal"/><!--req 0-all,1- post,2-
illegal(default to select all when use single FTP, select post when use double
FTP)-->
        <itemOrder
opt="devname,devno,devip,channelname,channelno,time,cardno,plateno,paltec color,la
nen, carspeed,picnumber,
carnumber,speedlimit,illegalcode,crossnumber,directionnumber"/>
        <delimiter opt="_"/><!--req separator??general: '_'-->
        <topDirMode
opt="deviceName,deviceNO,deviceIP,monitor,time,timeday,violatetype,direction,pla
ce ,channelName,channelNo,LaneNo,custom"/>
        <!--req 0x1 = use device name,0x2 = use device number,0x3 =use
device IP??0x4=use monitor point,0x5=use time(year month), 0x6=use time(year
month day),

```

```
0x7=violation type,0x8= direction,0x9=location,0xa=channel name,
0xb=channel number,0xc=lane number; 0xff=automatic-->
    <subDirMode
opt="deviceName,deviceNO,deviceIP,monitor,time,timeday,violatetype,direction,place
,channelName,channelNo,LaneNo,custom"/>
    <threeDirMode
opt="deviceName,deviceNO,deviceIP,monitor,time,timeday,violatetype,direction,place
,channelName,channelNo,LaneNo,custom"/>
    <fourDirMode
opt="deviceName,deviceNO,deviceIP,monitor,time,timeday,violatetype,direction,place
,channelName,channelNo,LaneNo,custom"/>
    <topCustomDirLen min="" max=""/><!--req customized length of level 1
directory-->
    <subCustomDirLen min="" max=""/><!--req customized length of level 2
directory -->
    <threeCustomDirLen min="" max=""/><!--req customized length of level 3
directory-->
    <fourCustomDirLen min="" max=""/><!--req customized length of level 4
directory-->
    </ItcFtpCfg>

</ITCFTP>

<GPSDATAACFG>
    <gpsDataMode opt="auto,manual"/>
    <!--Getting GPS data, 0-Auto??1-Manual-->
    <longitudeType opt="east,west"/><!--Longitude: 0- East longitude??1-
West longitude-->
    <latitudeType opt="north,south"/><!--latitude: 0-South latitude??1-
North latitude-->
    <Longitude>
        <degree min="0" max="180"/>
        <minute min="0" max="60"/>
        <sec min="0" max="60"/>
    </Longitude>
    <Latitude>
        <degree min="0" max="180"/>
        <minute min="0" max="60"/>
        <sec min="0" max="60"/>
    </Latitude>
</GPSDATAACFG>

</ITCAbility>
<ITSAbility>
    <cameraAmount>
        <!--req, xs:integer, the maximum number of supported cameras-->
    </cameraAmount>
    <storeAmount>
        <!--req, xs:integer, the maximum amount of supported data storage-->
    </storeAmount>
    <ImageCombine>
        <enabled>true</enabled>
```

```
<!--req, it supports picture mergence, no this node if not support-->
</ImageCombine>
<TrafficDataUpload>
  <UploadType1>
    <enabled>
      true</enabled>
      <!--req, it supports the uploading mode no.1 (that is, the type
supported by intelligent traffic camera, COMM_UPLOAD_PLATE_RESULT of arming
mode), no this node if not support-->
    </UploadType1>
    <UploadType2>
      <enabled>
        true</enabled>
        <!--req, it supports the uploading mode no.2 (that is, the type
supported by ITS, COMM_ITS_PLATE_RESULT of listening mode), no this node if not
support-->
      </UploadType2>
      <UploadType3>
        <enabled>
          true</enabled>
          <!--req, it supports the uploading mode no.2 (that is, the type
supported by intelligent traffic camera, COMM_ITS_PLATE_RESULT of arming mode),
no this node if not support-->
        </UploadType3>
      </TrafficDataUpload>
      <VideoLocalRecord>
        <enabled>
          true<!--req, it supports local recording, no this node if not support--
>
        </enabled>
      </VideoLocalRecord>
      <PicLocalStore>
        <enabled>
          true<!--req, it supports local storage of pictures, no this node if not
support-->
        </enabled>
      </PicLocalStore>
      <PlateRecognise>
        <enabled>
          true<!--req, whether it supports license plate recognition, no this
node if not support-->
        </enabled>
        <regionType opt="rect,polygon"/>
        <!-- the supported types of license plate recognition area-->
        <plateTypeopt
opt="standard92,standard02,armedPolice,police,standard92Back,embassy,agricultura
l,moto"/>
        <plateColor opt="blue,yellow,white,black,green"/>
        <vehicleType opt="small,big"/>
        <bodyworkColor
opt="white,silver,gray,black,red,darkblue,blue,yellow,green,brown,pink,violet"/>
```

```
</PlateRecognise>
<imageOverlayString>
  <model>
    <!--opt, the character overlay on the picture of original intelligent
traffic camera, no this node if not support-->
    <overlayStringType
opt="monitor,time,speed,speedRatio,speedLimit,plate,carColor,carLength,carType,l
aneNum,milliSecond,illegalInfo,redOnTime"/>
    <!--opt, monitoring site, time, speed, overspeed ratio, speed limit
sign, license plate, color of vehicle, length of vehicle,
    vehicle type, traffic lane ID, millisecond, violation information, the
time that the red light has been lighted-->
  </model>
  <mode2>
    <!--opt, the character overlay on the picture of ITS or camera of ITS,
no this node if not support-->
    <overlapType
opt="monitor,site,roadNum,instrumentNum,directionNum,directionDes,laneNum,laneDe
s,capTime,capTimeMilli,plate,carColor,carLength,carType,carBrand,speed,speedRati
o,speedLimit,illegalDes,redStart,redStop,redOnTime,securityCode,capCode"/>
    <!--opt, monitoring site, location, intersection ID, device ID,
direction ID, direction description, lane ID, lane description,
    the time of capturing, the time of capturing(millisecond), license
plate number, color of vehicle, length of vehicle, vehicle type,
    vehicle brands, speed, overspeed ratio, speed limit sign, violation
information, the starting time of red light, the ending time of
    red light, the time that the red light has been lighted, security code,
capturing ID-->
    <itemLength min="0" max=""/>
    <!--opt, length of character overlay-->
    <changeLineNum min="0" max="10"/>
    <!--opt, the number of line breaks-->
    <spaceNum min="0" max="255"/>
    <!--opt, the number of spaces-->
    <linePercent min="0" max="100"/>
    <!--opt, percentage of overlay line-->
    <itemsStlye opt="horizontal,vertical"/>
    <!--opt, overlay style: horizontal, vertical-->
    <startPosTop min="0" max="2448"/>
    <!--opt, the upper coordinate of starting point-->
    <startPosLeft min="0" max="2448"/>
    <!--opt, the left coordinate of starting point-->
    <charStyle opt="SongTi,WeiTi"/>
    <!--opt, font type: standard typeface of Chinese, typeface of Wei-->
    <charSize opt="16*16,32*32,48*48,64*64"/>
    <!--req, font size: 0-32*32(Chinese)/16*32(English), 1-64*64(Chinese)/
32*64(English), 2-48*48 -->
    <charInterval min="0" max="16"/>
    <!--opt, character spacing-->
    <ForeClorRGB>
      <enabled>
        true</enabled>
```



```
        <!--opt, whether it supports to adjust RGB value of foreground
color, no this node if not support-->
        </ForeClorRGB>
        <BackClorRGB>
        <enabled>
        true</enabled>
        <!--opt, whether it supports to adjust RGB value of background
color, no this node if not support-->
        </BackClorRGB>
        <ColorSelfAdapt>
        <enabled>
        true</enabled>
        <!--opt, whether the color is adaptive, no this node if not
support-->
        </ColorSelfAdapt>
    </mode2>
</imageOverlayString>
<TPSAlarm>
    <!--req, whether it supports uploading information of traffic counting,
no this node if not support-->
    <UploadType1>
    <enabled>
    true</enabled>
    <!--req, it supports the uploading mode no.2 (it supports the
uploading mode no.2(that is, the type supported by ITS,
COMM_ITS_TRAFFIC_COLLECT of listening mode), no this node if not support-->
    </UploadType1>
    <UploadType2>
    <enabled>
    true</enabled>
    <!--req, it supports the uploading mode no.2 (that is, the type
supported by intelligent traffic camera, COMM_ITS_TRAFFIC_COLLECT of arming
mode,no this node if not support-->
    </UploadType2>
</TPSAlarm>
<CameraSetup>
    <!--req, it supports camera mounting configuration, no this node if not
support-->
    <enabled>
    true</enabled>
</CameraSetup>
<!--req,park project -->
<LampCtrlInfo>
    <lampCtrlMode opt="inlayLamp,externalLamp"/>
    <!--req, lamp control mode??1-internal lamp??2-external lamp-->
    <ctrlChannelIndex min="" max=""/>
    <!--req alternate control channel number-->
    <inlayLampCtrlMode>
    <!--req internal lamp control mode -->
    <lampStateCtrlNum min="" max=""/>
    <!--req range of park space supported-->
    <parkInlayInfo>
```

```

        <enable opt="false,true"/>
        <!--req,whether enable, true-enable, false-disable-->
        <flicker opt="false,true"/>
        <!--req,whether flash, true-flash??false- no flash-->
        <lampColor opt="close,red,green,yellow,blue,magenta,cyan,white"/>
        <!--req 0- none 1-red 2-green 3-yellow 4-blue 5-pink 6-cyan 7-white--
>
    </parkInlayInfo>
</inlayLampCtrlMode>
<externalLampCtrlMode>
    <!--req external lamp control mode -->
    <maxParkNum min="" max=""/>
    <!--req park space number-->
    <parkInfoType
opt="normalParkIOstate,normalNoParkIOstate,specialParkIOstate,noSpecialParkIOsta
te"/>
        <parkExternalSubinfo>
            <enable opt="false,true"/>
            <!--req,whether enable, true-enable??false-disable-->
            <flicker opt="false,true"/>
            <!--req,whether flash, true-flash??false-no flash-->
            <IOstate opt="lowLevel,hightLevel"/>
            <!--req,level, 0-low level??1-high level(valid external lamp)-->
        </parkExternalSubinfo>
    </externalLampCtrlMode>
</LampCtrlInfo>

<parkSpaceAttributeParam>
    <maxParkNum min="" max=""/>
    <!--req park number-->
    <parkSpaceInfo>
        <parkSpaceAttribute opt="normalPack,specialPack"/>
        <!-- 0~normal pack 1~special pack-->
    </parkSpaceInfo>
</parkSpaceAttributeParam>

<lampExternalCfg>
    <enable opt="false,true"/>
    <!--req,whethe enable, true-enable??false-disable-->
    <lampState>
        <flicker opt="false,true"/>
        <!--req,whether flash, true-flash??false-no flash-->
        <IONo opt="IO1,IO2,IO3"/>
        <!--req 1~IO1,2~IO2,4~IO3 -->
    </lampState>
</lampExternalCfg>

<compelCaptureCfg>
    <parkIndex opt="parkPlace1,parkPlace2,parkPlace3,parkPlace4"/>
    <!-- park number from left to right 1,2,3,4-->
</compelCaptureCfg>

```

```
<externalControlAlarm>
  <lampState>
    <flicker opt="false,true"/>
    <!--req,whether flash, true-flash??false- no flash-->
    <IONo opt="IO1,IO2,IO3"/>
    <!--req 1~IO1,2~IO2,4~IO3 -->
  </lampState>
  <externalBeginTime>true</externalBeginTime>
  <!--req whether support external begin time uploading -->
</externalControlAlarm>
<ManualSnap>
<!-- req, 3.7 added-->
  <osdEnable>true</osdEnable>
<!-- req, capture OSD supports force closing-->
<laneNo min="1" max="6" />
  <!-- req, vehicle lane No.-->
</ManualSnap>
</ITSAbility>
</ITDeviceAbility>
```

### Example

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

## B.9 XML\_LPListAuditSearchDescription

LPListAuditSearchDescription message in XML format

```
<LPListAuditSearchDescription version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <searchID><!--req, xs: 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--></
searchID>
  <searchResultPosition><!--req, xs: integer, the end position of search result
in result list--></searchResultPosition>
  <maxResults><!--req, xs: integer, the maximum number of results obtained by
this search. Up to 2000 results can be obtained by each search--></maxResults>
  <type><!--optional, xs:string, license plate type: "blackList" (license plate
in blacklist), "whitelist" (license plate in allowlist), "allVehicleList"
(license plate in all lists), "otherVehicleList" (license plate in other
lists)--></type>
  <LicensePlate><!--optional, xs:string, license plate number, fuzzy search is
supported--></LicensePlate>
  <cardNo><!--optional, xs:string, card No.--></cardNo>
  <cardID><!--optional, xs:string, card ID (Wiegand protocol), the maximum
string size is 9 bytes--></cardID>
</LPListAuditSearchDescription>
```

## B.10 XML\_LPListAuditSearchResult

LPListAuditSearchResult message in XML format

```
<LPListAuditSearchResult version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <searchID><!--req, xs: string--></searchID>
  <responseStatus></responseStatus>
  <responseStatusStrg></responseStatusStrg>
  <numOfMatches><!--req, xs: integer--></numOfMatches>
  <totalMatches><!--req, xs: integer--></totalMatches>
  <LicensePlateInfoList>
    <LicensePlateInfo>
      <id><!--req, xs: string --></id>
      <LicensePlate><!--opt, xs: string--></LicensePlate>
      <type><!--opt, xs: string,
"blackList,whitelist,allVehicleList,otherVehicleList"--></type>
      <createTime><!--opt, xs: string, time in ISO8601 format--></
createTime>
      <direction><!--opt, xs: string, "forward,reverse,unknown"--></direction>
      <laneNo><!--opt, xs: integer, lane No.--></laneNo>
      <plateCategory><!--optional, xs:string, license plate type--></
plateCategory>
      <country><!--optional, xs:string, country/region--></country>
      <area><!--optional, xs:integer, area--></area>
      <effectiveTime><!--optional, xs:date, effective date of the blocklist and
allowlist--></effectiveTime>
      <countryIndex><!--optional, xs:integer, country/region index--></
countryIndex>
      <cardID><!--optional, xs:string, card ID (Wiegand protocol), the maximum
string size is 9 bytes--></cardID>
      <effectiveDateTime><!--optional, xs:datetime, effective date and time of
the blocklist and allowlist--></effectiveDateTime>
      <effectiveStartDate><!--optional, xs:string, start date of the effective
period--></effectiveStartDate>
    </LicensePlateInfo>
  </LicensePlateInfoList>
</LPListAuditSearchResult>
```

## B.11 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>
```

## B.12 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>
```

### B.13 XML\_TrafficChannelCap

Traffic channel capability message in XML format

```
<TrafficChannelCap version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <deviceTypeSupport>
    <!--optional, xs: integer, device type: 0-video monitoring, 1-loop
detection, 2-video analysis, 3-reserved, 4-video analysis (event + traffic
enforcement + traffic data collection), 5-video analysis + capture (event +
traffic enforcement + traffic data collection), 6-video structurization, 7-
speed dome for dynamic tracking and capture, 8-speed dome for non-dynamic
tracking and capture-->
  </deviceTypeSupport>
  <isPTZSupport><!--optional, xs:boolean, whether it supports configuring PTZ
type, PTZ address, PTZ control speed, and zooming speed--></isPTZSupport>
  <isEvidenceGettingSupport>
```

```
<!--optional, xs: boolean, whether it supports violation enforcement,
setting scenes, and scene auto-switch-->
</isEvidenceGettingSupport>
<isBasicSupport>
  <!--optional, xs: boolean, whether it supports setting basic parameters-->
</isBasicSupport>
<isImageMergeSupport>
  <!--optional, xs: boolean, whether it supports picture overlay-->
</isImageMergeSupport>
<isOverlapSupport>
  <!--optional, xs: boolean, whether it supports text overlay-->
</isOverlapSupport>
<isEdfAlgSupport>
  <!--optional, xs: boolean, whether it supports setting analysis parameters--
>
</isEdfAlgSupport>
<isAutoTraceSupport>
  <!--required, xs: boolean, whether it supports auto-tracking-->
</isAutoTraceSupport>
<isEdfManualItsCapSupport>
  <!--optional, xs: boolean, whether it supports setting manual enforcement
parameters-->
</isEdfManualItsCapSupport>
<isViolationTypeStdSupport>
  <!--optional, xs: boolean, whether it supports violation code-->
</isViolationTypeStdSupport>
<isEDFRemoteHostSupport>
  <!--optional, xs: boolean, whether it supports remote host-->
</isEDFRemoteHostSupport>
<isANRSupport>
  <!--optional, xs: boolean, whether it supports ANR-->
</isANRSupport>
<isvoiceTriggerSupport>
  <!--optional, xs: boolean, whether it supports audible linkage-->
</isvoiceTriggerSupport>
<uploadDataTypesSupport>
  <!--optional, xs: string, uploaded data types:
"illegalParking, wrongDirection, crossLane, congestion, parkingEvidence, crossLaneEvi
dence, wrongDirectionEvidence, laneChange, turnRound, laneChangeEvidence, turnRoundEv
idence, vehicleexist, vehicleexistEvidence, edfManualEvidence, objectDroppedDown, smo
ke"-->
</uploadDataTypesSupport>
<aidTypeSupport>
  <!--optional, xs:string, traffic incident types:
"illegalParking, wrongDirection, crossLane, laneChange, turnRound, congestion, vehicle
exist, pedestrian, objectDroppedDown, smoke, edfManual, trafficAccident, construction,
roadBlock, abandonedObject, fogDetection"-->
</aidTypeSupport>
<isIntellMonitorSupport>
  <!--optional, xs: boolean, whether it supports smart monitoring-->
</isIntellMonitorSupport>
<isVCRSupport>
```

```
<!--optional, xs: boolean, whether it supports vehicle statistics-->
</isVCRSupport>
<isEdfManualTrackSupport>
  <!--optional, xs: boolean, whether it support manual tracking and
enforcement-->
  </isEdfManualTrackSupport>
  <videoEvidenceTypeSupport
opt="illegalParking,wrongDirection,crossLane,laneChange,turnRound,vehicleexist,e
dfManualEvidence">
    <!--optional, xs: string, uploaded video evidence types-->
  </videoEvidenceTypeSupport>
  <isRecordParamSupport>
    <!--optional, xs: boolean, whether it supports setting recording
parameters-->
  </isRecordParamSupport>
  <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,
notSlowZebraCrossing, overtakeRightSide, lowSpeed, dragRacing,
changeLaneContinuously, SSharpDriving, largeVehicleOccupyLine, 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>
  <isLaneRuleSupport>
    <!--optional, xs: boolean, whether it supports setting lane rules-->
  </isLaneRuleSupport>
  <isSupportPlateListEvidence>
    <!--optional, xs: boolean, whether it supports setting parameters for
blocklist and allowlist ANPR enforcement-->
  </isSupportPlateListEvidence>
  <isSupportMixedTargetDetection>
```



```
<!--optional, xs: boolean, whether it supports multi-target-type detection-->
</isSupportMixedTargetDetection>
<isSupportVideoOverlays>
  <!--optional, xs: boolean, whether it supports overlaying information on
video-->
  </isSupportVideoOverlays>
  <isSupportAddrInfo>
    <!--optional, xs:boolean, whether it supports overlaying address
information-->
    </isSupportAddrInfo>
    <VehiclePositionControl><!--optional, whether it supports vehicle direction
control-->
      <license min="0" max="16"><!--required, xs:string, license plate number--></
license>
      <intervalTime min="0" max="65535" def="1"><!--required, xs:integer, upload
interval, unit: s, by default it is 1--></intervalTime>
    </VehiclePositionControl>
    <isSupportLicensePlateExposure><!--optional, xs:boolean, whether it supports
license plate exposure, return "true" if it supports, and this node will not be
returned if the device does not supports this function--></
isSupportLicensePlateExposure>
    <isSupportFiltration><!--optional, xs:boolean, whether it supports filtering
duplicated license plate, return "true" if it supports, and this node will not
be returned if the device does not support the function--></isSupportFiltration>
  </TrafficChannelCap>
```

### Example

#### TrafficChannelCap Message Example

```
<TrafficChannelCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <isEvidenceGettingSupport>false</isEvidenceGettingSupport>
  <isBasicSupport>true</isBasicSupport>
  <isImageMergeSupport>true</isImageMergeSupport>
  <isOverlapSupport>true</isOverlapSupport>
  <isEdfAlgSupport>true</isEdfAlgSupport>
  <isEdfManualItsCapSupport>false</isEdfManualItsCapSupport>
  <isEDFRemoteHostSupport>true</isEDFRemoteHostSupport>
  <isANRSupport>true</isANRSupport>
  <isvoiceTriggerSupport>false</isvoiceTriggerSupport>
  <uploadDataTypesSupport></uploadDataTypesSupport>

  <aidTypeSupport>wrongDirection,turnRound,vehicleexist,congestion,crossLane,laneC
hange,pedestrian,roadBlock,abandonedObject,construction,trafficAccident,fogDetec
tion</aidTypeSupport>
  <isIntellMonitorSupport>false</isIntellMonitorSupport>
  <isVCRSupport>false</isVCRSupport>
  <isEdfManualTrackSupport>false</isEdfManualTrackSupport>
  <videoEvidenceTypeSupport></videoEvidenceTypeSupport>
  <TFSEventSupport>wrongDirection,turnRound,vehicleexist,crossLane,laneChange</
TFSEventSupport>
```

```
<AIDEventSupport>congestion,pedestrian,roadBlock,abandonedObject,construction,tr  
afficAccident,fogDetection</AIDEventSupport>  
  <isVehicleStatisticsSupport>true</isVehicleStatisticsSupport>  
  <isLaneRuleSupport>true</isLaneRuleSupport>  
  <isSupportPlateListEvidence>true</isSupportPlateListEvidence>  
</TrafficChannelCap>
```

## Appendix C. Appendixes

### C.1 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_OPERNOPERMIT	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_ADM_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	Ddevice is in N+1 mode. Cloud storage is not supported.
NET_DVR_CLOUD_STORAGE_OPENED	823	Cloud storage mode is enbaled.
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_LIBRARY_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_TRIGGERS	1204	Getting trigger

Error Name	Error Code	Error Description
NET_ERR_ARMED_STATUS	1205	System is in arming status
NET_ERR_PROGRAM_MODE_STATUS	1206	System is in programming mode
NET_ERR_WALK_TEST_MODE_STATUS	1207	System is in pacing measuring mode
NET_ERR_BYPASS_STATUS	1208	Bypass status
NET_ERR_DISABLED_MODULE_STATUS	1209	Function not enabled
NET_ERR_NOT_SUPPORT_OPERATE_ZONE	1210	Operation is not supported by arming region
NET_ERR_NOT_SUPPORT_MOD_MODULE_ADDR	1211	Module address cannot be modified
NET_ERR_UNREGISTERED_MODULE	1212	Module is not registered
NET_ERR_PUBLIC_SUBSYSTEM_ASSOCIATE_SELF	1213	Public sub system associate with its self
NET_ERR_EXCEEDS_ASSOCIATE_SUBSYSTEM_NUM	1214	Number of associated public sub system reached upper limit
NET_ERR_BE_ASSOCIATED_BY_PUBLIC_SUBSYSTEM	1215	Sub system is associated by other public sub system
NET_ERR_ZONE_FAULT_STATUS	1216	Arming region is in failure status
NET_ERR_SAME_EVENT_TYPE	1217	Same event type exists in enable event trigger alarm output and disable event trigger alarm output
NET_ERR_ZONE_ALARM_STATUS	1218	Arming region is in alarm status
NET_ERR_EXPANSION_BUS_SHORT_CIRCUIT	1219	Extension bus short-circuit
NET_ERR_PWD_CONFLICT	1220	Password conflict, e.g., lock password is identical with duress password
NET_ERR_DETECTOR_GISTERED_BY_OTHER_ZONE	1221	Detector has been registered by other arming regions
NET_ERR_DETECTOR_GISTERED_BY_OTHER_PU	1222	Detector has been registered by other hosts
NET_ERR_DETECTOR_DISCONNECT	1223	Detector offline
NET_ERR_CALL_BUSY	1224	Device in call

Error Name	Error Code	Error Description
NET_ERR_FILE_NAME	1357	File name error, empty or invalid
NET_ERR_BROADCAST_BUSY	1358	Device in broadcast
NET_DVR_ERR_LANENUM_EXCEED	1400	Over the number of lanes.
NET_DVR_ERR_PRAREA_EXCEED	1401	Recognition area is too large.
NET_DVR_ERR_LIGHT_PARAM	1402	Signal lamp access parameters error.
NET_DVR_ERR_LANE_LINE_INVALID	1403	Lane configuration error.
NET_DVR_ERR_STOP_LINE_INVALID	1404	Stop line configuration error.
NET_DVR_ERR_LEFTORRIGHT_LINE_INVALID	1405	Turn left / right boundary configuration error.
NET_DVR_ERR_LANE_NO_REPEAT	1406	Overlay lane number repetition.
NET_DVR_ERR_PRAREA_INVALID	1407	The polygon does not meet the requirements.
NET_DVR_ERR_LIGHT_NUM_EXCEED	1408	Video detection of traffic light signal exceeds the maximum number of.
NET_DVR_ERR_SUBLIGHT_NUM_INVALID	1409	Video detection of traffic signal lamp lights are not legitimate
NET_DVR_ERR_LIGHT_AREASIZE_INVALID	1410	The size of the video detection of traffic light input signal lamp is not valid.
NET_DVR_ERR_LIGHT_COLOR_INVALID	1411	The color of the video detection of traffic light input signal lamp color is not legitimate.
NET_DVR_ERR_LIGHT_DIRECTION_INVALID	1412	The direction property of the video detection of traffic light input light is not valid.
NET_DVR_ERR_LACK_IOABLITY	1413	Lack of IO ability.
NET_DVR_ERR_FTP_PORT	1414	FTP port error.
NET_DVR_ERR_FTP_CATALOGUE	1415	FTP catalogue error.
NET_DVR_ERR_FTP_UPLOAD_TYPE	1416	FTP upload type error.
NET_DVR_ERR_FLASH_PARAM_WRITE	1417	Setting param flash write error.

Error Name	Error Code	Error Description
NET_DVR_ERR_FLASH_PARAM_READ	1418	Getting param flash read error.
NET_DVR_ERR_PICNAME_DELIMITER	1419	Pic name delimiter error.
NET_DVR_ERR_PICNAME_ITEM	1420	Pic name item error.
NET_DVR_ERR_PLATE_RECOGNIZE_TYPE	1421	Plate recognize type error.
NET_DVR_ERR_CAPTURE_TIMES	1422	Capture times error.
NET_DVR_ERR_LOOP_DISTANCE	1423	Loop distance error.
NET_DVR_ERR_LOOP_INPUT_STATUS	1424	Loop input status error.
NET_DVR_ERR_RELATE_IO_CONFLICT	1425	Related IO conflict.
NET_DVR_ERR_INTERVAL_TIME	1426	Interval time error.
NET_DVR_ERR_SIGN_SPEED	1427	Sign speed error.
NET_DVR_ERR_PIC_FLIP	1428	Flip is used.
NET_DVR_ERR_RELATE_LANE_NUMBER	1429	Related lane number error.
NET_DVR_ERR_TRIGGER_MODE	1430	Trigger mode error.
NET_DVR_ERR_DELAY_TIME	1431	Delay time error.
NET_DVR_ERR_EXCEED_RS485_COUNT	1432	Exceed RS485 count.
NET_DVR_ERR_RADAR_TYPE	1433	Radar type error.
NET_DVR_ERR_RADAR_ANGLE	1434	Radar angle error.
NET_DVR_ERR_RADAR_SPEED_VALID_TIME	1435	Radar speed valid time error.
NET_DVR_ERR_RADAR_LINE_CORRECT	1436	Radar line correct error.
NET_DVR_ERR_RADAR_CONST_CORRECT	1437	Radar const correct error.
NET_DVR_ERR_RECORD_PARAM	1438	Record param error.
NET_DVR_ERR_LIGHT_WITHOUT_COLOR_AND_DIRECTION	1439	Light number and other param error.

Error Name	Error Code	Error Description
NET_DVR_ERR_LIGHT_WITHOUT_DETECTION_REGION	1440	Light number and detection region error.
NET_DVR_ERR_RECOGNIZE_PROVINCE_PARAM	1441	Plate recognize Province param error.
NET_DVR_ERR_SPEED_TIMEOUT	1442	IO Speed TimeOut Param error.
NET_DVR_ERR_NTP_TIMEZONE	1443	NTP TimeZone Param error.
NET_DVR_ERR_NTP_INTERVAL_TIME	1444	NTP Interval Time error.
NET_DVR_ERR_NETWORK_CARD_NUM	1445	Network Card Num error.
NET_DVR_ERR_DEFAULT_ROUTE	1446	Default Route error.
NET_DVR_ERR_BONDING_WORK_MODE	1447	Banding Work Mode error.
NET_DVR_ERR_SLAVE_CARD	1448	Sub-Card error.
NET_DVR_ERR_PRIMARY_CARD	1449	Primary Card error.
NET_DVR_ERR_DHCP_PPOE_WORK	1450	DHCP and PPOE not Meanwhile start.
NET_DVR_ERR_NET_INTERFACE	1451	Net Interface invalid.
NET_DVR_ERR_MTU	1452	Invalid MTU parameters.
NET_DVR_ERR_NETMASK	1453	Netmask address invalid.
NET_DVR_ERR_IP_INVALID	1454	IP address invalid.
NET_DVR_ERR_MULTICAST_IP_INVALID	1455	Multicast IP address invalid.
NET_DVR_ERR_GATEWAY_INVALID	1456	Gateway address invalid.
NET_DVR_ERR_DNS_INVALID	1457	DNS Param invalid.
NET_DVR_ERR_ALARMHOST_IP_INVALID	1458	AlarmHost IP invalid.
NET_DVR_ERR_IP_CONFLICT	1459	IP address Conflict.
NET_DVR_ERR_NETWORK_SEGMENT	1460	IP not support Multi Network segment.
NET_DVR_ERR_NETPORT	1461	NetPort error.
NET_DVR_ERR_PPPOE_NOSUPPORT	1462	PPPoE is not supported.

Error Name	Error Code	Error Description
NET_DVR_ERR_DOMAINNAME_NOSUPPORT	1463	Not Support Domain Name.
NET_DVR_ERR_NO_SPEED	1464	Speed Not Enabled.
NET_DVR_ERR_IOSTATUS_INVALID	1465	IO Status invalid.
NET_DVR_ERR_BURST_INTERVAL_INVALID	1466	Burst Interval invalid.
NET_DVR_ERR_RESERVE_MODE	1467	Reserve Mode invalid.
NET_DVR_ERR_LANE_NO	1468	Lane No error.
NET_DVR_ERR_COIL_AREA_TYPE	1469	Coil Area Type error.
NET_DVR_ERR_TRIGGER_AREA_PARAM	1470	Trigger Area Param error.
NET_DVR_ERR_SPEED_LIMIT_PARAM	1471	Speed Limit Param error.
NET_DVR_ERR_LANE_PROTOCOL_TYPE	1472	Lane Protocol Type error.
NET_DVR_ERR_INTERVAL_TYPE	1473	Capture Interval Type error.
NET_DVR_ERR_INTERVAL_DISTANCE	1474	Capture Interval Distance error.
NET_DVR_ERR_RS485_ASSOCIATE_DEVTYPE	1475	Rs485 Associate DevType error.
NET_DVR_ERR_RS485_ASSOCIATE_LANENO	1476	Rs485 Associate LaneNo error.
NET_DVR_ERR_LANENO_ASSOCIATE_MULTIRS485	1477	LaneNo Associate 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 dese not exist.
NET_ERR_VQD_CHAN_NO_EXIST	1502	The channel dese 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_RADAR_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	Ringling
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 overlaid.
NET_ERR_FACE_CHAN_UNOVERLAP_EACH_OTHER	2114	The face channels cannot be overlaid.
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_DIFFERENCE	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_CERTIFICATE_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_PLAYLOAD	2128	Deleting is not allowed. The device is cluster host.
NET_DVR_ERR_CLUSTER_DEVNUM_OVER_UPPER_LIMIT	2129	No more devices can be added.

Error Name	Error Code	Error Description
NET_DVR_ERR_CLUSTER_DEVICE_TYPE_INCONFORMITY	2130	Device type mismatched.
NET_DVR_ERR_CLUSTER_DEVICE_VERSION_INCONFORMITY	2131	Device version mismatched.
NET_DVR_ERR_CLUSTER_IP_CONFLICT	2132	Cluster system IP address conflict: ipv4 address conflict, invalid ipv6.
NET_DVR_ERR_CLUSTER_IP_INVALID	2133	Invalid cluster system IP address: invalid ipv4, invalid ipv6.
NET_DVR_ERR_CLUSTER_PORT_CONFLICT	2134	Cluster system port conflict
NET_DVR_ERR_CLUSTER_PORT_INVALID	2135	Invalid cluster system port
NET_DVR_ERR_CLUSTER_USERNAEM_OR_PASSWORD_INVALID	2136	Invalid user name or password
NET_DVR_ERR_CLUSTER_DEVICE_ALREADY_EXIST	2137	The device already exists.
NET_DVR_ERR_CLUSTER_DEVICE_NOT_EXIST	2138	The device does not exist.
NET_DVR_ERR_CLUSTER_NON_CLUSTER_MODE	2139	The device working mode is not the cluster mode .
NET_DVR_ERR_CLUSTER_IP_NOT_SAME_LAN	2140	IP addresses are in different LAN. Building cluster or extending capacity for NVRs in different LAN is not allowed.
NET_DVR_ERR_IDENTITY_KEY	2147	Incorrect interaction password
NET_DVR_MISSING_IDENTITY_KEY	2148	Interaction password is missing
NET_DVR_ERR_CAPTURE_PACKAGE_FAILED	2141	Capturing packets failed.
NET_DVR_ERR_CAPTURE_PACKAGE_PROCESSING	2142	Capturing packet.
NET_DVR_ERR_SAFETY_HELMET_NO_RESOURCE	2143	No enough hard hat detection resource.

Error Name	Error Code	Error Description
NET_DVR_NO_SUPPORT_WITH_ABSTRACT	2144	This function is not supported. Video synopsis is already enabled.
NET_DVR_INSUFFICIENT_DEEP_LEARNING_RESOURCES	2146	No more deep learning resources can be added.
NET_DVR_NO_SUPPORT_WITH_PERSON_DENSITY_DETECT	2149	People gathering density is enabled, it is not supported
NET_DVR_IPC_RESOLUTION_OVERFLOW	2150	The network camera resolution is too large
NET_DVR_IPC_BITRATE_OVERFLOW	2151	The network camera bitrate is too large
NET_DVR_ERR_INVALID_TASKID	2152	Invalid taskID
NET_DVR_PANEL_MODE_NOT_CONFIG	2153	The ATM panel mode is not configured.
NET_DVR_NO_HUMAN_ENGINES_RESOURCE	2154	No enough engine resource
NET_DVR_ERR_TASK_NUMBER_OVERFLOW	2155	No more task data is allowed
NET_DVR_ERR_COLLISION_TIME_OVERFLOW	2156	Collision time is over the limit
NET_DVR_ERR_EVENT_NOTSUPPORT	2159	Subscribing alarm/event is not supported.
NET_DVR_IPC_NUM_REACHES_LIMIT	2184	The max. number of network camera channels reached.
NET_DVR_IOT_NUM_REACHES_LIMIT	2185	The max. number of IoT channels reached
NET_DVR_IOT_CHANNEL_DEVICE_EXIST	2186	Device of the IoT channel already exists.
NET_DVR_IOT_CHANNEL_DEVICE_NOT_EXIST	2187	Device of the IoT channel does not exist.
NET_DVR_INVALID_IOT_PROTOCOL_TYPE	2188	Invalid IoT protocol type
NET_DVR_INVALID_EZVIZ_SECRET_KEY	2189	Invalid verification code

Error Name	Error Code	Error Description
NET_DVR_DUPLICATE_IOT_DEVICE	2190	Duplicated IoT device
NET_DVR_ERROR_NEED_DOUBLE_VERIFICATION	2206	Double verification is required
NET_DVR_NO_DOUBLE_VERIFICATION_USER	2207	No double verification user
NET_DVR_TIMESPAN_NUM_OVER_LIMIT	2209	Max. number of time buckets reached
NET_DVR_CHANNEL_NUM_OVER_LIMIT	2210	Max. number of channels reached
NET_DVR_NO_SEARCH_ID_RESOURCE	2211	Insufficient searchID resources
NET_DVR_SWITCH_TIMEDIFF_LESS_LIMIT	2249	Time difference between power on and off should be less than 10 minutes.
NET_DVR_NO_SUPPORT_DELETE_STRANGER_LIB	2262	Deleting stranger library is not supported
NET_DVR_NO_SUPPORT_CREATE_STRANGER_LIB	2263	Creating stranger library is not supported
NET_DVR_SSD_FILE_SYSTEM_ERROR	2266	SSD file system error
NET_DVR_INSUFFICIENT_SSD__FOR_FPD	2267	Insufficient SSD space for person frequency detection
NET_DVR_SMRDISK_NOT_SUPPORT_RAID	2269	SMR disk does not support RAID.
NET_DVR_ERR_NOTSUPPORT_DEICING	3001	Device does not support deicing function under current status.(Deicing function is only supported under the power status of POE+, AC24V, and DC12V).
NET_DVR_ERR_THERMENABLE_CLOSE	3002	Temperature measurement function is not enabled. (The enable function in NET_DVR_THERMOMETRY_BASICPARAM is not turned on)
NET_DVR_ERR_PANORAMIC_LIMIT_OPERATED	3004	Panoramic map and limit cannot be operated at same time

Error Name	Error Code	Error Description
NET_DVR_ERR_SMARTH264_ROI_OPERATED	3005	SmartH264 and ROI cannot be enabled at the same time.
NET_DVR_ERR_RULENUM_LIMIT	3006	No more rules can be added.
NET_DVR_ERR_LASER_DEICING_OPERATED	3007	Laser and deicing function cannot be enabled at the same time.
NET_DVR_ERR_OFFDIGITALZOOM_OR_MINZOOMLIMIT	3008	Please disable the digital zoom function or set the zoom limit to the minimum value. Otherwise, when enabling smoke and fire detection, behavior analysis, ship detection, defective point correction, temperature measurement, smoke and fire shielding function, this error code will be prompted.
NET_DVR_SYNCHRONIZEFOV_ERROR	3010	Field of view synchronization failed.
NET_DVR_RULE_SHIELDMASK_CONFLICT_ERROR	3013	The rule region conflicts with the shielded area.
NET_DVR_ERR_NO_SAFETY_HELMET_REGION	3501	The hard hat detection area is not configured.
NET_DVR_ERR_UNCLOSED_SAFETY_HELMET	3502	The hard hat detection is enabled.
NET_DVR_UPLOAD_HBDLIBID_ERROR	3504	Incorrect ID of human body picture library (incorrect HBDID or customHBDID)

### RTSP Communication Library Related Errors

Error Name	Error Code	Error Description
NET_DVR_RTSP_ERROR_NOENOUGHPRI	401	Authentication failed: if server returns 401, it will change to this error code
NET_DVR_RTSP_ERROR_ALLOC_RESOURCE	402	Failed to allocate the resource
NET_DVR_RTSP_ERROR_PARAMETER	403	Parameter error



Error Name	Error Code	Error Description
NET_DVR_RTSP_ERROR_NO_URL	404	The assigned URL does not exist: when the server returns 404, SDK turns to this error code. E.g. the channel is not available, or the channel does not support sub stream
NET_DVR_RTSP_ERROR_FORCE_STOP	406	The user forces to exit midway
NET_DVR_RTSP_GETPORTFAILED	407	RTSP port getting error.
NET_DVR_RTSP_DESCRIBERROR	410	RTSP DESCRIBE communicate error
NET_DVR_RTSP_DESCRIBESENDDTIMEOUT	411	Sending "RTSP DESCRIBE" is timeout.
NET_DVR_RTSP_DESCRIBESENDERROR	412	Failed to send "RTSP DESCRIBE".
NET_DVR_RTSP_DESCRIBERECDTIMEOUT	413	Receiving "RTSP DESCRIBE" is timeout.
NET_DVR_RTSP_DESCRIBERECDATALOST	414	Receiving data of "RTSP DESCRIBE" error.
NET_DVR_RTSP_DESCRIBERECDERROR	415	Failed to receive "RTSP DESCRIBE".
NET_DVR_RTSP_DESCRIBESERVERERR	416	"RTSP DESCRIBE, 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_SETUPSENDDTIMEOUT	421	Sending "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPSENDERROR	422	Sending "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRECDTIMEOUT	423	Receiving "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPRECDATALOST	424	Receiving data of "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRECDERROR	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_PLAYERERROR	430	RTSP PLAY interaction error.
NET_DVR_RTSP_PLAYSENDTIMEOUT	431	Sending "RTSP PLAY" is timeout.
NET_DVR_RTSP_PLAYSENDERERROR	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_TEARDOWNSENDTIMEOUT	441	Sending "RTSP TEARDOWN" is timeout.
NET_DVR_RTSP_TEARDOWNSENDERERROR	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_OFFSCREEN_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_MEMERY	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 versio 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_STREAM_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.

**C.2 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.
updateDatabasError	0x3000000F	Updating database failed.

Sub Status Code	Error Code	Description
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.
trafficException	0x30001008	Monitoring point big data server exception.
faceException	0x30001009	Human face big data server exception.
SSDFileSystemsIsError	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.



Sub Status Code	Error Code	Description
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.
unsupportedStreamType	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).
improperPicSize	0x30002026	Improper picture size (The maximum size allowed is 5MB).
URLDownloadFailed	0x30002027	Downloading the picture via the URI failed.
unsupportedImageFormat	0x30002028	Invalid picture format (Only JPG is supported currently).
unsupportedPollingIntervalTime	0x30002029	Invalid polling interval (The interval should be more than 10s).
exceedImagesNumber	0x30002030	The number of pictures exceeds the limit (The platform can apply 1 to 100 picture URIs per time, the maximum number allowed is 100).

Sub Status Code	Error Code	Description
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 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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
fileAlreadyExists	0x40001007	The file already exists.
invalidID	0x40001008	Invalid ID.
backupnodeNotAllowe Log	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.
exportItemsExceedLimi t	0x40001010	No more items can be exported.
noFiles	0x40001011	The file does not exist.
beingExportedByAnoth erUser	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.
analysis unitFull	0x40001017	No more data analysis server can be added.
unitIDError	0x40001018	The data analysis server ID does not exist.
unitExit	0x40001019	The data analysis server already exists in the list.
unitSearch	0x4000101A	Searching data analysis server in the list failed.
unitNotOnline	0x4000101B	The data analysis server is offline.
unitInfoError	0x4000101C	Getting data analysis server information failed.
unitGetNodeInfoError	0x4000101D	Getting node information failed.
unitGetNetworkInfoErr or	0x4000101E	Getting the network information of data analysis server failed
unitSetNetworkInfoErr or	0x4000101F	Setting the network information of data analysis server failed

Sub Status Code	Error Code	Description
setSmartNodeInfoError	0x40001020	Setting node information failed.
setUnitNetworkInfoError	0x40001021	Setting data analysis server network information failed.
unitRestartCloseError	0x40001022	Rebooting or shutting down data analysis server failed.
virtualIPnotAllowed	0x40001023	Adding virtual IP address is not allowed.
unitInstalled	0x40001024	The data analysis server is already installed.
badSubnetMask	0x40001025	Invalid subnet mask.
uintVersionMismatched	0x40001026	Data analysis server version mismatches.
deviceModelMismatched	0x40001027	Adding failed. Device model mismatches.
unitAddNotSelf	0x40001028	Adding peripherals is not allowed.
noValidUnit	0x40001029	No valid data analysis server.
unitNameDuplicate	0x4000102A	Duplicated data analysis server name.
deleteUnitFirst	0x4000102B	Delete the added data analysis server of the node first.
getLocalInfoFailed	0x4000102C	Getting the server information failed.
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.

Sub Status Code	Error Code	Description
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.
userCannotCreatAdmin	0x40001046	The administrator cannot be created.
monitorCamExceed	0x40001048	Up to 3000 cameras can be added.
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.

Sub Status Code	Error Code	Description
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.
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.
TDACConnectFailed	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.

Sub Status Code	Error Code	Description
IDNotExist	0x40001068	The server ID does not exist.
serviceNameReachesLimit	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.
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.
cloudAnalyseOnline	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.



Sub Status Code	Error Code	Description
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.
armingRelationshipReachesLimit	0x4000108A	No more relation can be created.
duplicateArmingName	0x4000108B	The relation name already exists.
noMoreArmingListAdded	0x4000108C	No more blacklist 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.

Sub Status Code	Error Code	Description
searchByPictureTimed Out	0x40001094	Searching picture by picture timeout. Search again.
comparisonTimeRange Error	0x40001095	Comparison time period error.
selectMonitorNumber UpperLimit	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.
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 blacklist 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.

Sub Status Code	Error Code	Description
getProgressFailed	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.
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.
configStructuredAddressFailed	0x400010BA	Configuring IP address of structured server failed.
configPictureServerAddressFailed	0x400010BB	Configuring IP address of picture stored 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	Loopback address is not supported by the main or sub cluster.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
bigDataServerConnect Failed	0x400010D8	Connecting to big data server failed.
microVideoCloudRequ estInfoBuildFailed	0x400010D9	Adding response information of micro video cloud failed.
microVideoCloudRespo nseInfoBuildFailed	0x400010DA	Parsing response information of micro video cloud failed.
transcodingServerRequ estInfoBuildFailed	0x400010DB	Adding response information of transcoding server failed.
transcodingServerResp onseInfoParseFailed	0x400010DC	Parsing response information of transcoding server failed.
transcodingServerOffli ne	0x400010DD	Transcoding server is offline.
microVideoCloudOfflin e	0x400010DE	Micro video cloud is offline.
UPSServerOffline	0x400010DF	UPS monitor server is offline.
statisticReportRequestI nfoBuildFailed	0x400010E0	Adding response information of statistics report failed.
statisticReportRespons eInfoParseFailed	0x400010E1	Parsing response information of statistics report failed.
DisplayConfigInfoBuild Failed	0x400010E2	Adding display configuration information failed.
DisplayConfigInfoParse Failed	0x400010E3	Parsing display configuration information failed.
DisplayConfigInfoSaveF ailed	0x400010E4	Saving display configuration information failed.
notSupportDisplayConf igType	0x400010E5	The display configuration type is not supported.
passError	0x400010E7	Incorrect password.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
bigDataServerRequestInfoBuildFail	0x40001108	Creating request message of big data service failed.
analysisTimedOut	0x40001109	Analysis time out.
high-performanceModeDisabled.	0x4000110A	Please enable high-performance mode.
configuringUPSMonitoringServerTimedOut	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.

Sub Status Code	Error Code	Description
cloudAnalysisFacePictureQualityRatingFailed	0x40001110	Getting face quality grading of cloud analytic service failed.
cloudAnalysisExtractFeaturePointsFailed	0x40001111	Extracting feature of cloud analytic service failed.
cloudAnalysisExtractPropertyFailed	0x40001112	Extracting property of cloud analytic service failed.
getAddedNodeInformationFailed	0x40001113	Getting the added nodes information of data analysis server failed.
noMoreAnalysisUnitsAdded	0x40001114	No more data analysis servers can be added.
detectionAreaInvalid	0x40001115	Invalid detection region.
shieldAreaInvalid	0x40001116	Invalid shield region.
noMoreShieldAreasAdded	0x40001117	No more shield region can be drawn.
onlyAreaOfRectangleShapeAllowed	0x40001118	Only drawing rectangle is allowed in detection area.
numberReachedLimit	0x40001119	Number reached the limit.
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.



Sub Status Code	Error Code	Description
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.
timeRulesParseFailed	0x4000112E	Parsing time rule failed.
monitoInfoInvalid	0x4000112F	Invalid camera information.
addingFailedVersionMismatches	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.

Sub Status Code	Error Code	Description
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.
theArmingLinkageIdDoesNotExist	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.
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.

Sub Status Code	Error Code	Description
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.
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 service.

Sub Status Code	Error Code	Description
theCloudAnalyticsIsInterrupted	0x4000115D	The operation of cloud analytic service 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 service overtime.
noDataOfTheCurrentLibraryFound	0x40001167	No data in the current library. Make sure there is data in the Hbase.
noFacePictureLibraryIsArmed	0x40001168	No face picture library is armed for big data service.
noAvailableDataSlicingVersionInformationArmedFirstAndSliceTheData	0x40001169	Invalid standard version information.
duplicatedOperationDataSlicingIsExecuting	0x4000116A	Slicing failed. Duplicated operation.
slicingDataFailedNoArmedFacePictureLibrary	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.

Sub Status Code	Error Code	Description
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.
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 service error.
engineModuleError	0x4000117D	Engine module of cloud analytic service error.

Sub Status Code	Error Code	Description
streamingServiceError	0x4000117E	Streaming component of cloud analytic service error.
faceAnalysisModuleError	0x4000117F	Face analysis module of cloud analytic service error.
vehicleAnalysisModuleError	0x40001180	Vehicle pictures analytic module of cloud analytic service error.
videoStructuralAnalysisModuleError	0x40001181	Video structuring module of cloud analytic service error.
postprocessingModuleError	0x40001182	Post-processing module of cloud analytic service 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.
settingAlgorithmTypeForDeviceFailed	0x40001187	Allocating task resource failed.
gettingHighFrequencyPersonDetectionAlarmInformationFailed	0x40001188	Setting frequently appeared person alarm failed.
invalidSearchCondition	0x40001189	Invalid result.
theTaskIsNotCompleted	0x4000118B	The task is not completed.
resourceOverRemainLimit	0x4000118C	No more resource can be pre-allocated.
frequentlyAppearedPersonAlarmIsAlreadyConfiguredForTheCameraDisarmFirstAndTryAgain	0x4000118D	The frequently appeared person alarm of this camera is configured. Delete the arming information and try again.

Sub Status Code	Error Code	Description
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.
SMRDiskNotSupportRaid	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.
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.

Sub Status Code	Error Code	Description
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.
AITargetBPDuplicateName	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.
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.
packNotFound	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.



Sub Status Code	Error Code	Description
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.
audiosPlayingPleaseWait	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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
masterSlavelEnable	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.

Sub Status Code	Error Code	Description
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 (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.

Sub Status Code	Error Code	Description
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.
POSCannotAddHolidayPlan	0x400015FF	No POS events during holidays.
eventTypesTooLong	0x40001600	Event type is too long.
eventNamesTooLong	0x40001601	Event name is too long.
PerimeterEnginesNoResource	0x40001602	No more perimeter engines.
invalidProvinceCode	0x40001607	Invalid province code.

### StatusCode=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.
connectServerFail	0x60000009	Connecting to server failed.

Sub Status Code	Error Code	Description
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.
incorrentUserNameOrPassword	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.
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.

Sub Status Code	Error Code	Description
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
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
analysisEnginesNoResourceError	0x60001000	No analysis engine.
analysisEnginesUsageExcced	0x60001001	The engine usage is overloaded.

Sub Status Code	Error Code	Description
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.
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.



Sub Status Code	Error Code	Description
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.
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

Sub Status Code	Error Code	Description
timeSpanNumOverLimit	0x60001044	Max. number of time buckets reached
channelNumOverLimit	0x60001045	Max. number of channels reached
noSearchIDResource	0x60001046	Insufficient searchID resources
noSupportDeleteStrangerLib	0x60001051	Deleting stranger library is not supported
noSupportCreateStrangerLib	0x60001052	Creating stranger library is not supported
behaviorAnalysisRuleInfoError	0x60001053	Behavior analysis rule parameters error.
safetyHelmetParamError	0x60001054	Hard hat parameters error.
OneChannelOnlyCanBindOneEngine	0x60001077	No more engines can be bound.
engineTypeMismatch	0x60001079	Engine type mismatched.
badUpgradePackage	0x6000107A	Invalid upgrade package.
AudioFileNameDuplicate	0x60001135	Duplicated audio file name.
CurrentAudioFileAIRuleInUseAlreadyDelete	0x60001136	The AI rule linkage related to current audio file has been deleted.
TransitionUseEmmc	0x60002000	Starting device failed. The EMMC is overused.
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).

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
personInfoExtendValuesTooLong	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 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.

### C.3 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.	
deviceNotActivated	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.
deviceResourceNotEnough	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.
	0x00100103	The stream type is not supported.	Check the requested stream type.

Error String	Error Code	Description	Debugging Suggestion
	0x00100104	The number of connections exceeded limit.	Reduce the number of streaming clients and try again.
	0x00100105	Not enough bandwidth.	Reduce the number of remote streaming channels.

### User Function Module (Error Codes Range: from 0x00200001 to 0x002fffff)

Error String	Error Code	Description	Debugging Suggestion
passwordError	0x00200001	Incorrect user name or password.	Check if the password is correct.
userNameNotExist	0x00200002	The account does not exist.	Check if the account exists, or add the account.
userNameLocked	0x00200003	The account is locked.	Wait for the device to unlock.
userNumLimited	0x00200004	The number of users allowed to log in exceeded the upper limit.	Log out.
lowPrivilege	0x00200005	No permissions for this operation	<p>For users operations, check the following situations:</p> <ul style="list-style-type: none"> <li>• Deleting your own account is not allowed.</li> <li>• Editing your own level or permission is not allowed.</li> <li>• Getting information about users with higher permission is not allowed.</li> <li>• Elevating the user's level or permission is not allowed.</li> </ul> <p>For other operations, check according to the following measures: If operations unrelated to user's permission configuration failed, you can check the user type and permission, if not solved, contact the developers.</p>

Error String	Error Code	Description	Debugging Suggestion
incorrentUserNa meOrPassword	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.
passwordMustCo ntainMorethan8C haracters	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.
passwordLenNo MoreThan16	0x00200009	The password length cannot be greater than 16.	Check if the password length is greater than 16. If yes, change password again.
adminUserNotAll owedModify	0x0020000a	Editing admin information is not allowed.	Check if the edited account is admin.
confirmPassword Error	0x0020000b	Incorrect confirm password.	Check the confirm password.
passwordMustCo ntainMorethan2T ypes	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.
passwordContain UserName	0x0020000d	The password cannot contain the user name.	Check if the password contains the user name.
userPwdNameMi rror	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.
connectServerFail	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 0x005fffff)

Error String	Error Code	Description	Debugging Suggestion
upgradeXMLFormatError	0x00500001	Incorrect XML upgrading request.	Check if the upgrade file is correct. If the file is correct, try the local upgrade.
upgradeContentError	0x00500002	Incorrect upgrading request content.	Check if the upgrade file is correct. If the file is correct, try the local upgrade.
noUpgradePermission	0x00500003	No upgrade permission.	Switch to admin account or ask admin for advanced operation permission.
upgrading	0x00500004	Upgrading...	Wait for the upgrade to complete.
receiveUpgradePackageError	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
upgradePackageSizeMismatch	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 stream.
	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.	
codingFormatNotMatch	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.
destinationLongNumberError	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.
notFindStorageMedium	0x00a00002	No storage medium found.	
videoDownloadFailed	0x00a00003	Downloading videos failed.	



**Picture Storage Module (Error Codes Range: from 0x00b00001 to 0x00bfffff)**

Error String	Error Code	Description	Debugging Suggestion
	0x00b00001	Searching pictures failed.	No picture resource.

**IO Function Module (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 0x00dfffff)**

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 0x00ffffff)**

Error String	Error Code	Description	Debugging Suggestion
noMemory	0x00f00001	Insufficient device memory (heap space allocation failed).	Check the free memory and send logs to the developer for analysis.
deviceBusy	0x00f00002	The device is busy or the device is not responding.	Send logs to the developers for analysis.

Error String	Error Code	Description	Debugging Suggestion
			For fingerprint collection, face collection, file application, and file uploading services, check if the last operation is completed.
notSupport	0x00f00003	The URL is not supported by the device.	Capture the packets, check if the applied URL exists in the PMP platform. If yes, send the URL to the developer for analysis.
methodNotAllowed	0x00f00004	HTTP method is not allowed.	Capture the packets, check the method corresponding to the URL in the PMP platform.
invalidOperation	0x00f00005	Invalid operation of API command.	
IDNotExist	0x00f00006	The ID does not exist (the URL should contain ID, but the actual URL does not contain the ID).	Capture the packets and check if the ID included in the URL is correct.
invalidID	0x00f00007	Invalid ID (the ID in the URL exceeds the capability set or the ID format is invalid).	Capture the packets and check if the ID included in the URL is correct. Get the capabilities of URL and check the ID range.
invalidIURL	0x00f00008	The content after the "?" in the URL is wrong.	Capture the packets and check if the URL is correct.
deviceAckTimeout	0x00f00009	Device response timed out.	If the communication with the external module timed out, check if the external module is offline. When the above situation is eliminated, send logs to the developer for analysis.
badXmlFormat	0x00f0000a	XML format error	

Error String	Error Code	Description	Debugging Suggestion
badJsonFormat	0x00f0000b	JSON format error	
badURLFormat	0x00f0000c	URL format error	Get the URL and check if it is correct.
badXmlContent	0x00f0000d	XML message error: <ul style="list-style-type: none"> <li>The message contains only URL but no message body</li> <li>The required node is not configured.</li> <li>Node value exceeds the range limit (incorrect node value).</li> </ul>	
badJsonContent	0x00f0000e	JSON message error: <ul style="list-style-type: none"> <li>The message contains only URL but no message body</li> <li>The required node is not configured.</li> <li>Node value exceeds the range limit (incorrect node value).</li> </ul>	
messageParametersLack	0x00f0000f	The required node does not exist.	
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
multiAuthenticationFailed	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.
bondIpAddressNotMatch	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 0x015fffff)

Error String	Error Code	Description	Debugging Suggestion
materialDownloadFailed	0x01500001	Material download failed.	<ul style="list-style-type: none"> <li>Check if the network connection is normal.</li> <li>Check if the device is running normally.</li> <li>Check the log print.</li> </ul>
materialNumberIsOver	0x01500002	The number of materials in the program list reached the upper limit.	Check if the number of materials in applied program list exceeded the limit.

## C.4 Region Code

Here is the code list of the regions that are supported by the algorithm library for license plate recognition.

region (Region Code)	Description
ER	Russian Region
EU	Europe Region
EUandCIS	EU&CIS
ME	Middle East
other	Other
APAC	Asia-Pacific Region
AFandAM	Africa and America
THAandLA	Thailand and Laos
HKandMO	Hong Kong and Macao
India	India
All	All regions

## C.5 Country/Region Code

Here is the code list of the countries and regions that are supported by the algorithm library for license plate recognition.

CRIndex (Country/Region Code)	Description
0	Not supported by the algorithm
1	Czech Republic
2	France
3	Germany
4	Spain
5	Italy
6	Netherlands
7	Poland

CRIndex (Country/Region Code)	Description
8	Slovakia
9	Belarus
10	Moldova
11	Russia
12	Ukraine
13	Belgium
14	Bulgaria
15	Denmark
16	Finland
17	United Kingdom
18	Greece
19	Croatia
20	Hungary
21	Israel
22	Luxembourg
23	Macedonia (changed to North Macedonia in 2018)
24	Norway
25	Portuga
26	Romania
27	Serbia
28	Azerbaijan
29	Georgia
30	Kazakhstan
31	Lithuania
32	Turkmenistan
33	Uzbekistan
34	Latvia
35	Estonia

CRIndex (Country/Region Code)	Description
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	Montenegro
49	Kosovo
50	Andorra
51	Armenia
52	Monaco
53	Liechtenstein
54	San Marino
55	Reserved
56	Reserved
57	Reserved
58	Reserved
59	China
60	Bahrain
61	South Korea
62	Lebanon
63	Nepal



CRIndex (Country/Region Code)	Description
64	Thailand
65	Pakistan
66	United Arab Emirates
67	Bhutan
68	Oman
69	North Korea
70	Philippines
71	Cambodia
72	Qatar
73	Kyrgyzstan
74	Maldives
75	Malaysia
76	Mongolia
77	Saudi Arabia
78	Brunei
79	Laos
80	Japan
81	Turkey
82	Palestinian
83	Tajikistan
84	Kuwait
85	Syria
86	India
87	Indonesia
88	Afghanistan
89	Sri Lanka
90	Iraq
91	Vietnam

CRIndex (Country/Region Code)	Description
92	Iran
93	Yemen
94	Jordan
95	Myanmar
96	Sikkim
97	Bangladesh
98	Singapore
99	Democratic Republic of Timor-Leste
100	Reserved
101	Reserved
102	Reserved
103	Reserved
104	Egypt
105	Libya
106	Sudan
107	Tunisia
108	Algeria
109	Morocco
110	Ethiopia
111	Eritrea
112	Somalia Democratic
113	Djibouti
114	Kenya
115	Tanzania
116	Uganda
117	Rwanda
118	Burundi
119	Seychelles

CRIndex (Country/Region Code)	Description
120	Chad
121	Central African
122	Cameroon
123	Equatorial Guinea
124	Gabon
125	Congo
126	Democratic Republic of the Congo
127	Sao Tome and Principe
128	Mauritania
129	Western Sahara
130	Senegal
131	Gambia
132	Mali
133	Burkina Faso
134	Guinea
135	Guinea-Bissau
136	Cape Verde
137	Sierra Leone
138	Liberia
139	Ivory Coast
140	Ghana
141	Togo
142	Benin
143	Niger
144	Zambia
145	Angola
146	Zimbabwe
147	Malawi

CRIndex (Country/Region Code)	Description
148	Mozambique
149	Botswana
150	Namibia
151	South Africa
152	Swaziland
153	Lesotho
154	Madagascar
155	Comoros
156	Mauritius
157	Nigeria
158	South Sudan
159	Saint Helena
160	Mayotte
161	Reunion
162	Canary Islands
163	AZORES
164	Madeira
165	Reserved
166	Reserved
167	Reserved
168	Reserved
169	Canada
170	Greenland Nuuk
171	Pierre and Miquelon
172	United State
173	Bermuda
174	Mexico
175	Guatemala

CRIndex (Country/Region Code)	Description
176	Belize
177	El Salvador
178	Honduras
179	Nicaragua
180	Costa Rica
181	Panama
182	Bahamas
183	Turks and Caicos Islands
184	Cuba
185	Jamaica
186	Cayman Islands
187	Haiti
188	Dominican
189	Puerto Rico
190	United States Virgin Islands
191	British Virgin Islands
192	Anguilla
193	Antigua and Barbuda
194	Collectivité de Saint-Martin
195	Autonomous country
196	Saint-Barthélemy
197	Saint Kitts and Nevis
198	Montserrat
199	Guadeloupe
200	Dominica
201	Martinique
202	St. Lucia
203	Saint Vincent and the Grenadines

CRIndex (Country/Region Code)	Description
204	Grenada
205	Barbados
206	Trinidad and Tobago
207	Curaçao
208	Aruba
209	Netherlands Antilles
210	Colombia
211	Venezuela
212	Guyana
213	Suriname
214	Guyane Francaise
215	Ecuador
216	Peru
217	Bolivia
218	Paraguay
219	Chile
220	Brazil
221	Uruguay
222	Argentina
223	Reserved
224	Reserved
225	Reserved
226	Reserved
227	Australia
228	New Zealand
229	Papua New Guinea
230	Salomonen
231	Vanuatu

CRIndex (Country/Region Code)	Description
232	New Caledonia
233	Palau
234	Federated States of Micronesia
235	Marshall Island
236	Northern Mariana Islands
237	Guam
238	Nauru
239	Kiribati
240	Fidschi
241	Tonga
242	Tuvalu
243	Wallis et Futuna
244	Samoa
245	Eastern Samoa
246	Tokelau
247	Niue
248	Cook Islands
249	French Polynesia
250	Pitcairn Islands
251	Hawaii State
252	Reserved
253	Reserved
254	Unrecognized
255	ALL
256	Taiwan (China)
257	Hong Kong (China)
258	Macau (China)

