Device Network SDK (Video Intercom)

Developer Guide

Legal Information

Contents

Chapter 1 Overview 1
1.1 Introduction 1
1.2 Update History 1
Chapter 2 Start Video Intercom 3
Chapter 3 Manage Ring 5
Chapter 4 Get Registration Information of Door Station 7
Chapter 5 Configure Video Intercom Alarm or Event 9
Chapter 6 Alarm/Event Receiving 11
6.1 Receive Alarm/Event in Arming Mode 11
6.2 Receive Alarm/Event in Listening Mode 14
Chapter 7 Remote Configuration 18
Chapter 8 Module Configuration 22
Chapter 9 Maintenance 23
Chapter 10 API Reference 24
10.1 NET_DVR_Cleanup 24
10.2 NET_DVR_GetDeviceAbility 24
10.3 NET_DVR_GetDeviceConfig 25
10.4 NET_DVR_GetDVRConfig 26
10.5 NET_DVR_GetErrorMsg 27
10.6 NET_DVR_GetLastError 28
10.7 NET_DVR_GetNextRemoteConfig 28
10.8 NET_DVR_GetUploadState 29
10.9 NET_DVR_Init 31
10.10 NET_DVR_Login_V40 32
10.10.1 fLoginResultCallBack 32
10.11 NET_DVR_Logout 33

10.12 NET_DVR_SendRemoteConfig 33
10.13 NET_DVR_SetDeviceConfig 34
10.14 NET_DVR_SetDVRConfig 35
10.15 NET_DVR_SetSDKInitCfg 36
10.16 NET_DVR_StartRemoteConfig 38
10.16.1 fRemoteConfigCallback 39
10.17 NET_DVR_STDXMLConfig 40
10.18 NET_DVR_StopRemoteConfig 41
10.19 NET_DVR_UploadClose 42
10.20 NET_DVR_UploadFile_V40 42
10.21 NET_DVR_CloseAlarmChan_V30 43
10.22 NET_DVR_GetDVRConfig 43
10.23 NET_DVR_SetDVRConfig 45
10.24 NET_DVR_SetDVRMessageCallBack_V50 45
10.24.1 MSGCallBack 53
10.25 NET_DVR_SetupAlarmChan_V50 53
10.26 NET_DVR_StartListen_V30 54
10.27 NET_DVR_StopListen_V30 55
Appendix A. Appendixes 57
A.1 Data Structure 57
A.1.1 NET_DVR_AGAIN_RELATEDEV 57
A.1.2 NET_DVR_AUTH_INFO 58
A.1.3 NET_DVR_CALL_ROOM_CFG 58
A.1.4 NET_DVR_DEVICEINFO_V30 59
A.1.5 NET_DVR_DEVICEINFO_V40 63
A.1.6 NET_DVR_DIRECT_CONNECT_CHAN_INFO 66
A.1.7 NET_DVR_INDOOR_UNIT_DEVICEID 68
A.1.8 NET_DVR_INDOOR_UNIT_RELATEDEV 68

A.1.9 NET_DVR_INDOOR_UNIT_OPERATION_TIME_CFG 69
A.1.10 NET_DVR_INIT_CFG_ABILITY 70
A.1.11 NET_DVR_IPADDR_UNION 71
A.1.12 NET_DVR_LOCAL_SDK_PATH 71
A.1.13 NET_DVR_MANAGE_UNIT_DEVICEID 71
A.1.14 NET_DVR_MANAGE_UNIT_RELATEDEV 72
A.1.15 NET_DVR_MANAGE_UNIT_OPERATION_TIME_CFG 72
A.1.16 NET_DVR_MIME_UNIT 73
A.1.17 NET_DVR_NOTICEDATA_RECEIPT_INFO 74
A.1.18 NET_DVR_OUTDOOR_FENCE_DEVICEID 74
A.1.19 NET_DVR_OUTDOOR_UNIT_DEVICEID 75
A.1.20 NET_DVR_OUTDOOR_UNIT_RELATEDEV 76
A.1.21 NET_DVR_OUTDOOR_UNIT_OPERATION_TIME_CFG 76
A.1.22 NET_DVR_PRIVILEGE_PASSWORD_CFG 77
A.1.23 NET_DVR_PROTO_TYPE 78
A.1.24 NET_DVR_PU_STREAM_URL 81
A.1.25 NET_DVR_PU_STREAM_URL_CFG 82
A.1.26 NET_DVR_RING_SEARCH_CFG 84
A.1.27 NET_DVR_RING_SEARCH_COND 84
A.1.28 NET_DVR_RING_UPLOAD_CFG 85
A.1.29 NET_DVR_SEND_CARD_INFO 86
A.1.30 NET_DVR_STREAM_INFO 86
A.1.31 NET_DVR_STREAM_MODE_TYPE 87
A.1.32 NET_DVR_STREAM_SRC_CFG 88
A.1.33 NET_DVR_STREAM_SRC_INFO 88
A.1.34 NET_DVR_STREAM_TYPE_UNION 89
A.1.35 NET_DVR_TIME_EX 89
A.1.36 NET_DVR_UNLOCK_RECORD_INFO 90

A.1.37 NET_DVR_USER_LOGIN_INFO 92
A.1.38 NET_DVR_VIDEO_CALL_COND 93
A.1.39 NET_DVR_VIDEO_CALL_PARAM 93
A.1.40 NET_DVR_VIDEOINTERCOM_STREAM 94
A.1.41 NET_DVR_VIDEO_INTERCOM_ALARM 95
A.1.42 NET_DVR_VIDEO_INTERCOM_ALARM_CFG 96
A.1.43 NET_DVR_VIDEO_INTERCOM_ALARM_INFO_UNION 96
A.1.44 NET_DVR_VIDEO_INTERCOM_DEVICEID_CFG 97
A.1.45 NET_DVR_VIDEO_INTERCOM_EVENT 98
A.1.46 NET_DVR_VIDEO_INTERCOM_EVENT_INFO_UINON 99
A.1.47 NET_DVR_VIDEO_INTERCOM_IOIN_CFG 99
A.1.48 NET_DVR_VIDEO_INTERCOM_IOOUT_CFG 100
A.1.49 NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_CFG 100
A.1.50 NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_UNION 101
A.1.51 NET_DVR_VIDEO_INTERCOM_RELATEDEV_CFG 102
A.1.52 NET_DVR_VIDEO_INTERCOM_UNIT_DEVICEID_UNION 102
A.1.53 NET_DVR_VIDEO_INTERCOM_UNIT_RELATEDEV_UNION 104
A.1.54 NET_DVR_VIS_REGISTER_INFO 104
A.1.55 NET_DVR_XML_CONFIG_INPUT 105
A.1.56 NET_DVR_XML_CONFIG_OUTPUT 106
A.1.57 NET_DVR_ZONE_ALARM_INFO 107
A.1.58 NET_SDK_CALLBACK_STATUS_NORMAL 108
A.1.59 NET_SDK_UPLOAD_TYPE 108
A.1.60 NET_ALARM_CVR_SUBINFO_UNION 111
A.1.61 NET_ALARM_RECORD_EXCEPTION 112
A.1.62 NET_ALARM_RECORDFILE_LOSS 112
A.1.63 NET_ALARM_RESOURCE_USAGE 113
A.1.64 NET_ALARM_STREAM_EXCEPTION 113

A.1.65 NET_DVR_ALARMER 114
A.1.66 NET_DVR_ALARMINFO_DEV 114
A.1.67 NET_DVR_ALARMINFO_DEV_V40 115
A.1.68 NET_DVR_ALARMINFO_V30 116
A.1.69 NET_DVR_ALARMINFO_V40 117
A.1.70 NET_DVR_ALRAM_FIXED_HEADER 118
A.1.71 NET_DVR_ALARM_ISAPI_INFO 121
A.1.72 NET_DVR_ALARM_ISAPI_PICDATA 122
A.1.73 NET_DVR_ETHERNET_V30 123
A.1.74 NET_DVR_IPADDR_UNION 123
A.1.75 NET_DVR_NETCFG_V50 124
A.1.76 NET_DVR_PPPOECFG 125
A.1.77 NET_DVR_SETUPALARM_PARAM_V50 126
A.1.78 NET_DVR_TIME 128
A.1.79 NET_DVR_TIME_EX 129
A.2 Request URL 129
A.2.1 /ISAPI/VideoIntercom/callerInfo/capabilities?format=json 129
A.2.2 /ISAPI/VideoIntercom/callerInfo?format=json 130
A.2.3 /ISAPI/VideoIntercom/capabilities 130
A.2.4 /ISAPI/VideoIntercom/IntercomProtocolType 131
A.2.5 /ISAPI/VideoIntercom/IntercomProtocolType/capabilities 131
A.2.6 /ISAPI/VideoIntercom/keyCfg 132
A.2.7 /ISAPI/VideoIntercom/keyCfg/<ID> 132
A.2.8 /ISAPI/VideoIntercom/keyCfg/<ID>/capabilities 133
A.2.9 /ISAPI/VideoIntercom/passwordAuthentication 133
A.2.10 /ISAPI/VideoIntercom/passwordAuthentication/capabilities 134
A.2.11 /ISAPI/VideoIntercom/ring 134
A.2.12 /ISAPI/VideoIntercom/ring/capabilities 134

A.2.13 /ISAPI/VideoIntercom/ring/ringID/<ID> 135
A.2.14 /ISAPI/VideoIntercom/scene/nowMode 135
A.2.15 /ISAPI/VideoIntercom/scene/nowMode/capabilities 136
A.2.16 /ISAPI/VideoIntercom/SubModules/<ID>/CallCfg/capabilities?format=json 136
A.2.17 /ISAPI/VideoIntercom/SubModules/<ID>/CallCfg?format=json 137
A.2.18 /ISAPI/VideoIntercom/SubModules/<ID>/Configurations/capabilities?format=json 137
A.2.19 /ISAPI/VideoIntercom/SubModules/<ID>/Configurations?format=json 138
A.2.20 /ISAPI/VideoIntercom/SubModules/capabilities?format=json 139
A.2.21 /ISAPI/VideoIntercom/SubModules?format=json 139
A.2.22 /ISAPI/VideoIntercom/WorkStatus 139
A.3 Request and Response Message 140
A.3.1 JSON_CallCfg 140
A.3.2 JSON_CallerInfo 140
A.3.3 JSON_Cap_CallCfg 141
A.3.4 JSON_Cap_CallerInfo 142
A.3.5 JSON_Cap_SubModules 143
A.3.6 JSON_Cap_SubModulesCfg 143
A.3.7 JSON_EventNotificationAlert_Alarm/EventInfo 144
A.3.8 JSON_EventNotificationAlert_VideoIntercomEventMsg 145
A.3.9 JSON_ResponseStatus 147
A.3.10 JSON_SubModules 147
A.3.11 JSON_SubModulesCfg 148
A.3.12 XML_Cap_IntercomProtocolType 148
A.3.13 XML_Cap_KeyCfg 148
A.3.14 XML_Cap_PasswordAuthenticationCfg 149
A.3.15 XML_Cap_RingParam 149
A.3.16 XML_Cap_SceneNowMode 149

A.3.17 XML_EventNotificationAlert_AlarmEventInfo 149
A.3.18 XML_IntercomProtocolType 150
A.3.19 XML_IpViewDevAbility 150
A.3.20 XML_KeyCfg 157
A.3.21 XML_KeyCfgList 157
A.3.22 XML_PasswordAuthenticationCfg 157
A.3.23 XML_ResponseStatus 158
A.3.24 XML_SceneNowMode 158
A.3.25 XML_VideoIntercomCap 158
A.3.26 XML_WorkStatus 160
A.4 Device Network SDK Errors 160
A.5 Response Codes of Text Protocol 203

Chapter 1 Overview

1.1 Introduction

The video intercom is a stand-alone intercom system used to manage calls made at the entrance to a building (e.g., residential complex, detached family home, workplace, etc.) with access controlled by audiovisual communication between the inside and outside. This manual introduces the typical applications (e.g., starting video intercom, ring management, and so on) and video intercom APIs of device network SDK (hereafter referred to as "HCNetSDK").

1.2 Update History

The update history shows the summary of changes in HCNetSDK integrations according to different products or projects, and the updated time.

Summary of Changes in Version 6.1.4.35_May, 2020

Related Product: DS-K1T341B Series Face Recognition Terminal with Software Version 1.1
Extended configuration capability message ***XML_Cap_KeyCfg*** and parameter message ***XML_KeyCfg*** of pressing the button to call (related URIs: ***/ISAPI/VideoIntercom/keyCfg/<ID>/capabilities*** and ***/ISAPI/VideoIntercom/keyCfg/<ID>*** ; related API: ***NET_DVR_STDXMLConfig***): added a node <**callMethod**> (calling method).

Summary of Changes in Version 6.1.0.151_July, 2019

Related Product: DS-KH8350 Series, DS-KH6320 Series, and DS-KH8520 Series Indoor Station; DS-KD8003 Series Door Station

1. Extended operation time parameter structure ***NET_DVR_INDOOR_UNIT_OPERATION_TIME_CFG*** of indoor station (related APIs: ***NET_DVR_GetDVRConfig*** and ***NET_DVR_SetDVRConfig*** ; commands: 16004-"NET_DVR_GET_OPERATION_TIME_CFG" and 16005-"NET_DVR_SET_OPERATION_TIME_CFG"): added a member **dwRingDurationTime** (ringing timeout).
2. Extended capability message ***JSON_Cap_SubModules*** of getting sub module status and the sub module status message ***JSON_SubModules*** (related API: ***NET_DVR_STDXMLConfig*** ; URLs: ***/ISAPI/VideoIntercom/SubModules/capabilities?format=json*** and ***/ISAPI/VideoIntercom/SubModules?format=json***): added 12 values to the node "**moduleType**" (module type), i.e., "DS-1101M", "DS-1102M", "DS-1103M", "DS-1104M", "DS-1107M", "DS-1108M", "DS-1101MK", "DS-1102MK", "DS-1103MK", "DS-1104MK", "DS-1107MK", and "DS-1108MK".
3. Extended configuration capability ***JSON_Cap_CallCfg*** of nametag module and nametag module parameter message ***JSON_CallCfg*** (related API: ***NET_DVR_STDXMLConfig*** ; URLs: ***/ISAPI/***

VideoIntercom/SubModules/<ID>/CallCfg/capabilities?format=json and ***/ISAPI/VideoIntercom/SubModules/<ID>/CallCfg?format=json***):
added a node "**callNumber**" (call number).

4. Extended configuration capability ***JSON_Cap_SubModulesCfg*** of sub module and sub module parameter message ***JSON_SubModulesCfg*** (related API: ***NET_DVR_STDXMLConfig*** ; URLs: ***/ISAPI/VideoIntercom/SubModules/<ID>/Configurations/capabilities?format=json*** and ***/ISAPI/VideoIntercom/SubModules/<ID>/Configurations?format=json***):
added a node "**infoMode**" (indicator mode).
5. Extended parameter structure ***NET_DVR_CALL_ROOM_CFG*** of calling resident by button (related APIs: ***NET_DVR_GetDVRConfig*** and ***NET_DVR_SetDVRConfig*** ; commands: 16028-"NET_DVR_GET_CALL_ROOM_CFG" and 16029-"NET_DVR_SET_CALL_ROOM_CFG"):
added the support of member **byCalledName** in non-standard SIP mode (this member represents the room No.).
6. Extended video intercom device capability ***XML_IpViewDevAbility*** (related API: ***NET_DVR_GetDeviceAbility*** ; capability type: 0x014-"IP_VIEW_DEV_ABILITY"):
added a sub node <**dwRingDurationTime**> (ringing time duration) to the node <**OperationTime**>.

Summary of Changes in Version 6.0.2.30_July, 2019

Related Product: DS-KD-DIS Display Module

1. Extended capability message ***JSON_Cap_SubModules*** of getting sub module status and the sub module status message ***JSON_SubModules*** (related API: ***NET_DVR_STDXMLConfig*** ; URLs: ***/ISAPI/VideoIntercom/SubModules/capabilities?format=json*** and ***/ISAPI/VideoIntercom/SubModules?format=json***):
added a value "DS-KD-DIS" (display module) to the node "**moduleType**" (module type);
added a node "**version**" (software version No.).
2. Added the function of sub module configuration (related API: ***NET_DVR_STDXMLConfig***):
Get configuration capability: GET ***/ISAPI/VideoIntercom/SubModules/<ID>/Configurations/capabilities?format=json*** ;
Get parameters: GET ***/ISAPI/VideoIntercom/SubModules/<ID>/Configurations?format=json***
Set parameters: PUT ***/ISAPI/VideoIntercom/SubModules/<ID>/Configurations?format=json*** .
3. Extended video intercom capability ***XML_VideoIntercomCap*** (related API: ***NET_DVR_STDXMLConfig*** ; URL: ***/ISAPI/VideoIntercom/capabilities***):
added two nodes, i.e., <**isSupportSubModules**> (whether to support sub module management) and <**isSupportSubModulesCfg**> (whether to support configuring sub module).

Summary of Changes in Version 6.0.X.X_July, 2019

New document.

Chapter 2 Start Video Intercom

The interaction between video intercom device and software client depends on the persistent connection. The video and audio data will be sent or received via the connection to realize the video intercom.

Before You Start

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

Steps

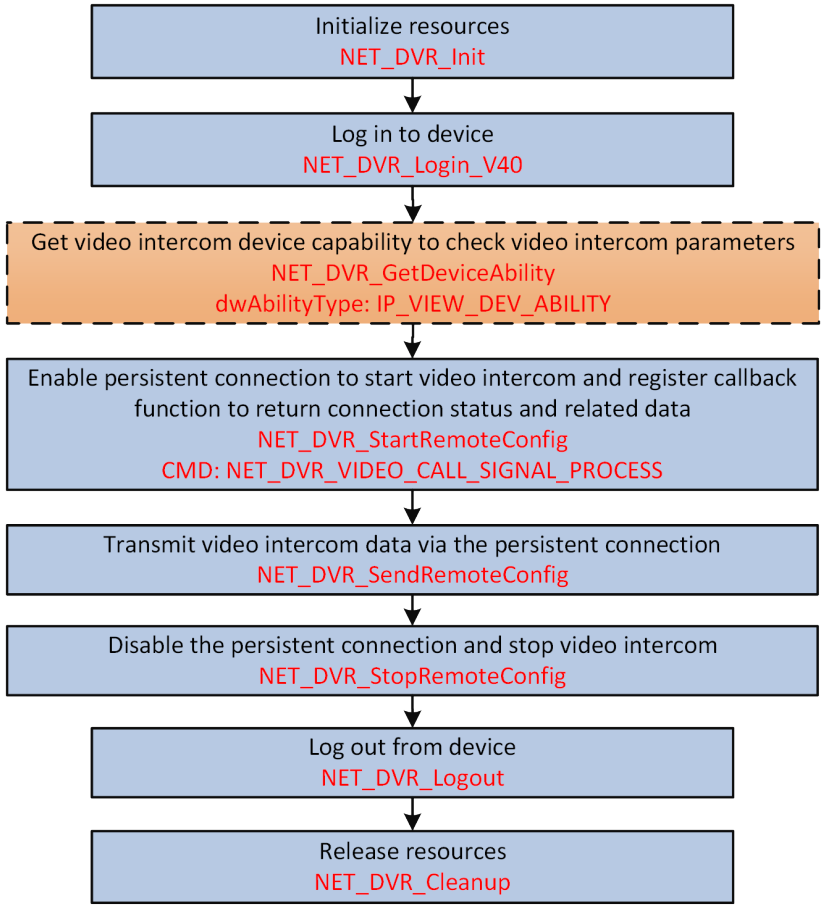

Figure 2-1 Programming Flow of Starting Video Intercom

1. **Optional:** Call ***NET_DVR_GetDeviceAbility*** and set **dwAbilityType** to "IP_VIEW_DEV_ABILITY" (macro definition value: 0x014) to get video intercom device capability for checking the supported video intercom parameters.

   The video intercom device capability is returned in the message ***XML_IpViewDevAbility*** , and the related node is <**VideoCall**>.

2. Call ***NET_DVR_StartRemoteConfig*** with "NET_DVR_VIDEO_CALL_SIGNAL_PROCESS" (command No.: 16032) and set **lpInBuffer** to ***NET_DVR_VIDEO_CALL_COND*** for enabling persistent connection to start video intercom and registering callback function (***fRemoteConfigCallback***) to return connection status or related data.

   The macro definition value of connection status (i.e., 0, 1, or 2) and related data (i.e., status value (1000, 1001, 1002), progress value, or data structure ***NET_DVR_VIDEO_CALL_PARAM***) are returned by **dwType** and **lpBuffer**, respectively, in the callback function.
3. Call ***NET_DVR_SendRemoteConfig*** , set **dwDataType** to "0", and set **pSendBuf** to ***NET_DVR_VIDEO_CALL_PARAM*** for transmitting video intercom data.
4. Call ***NET_DVR_StopRemoteConfig*** to disable the persistent connection and stop video intercom.

What to do next

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

Chapter 3 Manage Ring

Before You Start

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

Steps

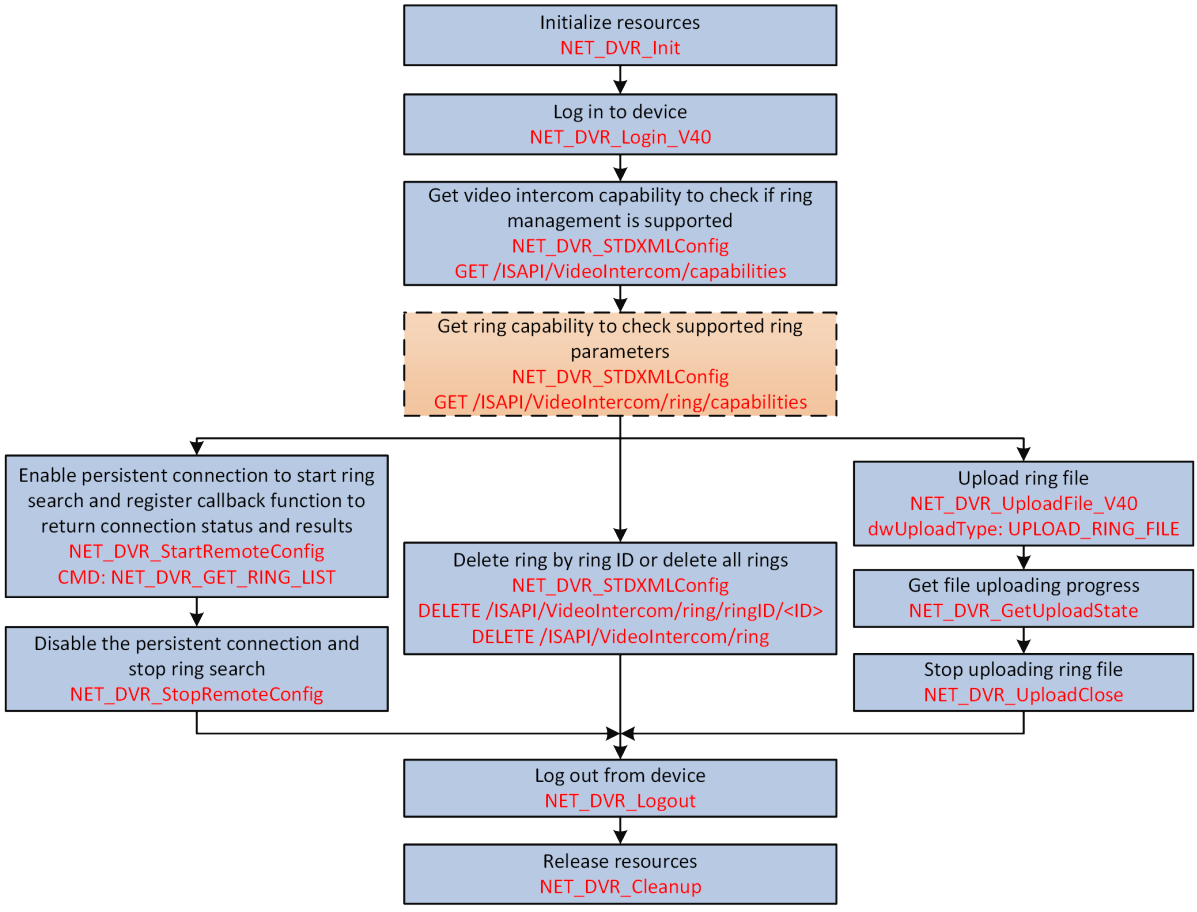

Figure 3-1 Programming Flow of Managing Ring

1. Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: GET ***/ISAPI/VideoIntercom/capabilities*** for getting video intercom capability to check if ring management is supported.

   The video intercom capability is returned in the message ***XML_VideoIntercomCap*** by **lpOutputParam**.

   If supports, the node <**isSupportRingManage**> is returned and its value is "true", and then you can perform the following steps.

   Otherwise, ring management is not supported by device, please end this task.

2. **Optional:** Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: GET ***/ISAPI/VideoIntercom/ring/capabilities*** for getting ring capability to check the supported ring configuration parameters.

   The ring capability is returned in the message ***XML_Cap_RingParam*** by **lpOutputParam**.

3. Perform the following operation(s) to manage ring, such as ring search, deleting rings, and uploading ring file.

   Ring Search

   a. Call ***NET_DVR_StartRemoteConfig*** with "NET_DVR_GET_RING_LIST" (command No.: 16039) and set **lpInBuffer** to ***NET_DVR_RING_SEARCH_COND*** for enabling persistent connection to start ring search and registering callback function (***fRemoteConfigCallback***) to return connection status and search results.

   Note

   The macro definition value of connection status (i.e., 0 or 2) and related data (i.e., status value (1000, 1001, 1002) or search result structure (***NET_DVR_RING_SEARCH_CFG***)) are returned by **dwType** and **lpBuffer**, respectively, in the callback function.

   b. Call ***NET_DVR_StopRemoteConfig*** to disable the persistent connection and stop ring search when the called back status value is "1000".

   Delete Ring by Ring ID

   Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: DELETE ***/ISAPI/VideoIntercom/ring/ringID/<ID>*** .

   Delete All Rings

   Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: DELETE ***/ISAPI/VideoIntercom/ring*** .

   Upload Ring File

   a. Call ***NET_DVR_UploadFile_V40*** , set **dwUploadType** to "UPLOAD_RING_FILE" (macro definition value: 20), and set **lpOutBuffer** to ***NET_DVR_RING_UPLOAD_CFG*** for uploading ring file.
   b. Call ***NET_DVR_GetUploadState*** to get file uploading progress.
   c. Call ***NET_DVR_UploadClose*** to stop uploading ring file when the returned uploading progress is "100".

What to do next

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

Chapter 4 Get Registration Information of Door Station

Before You Start

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

Steps

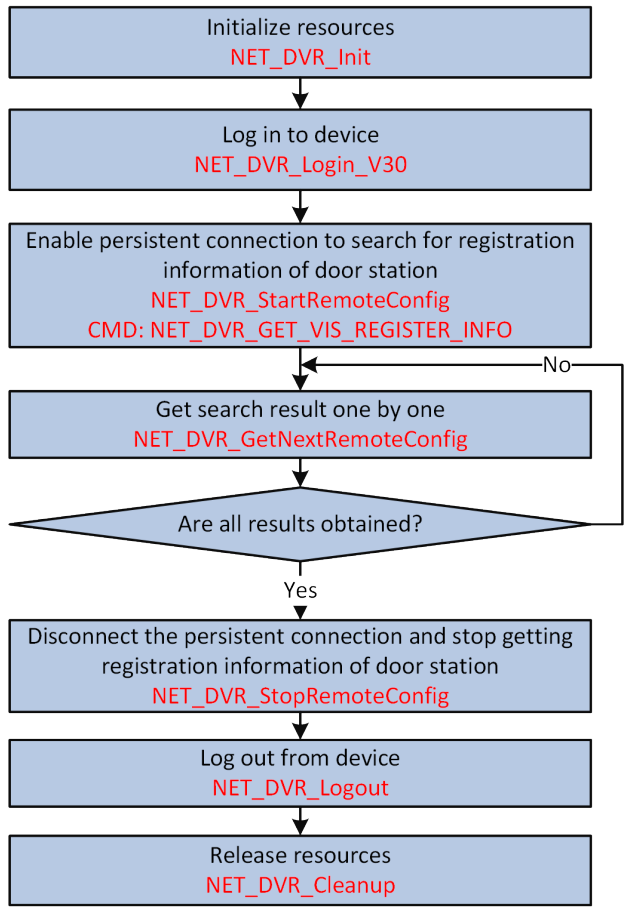

Figure 4-1 Programming Flow of Getting Registration Information of Door Station

1. Call ***NET_DVR_StartRemoteConfig*** with "NET_DVR_GET_VIS_REGISTER_INFO" (command No.: 16027) and set **cbStateCallback** to "null" for enabling persistent connection to search for registration information of door station.
2. Call ***NET_DVR_GetNextRemoteConfig*** to get search result one by one.

   Each search result is returned in the structure ***NET_DVR_VIS_REGISTER_INFO*** by **lpOutBuff**, and the getting status is also returned by the API.
3. Perform one of the following operations after checking the returned getting status.

- If the getting status value is "1000" or "1001", call ***NET_DVR_GetNextRemoteConfig*** to get the next result.
- If the getting status value is "1002" or "1003", continue to perform the following steps.

4. Call ***NET_DVR_StopRemoteConfig*** to disable the persistent connection and stop getting registration information of door station.

What to do next

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

Chapter 5 Configure Video Intercom Alarm or Event

The alarm triggered or event occurred during video intercom can be uploaded to central platform for further processing after setting alarm or event parameters and configuring arming or listening mode of alarm/event receiving.

Before You Start

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

Steps

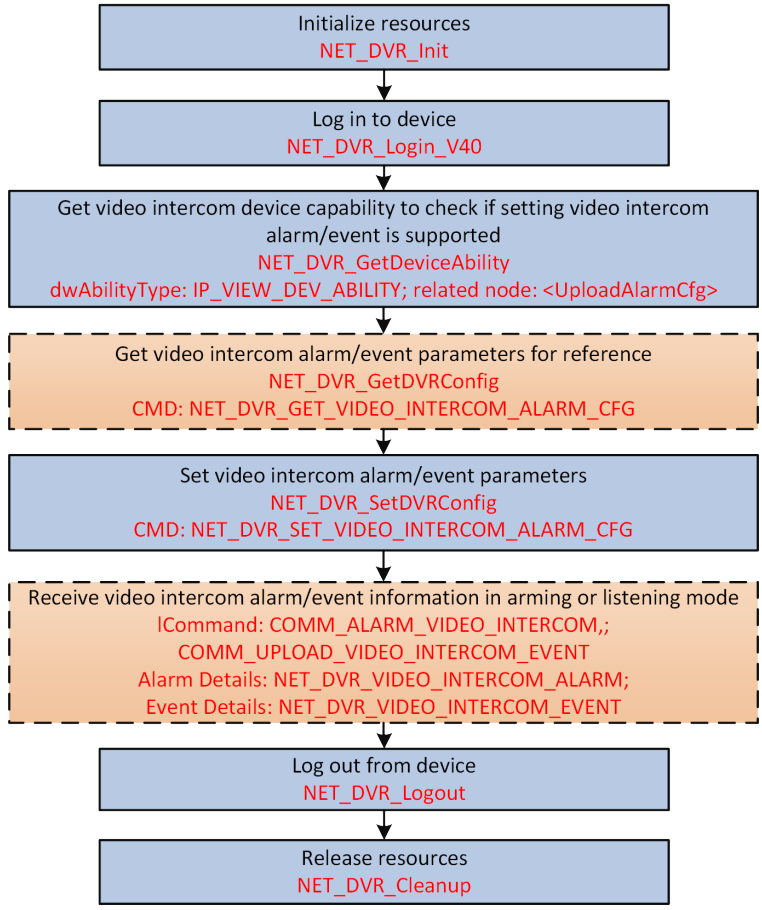

Figure 5-1 Programming Flow of Configuring Video Intercom Alarm or Event

1. Call ***NET_DVR_GetDeviceAbility*** and set **dwAbilityType** to "IP_VIEW_DEV_ABILITY" (macro definition value: 0x014) to get video intercom device capability for checking if setting video intercom alarm or event is supported.

   The video intercom device capability is returned in the message ***XML_IpViewDevAbility*** .

   If supports, the node <**UploadAlarmCfg**> is returned and you can perform the following steps.

2. **Optional:** Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_GET_VIDEO_INTERCOM_ALARM_CFG" (command No.: 16037) and set **lChannel** to "0xFFFFFFFF" for getting default or configured video intercom alarm or event parameters for reference.

   The alarm or event parameters of video intercom are returned in the structure ***NET_DVR_VIDEO_INTERCOM_ALARM_CFG*** by **lpOutBuffer**.
3. Call ***NET_DVR_SetDVRConfig*** with "NET_DVR_SET_VIDEO_INTERCOM_ALARM_CFG" (command No.: 16038), set **lChannel** to "0xFFFFFFFF", and set **lpInBuffer** to the structure ***NET_DVR_VIDEO_INTERCOM_ALARM_CFG*** for setting video intercom alarm or event parameters.
4. **Optional:** Respectively set **lCommand** in alarm/event callback function (***MSGCallBack***) to "COMM_ALARM_VIDEO_INTERCOM" (command No.: 0x1133) and "COMM_UPLOAD_VIDEO_INTERCOM_EVENT" (command No.:0x1132) for receiving video intercom alarm and event information in arming mode (refer to ***Receive Alarm/Event in Arming Mode***) or listening mode (refer to ***Receive Alarm/Event in Listening Mode***).

   The video intercom alarm information is called back in the structure ***NET_DVR_VIDEO_INTERCOM_ALARM*** ; and the video intercom event information is called back in the structure ***NET_DVR_VIDEO_INTERCOM_EVENT*** .

What to do next

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

Chapter 6 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.

6.1 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

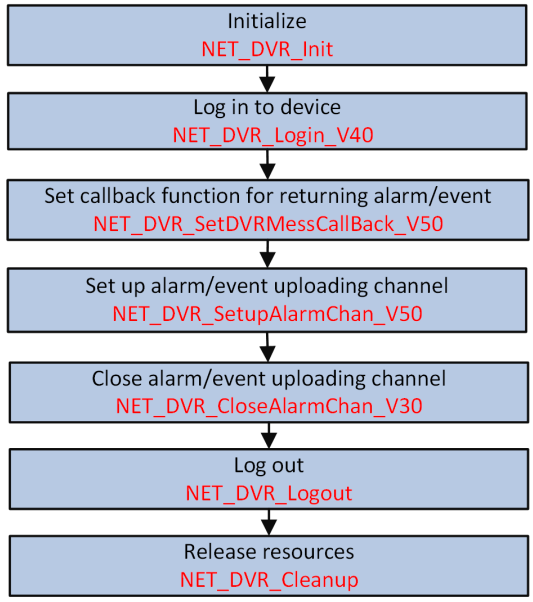

Figure 6-1 Programming Flow of Receiving Alarm/Event in Arming Mode

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
```

```
  lIHandle = 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.

6.2 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

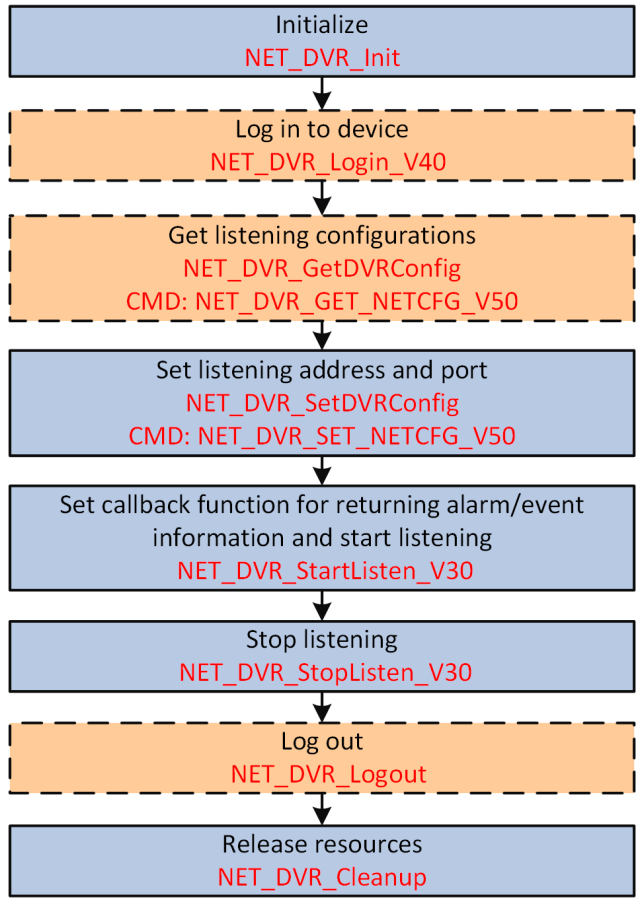

Figure 6-2 Programming Flow of Receiving Alarm/Event in Listening Mode

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 7 Remote Configuration

Device Settings

- Get video intercom device parameters, such as device No., type, registration type, and so on.
  Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_GET_VIDEO_INTERCOM_DEVICEID_CFG" (command No.: 16001) and set **lChannel** to "0xFFFFFFFF".
  The video intercom device parameters are returned in the structure ***NET_DVR_VIDEO_INTERCOM_DEVICEID_CFG*** by **lpOutBuffer**.
- Set video intercom device No.
  Call ***NET_DVR_SetDVRConfig*** with "NET_DVR_SET_VIDEO_INTERCOM_DEVICEID_CFG" (command No.: 16002), set **lChannel** to "0xFFFFFFFF", and set **lpInBuffer** to the structure ***NET_DVR_VIDEO_INTERCOM_DEVICEID_CFG*** .

Network Settings

- Get network parameters of video intercom device
  Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_GET_VIDEO_INTERCOM_RELATEDEV_CFG" (command No.: 16006) and set **lChannel** to "0xFFFFFFFF".
  The network parameters of video intercom device are returned in the structure ***NET_DVR_VIDEO_INTERCOM_RELATEDEV_CFG*** by **lpOutBuffer**.
- Set network parameters of video intercom device
  Call ***NET_DVR_SetDVRConfig*** with "NET_DVR_SET_VIDEO_INTERCOM_RELATEDEV_CFG" (command No.: 16007), set **lChannel** to "0xFFFFFFFF", and set **lpInBuffer** to the structure ***NET_DVR_VIDEO_INTERCOM_RELATEDEV_CFG*** .

Security Settings

- Set passwords for different permissions of video intercom device
  Call ***NET_DVR_SetDVRConfig*** with "NET_DVR_SET_PRIVILEGE_PASSWORD" (command No.: 16003), set **lChannel** to "0xFFFFFFFF", and set **lpInBuffer** to the structure ***NET_DVR_PRIVILEGE_PASSWORD_CFG*** .
- Get encryption verification capability of video intercom
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/passwordAuthentication/capabilities*** .
  The capability is returned in the message ***XML_Cap_PasswordAuthenticationCfg*** by **lpOutBuffer**.
- Verify video intercom encryption
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: PUT ***/ISAPI/VideoIntercom/passwordAuthentication*** and set lpInBuffer to the message ***XML_PasswordAuthenticationCfg*** .

Operation Time Settings

- Get operation time parameters of video intercom device

Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_GET_OPERATION_TIME_CFG" (command No.: 16004) and set **lChannel** to "0xFFFFFFFF".
The operation time parameters are returned in the structure ***NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_CFG*** by **lpOutBuffer**.

- Set operation time for video intercom device
  Call ***NET_DVR_SetDVRConfig*** with "NET_DVR_SET_OPERATION_TIME_CFG" (command No.: 16005), set **lChannel** to "0xFFFFFFFF", and set **lpInBuffer** to the structure ***NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_CFG*** .

Streaming Channel Settings

- Get streaming channel parameters of video intercom
  Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_GET_VIDEOINTERCOM_STREAM" (command No.: 16022) and set **lChannel** to the channel No.
  The stream channel parameters of video intercom are returned in the structure ***NET_DVR_VIDEOINTERCOM_STREAM*** by **lpOutBuffer**.
- Set streaming channel parameters of video intercom
  Call ***NET_DVR_SetDVRConfig*** with "NET_DVR_SET_VIDEOINTERCOM_STREAM" (command No.: 16023), set **lChannel** to the channel No., and set **lpInBuffer** to the structure ***NET_DVR_VIDEOINTERCOM_STREAM*** .

I/O Settings

- Get alarm input parameters of video intercom
  Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_GET_VIDEO_INTERCOM_IOIN_CFG" (command No.: 16016) and set **lChannel** to the alarm input No.
  The alarm input parameters of video intercom are returned in the structure ***NET_DVR_VIDEO_INTERCOM_IOIN_CFG*** by **lpOutBuffer**.
- Set alarm input parameters of video intercom
  Call ***NET_DVR_SetDVRConfig*** with "NET_DVR_SET_VIDEO_INTERCOM_IOIN_CFG" (command No.: 16017), set **lChannel** to the alarm input No., and set **lpInBuffer** to the structure ***NET_DVR_VIDEO_INTERCOM_IOIN_CFG*** .
- Get alarm output parameters of video intercom
  Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_GET_VIDEO_INTERCOM_IOOUT_CFG" (command No.: 16018) and set **lChannel** to the alarm output No.
  The alarm output parameters of video intercom are returned in the structure ***NET_DVR_VIDEO_INTERCOM_IOOUT_CFG*** by **lpOutBuffer**.
- Set alarm output parameters of video intercom
  Call ***NET_DVR_SetDVRConfig*** with "NET_DVR_SET_VIDEO_INTERCOM_IOOUT_CFG" (command No.: 16019), set **lChannel** to the alarm output No., and set **lpInBuffer** to the structure ***NET_DVR_VIDEO_INTERCOM_IOOUT_CFG*** .

Resident Calling Settings

- Get parameters of calling resident by pressing button

Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_GET_CALL_ROOM_CFG" (command No.: 16030) and set **lChannel** to the button No.
The parameters of calling resident by keypad are returned in the structure ***NET_DVR_CALL_ROOM_CFG*** by **lpOutBuffer**.

- Set parameters of calling resident by pressing button
  Call ***NET_DVR_SetDVRConfig*** with "NET_DVR_SET_CALL_ROOM_CFG" (command No.: 16031), set **lChannel** to the button No., and set **lpInBuffer** to the structure ***NET_DVR_CALL_ROOM_CFG*** .

Stream Source Settings

- Get stream source information
  Call ***NET_DVR_GetDeviceConfig*** with "NET_DVR_GET_STREAM_SRC_INFO" (command No.: 6017) and set **lpInBuffer** to the structure ***NET_DVR_STREAM_INFO*** .
  The stream source information is returned in the structure ***NET_DVR_STREAM_SRC_INFO*** by **lpOutBuffer**.
- Get information of all stream sources
  Call ***NET_DVR_GetDeviceConfig*** with "NET_DVR_GET_ALLSTREAM_SRC_INFO" (command No.: 9105).
  The information of each stream source is returned in the structure ***NET_DVR_STREAM_SRC_CFG*** by **lpOutBuffer**.
- Set stream source information
  Call ***NET_DVR_SetDeviceConfig*** with "NET_DVR_SET_STREAM_SRC_INFO" (command No.: 6018), set **lpInBuffer** to the structure ***NET_DVR_STREAM_INFO*** , and set **lpInParamBuffer** to the structure ***NET_DVR_STREAM_SRC_INFO*** .

Scene Settings

- Get scene configuration capability of video intercom
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/scene/nowMode/capabilities*** .
  The scene capability is returned in the message ***XML_Cap_SceneNowMode*** by **lpOutBuffer**.
- Get scene parameters of video intercom
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/scene/nowMode*** .
  The scene parameters are returned in the message ***XML_SceneNowMode*** by **lpOutBuffer**.
- Set scene parameters of video intercom
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: PUT ***/ISAPI/VideoIntercom/scene/nowMode*** and set **lpInBuffer** to the message ***XML_SceneNowMode*** .

Protocol Switching

- Get Configuration Capability for Switching Between Private Video Intercom Protocol and SIP (Session Initiation Protocol)
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/IntercomProtocolType/capabilities*** .

The configuration capability is returned in the message ***XML_Cap_IntercomProtocolType*** by **lpOutputParam**.

- Get Parameters for Switching Between Private Video Intercom Protocol and SIP (Session Initiation Protocol)
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/IntercomProtocolType*** .
  The parameters are returned in the message ***XML_IntercomProtocolType*** by **lpOutputParam**.
- Set Parameters for Switching Between Private Video Intercom Protocol and SIP (Session Initiation Protocol)
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: PUT ***/ISAPI/VideoIntercom/IntercomProtocolType*** and set **lpInputParam** to the message ***XML_IntercomProtocolType*** .

Settings for Pressing Button to Call

- Get configuration capability of pressing the button to call
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/keyCfg/<ID>/capabilities*** .
  The configuration capability is returned in the message ***XML_Cap_KeyCfg*** by **lpOutputParam**.
- Get parameters of multiple buttons for pressing the button to call
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/keyCfg*** .
  The parameters are returned in the message ***XML_KeyCfgList*** by **lpOutputParam**.
- Get parameters of a specific button for pressing the button to call
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/keyCfg/<ID>*** .
  The parameters are returned in the message ***XML_KeyCfg*** by **lpOutputParam**.
- Set parameters of a specific button for pressing the button to call
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: PUT ***/ISAPI/VideoIntercom/keyCfg/<ID>*** and set **lpInputParam** to the message ***XML_KeyCfg*** .

Chapter 8 Module Configuration

Sub Module Settings

- Get sub module configuration capability
  Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: GET ***/ISAPI/VideoIntercom/SubModules/<ID>/Configurations/capabilities?format=json*** .
  The configuration capability is returned in the message ***JSON_Cap_SubModulesCfg*** by **lpOutBuffer**.
- Get sub module parameters
  Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: GET ***/ISAPI/VideoIntercom/SubModules/<ID>/Configurations?format=json*** .
  The sub module parameters are returned in the message ***JSON_SubModulesCfg*** by **lpOutBuffer**.
- Set sub module parameters
  Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: PUT ***/ISAPI/VideoIntercom/SubModules/<ID>/Configurations?format=json*** and set **lpInBuffer** to the message ***JSON_SubModulesCfg*** .

Nametag Module Settings

- Get nametag module configuration capability
  Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: GET ***/ISAPI/VideoIntercom/SubModules/<ID>/CallCfg/capabilities?format=json*** .
  The configuration capability is returned in the message ***JSON_Cap_CallCfg*** by **lpOutBuffer**.
- Get nametag module parameters
  Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: GET ***/ISAPI/VideoIntercom/SubModules/<ID>/CallCfg?format=json*** .
  The nametag module parameters are returned in the message ***JSON_CallCfg*** by **lpOutBuffer**.
- Set nametag module parameters
  Call ***NET_DVR_STDXMLConfig*** to pass through the request URL: PUT ***/ISAPI/VideoIntercom/SubModules/<ID>/CallCfg?format=json*** and set **lpInBuffer** to the message ***JSON_CallCfg*** .

Chapter 9 Maintenance

Door Station Status

Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/WorkStatus*** . And the door station status is returned in the message ***XML_WorkStatus*** by **lpOutBuffer**.

Sub Module Status

- Get capability of getting sub module status
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/SubModules/capabilities?format=json*** .
  The capability is returned in the message ***JSON_Cap_SubModules*** by **lpOutBuffer**.
- Get sub module status
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/SubModules?format=json*** .
  The sub module status is returned in the message ***JSON_SubModules*** by **lpOutBuffer**.

Caller Information

- Get capability of getting caller information
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/callerInfo/capabilities?format=json*** .
  The capability is returned in the message ***JSON_Cap_CallerInfo*** by **lpOutBuffer**.
- Get caller information
  Call ***NET_DVR_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/VideoIntercom/callerInfo?format=json*** .
  The information is returned in the message ***JSON_CallerInfo*** by **lpOutBuffer**.

Chapter 10 API Reference

10.1 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.

10.2 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.

10.3 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

10.4 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

10.5 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.

10.6 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.

10.7 NET_DVR_GetNextRemoteConfig

Get the next search result.

API Definition

```
LONG NET_DVR_GetNextRemoteConfig(
  LONG    lHandle,
  void    *lpOutBuff,
  DWORD   dwOutBuffSize
);
```

Parameters

lHandle

[IN] Search handle, which is the value returned by ***NET_DVR_StartRemoteConfig*** .

lpOutBuff

[OUT] Output parameter buffer pointer, which relates to the commands (**dwCommand**) of ***NET_DVR_StartRemoteConfig*** .

dwOutBuffSize

[IN] Buffer size.

Return Values

Returns *-1* for failure, and returns other values for the current statuses, see details in the following table.

Status	Value	Description
NET_SDK_GET_NEXT_STATUS_SUCCESS	1000	The data is obtained. The API NET_DVR_GetNextRemoteConfig should be called again to get the next item of data.
NET_SDK_GET_NETX_STATUS_NEED_WAIT	1001	Waiting. The API NET_DVR_GetNextRemoteConfig can be called again.
NET_SDK_GET_NEXT_STATUS_FINISH	1002	All data is obtained. The API ***NET_DVR_StopRemoteConfig*** can be called to end.
NET_SDK_GET_NEXT_STATUS_FAILED	1003	Getting data exception. The API ***NET_DVR_StopRemoteConfig*** can be called to end.

If *-1* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

To get all information, you should call this API repeatedly.

10.8 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 10-1 Uploading Status Code

Return Value	Description
1	Uploaded successfully.
2	Uploading.
3	Uploading failed.
4	Network disconnected. Unknown status.
6	HDD error.
7	No HDD for saving inquest files.
8	Insufficient capacity.
9	Insufficient device resource.
10	No more files can be uploaded.
11	Too large file size.
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.

Return Value	Description
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 **customHumanID** of **FaceAppendData**).
38	Modeling failed. Device inner error.
39	Modeling failed. Face modeling error.
40	Modeling failed. Face score error.
41	Modeling failed. Feature collection error.
42	Modeling failed. Attribute collection error.
43	Picture data error.
44	Picture additional information error.
45	Certificate has already existed.

10.9 NET_DVR_Init

Initialize the programming environment before calling other APIs.

API Definition

```
BOOL NET_DVR_Init(
);
```

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 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

10.10 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

10.10.1 fLoginResultCallBack

Login Status Callback Function

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

10.11 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 *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, 3, 7, 8, 9, 10, 14, 17, 41, 44, 47, 72, and 73. See details in ***Device Network SDK Errors*** .

10.12 NET_DVR_SendRemoteConfig

Send data via the persistent connection.

API Definition

```
BOOL NET_DVR_SendRemoteConfig(
  LONG     lHandle,
  DWORD    dwDataType,
  char     *pSendBuf,
```

```
  DWORD    dwBufSize
);
```

Parameters

lHandle

Persistent configuration handle, which is returned by ***NET_DVR_StartRemoteConfig*** .

dwDataType

[IN] Data type, which relates to the commands of ***NET_DVR_StartRemoteConfig*** .

pSendBuf

[IN] Buffer for saving data to be sent, which relates to **dwDataType**.

dwBufSize

[IN] Size of data to be sent.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

Before calling this API, you must call ***NET_DVR_StartRemoteConfig*** to get the persistent connection handle.

10.13 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

10.14 NET_DVR_SetDVRConfig

Set the device parameters.

API Definition

```
BOOL NET_DVR_SetDVRConfig(
  LONG     lUserID,
```

```
  DWORD    dwCommand,
  LONG     lChannel,
  LPVOID   lpInBuffer,
  DWORD    dwInBufferSize
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwCommand

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

lChannel

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

lpInBuffer

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

dwInBufferSize

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

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.
The following error codes may be returned by this API: 0, 3, 6, 7, 8, 9, 10, 12, 17, 41, 43, 44, 47, 72, 73, and 76. See the corresponding error types and descriptions in the ***Device Network SDK Errors*** .

See Also

NET_DVR_GetDVRConfig

10.15 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 10-2 NET_SDK_INIT_CFG_TYPE

enumType	Value	Description	lpInBuff
NET_SDK_INIT_CFG_ABILITY	1	Capability supported by SDK.	***NET_DVR_INIT_CFG_ABILITY***
NET_SDK_INIT_CFG_SDK_PATH	2	Set loading path for component libraries (supported by both Linux and Windows system).	***NET_DVR_LOCAL_SDK_PATH***
NET_SDK_INIT_CFG_LIBEAY_PATH	3	Set path (including library name) for libeay32.dll (Windows), libcrypto.so (Linux), and libcrypto.dylib (Mac) of OpenSSL in version 1.1.1 and 1.0.2.	Path in string format, e.g., ***C:\\libeay32.dll***.
NET_SDK_INIT_CFG_SSLEAY_PATH	4	Set path (including library name) for ssleay32.dll (Windows), libssl.so (Linux), libssl.dylib (Mac) of OpenSSL in version 1.1.1 and 1.0.2.	Path in string format, e.g., ***C:\\ssleay32.dll***.

lpInBuff

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

Return Values

Returns *TURE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

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

10.16 NET_DVR_StartRemoteConfig

Enable remote configuration.

API Definition

```
LONG NET_DVR_StartRemoteConfig(
  LONG                lUserID,
  DWORD               dwCommand,
  LPVOID              lpInBuffer,
  DWORD               dwInBufferLen,
  fRemoteConfigCallback    cbStateCallback,
  LPVOID              pUserData
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwCommand

[IN] Configuration commands. For different functions, the commands and **lpInBuffer** are different.

lpInBuffer

Input parameter buffer pointer, which relates to the configuration command.

dwInBufferLen

[IN] Size of input buffer.

cbStateCallback

[IN] Status callback function, see the definition in ***fRemoteConfigCallback*** .

pUserData

[OUT] User data.

Return Values

Returns *-1* for failure, and returns other values for the handles of ***NET_DVR_GetNextRemoteConfig*** and ***NET_DVR_StopRemoteConfig*** .
If *-1* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

This API specifies the information to search. After calling this API, you can call ***NET_DVR_GetNextRemoteConfig*** to get the information one by one.

10.16.1 fRemoteConfigCallback

Function for calling back the persistent connection status and data to be transmitted.

Callback Function Definition

```
void(CALLBACK *fRemoteConfigCallback)(
  DWORD    dwType,
  void     *lpBuffer,
  DWORD    dwBufLen,
  void     *pUserData
);
```

Parameters

dwType

[OUT] Connection statuses, see the macro definitions below:

```
enum _NET_SDK_CALLBACK_TYPE_{
  NET_SDK_CALLBACK_TYPE_STATUS   = 0,
  NET_SDK_CALLBACK_TYPE_PROGRESS = 1,
  NET_SDK_CALLBACK_TYPE_DATA     = 2
}NET_SDK_CALLBACK_TYPE
```

NET_SDK_CALLBACK_TYPE_STATUS

Connection status.

NET_SDK_CALLBACK_TYPE_PROGRESS

Connection progress.

NET_SDK_CALLBACK_TYPE_DATA

Related data to be called back.

lpBuffer

[OUT] Pointer of buffer for saving progress, status, and related data to be called back, which relates to **dwType**, see details in the following table.

dwType	lpBuffer
NET_SDK_CALLBACK_TYPE_STATUS	If **dwBufLen** is 4, **lpBuffer** is 4-byte connection status; if **dwBufLen** is 8, **lpBuffer** consists of 4-byte connection status and 4-byte error code.

dwType	lpBuffer
	The connection status is enumerated in ***NET_SDK_CALLBACK_STATUS_NORMAL***
NET_SDK_CALLBACK_TYPE_PROGRESS	Connection progress value.
NET_SDK_CALLBACK_TYPE_DATA	Data structures to be returned, which are different according to different commands (**dwCommand**) in ***NET_DVR_StartRemoteConfig*** .

dwBufLen

[OUT] Buffer size.

pUserData

[OUT] User data.

10.17 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
lpInputParam	**lpRequestUrl** (see in structure ***NET_DVR_XML_CONFIG_INPUT***)	Method+URL E.g., GET /ISAPI/System/capabilities
	lpInBuffer (see in structure ***NET_DVR_XML_CONFIG_INPUT***)	Request Message
lpOutputParam	**lpOutBuffer** (see in structure ***NET_DVR_XML_CONFIG_OUTPUT***)	Response Message
	lpStatusBuffer (see in structure ***NET_DVR_XML_CONFIG_OUTPUT***)	Response Message

10.18 NET_DVR_StopRemoteConfig

Disconnect the persistent connection to stop remote configuration, and release resources.

API Definition

```
BOOL NET_DVR_StopRemoteConfig(
  LONG    lHandle
);
```

Parameters

lHandle

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

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

10.19 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.

10.20 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.

10.21 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 *TURE* 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*** .

10.22 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

10.23 NET_DVR_SetDVRConfig

Set the device parameters.

API Definition

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

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwCommand

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

lChannel

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

lpInBuffer

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

dwInBufferSize

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

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.
The following error codes may be returned by this API: 0, 3, 6, 7, 8, 9, 10, 12, 17, 41, 43, 44, 47, 72, 73, and 76. See the corresponding error types and descriptions in the ***Device Network SDK Errors*** .

See Also

NET_DVR_GetDVRConfig

10.24 NET_DVR_SetDVRMessageCallBack_V50

Set callback functions for getting the video data.

API Definition

```
BOOL NET_DVR_SetDVRMessageCallBack_V50(
  int          iIndex,
  MSGCallBack    fMessageCallBack,
  void          *pUser
);
```

Parameters

iIndex

[IN] Callback function index No., which ranges from 0 to 15.

fMessageCallBack

[IN] Callback function, see details in ***MSGCallBack*** .

pUser

[IN] User data.

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* returned, call ***NET_DVR_GetLastError*** to get the error code.

Remarks

- This API supports setting multiple callback functions for different channels (up to 16 channels are supported) at same time, and the configured callback functions are distinguished by the index No.
- All alarm/event information will be returned in each configured callback function, and you can distinguish the devices via the **pAlarmInfo** in the callback function (***MSGCallBack***).

Example

Sample Code of Setting Multiple Callback Functions to Receive Different Alarms/Events in Arming Mode

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

int iNum=0;
void CALLBACK MessageCallbackNo1(LONG lCommand, NET_DVR_ALARMER *pAlarmer, char *pAlarmInfo, DWORD
dwBufLen, void* pUser)
{
   int i=0;
   char filename[100];
   FILE *fSnapPic=NULL;
   FILE *fSnapPicPlate=NULL;
```

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

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

      switch(struPlateResult.struPlateInfo.byColor)//License plate color
      {
      case VCA_BLUE_PLATE:
        printf("Vehicle Color: Blue\n");
        break;
      case VCA_YELLOW_PLATE:
        printf("Vehicle Color: Yellow\n");
        break;
      case VCA_WHITE_PLATE:
        printf("Vehicle Color: White\n");
        break;
      case VCA_BLACK_PLATE:
        printf("Vehicle Color: Black\n");
        break;
      default:
        break;
      }
      //Scene picture
      if (struPlateResult.dwPicLen != 0 && struPlateResult.byResultType == 1 )
      {
```

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

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

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

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

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

   switch(lCommand)
   {
      case COMM_ALARM:
      {
         NET_DVR_ALARMINFO struAlarmInfo;
         memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_DVR_ALARMINFO));
         switch (struAlarmInfo.dwAlarmType)
         {
            case 3: //Motion detection alarm
               for (i=0; i<16; i++)   //#define MAX_CHANNUM   16  //The maximum number of channel
               {
                  if (struAlarmInfo.dwChannel[i] == 1)
                  {
                     printf("Channel No. with Motion Detection Alarm %d\n", i+1);
                  }
               }
               break;
            default:
               break;
         }
```

```
    break;
}
case COMM_UPLOAD_PLATE_RESULT:
{
    NET_DVR_PLATE_RESULT struPlateResult={0};
    memcpy(&struPlateResult, pAlarmInfo, sizeof(struPlateResult));
    printf("License Plate Number: %s\n", struPlateResult.struPlateInfo.sLicense);//License plate number

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

    for (i=0;i<struITSPlateResult.dwPicNum;i++)
```

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

void main() {

  //---------------------------------------
  //Initialize
```

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

//---------------------------------------
//Log in to device
LONG lUserID;
NET_DVR_DEVICEINFO_V30 struDeviceInfo;
lUserID = NET_DVR_Login_V30("172.0.0.100", 8000, "admin", "12345", &struDeviceInfo);
if (lUserID < 0)
{
    printf("Login error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

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

//Enable arming
NET_DVR_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

10.24.1 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 **lCommand**.

dwBufLen

[OUT] Size of alarm/event information buffer.

pUser

[OUT] User data.

10.25 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**.

10.26 NET_DVR_StartListen_V30

Register callback function for receiving alarm/event information and start listening (supports multiple threads).

API Definition

```
LONG NET_DVR_StartListen_V30(
  char         *sLocalIP,
  WORD          wLocalPort,
  MSGCallBack    DataCallback,
  void         *pUserData
);
```

Parameters

sLocalIP

[IN] IP address of local PC. It can be set to null.

wLocalPort

[IN] Listening port No. of local PC. It is configured by user, and it should be the same with that of device.

DataCallback

[IN] Alarm/event information callback function, see details in ***MSGCallBack*** .

pUserData

[IN] User data.

Return Values

Return *-1* for failure, and return other values for the handle parameters of ***NET_DVR_StopListen_V30*** .
If *-1* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.
The available error codes of this API are 0, 3, 6, 12, 17, 41, 44, 47, 72, and 75. See details in the ***Device Network SDK Errors*** .

Remarks

- To receive the alarm/event information sent by device, you should set the management host server address or listening host server address of device to the IP address of PC (which is same with the **sLocalIP**), or set the management host server port or listening host server port to the listening port No. of PC (which is same with the **wLocalPort**).
- The callback function in this API is prior to other callback functions, that is, if the callback function is configured in this API, other callback functions will not receive the alarm information. All the device alarm information is returned in same callback function, and you can distinguish the devices via the alarm device information (**pAlarmInfo**).

10.27 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*** .

Appendix A. Appendixes

A.1 Data Structure

A.1.1 NET_DVR_AGAIN_RELATEDEV

Parameter structure of linked network device of doorphone

Structure Definition

```
struct{
  NET_DVR_IPADDR    struSIPServer;
  NET_DVR_IPADDR    struCenterAddr;
  WORD              wCenterPort;
  BYTE              byRes1[2];
  NET_DVR_IPADDR    struIndoorUnit;
  NET_DVR_IPADDR    struAgainAddr;
  BYTE              byRes[444];
}NET_DVR_AGAIN_RELATEDEV,*LPNET_DVR_AGAIN_RELATEDEV;
```

Members

struSIPServer

IP address of SIP server, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

struCenterAddr

IP address of platform or system, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

byRes

Reserved, set to 0.

struIndoorUnit

IP address of indoor station, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

struAgainAddr

IP address of main doorphone, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

byRes

Reserved, set to 0.

See Also

A.1.2 NET_DVR_AUTH_INFO

Authentication information structure

Structure Definition

```
struct{
  BYTE          byAuthResult;
  BYTE          byAuthType;
  BYTE          byRes1[2];
  BYTE          byCardNo[ACS_CARD_NO_LEN/*32*/];
  DWORD           dwPicDataLen;
  BYTE          *pImage;
  BYTE          byRes[212];
}NET_DVR_AUTH_INFO, *LPNET_DVR_AUTH_INFO;
```

Members

byAuthResult

Authentication result: 0-invalid, 1-authenticated, 2-failed.

byAuthType

Authentication mode: 0- invalid, 1-fingerprint, 2-face.

byRes1

Reserved, set to 0.

byCardNo

Card No.

dwPicDataLen

Picture data size, it is valid when **byAuthType** equals to 2.

pImage

Picture pointer, it is valid when **byAuthType** equals to 2.

byRes

Reserved, set to 0.

See Also

NET_DVR_VIDEO_INTERCOM_EVENT_INFO_UINON

A.1.3 NET_DVR_CALL_ROOM_CFG

Configuration structure for calling resident by pressing button.

Structure Definition

```
struct{
  DWORD    dwSize;
  SHORT    nFloorNumber;
  WORD     wRoomNumber;
  BYTE     byManageCenter;
  BYTE     byRes1[3];
  BYTE     byCalledName[64];
  BYTE     byRes[60];
}NET_DVR_CALL_ROOM_CFG,*LPNET_DVR_CALL_ROOM_CFG;
```

Members

dwSize

Structure size.

nFloorNumber

Floor No.

wRoomNumber

Room No.

byManageCenter

Whether to set as call management center: 0-no, 1-yes.

byRes1

Reserved, set to 0.

byCalledName

Resident name to be called, it can contain digits, letters, @, and dots. This member is valid in standard SIP mode.

byRes

Reserved, set to 0.

A.1.4 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

Member	Data Type	Description
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-Hikvision Private Protocol (default), 1-RTSP, 2-Hikvision Private Protocol+RTSP
bySubProto	BYTE	Transmission protocol type of sub-stream: 0-Hikvision Private Protocol (default), 1-RTSP, 2-Hikvision 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. • bySupport&0x1: whether supports VCA search. • bySupport&0x2: whether supports backup. • bySupport&0x4: whether supports getting encoding parameters. • bySupport&0x8: whether supports dual-NIC. • bySupport&0x10: whether supports remote SADP. • bySupport&0x20: whether supports RAID card. • bySupport&0x40: whether supports searching in IPSAN directory. • bySupport&0x80: whether supports RTP over RTSP.
bySupport1	BYTE	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.

Member	Data Type	Description
		• bySupport1&0x1: whether supports SNMP with version 30. • bySupport1&0x2: whether supports playback and downloading video files. • bySupport1&0x4: whether supports setting the arming priority. • bySupport1&0x8: whether supports extending the arming time period. • bySupport1&0x10: whether supports multiple HDDs (more than 33). • bySupport1&0x20: whether supports RTP over RTSP. • bySupport1&0x80: whether supports license plate recognition alarm.
bySupport2	BYTE	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. • bySupport2&0x1: whether supports getting stream via URL. • bySupport2&0x2: whether supports FTP with version 40. • bySupport2&0x4: whether supports ANR. • bySupport2&0x20: whether supports getting device status. • bySupport2&0x40: whether supports encrypting stream.
wDevType	WORD	Device model
bySupport3	BYTE	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. • bySupport3&0x1: whether supports multi-stream. • bySupport3&0x4: whether supports configuring by group (e.g., image, alarm input, alarm output, user, device status, JPEG picture capture, continuous and scheduled

Member	Data Type	Description
		capture, .HDD group management, and so on). • bySupport3&0x20: whether supports getting stream via DDNS.
byMultiStreamProto	BYTE	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. • byMultiStreamProto&0x1: whether supports third-stream. • byMultiStreamProto&0x2: whether supports fourth-stream. • byMultiStreamProto&0x40: whether supports main stream. • byMultiStreamProto&0x80: whether supports sub-stream.
byStartDChan	BYTE	Start No. of digital channel, 0-no digital channel (e.g., DVR, network camera).
byStartDTalkChan	BYTE	Start No. of two-way audio channel, 0-no two-way audio channel.
byHighDChanNum	BYTE	Number of digital channels, high 8-bit.
bySupport4	BYTE	Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported. • bySupport4&0x01: whether all stream types support RTSP and Hikvision Private Protocol. • bySupport4&0x02: whether the device supports transmitting form format data via API (NET_DVR_STDXMLConfig). • bySupport4&0x10: whether supports loading network disk by domain name.
byLanguageType	BYTE	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.

Member	Data Type	Description
		• byLanguageType ==0: this field is not supported by device. • byLanguageType&0x1: whether supports Chinese. • byLanguageType&0x2: whether supports English.
byVoiceInChanNum	BYTE	Number of audio input channels
byStartVoiceInChanNo	BYTE	Start No. of audio input channel, 0-invalid.
byRes3	Array of BYTE	Reserved, set to 0.
byMirrorChanNum	BYTE	Number of mirror channels
wStartMirrorChanNo	WORD	Start No. of mirror channel
byRes2	Array of BYTE	Reserved, set to 0.

Remarks

- The maximum number of digital channels equal to byIPChanNum+byHighDChanNum*256.
- For login via text protocol, the following parameters are not supported: **byMainProto**, **bySubProto**, **bySupport**, **bySupport1**, **bySupport2**, **bySupport3**, **bySupport4**, **bySupport5**, **bySupport6**, **bySupport7**, **byMultiStreamProto**, **byStartDTalkChan**, **byVoiceInChanNum**, **byStartVoiceInChanNo**, **byMirrorChanNum**, and **wStartMirrorChanNo**.

See Also

NET_DVR_DEVICEINFO_V40

A.1.5 NET_DVR_DEVICEINFO_V40

Device Parameter Structure (V40)

Member	Data Type	Description
struDeviceV30	***NET_DVR_DEVICEINFO_V30***	Device parameters
bySupportLock	BYTE	Whether supports locking function: 1-support.
byRetryLoginTime	BYTE	Remaining login attempts, it is valid when the user name or password is incorrect and the **bySupportLock** is 1.

Member	Data Type	Description
byPasswordLevel	BYTE	Password strength: 0-invalid, 1-default password, 2-valid password, 3-risky password. For default password or risky password, the users are reminded to change password.
byProxyType	BYTE	Proxy type: 0-no proxy, 1-standard proxy, 2-EHome proxy.
dwSurplusLockTime	DWORD	Remaining locking time, unit: second. It is valid only when **bySupportLock** is 1. During the locing time, if the user try to log in to again, the remaining locking time will resume to 30 minutes.
byCharEncodeType	BYTE	Character encodings. 0-no decoding information, 1-GB2312 (Simplified Chinese), 2-GBK, 3-BIG5 (Traditional Chinese), 4-Shift_JIS (Japanese), 5-EUC-KR (Korean), 6-UTF-8, 7-ISO8859-1, 8-ISO8859-2, 9-ISO8859-3, …, 21-ISO8859-15 (Western European)
bySupportDev5	BYTE	Whether to support getting the parameters of devices that support HCNetSDK version 5.0 or above, the size of device name and type name are extended to 64 bytes.
bySupport	BYTE	Whether it supports uploading changes, it depends on the result of bitwise AND (&) operation: 0-not support, 1-support. The result of **bySupport**&0x1 indicates that this member is reserved; the result of **bySupport**&0x2 indicates that whether it supports uploading changes: 0-not support, 1-support. This member is the capability set extension.
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

Member	Data Type	Description
		indicates that the password being used has expired. For example, if -3 is returned, it indicates that the password being used has expired for three days.
byResidualValidity	BYTE	Whether the member **iResidualValidity** is valid: 0-invalid, 1-valid.
bySingleStartDTalkChan	BYTE	Start channel No. for connecting independent audio tracks to the device. The value 0 is reserved and invalid. The channel No. of audio tracks cannot start from 0.
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 has been changed by the administrator, the non-admin user will be prompted "Please set a new password" each time he/she logs in to the device until he/she changes the password).
bySupportStreamEncrypt	BYTE	Whether it supports stream encryption, it depends on the result of bitwise AND (&) operation: 0-no, 1-yes. The result of **bySupportStreamEncrypt**&0x1 indicates whether to support RTP/TLS streaming, the result of **bySupportStreamEncrypt**&0x2 indicates whether to support SRTP/UDP streaming, and the result of **bySupportStreamEncrypt**&0x4 indicates whether to support SRTP/MULTICAST streaming.
byRes2	Array of BYTE	Reserved, set to 0.

Remarks

- Four character types are allowed in the password, including digits, lowercase letters, uppercase letters and symbols. The maximum password length is 16 bits, and there are four password strength levels, see details below:
  - Level 0 (Risky Password): The password length is less than 8 bits, or only contains one kind of the character types. Or the password is the same with the user name, or is the mirror writing of the user name.
  - Level 1 (Weak Password): The password length is more than or equal to 8 bits, and contains two kinds of the character types. Meanwhile, the combination should be (digits + lowercase letters) or (digits + uppercase letters).
  - Level 2 (Medium Password): The password length is more than or equal to 8 bits, and contains two kinds of the character types. Meanwhile, the combination cannot be (digits + lowercase letters) and (digits + uppercase letters).
  - Level 3 (Strong Password): The password length is more than or equal to 8 bits, and at least contains three kinds of the character types.
- For login via text protocol, the following parameters are not supported: **bySupportLock**, **byRetryLoginTime**, **byPasswordLevel**, **byProxyType**, **dwSurplusLockTime**, **byCharEncodeType**, and **bySupportDev5**.

A.1.6 NET_DVR_DIRECT_CONNECT_CHAN_INFO

Digital channel information structure

Structure Definition

```
struct{
  BYTE          byEnable;
  BYTE          byProType;
  BYTE          byZeroChan;
  BYTE          byPriority;
  BYTE          sUserName[NAME_LEN/*32*/];
  BYTE          sPassword[PASSWD_LEN/*16*/];
  BYTE          byDomain[MAX_DOMAIN_NAME/*64*/];
  NET_DVR_IPADDR    struIP;
  WORD          wDVRPort;
  BYTE          byStreamType;
  BYTE          byOnline;
  DWORD         dwChannel;
  BYTE          byTransProtocol;
  BYTE          byLocalBackUp;
  WORD          wDirectLastTime;
  BYTE          byChanNo[CHAN_NO_LEN/*24*/];
}NET_DVR_DIRECT_CONNECT_CHAN_INFO, *LPNET_DVR_DIRECT_CONNECT_CHAN_INFO;
```

Members

byEnable

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

byProType

Protocol type, refer to the structure ***NET_DVR_PROTO_TYPE*** for details, by default, the protocol type is 0-Hikvision private protocol.

byZeroChan

Whether it is channel-zero: 0-no, 1-yes.

byPriority

Priority, whose value is between 1 and 5.

sUserName

User name.

sPassword

Password.

byDomain

Device domain name.

struIP

Device IP address, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

wDVRPort

Device port No.

byStreamType

Stream type: 0-main stream, 1-sub-stream

byOnline

Online status (read-only): 0-offline, 1-online.

dwChannel

Channel No.

byTransProtocol

Transport protocol type: 0-TCP, 1-UDP, 2-multi-slot

byLocalBackUp

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

wDirectLastTime

Broadcast time duration, unit: second, the value range should be obtained via capability.

byChanNo

Channel No. for getting stream via VAG.

A.1.7 NET_DVR_INDOOR_UNIT_DEVICEID

Parameter structure of indoor station No.

Structure Definition

```
struct{
  SHORT    wFloorNumber;
  WORD     wRoomNumber;
  WORD     wDevIndex;
  BYTE     byRes[122];
}NET_DVR_INDOOR_UNIT_DEVICEID, *LPNET_DVR_INDOOR_UNIT_DEVICEID;
```

Members

wFloorNumber

Floor No.

wRoomNumber

Room No.

wDevIndex

Indoor station No., which is between 0 and 10.

byRes

Reserved, set to 0.

See Also

NET_DVR_VIDEO_INTERCOM_UNIT_DEVICEID_UNION

A.1.8 NET_DVR_INDOOR_UNIT_RELATEDEV

Parameter structure of linked network device of indoor station.

Structure Definition

```
struct{
  NET_DVR_IPADDR   struOutdoorUnit;
  NET_DVR_IPADDR   struManageUnit;
  NET_DVR_IPADDR   struSIPServer;
  NET_DVR_IPADDR   struAgainUnit;
  BYTE        byOutDoorType;
  BYTE          byOutInConnectMode;
  BYTE          byIndoorConnectMode;
  BYTE          byRes1;
  NET_DVR_IPADDR   struIndoorUnit;
  BYTE          byRes[300];
}NET_DVR_INDOOR_UNIT_RELATEDEV,*LPNET_DVR_INDOOR_UNIT_RELATEDEV;
```

Members

struOutdoorUnit

IP address of main door station, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

struManageUnit

IP address of master station, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

struSIPServer

IP address of SIP server, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

struAgainUnit

Doorphone IP address, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

byOutDoorType

Main door station type: 0-reserved, 1-main door station, 2-villa door station

byOutInConnectMode

Network connection mode of door station and indoor station: 1-in same LAN, 2-in different LAN.

byIndoorConnectMode

Network connection mode of indoor station and sub indoor station: 1-by wireless NIC, 2-by wired NIC

byRes1

Reserved, set to 0.

struIndoorUnit

IP address of indoor station, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

byRes

Reserved, set to 0.

See Also

A.1.9 NET_DVR_INDOOR_UNIT_OPERATION_TIME_CFG

Operation time parameter structure of indoor station

Structure Definition

```
struct{
  DWORD  dwMaxMonitoringTime;
  DWORD  dwMaxRingTime;
  DWORD  dwCallForwardingTime;
  DWORD  dwRingDurationTime;
```

```
  BYTE   byRes[112];
}NET_DVR_INDOOR_UNIT_OPERATION_TIME_CFG,*LPNET_DVR_INDOOR_UNIT_OPERATION_TIME_CFG;
```

Members

dwMaxMonitoringTime

Maximum time duration of video intercom, range: [10, 60], unit: second.

dwMaxRingTime

Maximum ringing time duration, range: [15, 60], unit: second.

dwCallForwardingTime

Timeout of call forwarding, range: [0, 20], unit: second.

dwRingDurationTime

Ringing timeout, range: [30, 60], unit: second.

byRes

Reserved.

A.1.10 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: 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
enumMaxAlarmNum	INIT_CFG_MAX_NUM	Maximum number of alarm channels, see details below: 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
byRes	Array of BYTE	Reserved, set to 0.

Remarks

By default, up to 2048 channels are supported. More channels require higher computer performance and network bandwidth.

See Also

NET_DVR_SetSDKInitCfg

A.1.11 NET_DVR_IPADDR_UNION

IP Address Union

Member	Data Type	Description
szIPv4	char[]	IPv4 address. The maximum length is 16 bytes.
szIPv6	char[]	IPv6 address. The maximum length is 256 bytes.

A.1.12 NET_DVR_LOCAL_SDK_PATH

Path Information Structure for Loading Component Libraries

Member	Data Type	Description
sPath	Array of char	Component libraries' addresses
byRes	Array of BYTE	Reserved.

Remarks

If the path of HCNetSDKCom folder and HCNetSDK libraries are same, but the path of executable programs are different, you can call ***NET_DVR_SetSDKInitCfg*** to specify the path of HCNetSDKCom folder to make sure the component libraries are loaded normally.

A.1.13 NET_DVR_MANAGE_UNIT_DEVICEID

Parameter structure of master station No.

Structure Definition

```
struct{
  DWORD    wPeriod;
  DWORD    wDevIndex;
  BYTE     byRes[124];
}NET_DVR_MANAGE_UNIT_DEVICEID, *LPNET_DVR_MANAGE_UNIT_DEVICEID;
```

Members

wPeriod

Community No., range: [0,9].

wDevIndex

Outer door station No., which is unique in each floor, and it starts from 0.

byRes

Reserved, set to 0.

See Also

NET_DVR_VIDEO_INTERCOM_UNIT_DEVICEID_UNION

A.1.14 NET_DVR_MANAGE_UNIT_RELATEDEV

Parameter structure of linked network device of master station.

Structure Definition

```
struct{
  NET_DVR_IPADDR    struSIPServer;
  BYTE              byRes[880];
}NET_DVR_MANAGE_UNIT_RELATEDEV,*LPNET_DVR_MANAGE_UNIT_RELATEDEV;
```

Members

struSIPServer

IP address of SIP server, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

byRes

Reserved, set to 0.

See Also

A.1.15 NET_DVR_MANAGE_UNIT_OPERATION_TIME_CFG

Operation time parameter structure of master station

Structure Definition

```
struct{
  DWORD    dwMaxMonitoringTime;
  DWORD    dwMaxRingTime;
  DWORD    dwMaxTalkTime;
```

```
  BYTE    byRes[116];
}NET_DVR_MANAGE_UNIT_OPERATION_TIME_CFG,*LPNET_DVR_MANAGE_UNIT_OPERATION_TIME_CFG;
```

Members

dwMaxMonitoringTime

Maximum video intercom time duration, range: [10,60], unit: second.

dwMaxRingTime

Maximum ringing time duration, range: [15,60], unit: second.

dwMaxTalkTime

Maximum talking time duration, unit: second

byRes

Reserved, set to 0.

See Also

NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_UNION

A.1.16 NET_DVR_MIME_UNIT

Input Content Details Structure of Message Transmission API (NET_DVR_STDXMLConfig)

Member	Data Type	Description
szContentType	Array of char	Content type (corresponds to **Content-Type** field in the message), e.g., text/json. text/xml, and so on. The content format must be supported by HTTP.
szName	Array of char	Content name (corresponds to **name** field in the message), e.g., name="upload".
szFilename	Array of char	Content file name (corresponds to **filename** field in the message), e.g., filename="C:\Users \test\Desktop\11.txt".
dwContentLen	DWORD	Content size
pContent	char*	Data point
bySelfRead	BYTE	0-External file, 1-Internal data, whose address is specified by **szFilename**.
byRes	Array of BYTE	Reserved. Set to 0. Maximum: 15 bytes.

See Also

NET_DVR_XML_CONFIG_INPUT

A.1.17 NET_DVR_NOTICEDATA_RECEIPT_INFO

Announcement reading receipt information structure

Structure Definition

```
struct{
  BYTE    byNoticeNumber[MAX_NOTICE_NUMBER_LEN/*32*/];
  BYTE    byRes[224];
}NET_DVR_NOTICEDATA_RECEIPT_INFO, *LPNET_DVR_NOTICEDATA_RECEIPT_INFO;
```

Members

byNoticeNumber

Announcement No.

byRes

Reserved, set to 0.

See Also

NET_DVR_VIDEO_INTERCOM_EVENT_INFO_UINON

A.1.18 NET_DVR_OUTDOOR_FENCE_DEVICEID

Parameter structure of outer door station No.

Structure Definition

```
struct{
  DWORD    wPeriod;
  DWORD    wDevIndex;
  BYTE     byRes[124];
}NET_DVR_OUTDOOR_FENCE_DEVICEID, *LPNET_DVR_OUTDOOR_FENCE_DEVICEID;
```

Members

wPeriod

Community No., range: [0,9].

wDevIndex

Outer door station No., which starts from 0.

byRes

Reserved, set to 0.

See Also

NET_DVR_VIDEO_INTERCOM_UNIT_DEVICEID_UNION

A.1.19 NET_DVR_OUTDOOR_UNIT_DEVICEID

Parameter structure of door station (or intelligent access control device) No.

Structure Definition

```
struct{
  DWORD   wPeriod;
  DWORD   wBuildingNumber;
  DWORD   wUnitNumber
  DWORD   wFloorNumber
  DWORD   wDevIndex
  DWORD   byRes[118];
}NET_DVR_OUTDOOR_UNIT_DEVICEID, *LPNET_DVR_OUTDOOR_UNIT_DEVICEID;
```

Members

wPeriod

Project No., range: [0,9].

wBuildingNumber

Building No.

wUnitNumber

Unit No.

wFloorNumber

Floor No.

wDevIndex

Door station No., which is unique in each floor, and it starts from 0.

byRes

Reserved, set to 0.

See Also

NET_DVR_VIDEO_INTERCOM_UNIT_DEVICEID_UNION

A.1.20 NET_DVR_OUTDOOR_UNIT_RELATEDEV

Parameter structure of linked network device of door station, villa door station, or intelligent access control device.

Structure Definition

```
struct{
  NET_DVR_IPADDR    struMainOutdoorUnit;
  NET_DVR_IPADDR    struManageUnit;
  NET_DVR_IPADDR    struSIPServer;
  BYTE              byManageCenterID[32];
  BYTE              byRes[560];
}NET_DVR_OUTDOOR_UNIT_RELATEDEV,*LPNET_DVR_OUTDOOR_UNIT_RELATEDEV;
```

Members

struMainOutdoorUnit

IP address of door station, it is valid when sub door station exists, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

struManageUnit

IP address of master station, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

struSIPServer

IP address of SIP server, it is valid when the sub door station exits, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

byManageCenterID

Management center ID, which is valid in SIP mode, and it should contains digits, letters, @, and dots.

byRes

Reserved, set to 0.

See Also

A.1.21 NET_DVR_OUTDOOR_UNIT_OPERATION_TIME_CFG

Operation time parameter structure of door station, outer door station, villa door station, and doorphone

Structure Definition

```
struct{
  DWORD    dwMaxMessageTime;
```

```
 DWORD    dwMaxTalkTime;
 BYTE     byRes[120];
}NET_DVR_OUTDOOR_UNIT_OPERATION_TIME_CFG,*LPNET_DVR_OUTDOOR_UNIT_OPERATION_TIME_CFG;
```

Members

dwMaxMessageTime

Maximum messaging time duration, range: [30,60], unit: second.

dwMaxTalkTime

Maximum talking time duration, range: [90,120], unit: second.

byRes

Reserved, set to 0.

See Also

NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_UNION

A.1.22 NET_DVR_PRIVILEGE_PASSWORD_CFG

Structure for configuring permission password.

Structure Definition

```
struct{
 DWORD    dwSize;
 BYTE     byPwdType;
 BYTE     byRes1[3];
 BYTE     byOldPassword[PASSWD_LEN/*16*/];
 BYTE     byNewPassword[PASSWD_LEN/*16*/];
 BYTE     byRes2[128];
}NET_DVR_PRIVILEGE_PASSWORD_CFG, *LPNET_DVR_PRIVILEGE_PASSWORD_CFG;
```

Members

dwSize

Structure size.

byPwdType

Password type, see the definition below:

```
enum _PRIVILEGE_PASSWORD_TYPE_ENUM_{
 ENUM_PRIVILEGE_PASSWORD_ENGINEERING      = 1,
 ENUM_PRIVILEGE_PASSWORD_SETUPALARM       = 2,
 ENUM_PRIVILEGE_PASSWORD_HOUSEHOLDER_UNLOCK = 3,
 ENUM_PRIVILEGE_PASSWORD_ANTI_HIJACKING    = 4,
 ENUM_PRIVILEGE_PASSWORD_PUBLIC1          = 5,
 ENUM_PRIVILEGE_PASSWORD_PUBLIC2          = 6,
 ENUM_PRIVILEGE_PASSWORD_PUBLIC3          = 7,
```

```
 ENUM_PRIVILEGE_PASSWORD_SENDCARD        = 8
}PRIVILEGE_PASSWORD_TYPE_ENUM
```

ENUM_PRIVILEGE_PASSWORD_ENGINEERING

Configuration Password

ENUM_PRIVILEGE_PASSWORD_SETUPALARM

Arming/Disarming Password

ENUM_PRIVILEGE_PASSWORD_HOUSEHOLDER_UNLOCK

Unlocking Password

ENUM_PRIVILEGE_PASSWORD_ANTI_HIJACKING

Duress Code

ENUM_PRIVILEGE_PASSWORD_PUBLIC1

Public Password 1

ENUM_PRIVILEGE_PASSWORD_PUBLIC2

Public Password 2

ENUM_PRIVILEGE_PASSWORD_PUBLIC3

Public Password 3

ENUM_PRIVILEGE_PASSWORD_SENDCARD

Card Activation Password

byRes1

Reserved, set to 0.

byOldPassword

Old password.

byNewPassword

New password.

byRes2

Reserved, set to 0.

A.1.23 NET_DVR_PROTO_TYPE

Structure about protocol parameters.

Structure Definition

```
struct{
 DWORD   dwType;
 BYTE    byDescribe[DESC_LEN/*16*/];
}NET_DVR_PROTO_TYPE,*LPNET_DVR_PROTO_TYPE;
```

Members

dwType

Protocol value

byDescribe

Protocol description

Remarks

- The descriptions (**byDescribe**) of different protocol types (**dwType**) are shown in the table below.

```
enum tagNET_DVR_IPC_ENUM_UNIFY{
  ENUM_IPC_PROTOCOL_INVALID = -1,
  ENUM_IPC_PROTOCOL_HIKVISION = 0,      //HIKVISION
  ENUM_IPC_PROTOCOL_PANASONIC = 1,      //PANASONIC
  ENUM_IPC_PROTOCOL_SONY = 2,           //SONY
  ENUM_IPC_PROTOCOL_AXIS = 4,           //AXIS
  ENUM_IPC_PROTOCOL_SANYO = 5,          //SANYO
  ENUM_IPC_PROTOCOL_BOSCH = 6,          //BOSCH
  ENUM_IPC_PROTOCOL_ZAVIO = 7,          //ZAVIO
  ENUM_IPC_PROTOCOL_GRANDEYE,           //GRANDEYE
  ENUM_IPC_PROTOCOL_PROVIDEO = 16,      //PROVIDEO
  ENUM_IPC_PROTOCOL_ARECONT = 10,       //ARECONT
  ENUM_IPC_PROTOCOL_ACTI = 11,          //ACTI
  ENUM_IPC_PROTOCOL_PELCO = 12,         //PELCO
  ENUM_IPC_PROTOCOL_VIVOTEK = 13,       //VIVOTEK
  ENUM_IPC_PROTOCOL_DAHUA = 3,          //DAHUA
  ENUM_IPC_PROTOCOL_SAMSUNG = 15,       //SAMSUNG
  ENUM_IPC_PROTOCOL_PSIA = 17,          //PSIA
  ENUM_IPC_PROTOCOL_ONVIF = 18,         //ONVIF
  ENUM_IPC_PROTOCOL_BRICKCOM = 19,      //BRICKCOM
  ENUM_IPC_PROTOCOL_CANON = 23,         //CANON
  ENUM_IPC_PROTOCOL_HUINT = 32,         //HUINT
  ENUM_IPC_PROTOCOL_INFINOVA = 14,      //INFINOVA
  ENUM_IPC_PROTOCOL_HIK_STD_H264,       //HIKVISION STANDARD H.264
  ENUM_IPC_PROTOCOL_HIK_STD_MPEG4,      //HIKVISION STANDARD MPEG4
  ENUM_IPC_PROTOCOL_SUNELL,             //SUNELL
  ENUM_IPC_PROTOCOL_ATEME,              //ATEME
  ENUM_IPC_PROTOCOL_LAUNCH,             //LAUNCH
  ENUM_IPC_PROTOCOL_YAAN,               //YAAN
  ENUM_IPC_PROTOCOL_BLUESKY,            //BLUESKY
  ENUM_IPC_PROTOCOL_BLUESKYLIMIT,       //BLUESKYLIMIT
  ENUM_IPC_PROTOCOL_TDWY,               //TIANDY
  ENUM_IPC_PROTOCOL_HBGK,               //HBGK
  ENUM_IPC_PROTOCOL_SANTACHI,           //SANTACHI
  ENUM_IPC_PROTOCOL_HIGHEASY,           //HIGHEASY
  ENUM_IPC_PROTOCOL_HANBANG,            //HANBANG
  ENUM_IPC_PROTOCOL_SAMSUNG_3120,       //SAMSUNG 3120
  ENUM_IPC_PROTOCOL_SAMSUNG_3080,       //SAMSUNG 3080
  ENUM_IPC_PROTOCOL_SAMSUNG_2000,       //SAMSUNG 2000
  ENUM_IPC_PROTOCOL_SAMSUNG_5200,       //SAMSUNG 5200
```

```
ENUM_IPC_PROTOCOL_JINGYUAN,          //JINGYUAN
ENUM_IPC_PROTOCOL_VIDEOTREC,         //VIDEOTREC
ENUM_IPC_PROTOCOL_CHENOVA,           //CHENOVA
ENUM_IPC_PROTOCOL_FENGHUO,           //FENGHUO
ENUM_IPC_PROTOCOL_ZB_5301,           //ZB_5301
ENUM_IPC_PROTOCOL_ZB_5401,           //ZB_5401
ENUM_IPC_PROTOCOL_HAIXIN,            //HISENSE
ENUM_IPC_PROTOCOL_ZHONGYINGXIN,      //ZHONGYINGXIN
ENUM_IPC_PROTOCOL_AVUN,              //AVUN
ENUM_IPC_PROTOCOL_GOVTY,             //GOVTY
ENUM_IPC_PROTOCOL_SAE,               //SAE
ENUM_IPC_PROTOCOL_DONGFANGWANGLI,    //NETPOSA
ENUM_IPC_PROTOCOL_CHANGHONG,         //CHANGHONG
ENUM_IPC_PROTOCOL_H3C,               //H3C
ENUM_IPC_PROTOCOL_BAIAN,             //BAIAN
ENUM_IPC_PROTOCOL_HAT,               //HAT
ENUM_IPC_PROTOCOL_YUANYE,            //YUANYE
ENUM_IPC_PROTOCOL_HIKCARD,           //HIKVISION BOARD CARD
ENUM_IPC_PROTOCOL_HAIXINCAP,         //HISENSE CAPTURE CAMERA
ENUM_IPC_PROTOCOL_WENANCAP,          //WENAN CAPTURE CAMERA
ENUM_IPC_PROTOCOL_XUNMEI,            //XUNMEI
ENUM_IPC_PROTOCOL_BAIWO,             //BAIWO
ENUM_IPC_PROTOCOL_APD,               //APD
ENUM_IPC_PROTOCOL_REACHDEV,          //REACHDEV
ENUM_IPC_PROTOCOL_XUNMEI_DAHUA,      //XUNMEI_DAHUA OEM
ENUM_IPC_PROTOCOL_HUANGHE,           //HUANGHE
ENUM_IPC_PROTOCOL_LIANCHEN,          //LIANCHEN
ENUM_IPC_PROTOCOL_CHENGYE,           //CHENGYE
ENUM_IPC_PROTOCOL_VISIONDIGI,        //VISIONDIGI
ENUM_IPC_PROTOCOL_HENGHE,            //HENGHE
ENUM_IPC_PROTOCOL_KODAK,             //KODAK
ENUM_IPC_PROTOCOL_AIRONIX,           //AIRONIX
ENUM_IPC_PROTOCOL_LG,                //LG
ENUM_IPC_PROTOCOL_HASEE,             //HASEE
ENUM_IPC_PROTOCOL_8000ME,            //8000ME
ENUM_IPC_PROTOCOL_POVITEL,           //POVITEL
ENUM_IPC_PROTOCOL_YIVIEW,            //YIVIEW
ENUM_IPC_PROTOCOL_TIANYUE,           //TIANYUE
ENUM_IPC_PROTOCOL_HOWELL,            //HOWELL
ENUM_IPC_PROTOCOL_WAPA,              //WAPA
ENUM_IPC_PROTOCOL_SANLE,             //SANLE
ENUM_IPC_PROTOCOL_HIKCARD_ENCRYPTION,  //ENCRYPTED HIKVISION BOARD CARD
ENUM_IPC_PROTOCOL_JUNSDA,            //JUNSDA
ENUM_IPC_PROTOCOL_LIYUAN,            //LIYUAN
ENUM_IPC_PROTOCOL_XINCHAN,           //XINCHAN
ENUM_IPC_PROTOCOL_BITE,              //BITE
ENUM_IPC_PROTOCOL_MEIAN,             //MEIAN
ENUM_IPC_PROTOCOL_ROSEEK,            //ROSEEK
ENUM_IPC_PROTOCOL_AEBELL,            //AEBELL
ENUM_IPC_PROTOCOL_JSL_ST,            //JSL ST
ENUM_IPC_PROTOCOL_VIMICRO,           //VIMICRO
```

```
  ENUM_IPC_PROTOCOL_TYPE,              //MAX MANUFACTURER TYPE
}NET_DVR_IPC_ENUM_UNIFY

enum _NET_DVR_IPC_ENUM_{
  ENUM_BUSINESS_INVALID = -1,
  ENUM_BUSINESS_HIKVISION = 0,
  ENUM_BUSINESS_PANASONIC,
  ENUM_BUSINESS_SONY,
  ENUM_BUSINESS_AXIS,
  ENUM_BUSINESS_SANYO,
  ENUM_BUSINESS_BOSCH,
  ENUM_BUSINESS_ZAVIO,
  ENUM_BUSINESS_GRANDEYE,
  ENUM_BUSINESS_PROVIDEO,
  ENUM_BUSINESS_ARECONT,       //9
  ENUM_BUSINESS_ACTI,
  ENUM_BUSINESS_PELCO,
  ENUM_BUSINESS_VIVOTEK,
  ENUM_BUSINESS_INFINOVA,
  ENUM_BUSINESS_DAHUA,         //14
  ENUM_BUSINESS_HIK_STD_H264 = 0x20,
  ENUM_BUSINESS_HIK_STD_MPEG4,
  ENUM_BUSINESS_SUNELL,        //SUNELL
  ENUM_BUSINESS_ATEME,
  ENUM_BUSINESS_LAUNCH,        //LAUNCH
  ENUM_BUSINESS_YAAN,          //YAAN
  ENUM_BUSINESS_BLUESKY,       //BLUESKY
  ENUM_BUSINESS_BLUESKYLIMIT,  //BLUESKYLIMIT
  ENUM_BUSINESS_TDWY,          //TIANDY
  ENUM_BUSINESS_HBGK,          //HBGK
  ENUM_BUSINESS_SANTACHI,      //SANTACHI
  ENUM_BUSINESS_HIGHEASY,      //HIGHEASY
  ENUM_BUSINESS_SAMSUNG,
  ENUM_BUSINESS_URL_RTSP = 0x40,//Streaming via URL
  ENUM_BUSINESS_ONVIF,
  ENUM_MAX_BUSINESS_TYPE,      //Maximum manufacturer type
}NET_DVR_IPC_ENUM
```

- If the device supports unified network camera protocol, the supported protocol types are enumerated in NET_DVR_IPC_ENUM_UNIFY; otherwise, the supported protocol types are enumerated in NET_DVR_IPC_ENUM.

A.1.24 NET_DVR_PU_STREAM_URL

Configuration parameter structure about getting stream by URL.

Structure Definition

```
struct{
  BYTE   byEnable;
  BYTE   strURL[240];
```

```
  BYTE    byTransPortocol;
  WORD    wIPID;
  BYTE    byChannel;
  BYTE    byRes[7];
}NET_DVR_PU_STREAM_URL,*LPNET_DVR_PU_STREAM_URL;
```

Members

byEnable

Enable/disable getting stream by URL: 0-disable, 1-enable.

strURL

Stream URL

byTransPortocol

Transfer protocol type: 0-TCP, 1-UDP

wIPID

Device ID= iDevInfoIndex + iGroupNO*64 +1

byChannel

Device channel No.

byRes

Reserved, set to 0.

Remarks

The stream URL format is {rtsp://ip[:port]/urlExtension}[?username=username][?password=password][?linkmode=linkmode]. You can also customize the URL format if the network camera supports custom URL.

A.1.25 NET_DVR_PU_STREAM_URL_CFG

Streaming URL structure

Structure Definition

```
struct{
  BYTE    byEnable;
  BYTE    byRes[3];
  BYTE    byStreamMediaIP[64];
  WORD    wStreamMediaPort;
  BYTE    byTransmitType;
  BYTE    byRes1[33];
  BYTE    byDevIP[64];
  WORD    wDevPort;
  BYTE    byChannel;
  BYTE    byTransMode;
  BYTE    byProType;
```

```
  BYTE    byTransProtocol;
  BYTE    byRes3[2];
  BYTE    sUserName[NAME_LEN/*32*/];
  BYTE    sPassWord[PASSWD_LEN/*16*/];
  BYTE    byRes2[28];
}NET_DVR_PU_STREAM_URL_CFG,*LPNET_DVR_PU_STREAM_URL_CFG;
```

Members

byEnable

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

byRes

Reserved, set to 0.

byStreamMediaIP

IP address of stream media server.

wStreamMediaPort

Port No. of stream media server.

byTransmitType

Transport protocol of steam media server: 0-TCP, 1-UDP, 2-RTSP

byRes1

Reserved, set to 0.

byDevIP

Device IP address.

wDevPort

Device port No.

byChannel

Device channel No.

byTransMode

Transmission mode: 0-main stream, 1-sub-stream

byProType

Manufacturer protocol type, refer to the structure ***NET_DVR_PROTO_TYPE*** for details.

byTransProtocol

Transport protocol type: 0-TCP, 1-UDP, 2-multicast, 3-RTP

byRes3

Reserved, set to 0.

sUserName

User name for logging in to device.

sPassWord

Password for logging in to device.

byRes

Reserved, set to 0.

A.1.26 NET_DVR_RING_SEARCH_CFG

Result structure of ring search.

Structure Definition

```
struct{
  DWORD    dwSize;
  DWORD    dwRingID;
  BYTE     byRingName[MAX_RING_NAME_LEN_128/*128*/];
  DWORD    dwRingSize;
  BYTE     byRingType;
  BYTE     byRes[303];
}NET_DVR_RING_SEARCH_CFG,*LPNET_DVR_RING_SEARCH_CFG;
```

Members

dwSize

Structure size.

dwRingID

Ring ID, value range: [1,8].

byRingName

Ring name.

dwRingSize

Ring file size.

byRingType

Ring file format: 0-WAV

byRes

Reserved, set to 0.

A.1.27 NET_DVR_RING_SEARCH_COND

Condition structure for ring search.

Structure Definition

```
struct{
  DWORD    dwSize;
```

```
  DWORD    dwRingID;
  BYTE     byRes[300];
}NET_DVR_RING_SEARCH_COND,*LPNET_DVR_RING_SEARCH_COND;
```

Members

dwSize

Structure size.

dwRingID

Ring ID, if this member is set to 0, all rings will be searched.

byRes

Reserved. set to 0.

A.1.28 NET_DVR_RING_UPLOAD_CFG

Parameter structure for uploading ring file.

Structure Definition

```
struct{
  DWORD    dwSize;
  DWORD    dwRingID;
  BYTE     byRingName[MAX_RING_NAME_LEN_128/*128*/];
  DWORD    dwRingSize;
  BYTE     byRingType;
  BYTE     byRes[363];
}NET_DVR_RING_UPLOAD_CFG,*LPNET_DVR_RING_UPLOAD_CFG;
```

Members

dwSize

Structure size.

dwRingID

Ring ID, value range: [1,8].

byRingName

Ring name.

dwRingSize

Ring file size, unit: byte.

byRingType

Ring file format: 0-WAV

byRes

Reserved, set to 0.

A.1.29 NET_DVR_SEND_CARD_INFO

Card issuing event information structure of video intercom

Structure Definition

```
struct{
  BYTE    byCardNo[ACS_CARD_NO_LEN/*32*/];
  BYTE    byRes[224];
}NET_DVR_SEND_CARD_INFO, *LPNET_DVR_SEND_CARD_INFO;
```

Members

byCardNo

Card No.

byRes

Reserved, set to 0.

See Also

NET_DVR_VIDEO_INTERCOM_EVENT_INFO_UINON

A.1.30 NET_DVR_STREAM_INFO

Stream information structure.

Structure Definition

```
struct{
  DWORD    dwSize;
  BYTE     byID[STREAM_ID_LEN/*32*/];
  DWORD    dwChannel;
  BYTE     byRes[32];
}NET_DVR_STREAM_INFO,*LPNET_DVR_STREAM_INFO;
```

Members

dwSize

Structure size.

byID

Stream ID, which consists of letters, digits, and dashes, 0-invalid.

dwChannel

Linked device channel. When it is 0xffffffff, if setting the stream source, this parameter indicates that no device channel is linked; if setting configuration condition, this parameter is invalid.

byRes

Reserved, set to 0.

Remarks

- If the device does not support marking stream ID, e.g., DVR, the parameter **byID** should be set to 0.
- For transcoder, when setting the stream source, only one of **byID** and **dwChannel** can be valid; when transcoding, both the **byID** and **dwChannel** can be invalid, the transcoding channel or stream ID is automatically allocated by device.
- For other devices (e.g., CVR), when this structure is inputted as configuration condition, if both the **byID** and **dwChannel** are invalid, error code (17) will be returned, if they are valid, but mismatched, error may also be returned, so only setting one of these two parameters is suggested.

A.1.31 NET_DVR_STREAM_MODE_TYPE

Streaming mode structure

Structure Definition

```
struct{
  DWORD                  byGetStreamType;
  BYTE                   byRes[3];
  NET_DVR_STREAM_TYPE_UNION   uGetStream;
}NET_DVR_STREAM_MODE_TYPE,*LPNET_DVR_STREAM_MODE_TYPE;
```

Members

byGetStreamType

Streaming mode: 0-get stream from device directly; 1-get stream from stream media server; 2-get stream from device after getting the IP address by IP server; 3-get the device IP address by IP server, and then get stream from stream media server; 4-get stream from stream media server by URL; 6-get stream via URL combined by data structure (contains user name and password).

byRes

Reserved, set to 0.

uGetStream

Streaming mode, refer to the union ***NET_DVR_STREAM_TYPE_UNION*** for details.

See Also

NET_DVR_STREAM_SRC_INFO

A.1.32 NET_DVR_STREAM_SRC_CFG

Stream source information structure.

Structure Definition

```
struct{
  DWORD                 dwSize;
  NET_DVR_STREAM_INFO       struStreamID;
  NET_DVR_STREAM_SRC_INFO    struStreamSrcInfo;
}NET_DVR_STREAM_SRC_CFG,*LPNET_DVR_STREAM_SRC_CFG;
```

Members

dwSize

Structure size.

struStreamInfo

Stream source information, refer to the structure ***NET_DVR_STREAM_INFO*** for details.

struStreamSrcInfo

Stream source information, refer to the structure ***NET_DVR_STREAM_SRC_INFO*** for details.

A.1.33 NET_DVR_STREAM_SRC_INFO

Structure of stream source information

Structure Definition

```
struct{
  DWORD                 dwSize;
  NET_DVR_STREAM_INFO       struStreamInfo;
  NET_DVR_STREAM_MODE_TYPE    struStreamSrcInfo;
}NET_DVR_STREAM_SRC_INFO,*LPNET_DVR_STREAM_SRC_INFO;
```

Members

dwSize

Structure size.

struStreamInfo

Stream information, refer to the structure ***NET_DVR_STREAM_INFO*** for details.

struStreamSrcInfo

Stream source information, refer to the structure ***NET_DVR_STREAM_MODE_TYPE*** for details.

A.1.34 NET_DVR_STREAM_TYPE_UNION

Streaming mode union

Structure Definition

```
union{
  NET_DVR_DIRECT_CONNECT_CHAN_INFO    struChanInfo;
  NET_DVR_PU_STREAM_URL               struStreamUrl;
  NET_DVR_PU_STREAM_URL_CFG           struStreamUrlCfg;
}NET_DVR_STREAM_TYPE_UNION,*LPNET_DVR_STREAM_TYPE_UNION;
```

Members

struChanInfo

Digital channel information, refer to the structure ***NET_DVR_DIRECT_CONNECT_CHAN_INFO*** for details.

struStreamUrl

Parameters for getting steam by URL, refer to the structure ***NET_DVR_PU_STREAM_URL*** for details.

struStreamUrlCfg

Parameters for getting stream by URL structure, refer to the structure ***NET_DVR_PU_STREAM_URL_CFG*** for details.

A.1.35 NET_DVR_TIME_EX

Extended Time Parameter Structure

Member	Data Type	Description
wYear	WORD	Year
byMonth	BYTE	Month
byDay	BYTE	Day
byHour	BYTE	Hour
byMinute	BYTE	Minute
bySecond	BYTE	Second
byRes	BYTE	Reserved.

A.1.36 NET_DVR_UNLOCK_RECORD_INFO

Unlocking event information structure of video intercom

Structure Definition

```
struct{
  BYTE    byUnlockType;
  BYTE    byRes1[3];
  BYTE    byControlSrc[NAME_LEN/*32*/];
  DWORD   dwPicDataLen;
  BYTE    *pImage;
  DWORD   dwCardUserID;
  SHORT   nFloorNumber;
  WORD    wRoomNumber;
  WORD    wLockID;
  BYTE    byRes2[2];
  BYTE    byLockName[LOCK_NAME_LEN/*32*/];
  BYTE    byRes[168];
}NET_DVR_UNLOCK_RECORD_INFO, *LPNET_DVR_UNLOCK_RECORD_INFO;
```

Members

byUnlockType

Mode to open door

```
enum _UNLOCK_TYPE_ENUM_{
  ENUM_UNLOCK_PASSWORD       = 1,
  ENUM_UNLOCK_HIJACKING      = 2,
  ENUM_UNLOCK_CARD           = 3,
  ENUM_UNLOCK_HOUSEHOLDER    = 4,
  ENUM_UNLOCK_CENTER_PLATFORM = 5,
  ENUM_UNLOCK_BLUETOOTH      = 6,
  ENUM_UNLOCK_QR_CODE        = 7,
  ENUM_UNLOCK_FACE           = 8,
  ENUM_UNLOCK_FINGERPRINT    = 9
}UNLOCK_TYPE_ENUM
```

ENUM_UNLOCK_PASSWORD

Open door by password

ENUM_UNLOCK_HIJACKING

Open door by duress code

ENUM_UNLOCK_CARD

Open door by card

ENUM_UNLOCK_HOUSEHOLDER

Open door by household

ENUM_UNLOCK_CENTER_PLATFORM

Open door by central platform

ENUM_UNLOCK_BLUETOOTH

Open door by bluetooth

ENUM_UNLOCK_QR_CODE

Open door by QR code

ENUM_UNLOCK_FACE

Open door by face

ENUM_UNLOCK_FINGERPRINT

Open door by fingerprint

byRes1

Reserved, set to 0.

byControlSrc

Operation source information, when unlock by card, it represents the card No., otherwise, it represents the device No.

dwPicDataLen

Picture data size.

pImage

Buffer pointer, which is used to save picture data, and the default picture format is JPEG.

dwCardUserID

Card holder ID.

nFloorNumber

Floor No., it is valid when opening door by card.

wRoomNumber

Room No., it is valid when opening door by card.

wLockID

Lock ID, for door station: 0-lock of door station, 1-locak of external access control module.

byRes2

Reserved, set to 0.

byLockName

Lock name, it is valid when opening door by card.

byRes

Reserved, set to 0.

See Also

NET_DVR_VIDEO_INTERCOM_EVENT_INFO_UINON

A.1.37 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	***fLoginResultCallBack***	Callback function used to return login status, it is valid only when **bUseAsynLogin** is "1".
pUser	void*	User data.
bUseAsynLogin	BOOL	Whether to enable asynchronous login: 0-no, 1-yes.
byProxyType	BYTE	Proxy server type: 0-no proxy, 1-standard proxy, 2-EHome proxy.
byUseUTCTime	BYTE	0-not convert (default), 1-input or output UTC time, 2-input or output local time.
byLoginMode	BYTE	Login mode: 0-login by private protocol, 1-login by text protocol, 2-self-adaptive (it is available when the protocol type supported by device is unknown, and this mode does not support asynchronous login).
byHttps	BYTE	Whether to enable TLS for login (by private protocol or by text protocol): 0-no, 1-yes, 2-self-adaptive (which is usually used when the protocol type supported by device is unknown. Both HTTP and HTTPS requests will be sent).
iProxyID	LONG	Proxy server No.

Member	Data Type	Description
byVerifyMode	BYTE	Whether to enable verification mode: 0-no, 1-bidirectional verification (currently not available), 2-unidirectional verification (it is valid when **byLoginMode** is 0 and **byHttps** is 1); when **byVerifyMode** is 0, CA certificate is not required, when **byVerifyMode** is 2, you should call NET_DVR_SetSDKLocalCfg to load CA certificate, and the enumeration value is "NET_SDK_LOCAL_CFG_CERTIFICATION".
byRes3	BYTE[]	Reserved, the maximum length is 119 bytes.

A.1.38 NET_DVR_VIDEO_CALL_COND

Condition structure for starting video intercom.

Structure Definition

```
struct{
  DWORD    dwSize;
  BYTE     byRes[128];
}NET_DVR_VIDEO_CALL_COND, *LPNET_DVR_VIDEO_CALL_COND;
```

Members

dwSize

Structure size.

byRes

Reserved, set to 0.

A.1.39 NET_DVR_VIDEO_CALL_PARAM

Parameter structure for video intercom interaction.

Structure Definition

```
struct{
  DWORD    dwSize;
  DWORD    dwCmdType;
  WORD     wPeriod;
  WORD     wBuildingNumber;
  WORD     wUnitNumber;
  SHORT    wFloorNumber;
  WORD     wRoomNumber;
```

```
  BYTE    byRes[118];
}NET_DVR_VIDEO_CALL_PARAM, *LPNET_DVR_VIDEO_CALL_PARAM;
```

Members

dwSize

Structure size.

dwCmdType

Command type: 0-request for call, 1-cancel the call, 2-answer, 3-decline, 4-time out, 5-end the call, 6-device is in call, 7-client is in call.

wPeriod

Community No., value range: [0,9].

wBuildingNumber

Building No.

wUnitNumber

Unit No.

wFloorNumber

Floor No.

wRoomNumber

Room No.

byRes

Reserved, set to 0.

A.1.40 NET_DVR_VIDEOINTERCOM_STREAM

Stream channel parameter structure of video intercom

Structure Definition

```
struct{
  DWORD    dwSize;
  BYTE    byVisDevID[MAX_NAMELEN/*16*/];
  DWORD    byDeviceName[NAME_LEN/*32*/];
  BYTE    bySourceType;
  BYTE    byRes[255];
}NET_DVR_VIDEOINTERCOM_STREAM, *LPNET_DVR_VIDEOINTERCOM_STREAM;
```

Members

dwSize

Structure size.

byVisDevID

Video intercom device No., and all digits in the No. are 0 when the video source type is network camera.

byDeviceName

Device name.

bySourceType

Video source type: 0-invalid, 1-network camera, 2-DVR/DVS/NVR, 3-door station, 4-outer door station, 5-doorphone.

byRes

Reserved, set to 0.

A.1.41 NET_DVR_VIDEO_INTERCOM_ALARM

Video intercom alarm information structure

Structure Definition

```
struct{
  DWORD                          dwSize;
  NET_DVR_TIME_EX                    struTime;
  BYTE                           byDevNumber[MAX_DEV_NUMBER_LEN/*32*/];
  BYTE                           byAlarmType;
  BYTE                           byRes1[3];
  NET_DVR_VIDEO_INTERCOM_ALARM_INFO_UNION    uAlarmInfo;
  BYTE                           wLockID
  BYTE                           byRes2[254];
}NET_DVR_VIDEO_INTERCOM_ALARM, *LPNET_DVR_VIDEO_INTERCOM_ALARM;
```

Members

dwSize

Structure size.

struTime

Alarm time, refer to the structure ***NET_DVR_TIME_EX*** for time format.

byDevNumber

Device No.

byAlarmType

Alarm type: 1-zone alarm, 2-tampering alarm, 3-duress alarm, 4-opening door by password failed for multiple times, 5-opening door failed, 6-closing door failed, 7-panic alarm), 8-intercom alarm, 9-smart lock: fingerprint duress alarm, 10-smart lock: password duress alarm, 11-smart lock: tampering alarm, 12-smart lock: locked alarm, 13-smart lock: low battery alarm, 14-blacklist alarm, 15-smart lock disconnected, 16-access control module: anti-tampering alarm, 17-video intercom started, 18-video intercom stopped.

byRes1

Reserved, set to 0.

uAlarmInfo

Alarm information, it is valid when **byAlarmType** equals to 1. Refer to the union ***NET_DVR_VIDEO_INTERCOM_ALARM_INFO_UNION*** for details.

wLockID

Lock ID, it is valid when alarm type is 5 or 6. When the ID is 0, it represents the lock of door station, when the ID is 1, it represents the lock of external access control module.

byRes2

Reserved, set to 0.

A.1.42 NET_DVR_VIDEO_INTERCOM_ALARM_CFG

Alarm or event parameter structure of video intercom

Structure Definition

```
struct{
  DWORD    dwSize;
  BYTE     byDoorNotCloseAlarm;
  BYTE     byRes[603];
}NET_DVR_VIDEO_INTERCOM_ALARM_CFG, *LPNET_DVR_VIDEO_INTERCOM_ALARM_CFG;
```

Members

dwSize

Structure size.

byDoorNotCloseAlarm

Whether to upload door open alarm: 0-no, 1-yes.

byRes

Reserved, set to 0.

A.1.43 NET_DVR_VIDEO_INTERCOM_ALARM_INFO_UNION

Video intercom alarm information union

Structure Definition

```
union{
  BYTE                  byLen[256];
  NET_DVR_ZONE_ALARM_INFO    struZoneAlarm;
}NET_DVR_VIDEO_INTERCOM_ALARM_INFO_UNION, *LPNET_DVR_VIDEO_INTERCOM_ALARM_INFO_UNION;
```

Members

byLen

Union size, which is 256 bytes.

struZoneAlarm

Zone alarm information, it is valid when the **byAlarmType** equals to 1. Refer to the structure ***NET_DVR_ZONE_ALARM_INFO*** for details.

See Also

NET_DVR_VIDEO_INTERCOM_ALARM

A.1.44 NET_DVR_VIDEO_INTERCOM_DEVICEID_CFG

Parameter structure for configuring video intercom device No.

Structure Definition

```
struct{
  DWORD                                 bySize;
  BYTE                                  byUnitType;;
  BYTE                                  byIsAutoReg;
  BYTE                                  byRes1[2];
  NET_DVR_VIDEO_INTERCOM_UNIT_DEVICEID_UNION    uVideoIntercomUnit;
  BYTE                                  byRes2[128];
}NET_DVR_VIDEO_INTERCOM_DEVICEID_CFG, *LPNET_DVR_VIDEO_INTERCOM_DEVICEID_CFG;
```

Members

dwSize

Structure size.

byUnitType

Device type: 1-door station, 2-master station, 4-outer station, 5-villa door station, 6-doorphone, 7-intelligent access control device.

byIsAutoReg

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

byRes1

Reserved, set to 0.

uVideoIntercomUnit

Device No., refer to the data union ***NET_DVR_VIDEO_INTERCOM_UNIT_DEVICEID_UNION*** for details.

byRes2

Reserved, set to 0.

Remarks

Configuring No. for doorphone is not required.

A.1.45 NET_DVR_VIDEO_INTERCOM_EVENT

Video intercom event information structure

Structure Definition

```
struct{
  DWORD                          dwSize;
  NET_DVR_TIME_EX                    struTime;
  BYTE                           byDevNumber[MAX_DEV_NUMBER_LEN/*32*/];
  BYTE                           byEventType;
  BYTE                           byRes1[3];
  NET_DVR_VIDEO_INTERCOM_EVENT_INFO_UINON    uEventInfo;
  BYTE                           byRes2[256];
}NET_DVR_VIDEO_INTERCOM_EVENT, *LPNET_DVR_VIDEO_INTERCOM_EVENT;
```

Members

dwSize

Structure size.

struTime

Event occurred time, refer to the structure ***NET_DVR_TIME_EX*** for time format.

byDevNumber

Device No.

byEventType

Event type: 1-unlocking, 2-announcement reading receipt, 3-authentication, 5-illegal card swiping, 6-card issuing of door station.

byRes1

Reserved, set to 0.

uEventInfo

Event information, and different types correspond to different data structures. Refer to the union ***NET_DVR_VIDEO_INTERCOM_EVENT_INFO_UINON*** for details.

byRes2

Reserved, set to 0.

A.1.46 NET_DVR_VIDEO_INTERCOM_EVENT_INFO_UINON

Video intercom event information union

Union Definition

```
union{
  BYTE                        byLen[256];
  NET_DVR_UNLOCK_RECORD_INFO       struUnlockRecord;
  NET_DVR_NOTICEDATA_RECEIPT_INFO    struNoticedataReceipt;
  NET_DVR_SEND_CARD_INFO          struSendCardInfo;
  NET_DVR_AUTH_INFO               struAuthInfo;
}NET_DVR_VIDEO_INTERCOM_EVENT_INFO_UINON, *LPNET_DVR_VIDEO_INTERCOM_EVENT_INFO_UINON;
```

Members

byLen

Union size, which is 256 bytes.

struUnlockRecord

Unlocking event information, refer to the structure ***NET_DVR_UNLOCK_RECORD_INFO*** for details.

struNoticedataReceipt

Announcement reading receipt information, refer to the structure ***NET_DVR_NOTICEDATA_RECEIPT_INFO*** for details.

struSendCardInfo

Card issuing event information, it is valid when **byEventType** equals to 5 or 6. Refer to the structure ***NET_DVR_SEND_CARD_INFO*** for details.

struAuthInfo

Authentication information, refer to the structure ***NET_DVR_AUTH_INFO*** for details.

A.1.47 NET_DVR_VIDEO_INTERCOM_IOIN_CFG

Alarm input parameter structure of video intercom

Structure Definition

```
struct{
  DWORD    dwSize;
  BYTE     byIOUseType;
  BYTE     byRes[63];
}NET_DVR_VIDEO_INTERCOM_IOIN_CFG, *LPNET_DVR_VIDEO_INTERCOM_IOIN_CFG;
```

Members

dwSize

Structure size.

byIOUseType

Alarm input usage: 0-disabled, 1-for unlocking, 2-represents door status, 0xff-custom.

byRes

Reserved, set to 0.

A.1.48 NET_DVR_VIDEO_INTERCOM_IOOUT_CFG

Alarm output parameter structure of video intercom

Structure Definition

```
struct{
  DWORD    dwSize;
  BYTE     byIOUseType;
  BYTE     byRes[63];
}NET_DVR_VIDEO_INTERCOM_IOOUT_CFG, *LPNET_DVR_VIDEO_INTERCOM_IOOUT_CFG;
```

Members

dwSize

Structure size.

byIOUseType

Alarm output usage: 0-disabled, 1-for electronic lock, 0xff-custom.

byRes

Reserved, set to 0.

A.1.49 NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_CFG

Operation time configuration structure

Structure Definition

```
struct{
  DWORD                                dwSize;
  BYTE                                 byUnitType;
  BYTE                                 byRes1[3];
  NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_UNION    uVideoIntercomUnit;
  BYTE                                 byRes2[128];
}NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_CFG, *LPNET_DVR_VIDEO_INTERCOM_OPERATION_TIME_CFG;
```

Members

dwSize

Structure size.

byUnitType

Device type: 1-door station, 2-master station, 4-outer door station, 5-villa door station, 6-doorphone.

byRes1

Reserved, set to 0.

uVideoIntercomUnit

Operation time configuration union, which contains different structures for different devices, refer to the union ***NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_UNION*** for details.

byRes2

Reserved, set to 0.

A.1.50 NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_UNION

Operation time configuration union of video intercom

Structure Definition

```
union{
  BYTE                              byLen[128];
  NET_DVR_INDOOR_UNIT_OPERATION_TIME_CFG     struIndoorUnit;
  NET_DVR_OUTDOOR_UNIT_OPERATION_TIME_CFG    struOutdoorUnit;
  NET_DVR_MANAGE_UNIT_OPERATION_TIME_CFG     struManageUnit;
}NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_UNION,
*LPNET_DVR_VIDEO_INTERCOM_OPERATION_TIME_UNION;
```

Members

byLen

Union size, which is 128 bytes.

struIndoorUnit

Operation time parameters of indoor station, refer to the structure ***NET_DVR_INDOOR_UNIT_OPERATION_TIME_CFG*** for details.

struOutdoorUnit

Operation time parameters of door station, outer door station, villa door station, and doorphone, refer to the structure ***NET_DVR_OUTDOOR_UNIT_OPERATION_TIME_CFG*** for details.

struManageUnit

Operation time parameters of master station, refer to the structure ***NET_DVR_MANAGE_UNIT_OPERATION_TIME_CFG*** for details.

See Also

NET_DVR_VIDEO_INTERCOM_OPERATION_TIME_CFG

A.1.51 NET_DVR_VIDEO_INTERCOM_RELATEDEV_CFG

Configuration structure for linked network devices of video intercom.

Structure Definition

```
struct{
  DWORD                                  dwSize;
  BYTE                                   byUnitType;
  BYTE                                   byRes1[3];
  NET_DVR_VIDEO_INTERCOM_UNIT_RELATEDEV_UNION    uVideoIntercomUnit;
  BYTE                                   byRes2[128];
}NET_DVR_VIDEO_INTERCOM_RELATEDEV_CFG, *LPNET_DVR_VIDEO_INTERCOM_RELATEDEV_CFG;
```

Members

dwSize

Structure size.

byUnitType

Device type: 1-door station, 2-master station, 4-outer station, 5-villa door station, 6-doorphone, 7-intelligent access control device.

byRes1

Reserved, set to 0.

uVideoIntercomUnit

Parameter union of linked network device, refer to the union ***NET_DVR_VIDEO_INTERCOM_UNIT_RELATEDEV_UNION*** for details.

byRes2

Reserved, set to 0.

A.1.52 NET_DVR_VIDEO_INTERCOM_UNIT_DEVICEID_UNION

Parameter union of video intercom device No.

Structure Definition

```
union{
  BYTE                    byLen[128];
```

```
  NET_DVR_INDOOR_UNIT_DEVICEID     struIndoorUnit;
  NET_DVR_OUTDOOR_UNIT_DEVICEID    struOutdoorUnit;
  NET_DVR_MANAGE_UNIT_DEVICEID     struManageUnit;
  NET_DVR_OUTDOOR_FENCE_DEVICEID   struFenceUnit;
  NET_DVR_OUTDOOR_UNIT_DEVICEID    struVillaOutdoorUnit;
  NET_DVR_OUTDOOR_UNIT_DEVICEID    struAgainConfirmUnit;
}NET_DVR_VIDEO_INTERCOM_UNIT_DEVICEID_UNION, *LPNET_DVR_VIDEO_INTERCOM_UNIT_DEVICEID_UNION;
```

Members

byLen

Union size, which is 128 bytes.

struIndoorUnit

Indoor station No., refer to the structure ***NET_DVR_INDOOR_UNIT_DEVICEID*** for details.

struOutdoorUnit

Door station (or intelligent access control device) No., refer to the structure ***NET_DVR_OUTDOOR_UNIT_DEVICEID*** for details.

struManageUnit

Master station No., refer to the structure ***NET_DVR_MANAGE_UNIT_DEVICEID*** for details.

struFenceUnit

Outer door station No., refer to the structure ***NET_DVR_OUTDOOR_FENCE_DEVICEID*** for details.

struVillaOutdoorUnit

Villa door station No., refer to the structure ***NET_DVR_OUTDOOR_UNIT_DEVICEID*** for details.

struAgainConfirmUnit

Doorphone No., refer to the structure ***NET_DVR_OUTDOOR_UNIT_DEVICEID*** for details.

Remarks

- The rules to generate device No. are as the follows:
  - The long No. of master station is "*00000001XX": "*"-project No., "001"-master station, "XX"-master station No. (01, 02, …, increased by 1).
  - The long No. of door station, villa door station, or doorphone is "*XXXXX000$$": "*"-project No.; "XXX"-building No.; "XX"-unit No.; "000"-door station; "$$"-door station No. ("00"-main door station, other values-sub door station).
  - The long No. of indoor station is "*XXXXX$$$$$": "*"-project No.; "XXX"-building No.; "XX"-unit No.; "$$$"-floor No.; "$$"-room No.
  - The long No. of outer door station is "*00000002XX": "*"-project No., "002"-outer door station, "XX"-outer door station No. (01, 02, …, increased by 1).
- In the actual application, one master station may belong to multiple communities, so the project No. of the device can be any of the existing project No., as long as the generated device No. is unique.

A.1.53 NET_DVR_VIDEO_INTERCOM_UNIT_RELATEDEV_UNION

Configuration union of the network device linked with video intercom.

Structure Definition

```
union{
  DWORD                    dwRes[256];
  NET_DVR_INDOOR_UNIT_RELATEDEV   struIndoorUnit;
  NET_DVR_OUTDOOR_UNIT_RELATEDEV  struOutdoorUnit;
  NET_DVR_MANAGE_UNIT_RELATEDEV   struManageUnit;
  NET_DVR_OUTDOOR_UNIT_RELATEDEV  struVillaUnit;
  NET_DVR_AGAIN_RELATEDEV       struAgainUnit;
}NET_DVR_VIDEO_INTERCOM_UNIT_RELATEDEV_UNION,*LPNET_DVR_VIDEO_INTERCOM_UNIT_RELATEDEV_UNION;
```

Members

dwRes

Union size, which is 1024 bytes (256 × 4).

struIndoorUnit

Linked network device parameters of indoor station, refer to the structure ***NET_DVR_INDOOR_UNIT_RELATEDEV*** for details.

struOutdoorUnit

Linked network device parameters of door station, outer door station, or intelligent access control device, refer to the structure ***NET_DVR_OUTDOOR_UNIT_RELATEDEV*** for details.

struManageUnit

Linked network device parameters of master station, refer to the structure ***NET_DVR_MANAGE_UNIT_RELATEDEV*** for details.

struVillaUnit

Linked network device parameters of villa door station, refer to the structure ***NET_DVR_OUTDOOR_UNIT_RELATEDEV*** for details.

struAgainUnit

Linked network device parameters of doorphone, refer to the structure ***NET_DVR_AGAIN_RELATEDEV*** for details.

See Also

NET_DVR_VIDEO_INTERCOM_RELATEDEV_CFG

A.1.54 NET_DVR_VIS_REGISTER_INFO

Registration information structure of door station.

Structure Definition

```
struct{
  DWORD          dwSize;
  DWORD          dwID;
  BYTE           szDevNumber[MAX_DEV_NUMBER_LEN/*328*/];
  DWORD          byMACAddr[MACADDR_LEN/*6*/];
  BYTE           byRes1[2];
  DWORD          sSerialNumber[SERIALNO_LEN/*48*/];
  NET_DVR_IPADDR    struDevIP;
  NET_DVR_TIME_EX   struRegisterTime;
  BYTE           byRegisterType;
  BYTE           byRes[127];
}NET_DVR_VIS_REGISTER_INFO, *LPNET_DVR_VIS_REGISTER_INFO;
```

Members

dwSize

Structure size.

dwID

Registration ID.

szDevNumber

Device No.

byMACAddr

MAC address.

byRes1

Reserved, set to 0.

sSerialNumber

Device serial No.

struDevIP

Device IP address, refer to the structure ***NET_DVR_IPADDR_UNION*** for details.

struRegisterTime

Registration time, for time format, refer to the structure ***NET_DVR_TIME_EX*** .

byRegisterType

Registration type: 0-reserved, 1-door station, 2-master station, 4-outer door station, 5-villa door station, 6-doorphone, 7-Infosight Client Software, 8-iVMS-4200 Client Software.

byRes

Reserved, set to 0.

A.1.55 NET_DVR_XML_CONFIG_INPUT

Input Parameter Structure of Message Transmission API (NET_DVR_STDXMLConfig)

Member	Data Type	Description
dwSize	DWORD	Structure size.
lpRequestUrl	void*	Request URL (command) for implement different functions, and it is in string format.
dwRequestUrlLen	DWORD	Request URL size.
lpInBuffer	void*	Buffer for storing input parameters (request messages), see the input content details structure in ***NET_DVR_MIME_UNIT*** .
dwInBufferSize	DWORD	Input buffer size.
dwRecvTimeOut	DWORD	Receiving timeout, unit: ms, 0-5000ms (default).
byForceEncrpt	BYTE	Whether to enable force encryption (the messages will be encrypted by AES algorithm for transmission): 0-no, 1-yes.
byNumOfMultiPart	BYTE	Number of message segments: 0-invalid; other values-number of message segments, which is transmitted by the parameter **lpInBuffer** in the structure ***NET_DVR_MIME_UNIT*** .
byRes	Array of BYTE	Reserved, set to 0.

Related API

NET_DVR_STDXMLConfig

A.1.56 NET_DVR_XML_CONFIG_OUTPUT

Output Parameter Structure of Message Transmission API (NET_DVR_STDXMLConfig)

Member	Data Type	Description
dwSize	DWORD	Structure size.
lpOutBuffer	void*	Buffer for storing output parameters (response messages), which is allocated when passing through URL by GET method.
dwOutBufferSize	DWORD	Output buffer size.
dwReturnedXMLSize	DWORD	Actual size of response message.

Member	Data Type	Description
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.
byRes	Array of BYTE	Reserved, set to 0.

Related API

NET_DVR_STDXMLConfig

A.1.57 NET_DVR_ZONE_ALARM_INFO

Zone alarm information structure

Structure Definition

```
struct{
  BYTE    byZoneName[NAME_LEN/*32*/];
  DWORD   dwZonendex;
  BYTE    byZoneType;
  BYTE    byRes[219];
}NET_DVR_ZONE_ALARM_INFO, *LPNET_DVR_ZONE_ALARM_INFO;
```

Members

byZoneName

Zone name.

dwZonendex

Zone No.

byZoneType

Zone types, which are enumerated below.

```
enum _ALARM_ZONE_TYPE_ENUM_{
  ENUM_ALARM_ZONE_TYPE_MANUAL         = 0,
  ENUM_ALARM_ZONE_TYPE_MAGNETIC       = 1,
  ENUM_ALARM_ZONE_TYPE_SMOKE          = 2,
  ENUM_ALARM_ZONE_TYPE_ACTIVE_INFRARED  = 3,
  ENUM_ALARM_ZONE_TYPE_PASSIVE_INFRARED = 4,
  ENUM_ALARM_ZONE_TYPE_GAS            = 11
}ALARM_ZONE_TYPE_ENUM
```

ENUM_ALARM_ZONE_TYPE_MANUAL

Panic alarm

ENUM_ALARM_ZONE_TYPE_MAGNETIC

Door magnetic alarm

ENUM_ALARM_ZONE_TYPE_SMOKE

Smoke detection alarm

ENUM_ALARM_ZONE_TYPE_ACTIVE_INFRARED

Active IR alarm

ENUM_ALARM_ZONE_TYPE_PASSIVE_INFRARED

Passive IR alarm

ENUM_ALARM_ZONE_TYPE_GAS

Gas detection alarm

byRes

Reserved, set to 0.

See Also

NET_DVR_VIDEO_INTERCOM_ALARM_INFO_UNION

A.1.58 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 **lpBuffer** is 4-byte status.
NET_SDK_CALLBACK_STATUS_FAILED	1002	Failed. The **lpBuffer** is the value of 4-byte status and 4-byte error code.

A.1.59 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_FIL	8	Timing generator file.
UPLOAD_DATA_TO_DB	9	File to be uploaded to picture and video library.
UPLOAD_BACKGROUND_PIC	10	Background picture.
UPLOAD_CALIBRATION_FILE	11	Calibration file.
UPLOAD_TME_FILE	12	Entrance and exiting management file.
UPLOAD_VEHICLE_BLACKWHITELST_FILE	13	Vehicle blacklist file.
UPLOAD_PICTURE_TO_CLOUD	15	Picture file to be uploaded to cloud storage.
UPLOAD_VIDEO_FILE	16	Video file.
UPLOAD_SCREEN_FILE	17	Screen server file.
UPLOAD_PUBLISH_MATERIAL	18	Local material file of information release system.
UPLOAD_PUBLISH_UPGRADE_FILE	19	Upgrade file of information release system.
UPLOAD_RING_FILE	20	Ringtone file.
UPLOAD_ENCRYPT_CERT	21	Encryption certificate.
UPLOAD_THERMOMETRIC_FILE	22	Calibration file for temperature measurement.

Enumeration Type	Macro Definition Value	Description
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_BLACK_LIST	32	ID card blacklist.
UPLOAD_HD_CAMERA_CORRECT_TABLE	33	HD camera correction sheet (CAL format).
UPLOAD_FD_DATA	35	Face data file to be imported to face picture library.
UPLOAD_FACE_DATA	36	Face picture file to be imported to face picture library.
UPLOAD_FACE_ANALYSIS_DATA	37	Picture file to be imported to picture recognition target.
UPLOAD_FILEVOLUME_DATA	38	File volume file
IMPORT_DATA_TO_FACELIB	39	Face data (face picture and picture additional information) to be imported to face picture library of device.
UPLOAD_LEFTEYE_4K_CALIBFILE	40	Camera calibration parameter file.
UPLOAD_SECURITY_CFG_FILE	41	Configuration file to be securely imported.
UPLOAD_RIGHT_CONTROLLER_AUDIO	42	Audio file of main controller.
UPLOAD_MODBUS_CFG_FILE	43	Configuration file of Modbus protocol.
UPLOAD_NOTICE_VIDEO_DATA	44	Bulletin video file.

Enumeration Type	Macro Definition Value	Description
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.

A.1.60 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	***NET_ALARM_RECORD FILE_LOSS***	Video loss alarm information, the value of **dwAlarmType** in ***NET_DVR_ALARMINFO_DEV_V40*** is 8.
struStreamException	***NET_ALARM_STREAM_EXCEPTION***	Streaming exception alarm information, the value of **dwAlarmType** in ***NET_DVR_ALARMINFO_DEV_V40*** is 9.

Member	Data Type	Description
struResourceUsage	***NET_ALARM_RESOURCE_USAGE***	Resource usage alarm information, the value of **dwAlarmType** in ***NET_DVR_ALARMINFO_DEV_V40*** is 10.
struRecordException	***NET_ALARM_RECORD_EXCEPTION***	Recording exception alarm information, the value of **dwAlarmType** in ***NET_DVR_ALARMINFO_DEV_V40*** is 12.

A.1.61 NET_ALARM_RECORD_EXCEPTION

Structure about Recording Exception Alarm Information

Member	Data Type	Description
byReason	BYTE	Exception reason: 0-video volume full, 1-video volume exception, 2-no available video volume.
byRes1	BYTE[]	Reserved, set to 0. The maximum array length is 3 bytes.
sVolumeName	BYTE[]	Video volume name, the maximum array length is "MAX_VOLUMENAME_LEN" (32 bytes).
dwVolumeID	DWORD	Video volume ID, or HDD No.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 452 bytes.

A.1.62 NET_ALARM_RECORDFILE_LOSS

Structure about Video Loss Alarm Information

Member	Data Type	Description
struInspectStart	***NET_DVR_TIME_EX***	Start time of video loss check.
struInspectEnd	***NET_DVR_TIME_EX***	End time of video loss check.
struIP	***NET_DVR_IPADDR_UNION***	IP address of video loss channel.
dwChanNo	DWORD	Channel No.
dwIDIndex	DWORD	Encoder ID.

Member	Data Type	Description
sName	BYTE[]	Encoder name, the maximum array length is "STREAM_ID_LEN" (32 bytes).
struLossStartTime	***NET_DVR_TIME_EX***	Start time of video loss.
struLossEndTime	***NET_DVR_TIME_EX***	End time of video loss.
dwLostNum	DWORD	Number of lost video files, 0xffffffff-all video files are lost.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 240 bytes.

A.1.63 NET_ALARM_RESOURCE_USAGE

Structure about Resource Usage Alarm Information

Member	Data Type	Description
byLevel	BYTE	Usage alarm level: 0-normal, 1-alarm level 1, 2-alarm level 2, 3-alarm level 3.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 491 bytes.

A.1.64 NET_ALARM_STREAM_EXCEPTION

Structure about Video Exception Alarm Information

Member	Data Type	Description
struIP	***NET_DVR_IPADDR_UNION***	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.

A.1.65 NET_DVR_ALARMER

Alarm Device Information Structure

Member	Data Type	Description
byUserIDValid	BYTE	Whether the user ID is valid: 0-no, 1-yes
bySerialValid	BYTE	Whether the serial No. is valid: 0-no, 1-yes
byVersionValid	BYTE	Whether the version No. is valid: 0-no, 1-yes
byDeviceNameValid	BYTE	Whether the device name is valid: 0-no, 1-yes
byMacAddrValid	BYTE	Whether the MAC address is valid: 0-no, 1-yes
byLinkPortValid	BYTE	Whether the login port No. is valid: 0-no, 1-yes
byDeviceIPValid	BYTE	Whether the device IP address is valid: 0-no, 1-yes
bySocketIPValid	BYTE	Whether the Socket IP address is valid: 0-no, 1-yes
lUserID	LONG	Value returned by ***NET_DVR_Login_V40*** , it is valid when arming.
sSerialNumber	Array of BYTE	Serial No.
dwDeviceVersion	DWORD	Version information
sDeviceName	Array of char	Device name
byMacAddr	Array of BYTE	MAC address
wLinkPort	WORD	Device communication port No.
sDeviceIP	Array of char	Device IP address
sSocketIP	Array of char	Socket IP address when actively uploading alarm.
byIpProtocol	BYTE	Network protocol: 0-IPv4, 1-IPv6
byRes2	Array of BYTE	Reserved, set to 0.

A.1.66 NET_DVR_ALARMINFO_DEV

Device Alarm Information Structure

Memeber	Data Type	Description
dwAlarmType	DWORD	Alarm types: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel.
struTime		Alarm time
byRes	Array of BYTE	Reserved, set to 0.
dwNumber	DWORD	Number of alarm triggered channels.
pNO	WORD*	Channel No. or disk No., which ranges from 0 to 65535.

Remarks

For **pNO**: if **dwAlarmType** is 0, 3, 6, or 7, it may be channel No.; if **dwAlarmType** is 5, it may be disk No.

A.1.67 NET_DVR_ALARMINFO_DEV_V40

Structure about CVR Alarm Information

Member	Data Type	Description
dwAlarmType	DWORD	Alarm categories: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel, 8-video loss alarm, 9-real-time health monitoring alarm, 10-usage alarm, 11-CVR exception recovered, 12-recording exception.
struTime	***NET_DVR_TIME***	Alarm time

Member	Data Type	Description
uSubAlarmInfo	***NET_ALARM_CVR_SUBINFO_UNION***	CVR alarm information structure, and it is valid when the alarm type is 8, 9, 10, and 12.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 256 bytes.
dwNumber	DWORD	Number of alarm triggered channels.
pNO	WORD*	Channel No. or disk No., which ranges from 0 to 65535.

Remarks

For **pNO**: if **dwAlarmType** is 0, 3, 6, or 7, it may be channel No.; if **dwAlarmType** is 5, it may be disk No.

A.1.68 NET_DVR_ALARMINFO_V30

Structure About Uploaded Alarm Information

Member	Data Type	Description
dwAlarmType	DWORD	Alarm types: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel, 8-video loss alarm, 9-real-time health monitoring alarm, 10-usage alarm, 11-CVR exception recovered, 12-recording exception.
dwAlarmInputNumber	DWORD	Alarm input No., it is valid when alarm type is 0 or 23
byAlarmOutputNumber	Array of BYTE	The triggered alarm output No. E.g. dwAlarmOutputNumber[0]==1 indicates that alarm output No.1 is triggered; dwAlarmOutputNumber[1]==1 indicates that alarm output No.2 is triggered.
byAlarmRelateChannel	Array of BYTE	The triggered recording channel No.: 0-not triggered, 1-triggered. E.g.

Member	Data Type	Description
		dwAlarmRelateChannel[0]==1 indicates that the channel No.1 is triggered to record.
byChannel	Array of BYTE	Alarm channel, it is valid when alarm type is 2, 3, 6, 9, 10 or 11. E.g. dwChannel[0]==1 indicates that the channel No. is in alarm.
byDiskNumber	Array of BYTE	Alarm HDD, it is valid when alarm type is 1, 4, or 5. E.g. dwDiskNumber [0]==1 indicates that the HDD No.1 is abnormal.

Remarks

The time interval to upload the alarm of face picture library changed is 1 hour; for other alarm type, the alarm information is uploaded in real-time, and the time interval is 1s. Currently, editing the time interval is not supported.

A.1.69 NET_DVR_ALARMINFO_V40

Structure About Uploaded Alarm Information

Member	Data Type	Description
struAlarmFixedHeader	***NET_DVR_ALRAM_FIXED_HEADER***	Constant content in alarm information, see details in the structure .
pAlarmData	DWORD*	Variable content in alarm information

Remarks

- The time interval to upload the alarm of face picture library changed is 1 hour; for other alarm type, the alarm information is uploaded in real-time, and the time interval is 1s. Currently, editing the time interval is not supported.
- The content of **pAlarmData** varies with the value of **dwAlarmType** in the structure ***NET_DVR_ALRAM_FIXED_HEADER*** , see details in the table below:

Table A-1 Relations Between pAlarmData and dwAlarmType

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

A.1.70 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	***NET_DVR_TIME_EX***	Alarm time
uStruAlarm	Union (***Table 11-2***)	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 **byISO8601** is "1".
cTimeDifferenceM	char	Time difference between time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when **byISO8601** is "1".
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 5 bytes.

Table A-2 Union about Alarm Information Structures (uStruAlarm)

Member	Data Type	Description
byUnionLen	Array of BYTE	Union size, which is 116 bytes.
struIOAlarm	Struct (***Table 11-3***)	Structure about alarm input parameters
struAlarmChannel	Struct (***Table 11-4***)	Structure about alarm channel parameters
struAlarmHardDisk	Struct (***Table 11-5***)	Structure about HDD alarm parameters
struRecordingHost	Struct (***Table 11-6***)	Structure about alarm parameters of education sharing system
struVoltageInstable	Struct (***Table 11-7***)	Structure about alarm parameters of supply voltage exception
struPTLocking	Struct (***Table 11-8***	Structure about parameters of PTZ locked alarm

Table A-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

Member	Data Type	Description
		alarm outputs by **pAlarmData** in ***NET_DVR_ALARMINFO_V40*** , each alarm output is represented by 4 bytes.
dwTrigerRecordChanNum	DWORD	The number of triggered recording channels. It is used for calculating the number of all triggered recording channels by **pAlarmData** of ***NET_DVR_ALARMINFO_V40*** , each channel is represented by 4 bytes.

Table A-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 **pAlarmData** of ***NET_DVR_ALARMINFO_V40*** , each alarm channel is represented by 4 bytes.
dwPicLen	DWORD	Size of JPEG picture.
byPicURL	BYTE	Picture data format: 0-binary data, 1-URL.
byTarget	BYTE	Detection target type: 0-not supported, 1-person, 2-vehicle.
byRes1	Array of BYTE	Reserved, the maximum size is 2 bytes.
pDataBuff	char*	Alarm picture data or URL. The pointer size is 8 bytes.
byRes3	Array of BYTE	Reserved, the maximum size is 4 bytes. This member is only available for 64-bit Window operating system and 64-bit Linux operating system.

Table A-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 **pAlarmData** of ***NET_DVR_ALARMINFO_V40*** , each alarm HDD is represented by 4 bytes.

Table A-6 Structure about Alarm Parameters of Education Sharing System (struRecordingHost)

Member	Data Type	Description
bySubAlarmType	BYTE	Alarm minor type: 1-one-touch post-record
byRes1	Array of BYTE	Reserved, set to 0. The maximum size is 3 bytes.
struRecordEndTime	***NET_DVR_TIME_EX***	Recording end time.

Table A-7 Structure about Alarm Parameters of Supply Voltage Exception (struVoltageInstable)

Member	Data Type	Description
fVoltageValue	float	Supply voltage, unit: V, corrects to one decimal place.
byVoltageAlarmType	BYTE	Supply voltage exception type: 0-high supply voltage, 1-low supply voltage
byRes1	Array of BYTE	Reserved, set to 0. The maximum size is 3 bytes.

Table A-8 Structure about Parameters of PTZ Locked Alarm (struPTLocking)

Member	Data Type	Description
fTemperature	float	Sensor temperature, which is accurate to one decimal place.
dwCustomInfoLength	DWORD	Custom information length.
pCustomInfo	BYTE*	Custom information.
byType	BYTE	PTZ locked direction: 1-panning is locked, 2-tilting is locked.
byDeicingEnabled	BYTE	Whether to enable heat for PTZ: 0-no, 1-yes.

Remarks

dwAlarmType==0, 23 corresponds to the structure struIOAlarm; **dwAlarmType**== 2/3/6/9/10/11/13/15/16/28 corresponds to the structure struAlarmChannel; **dwAlarmType**== 1/4/5 corresponds to the structure struAlarmHardDisk; **dwAlarmType**== 17 corresponds to the structure struRecordingHost; **dwAlarmType**== 31 corresponds to the structure struVoltageInstable; for other value, the union is not available.

A.1.71 NET_DVR_ALARM_ISAPI_INFO

Structure about Alarm Information Transmitted Based on Text Protocol

Member	Data Type	Description
pAlarmData	char*	Alarm information based on text protocol (XML or JSON message without binary data).
dwAlarmDataLen	DWORD	Alarm data length.
byDataType	BYTE	Alarm data type: 0-invalid, 1-XML, 2-JSON.
byPicturesNumber	BYTE	The number of pictures (number of **pPicPackData** returned). When this member is 1, only one structure of ***NET_DVR_ALARM_ISAPI_PICDATA*** will be returned by **pPicPackData**. When this member is larger than 1, multiple structures of ***NET_DVR_ALARM_ISAPI_PICDATA*** will be returned by **pPicPackData**.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 2 bytes.
pPicPackData	void*	Alarm picture structure, see ***NET_DVR_ALARM_ISAPI_PICDATA*** for details.
byRes	Array of BYTE	Reserved. The maximum size is 32 bytes.

Remarks

When enabling the listening mode, you should call the network configuration API based on text protocol to set the IP address for the listening service.

A.1.72 NET_DVR_ALARM_ISAPI_PICDATA

Structure about Alarm Picture Data Transmitted Based on Text Protocol

Member	Data Type	Description
dwPicLen	DWORD	Alarm picture data length.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 4 bytes.
szFilename	Array of char	Picture file saving path, including file name. The maximum size is 256 bytes.
pPicData	BYTE*	Pointer that pointing to the uploaded image data.

A.1.73 NET_DVR_ETHERNET_V30

Ethernet Configuration Structure

Member	Data Type	Description
struDVRIP	***NET_DVR_IPADDR_UNION***	Device IP address
struDVRIPMask	***NET_DVR_IPADDR_UNION***	Mask of device IP address
dwNetInterface	DWORD	Network interface type: 1-10MBase-T; 2-10MBase-T (full duplex); 3-100MBase-TX; 4-100M (full duplex); 5-10M/100M/1000M (self-adaptive); 6-1000M (full duplex)
wDVRPort	WORD	Device port No.
wMTU	WORD	MTU settings, the default is 1500.
byMACAddr	Array of BYTE	Device physical address.
byEthernetPortNo	BYTE	Network interface No.: 0-invalid, 1-interface 0, 2-interface 1, and so on. This parameter is read-only.
byRes	Array of BYTE	Reserved.

A.1.74 NET_DVR_IPADDR_UNION

IP Address Union

Member	Data Type	Description
szIPv4	char[]	IPv4 address. The maximum length is 16 bytes.
szIPv6	char[]	IPv6 address. The maximum length is 256 bytes.

A.1.75 NET_DVR_NETCFG_V50

Network Configuration Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
struEtherNet	Array of ***NET_DVR_ETHERNET_V30***	Ethernet interface
struRes1	Array of	Reserved, set to 0.
struAlarmHostIpAddr	***NET_DVR_IPADDR_UNION***	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	***NET_DVR_IPADDR_UNION***	IP address of domain name server 1
struDnsServer2IpAddr	***NET_DVR_IPADDR_UNION***	IP address of domain name server 2
byIpResolver	Array of BYTE	IP resolver domain name or IP address (if the port No. of device is 8000, the domain name is not supported).
wIpResolverPort	WORD	IP resolver port No.
wHttpPortNo	WORD	HTTP port No.

Member	Data Type	Description
struMulticastIpAddr	***NET_DVR_IPADDR_UNION***	Multicast group address
struGatewayIpAddr	***NET_DVR_IPADDR_UNION***	Gateway address
struPPPoE	***NET_DVR_PPPOECFG***	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	***NET_DVR_IPADDR_UNION***	IP address of listening host 2
byEnableDNS	BYTE	DNS address setting mode: 0-automatically get, 1-manually set.
byRes	Array of BYTE	Reserved, set to 0

Remarks

- For device only supports the private protocol with version 3.0 or lower, when the parameter **byUseDhcp**="0xff", you should set the device IP address to null, and then the device will automatically get the DHCP information.
- When the parameter **byIPv6Mode** is set to 0 or 2, setting IPv6 address in the parameter **struEtherNet** is not required, it will be obtained automatically by the device; when **byIPv6Mode** is set to 1, you should set IPv6 address. As there are multiple IPv6 addresses, the IPv6 address of current logged-in device may be different with that in **struEtherNet**.

A.1.76 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	***NET_DVR_IPADDR_UNION***	PPPoE IP address

A.1.77 NET_DVR_SETUPALARM_PARAM_V50

Arming Parameter Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
byLevel	BYTE	Arming priority: 0-high, 1-medium, 2-low.
byAlarmInfoType	BYTE	Intelligent traffic alarm information type: 0-old (NET_DVR_PLATE_RESULT),1-new (NET_ITS_PLATE_RESULT).
byRetAlarmTypeV40	BYTE	0-the motion detection, video loss, video tampering, and alarm input alarm information is uploaded in normal mode (alarm type: COMM_ALARM_V30, alarm information structure: ***NET_DVR_ALARMINFO_V30***); 1-alarm information is uploaded in variable size (alarm type: COMM_ALARM_V40, alarm information structure: ***NET_DVR_ALARMINFO_V40***).
byRetDevInfoVersion	BYTE	Alarm types of CVR: 0-COMM_ALARM_DEVICE (alarm information structure: ***NET_DVR_ALARMINFO_DEV***), 1-COMM_ALARM_DEVICE_V40 (alarm information structure: ***NET_DVR_ALARMINFO_DEV_V40***).
byRetVQDAlarmType	BYTE	VQD alarm types: 0-COMM_ALARM_VQD (alarm information structure: NET_DVR_VQD_DIAGNOSE_INFO), 1-COMM_ALARM_VQD_EX (alarm information structure: NET_DVR_VQD_ALARM, including camera information and captured pictures)
byFaceAlarmDetection	BYTE	Face detection alarm types: 1-face detection alarm (alarm type: COMM_ALARM_FACE_DETECTION, alarm information structure: NET_DVR_FACE_DETECTION), 0-face capture alarm (alarm type: COMM_UPLOAD_FACESNAP_RESULT, alarm information structure: NET_VCA_FACESNAP_RESULT).
bySupport	BYTE	Capabilities, which is represented by bit:

Member	Data Type	Description
		• bit0-whether to upload picture: 0-yes, 1-no • bit1-whether to enable ANR: 0-no, 1-yes • bit4-whether to upload behavior analysis events of all detection targets: 0-no, 1-yes. It is used to enable the NVR to get events of all targets detected by network cameras. • bit5-whether to enable all-day event or alarm uploading: 0-no, 1-yes. It is used to enable the NVR to receive all alarms from network cameras.
byBrokenNetHttp	BYTE	ANR type, which is represented by bit and should be supported by device: • bit0-whether to enable ANR for ANPR: 0-no, 1-yes. • bit1-whether to enable ANR for people counting: 0-no, 1-yes. • bit2-whetehr to enable ANR for heat map: 0-no, 1-yes. • bit3-whether to enable ANR for face capture: 0-no, 1-yes. • bit4-whether to enable ANR for face picture comparison: 0-no, 1-yes. • bit5-whether to enable ANR for JSON message transmission: 0-no, 1-yes. • bit6: whether to enable ANR for uploading heat map data by dwell time duration and by people quantity: 0-no, 1-yes. • bit7: whether to enable ANR for uploading intersection analysis result: 0-no, 1-yes.
wTaskNo	BYTE	Task No.
byDeployType	BYTE	Arming type: 0-arm via client software, 1-real-time arming.
bySubScription	BYTE	Subscription parameters, which is represent by bit. Bit7-whether to upload picture after subscribing motion detection alarm by person or vehicle: 0-no, 1-yes.
byRes1	Array [BYTE]	Reserved, set to 0. The maximum size is 2 bytes.

Member	Data Type	Description
byAlarmTypeURL	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. • bit0-type of captured face pictures: 0-binary data, 1-URL • bit1-type of picture uploaded in message: 0-binary, 1-URL • bit2-type of picture uploaded for face picture comparison: 0-binary, 1-URL
byCustomCtrl	BYTE	Custom control type, which is represented by bit, bit0-whether to upload the face thumbnail of the front passenger: 0-no, 1-yes
byRes4	Array [BYTE]	Reserved, set to 0. The maximum size is 128 bytes.

Remarks

- The parameters **byLevel** and **byAlarmInfoType** are available for traffic cameras. Up to 1 cameras can be armed in the priority of level 0, up to 3 cameras can be armed in the priority of level 1, and up to 5 cameras can be armed in the priority of level 3, the alarm/event information from the camera in highest priority will be uploaded first.
- For arming via client software, only supports arming one channel, and supports uploading the alarm/event when device is offline; for real-time arming, up to four channels can be armed at same time, but uploading alarm/event when device is offline is not supported.
- The parameter **wTaskNo** is used to distinguish different arming connections. If the value of this parameter in different arming connections is same, error will be returned.

A.1.78 NET_DVR_TIME

Time Parameter Structure

Member	Data Type	Description
dwYear	DWORD	Year
dwMonth	DWORD	Month
dwDay	DWORD	Day
dwHour	DWORD	Hour
dwMinute	DWORD	Minute
dwSecond	DWORD	Second

A.1.79 NET_DVR_TIME_EX

Extended Time Parameter Structure

Member	Data Type	Description
wYear	WORD	Year
byMonth	BYTE	Month
byDay	BYTE	Day
byHour	BYTE	Hour
byMinute	BYTE	Minute
bySecond	BYTE	Second
byRes	BYTE	Reserved.

A.2 Request URL

The request URLs for implementing the functions in this manual are listed here for reference. You can search for the URLs and view their definitions.

A.2.1 /ISAPI/VideoIntercom/callerInfo/capabilities?format=json

Get the capability of getting the caller information.

Request URI Definition

Table A-9 GET /ISAPI/VideoIntercom/callerInfo/capabilities?format=json

Method	GET
Description	Get the capability of getting the caller information.
Query	**format**: determine the format of request or response message.
Request	None.
Response	Succeeded: ***JSON_Cap_CallerInfo*** Failed: ***JSON_ResponseStatus***

A.2.2 /ISAPI/VideoIntercom/callerInfo?format=json

Get the caller information.

Request URI Definition

Table A-10 GET /ISAPI/VideoIntercom/callerInfo?format=json

Method	GET
Description	Get the caller information.
Query	**format**: determine the format of request or response message.
Request	None.
Response	Succeeded: ***JSON_CallerInfo*** Failed: ***JSON_ResponseStatus***

A.2.3 /ISAPI/VideoIntercom/capabilities

Get video intercom capability.

Request URI Definition

Table A-11 GET /ISAPI/VideoIntercom/capabilities

Method	GET
Description	Get video intercom capability.
Query	None.

Request	None.
Response	Succeeded: ***XML_VideoIntercomCap*** Failed: ***XML_ResponseStatus***

A.2.4 /ISAPI/VideoIntercom/IntercomProtocolType

Get or set parameters for switching between the private video intercom protocol and the standard SIP (Session Initiation Protocol).

Request URI Definition

Table A-12 GET /ISAPI/VideoIntercom/IntercomProtocolType

Method	GET
Description	Get parameters for switching between the private video intercom protocol and the standard SIP.
Query	None.
Request	None.
Response	Succeeded: ***XML_IntercomProtocolType*** Failed: ***XML_ResponseStatus***

Table A-13 PUT /ISAPI/VideoIntercom/IntercomProtocolType

Method	PUT
Description	Set parameters for switching between the private video intercom protocol and the standard SIP.
Query	None.
Request	***XML_IntercomProtocolType***
Response	***XML_ResponseStatus***

A.2.5 /ISAPI/VideoIntercom/IntercomProtocolType/capabilities

Get the configuration capability for switching between the private protocol and the standard SIP (Session Initiation Protocol).

Request URI Definition

Table A-14 GET /ISAPI/VideoIntercom/IntercomProtocolType/capabilities

Method	GET
Description	Get the configuration capability for switching between the private protocol and the standard SIP (Session Initiation Protocol).
Query	None.
Request	None.
Response	Succeeded: ***XML_Cap_IntercomProtocolType*** Failed: ***XML_ResponseStatus***

A.2.6 /ISAPI/VideoIntercom/keyCfg

Get the parameters of pressing the button to call of multiple buttons.

Request URI Definition

Table A-15 GET /ISAPI/VideoIntercom/keyCfg

Method	GET
Description	Get the parameters of pressing the button to call of multiple buttons.
Query	None.
Request	None.
Response	Succeeded: ***XML_KeyCfgList*** Failed: ***XML_ResponseStatus***

A.2.7 /ISAPI/VideoIntercom/keyCfg/<ID>

Get or set the parameters of pressing the button to call for a specific button.

Request URI Definition

Table A-16 GET /ISAPI/VideoIntercom/keyCfg/<ID>

Method	GET
Description	Get the parameters of pressing the button to call for a specific button.
Query	None.

Request	None.
Response	Succeeded: ***XML_KeyCfg*** Failed: ***XML_ResponseStatus***

Table A-17 PUT /ISAPI/VideoIntercom/keyCfg/<ID>

Method	PUT
Description	Set the parameters of pressing the button to call for a specific button.
Query	None.
Request	***XML_KeyCfg***
Response	***XML_ResponseStatus***

Remarks

The <**ID**> in the request URI refers to the button No.

A.2.8 /ISAPI/VideoIntercom/keyCfg/<ID>/capabilities

Get the configuration capability of pressing the button to call.

Request URI Definition

Table A-18 GET /ISAPI/VideoIntercom/keyCfg/<ID>/capabilities

Method	GET
Description	Get the configuration capability of pressing the button to call.
Query	None.
Request	None.
Response	Succeeded: ***XML_Cap_KeyCfg*** Failed: ***XML_ResponseStatus***

A.2.9 /ISAPI/VideoIntercom/passwordAuthentication

Verify video intercom encryption.

Request URI Definition

Table A-19 PUT /ISAPI/VideoIntercom/passwordAuthentication

Method	PUT
Description	Verify video intercom encryption.

Query	None.
Request	***XML_PasswordAuthenticationCfg***
Response	***XML_ResponseStatus***

A.2.10 /ISAPI/VideoIntercom/passwordAuthentication/capabilities

Get encryption verification capability of video intercom.

Request URI Definition

Table A-20 GET /ISAPI/VideoIntercom/passwordAuthentication/capabilities

Method	GET
Description	Get encryption verification capability of video intercom.
Query	None.
Request	None.
Response	Succeeded: ***XML_Cap_PasswordAuthenticationCfg*** Failed: ***XML_ResponseStatus***

A.2.11 /ISAPI/VideoIntercom/ring

Delete all rings.

Request URI Definition

Table A-21 DELETE /ISAPI/VideoIntercom/ring

Method	DELETE
Description	Delete all rings.
Query	None.
Request	None.
Response	***XML_ResponseStatus***

A.2.12 /ISAPI/VideoIntercom/ring/capabilities

Get ring configuration capability.

Request URI Definition

Table A-22 GET /ISAPI/VideoIntercom/ring/capabilities

Method	GET
Description	Get ring configuration capability.
Query	None.
Request	None.
Response	Succeeded: ***XML_Cap_RingParam*** Failed: ***XML_ResponseStatus***

A.2.13 /ISAPI/VideoIntercom/ring/ringID/<ID>

Delete ring by ring ID.

Request URI Definition

Table A-23 DELETE /ISAPI/VideoIntercom/ring/ringID/<ID>

Method	DELETE
Description	Delete ring by ring ID.
Query	None.
Request	None.
Response	***XML_ResponseStatus***

Remarks

The <**ID**> in the request URI refers to the ring ID.

A.2.14 /ISAPI/VideoIntercom/scene/nowMode

Get or set scene parameters of video intercom.

Request URI Definition

Table A-24 GET /ISAPI/VideoIntercom/scene/nowMode

Method	GET
Description	Get scene parameters of video intercom.
Query	None.

Request	None.
Response	Succeeded: ***XML_SceneNowMode*** Failed: ***XML_ResponseStatus***

Table A-25 PUT /ISAPI/VideoIntercom/scene/nowMode

Method	PUT
Description	Set scene parameters of video intercom.
Query	None.
Request	***XML_SceneNowMode***
Response	***XML_ResponseStatus***

A.2.15 /ISAPI/VideoIntercom/scene/nowMode/capabilities

Get scene configuration capability of video intercom.

Request URI Definition

Table A-26 GET /ISAPI/VideoIntercom/scene/nowMode/capabilities

Method	GET
Description	Get scene configuration capability of video intercom.
Query	None.
Request	None.
Response	Succeeded: ***XML_Cap_SceneNowMode*** Failed: ***XML_ResponseStatus***

A.2.16 /ISAPI/VideoIntercom/SubModules/<ID>/CallCfg/capabilities?format=json

Get configuration capability of nametag module.

Request URI Definition

Table A-27 GET /ISAPI/VideoIntercom/SubModules/<ID>/CallCfg/capabilities?format=json

Method	GET
Description	Get configuration capability of nametag module.
Query	**format**: determine the format of request or response message.

Request	None.
Response	Succeeded: ***JSON_Cap_CallCfg*** Failed: ***JSON_ResponseStatus***

Remarks

The <**ID**> in the request URI refers to the sub module ID or DIP switch address, which should be larger than 0.

A.2.17 /ISAPI/VideoIntercom/SubModules/<ID>/CallCfg?format=json

Get or set nametag module parameters.

Request URI Definition

Table A-28 GET /ISAPI/VideoIntercom/SubModules/<ID>/CallCfg?format=json

Method	GET
Description	Get nametag module parameters.
Query	**format**: determine the format of request or response message.
Request	None.
Response	Succeeded: ***JSON_CallCfg*** Failed: ***JSON_ResponseStatus***

Table A-29 PUT /ISAPI/VideoIntercom/SubModules/<ID>/CallCfg?format=json

Method	PUT
Description	Set nametag module parameters.
Query	**format**: determine the format of request or response message.
Request	***JSON_CallCfg***
Response	***JSON_ResponseStatus***

Remarks

The <**ID**> in the request URI refers to the sub module ID or DIP switch address, which should be larger than 0.

A.2.18 /ISAPI/VideoIntercom/SubModules/<ID>/Configurations/capabilities?format=json

Get configuration capability of sub module.

Request URI Definition

Table A-30 GET /ISAPI/VideoIntercom/SubModules/<ID>/Configurations/capabilities?format=json

Method	GET
Description	Get configuration capability of sub module.
Query	**format**: determine the format of request or response message.
Request	None.
Response	Succeeded: ***JSON_Cap_SubModulesCfg*** Failed: ***JSON_ResponseStatus***

Remarks

The <**ID**> in the request URI refers to the sub module ID or DIP switch address, which should be larger than 0.

A.2.19 /ISAPI/VideoIntercom/SubModules/<ID>/Configurations?format=json

Get or set sub module parameters.

Request URI Definition

Table A-31 GET /ISAPI/VideoIntercom/SubModules/<ID>/Configurations?format=json

Method	GET
Description	Get sub module parameters.
Query	**format**: determine the format of request or response message.
Request	None.
Response	Succeeded: ***JSON_SubModulesCfg*** Failed: ***JSON_ResponseStatus***

Table A-32 PUT /ISAPI/VideoIntercom/SubModules/<ID>/Configurations?format=json

Method	PUT
Description	Set sub module parameters.
Query	**format**: determine the format of request or response message.
Request	***JSON_SubModulesCfg***
Response	***JSON_ResponseStatus***

Remarks

The <**ID**> in the request URI refers to the sub module ID or DIP switch address, which should be larger than 0.

A.2.20 /ISAPI/VideoIntercom/SubModules/capabilities?format=json

Get capability of getting sub module status.

Request URI Definition

Table A-33 GET /ISAPI/VideoIntercom/SubModules/capabilities?format=json

Method	GET
Description	Get capability of searching for sub module status.
Query	**format**: determine the format of request or response message.
Request	None.
Response	Succeeded: ***JSON_Cap_SubModules*** Failed: ***JSON_ResponseStatus***

A.2.21 /ISAPI/VideoIntercom/SubModules?format=json

Get sub module status.

Request URI Definition

Table A-34 GET /ISAPI/VideoIntercom/SubModules?format=json

Method	GET
Description	Get sub module status.
Query	**format**: determine the format of request or response message.
Request	None.
Response	Succeeded: ***JSON_SubModules*** Failed: ***JSON_ResponseStatus***

A.2.22 /ISAPI/VideoIntercom/WorkStatus

Get working status of door station.

Request URI Definition

Table A-35 GET /ISAPI/VideoIntercom/WorkStatus

Method	GET
Description	Get working status of door station.
Query	None.
Request	None.
Response	Succeeded: ***XML_WorkStatus*** Failed: ***XML_ResponseStatus***

A.3 Request and Response Message

The request and response messages in XML or JSON format of each request URL are listed here for reference. You can search for the parameters by the message name.

A.3.1 JSON_CallCfg

JSON message about nametag module parameters

```
{
  "CallCfg": {
    "KeyBindings": [{
/*this node is valid for the nametag module*/
      "buttonNumber": ,
/*optional, integer type, button No. which is valid for nametag module only*/
      "roomNumber": ,
/*optional, integer type, room No. which is valid for nametag module only*/
      "callNumber": ""
/*optional, string type, call number which is valid for nametag module only*/
    }]
  }
}
```

A.3.2 JSON_CallerInfo

Message about the caller information in JSON format.

```
{
  "CallerInfo":{
    "buildingNo": ,
/*optional, int, building No.*/
    "floorNo": ,
```

```
/*optional, int, floor No.*/
   "zoneNo": ,
/*optional, int, project No.*/
   "unitNo": ,
/*optional, int, unit No.*/
   "devNo": ,
/*optional, int, device serial No.*/
   "devType": ,
/*required, int, device type: 1-door station, 2-master station, 3-indoor station, 4-outer door station, 5-villa door station, 6-doorphone, 7-Infosight Client Software, 8-iVMS-4200 Client Software, 9-APP, 10-doorbell, 11-VOIP Client Software, 12-network camera, 13-access control device*/
   "lockNum": ,
/*optional, int, number of locks*/
   "voipNo": ,
/*optional, string, VOIP Client Software number, this field is valid when devType is 11*/
   "status":
/*required, string, status: "idle", "ring", "ring"-the call is in progress*/
  }
}
```

A.3.3 JSON_Cap_CallCfg

JSON message about the configuration capability of the nametag module

```
{
  "CallCfg": {
    "KeyBindings": {
      "maxSize": 20,
/*maximum number of buttons, which can be same as the maximum number of nametag modules*/
      "roomNumber": {
/*required, integer type, room No., which is valid only for nametag module*/
        "@min": 1,
        "@max": 65535
      },
      "buttonNumber": {
/*required, integer type, button No., which is valid only for nametag module*/
        "@min": 1,
        "@max": 16
      },
      "callNumber": {
/*optional, string type, call number, which is valid only for nametag module; if this node is not supported, it will not be returned*/
        "@min": ,
        "@max":
      }
    }
  }
}
```

A.3.4 JSON_Cap_CallerInfo

Message about the capability of getting the caller information in JSON format.

```
{
  "CallerInfo":{
    "buildingNo":{
/*optional, int, building No.*/
      "@min":0,
      "@max":16
    },
    "floorNo":{
/*optional, int, floor No.*/
      "@min":0,
      "@max":16
    },
    "zoneNo":{
/*optional, int, project No.*/
      "@min":0,
      "@max":16
    },
    "unitNo":{
/*optional, int, unit No.*/
      "@min":0,
      "@max":16
    },
    "devNo":{
/*optional, int, device serial No.*/
      "@min":0,
      "@max":16
    },
    "devType":{
/*required, int, device type: 1-door station, 2-master station, 3-indoor station, 4-outer door station, 5-villa door station, 6-doorphone, 7-Infosight Client Software, 8-iVMS-4200 Client Software, 9-APP, 10-doorbell, 11-VOIP Client Software, 12-network camera, 13-access control device*/
      "@opt":"1,2,3,4,5,6,7,8,9,10,11,12,13"
    },
    "lockNum":{
/*optional, int, number of locks*/
      "@min":0,
      "@max":16
    },
    "voipNo":{
/*optional, string, VOIP Client Software number, this field is valid when devType is 11*/
      "@min":0,
      "@max":16
    },
    "status":{
/*required, string, status: "idle", "ring", "ring"-the call is in progress*/
      "@opt":"idle,ring,onCall"
    }
```

```
  }
}
```

A.3.5 JSON_Cap_SubModules

JSON message about the capability of getting the sub module status

```
{
  "SubModules": {
    "maxSize": 20,
/*maximum number of sub modules*/
    "id": {
/*required, integer type, sub module ID (or DIP switch address)*/
      "@min": 1,
      "@max": 10000
    },
    "moduleType": {
/*required, string type, "DS-KD-KK": nametag module, "DS-KD-KP": keypad module, "DS-KD-M": card swiping module (13.56MHz), "DS-KD-E": card swiping module (125KHz), "DS-KD-SG": indicator module, "DS-KD-DIS": display module, "DS-1101M, DS-1102M, DS-1103M, DS-1104M, DS-1107M, DS-1108M, DS-1101MK, DS-1102MK, DS-1103MK, DS-1104MK, DS-1107MK, DS-1108MK": card reader*/
      "@opt": "DS-KD-KK,DS-KD-KP, DS-KD-M,DS-KD-E,DS-KD-SG, DS-KD-DIS, DS-1101M,DS-1102M,DS-1103M,DS-1104M,DS-1107M,DS-1108M,DS-1101MK,DS-1102MK,DS-1103MK,DS-1104MK,DS-1107MK,DS-1108MK"
    },
    "status": {
/required, string type, status*/
      "@opt": "online, offline, fault"
    },
    "numOfButtons": {
/*optional, integer type, number of buttons, this node is valid for nametag module only*/
      "@min": 1,
      "@max": 16
    },
    "version": {
/*optional, string type, software version*/
      "@min": 1,
      "@max": 16
    }
  }
}
```

A.3.6 JSON_Cap_SubModulesCfg

JSON message about the configuration capability of the sub module

```
{
  "SubModulesCfg": {
    "Backlight": {
```

```
    "brightness": {
/*optional, integer, backlight brightness whose value is between 0 and 100*/
      "@min": 0,
      "@max": 100
    }
   },
   "enableKeypadTone":"true,false",
/*optional, boolean, whether to enable tough sound*/
   "infoMode": "on,off,auto"
/*optional, string, indicator mode: "on"-remain on, "off"-remain off, "auto"*/
  }
}
```

A.3.7 JSON_EventNotificationAlert_Alarm/EventInfo

EventNotificationAlert message with alarm or event information in JSON format.

```
{
  "ipAddress": "",
/*required, device IPv4 address , string, the maximum size is 32 bytes*/
  "ipv6Address": "",
/*optional, device IPv6 address, string, the maximum size is 128 bytes*/
  "portNo": ,
/*optional, device port No., integer32*/
  "protocol": "",
/*optional, protocol type, "HTTP, HTTPS", string, the maximum size is 32 bytes*/
  "macAddress": "",
/*optional, MAC address, string, the maximum size is 32 bytes, e.g., 01:17:24:45:D9:F4*/
  "channelID": "",
/*optional, device channel No., integer32*/
  "dateTime": "",
/*optional, string, alarm/event triggered or occurred time based on ISO8601, the maximum size is 32 bytes, e.g.,
2009-11-14T15:27Z*/
  "activePostCount": "",
/*required, alarm/event frequency, integer32*/
  "eventType": "",
/*required, alarm/event type, "captureResult, faceCapture,...", string, the maximum size is 128 bytes*/
  "eventState": "",
/*required, string, the maximum size is 32 bytes, durative alarm/event status: "active"-valid, "inactive"-invalid*/
  "eventDescription": "",
/*required, event description, string, the maximum size is 128 bytes*/
  "deviceID":"",
/*string type, device ID*/
  "uuid":"",
/*string type, event UUID, which is used to uniquely identify an event, the standard UUID format is xxxxxxxx-xxxx-xxxx-
xxxx-xxxxxxxxxxxx*/
  ...
/*optional, for different alarm/event types, the nodes are different, see the message examples in different
applications*/
}
```

A.3.8 JSON_EventNotificationAlert_VideoIntercomEventMsg

The information about the video intercom event is uploaded in JSON format of EventNotificationAlert message.

```
{
  "ipAddress":172.6.64.7,
/*required, string, IPv4 address of the alarm device, the maximum size is 32 bytes*/
  "ipv6Address":"",
/*optional, string, IPv6 address of the alarm device, the maximum size is 128 bytes*/
  "portNo":80,
/*optional, integer32, port No. of the alarm device*/
  "protocol":"HTTP",
/*optional, string, protocol type: "HTTP", "HTTPS", the maximum size is 32 bytes*/
  "macAddress":"01:17:24:45:D9:F4",
/*optional, string, MAC address, the maximum size is 32 bytes*/
  "channelID":"1",
/*optional, integer32, device channel No. that triggered alarm*/
  "dateTime":"2016-12-12T17:30:08+08:00",
/*required, string, time when the alarm is triggered (UTC time, e.g., "2017-04-22T15:39:01+08:00"), the maximum size is 32 bytes*/
  "activePostCount":1,
/*required, integer32, times that the same alarm has been uploaded*/
  "eventType":"voiceTalkEvent",
/*required, string, triggered event type, here it should be set to "voiceTalkEvent", the maximum size is 128 bytes*/
  "eventState":"active",
/*required, string, event triggering status: "active"-triggered, "inactive"-not triggered, the maximum size is 32 bytes. For continuous events, this field is used to check whether uploading the continuous events is ended. For example, when the motion detection event occurs, it will be uploaded continuously until an message with eventType "inactive" is uploaded ("invalid" indicates an invalid event)*/
  "eventDescription":"Voice Talk Interactive Event",
/*required, event description*/
  "deviceID":"",
/*optional, string ,device ID, this field is required for devices using ISUP*/
  "VoiceTalkEvent":{
    "deviceName":"",
/*optional, string, device name*/
    "deviceId": ,
/*optional, string, device No. which consists of 0 to 32 characters*/
    "cmdType":"",
/*required, string, operation type: "request"-request for calling, "cancel"-cancel this calling, "answer"-answer this calling, "reject"-reject this calling, "bellTimeout"-the bell ringing of the callee timed out, "hangUp"-end this calling, "deviceOnCall"-the calling is in progress*/
    "src":{
/*optional*/
      "periodNumber": ,
/*optional, int, project No.*/
      "buildingNumber": ,
/*optional, int, building No.*/
      "unitNumber": ,
/*optional, int, unit No.*/
```

```
    "floorNumber": ,
/*optional, int, floor No.*/
    "roomNumber": ,
/*optional, int, room No.*/
    "devIndex": ,
/*optional, int, device serial No.*/
    "communityNumber":"",
/*optional, string, community No.*/
    "unitType":""
/*optional, string, unit type: "indoor"-indoor station, "villa"-villa door station, "confirm"-doorphone, "outdoor"-door station, "fence"-outer door station, "doorbell"-doorbell, "manage"-master station, "acs"-access control device*/
  },
  "target":{
/*optional*/
    "periodNumber": ,
/*optional, int, project No.*/
    "buildingNumber": ,
/*optional, int, building No.*/
    "unitNumber": ,
/*optional, int, unit No.*/
    "floorNumber": ,
/*optional, int, floor No.*/
    "roomNumber": ,
/*optional, int, room No.*/
    "devIndex": ,
/*optional, int, device serial No.*/
    "communityNumber":"",
/*optional, string, community No.*/
    "unitType":""
/*optional, string, unit type: "indoor"-indoor station, "villa"-villa door station, "confirm"-doorphone, "outdoor"-door station, "fence"-outer door station, "doorbell"-doorbell, "manage"-master station, "acs"-access control device*/
  },
  "serialNo": ,
/*optional, int, event serial No.*/
  "currentEvent": ,
/*optional, boolean, whether it is a real-time event: true-yes (real-time event), false-no (offline event)*/
  "frontSerialNo": ,
/*optional, int, serial No. of the previous event. If this field is not returned, the device will check whether event loss occurs by serialNo; if this field is returned, the device will check whether event loss occurs by this field and serialNo together. This field is mainly used to solve the problem that the serialNo is noncontinuous after subscribing alarms*/
  "pictureURL":"",
/*optional, string, picture URL*/
  "picturesNumber":
/*optional, int, number of pictures. If there is no picture or the picture is uploaded by picture URL, this field is 0 or not returned*/
 }
}
```

A.3.9 JSON_ResponseStatus

JSON message about response status

```
{
  "requestURL":"",
/*optional, string, request URL*/
  "statusCode": ,
/*required, int, status code*/
  "statusString":"",
/*required, string, status description*/
  "subStatusCode":"",
/*required, string, sub status code*/
  "errorCode": ,
/*optional, int, error code, which corresponds to subStatusCode, this field is required when statusCode is not 1. The returned value is the transformed decimal number*/
  "errorMsg":"",
/*optional, string, error details, this field is required when statusCode is not 1*/
}
```

Note

See ***Response Codes of Text Protocol*** for details about the status codes, sub status codes, error codes, and error descriptions.

A.3.10 JSON_SubModules

JSON message about the sub module status

```
{
  "SubModules": [{
    "id": ,
/*required, integer type, sub module ID (or DIP switch address)*/
    "moduleType": "",
/*required, string type, "DS-KD-KK": nametag module, "DS-KD-KP": keypad module, "DS-KD-M": card swiping module (13.56MHz), "DS-KD-E": card swiping module (125KHz), "DS-KD-SG": indicator module, "DS-KD-DIS": display module, "DS-1101M, DS-1102M, DS-1103M, DS-1104M, DS-1107M, DS-1108M, DS-1101MK, DS-1102MK, DS-1103MK, DS-1104MK, DS-1107MK, DS-1108MK": card reader*/
    "status": "",
/*required, string type, status: "online", "offline", "fault"*/
    "numOfButtons": ,
/*optional, integer type, number of buttons, this node is valid for nametag module only*/
    "version":""
/*optional, string type, software version*/
  }]
}
```

A.3.11 JSON_SubModulesCfg

JSON message about sub module parameters

```
{
  "SubModulesCfg": {
    "Backlight": {
      "brightness":
/*optional, integer type, backlight brightness whose value is between 0 and 100, this node is valid only for display module*/
    },
    "enableKeypadTone": ,
/*optional, boolean type, whether to enable touch sound: true-yes, false-no, this node is valid only for display module*/
    "infoMode": ""
/*optional, string type, indicator mode: "on"-remain on, "off"-remain off, "auto"*/
  }
}
```

A.3.12 XML_Cap_IntercomProtocolType

Message about the configuration capability for switching between the private protocol and the standard SIP in XML format.

```
<IntercomProtocolType version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <protocolType opt="Private, StandardSIP"><!--required, xs:string, protocol type: "Private"-private video intercom protocol, "StandardSIP"-standard SIP protocol--></protocolType>
</IntercomProtocolType>
```

A.3.13 XML_Cap_KeyCfg

Message about the configuration capability of pressing the button to call in XML format.

```
<KeyCfg version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id min="" max=""><!--required, xs:integer, button No. which corresponds to the <ID> in the request URI /ISAPI/VideoIntercom/keyCfg/<ID>/capabilities--></id>
  <module opt="main,sub"><!--optional, xs:string, module to be configured: "main"-main module (default), "sub"-sub module--></module>
  <moduleId min="" max=""><!--dependent, xs:integer, sub module ID, this node is valid when <module> is "sub" and is used to specify that the button information of which module will be configured--></moduleId>
  <callNumber min="" max=""><!--optional, xs:string, called number--></callNumber>
  <enableCallCenter opt="true,false"><!--optional, xs:boolean, whether to call the management center--></enableCallCenter>
  <callMethod opt=" callNumber,manageCenter,app"><!--optional, xs:string, calling method: "callNumber"-call by number, "manageCenter"-call the management center, "app"-call APP. The empty string is invalid, and if this node is configured, the node <enableCallCenter> is not required--></callMethod>
</KeyCfg>
```

A.3.14 XML_Cap_PasswordAuthenticationCfg

XML message about the encryption verification capability of video intercom

```
<PasswordAuthenticationCfg version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
 <password min="" max="">
  <!--required, xs:string, password-->
 </password>
</PasswordAuthenticationCfg>
```

A.3.15 XML_Cap_RingParam

XML message about the ring configuration capability

```
<RingParam version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
 <ringID min="" max=""><!--required, xs:integer, ring ID--></ringID>
 <ringName min="" max=""><!--required, xs:string, ring name--></ringName>
 <ringSize min="0" max="100"><!--required, xs:integer, ring file size--></ringSize>
 <ringType opt="wav"><!--required, xs:string, ring type--></ringType>
</RingParam>
```

A.3.16 XML_Cap_SceneNowMode

XML message about the scene configuration capability of video intercom

```
<SceneNowMode version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
 <nowMode opt="atHome,goOut,goToBed,custom">
  <!--required, xs:string, current scene mode-->
 </nowMode>
</SceneNowMode>
```

A.3.17 XML_EventNotificationAlert_AlarmEventInfo

EventNotificationAlert message with alarm/event information in XML format.

```
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
 <ipAddress><!--dep, xs:string, device IPv4 address--></ipAddress>
 <ipv6Address><!--dep, xs:string, device IPv6 address--></ipv6Address>
 <portNo><!--opt, xs:integer, device port number--></portNo>
 <protocol><!--opt, xs:string, protocol type for uploading alarm/event information, "HTTP,HTTPS"--></protocol>
 <macAddress><!--opt, xs:string, MAC address--></macAddress>
 <channelID><!--dep, xs:string, device channel No., starts from 1--></channelID>
 <dateTime><!--req, alarm/event triggered or occurred time, format: 2017-07-19T10:06:41+08:00--></dateTime>
 <activePostCount><!--req, xs:integer, alarm/event frequency, starts from 1--></activePostCount>
 <eventType><!--req, xs:string, alarm/event type, "peopleCounting, ANPR,..."--></eventType>
 <eventState>
```

```
  <!--req, xs:string, durative alarm/event status: "active"-valid, "inactive"-invalid, e.g., when a moving target is detected,
    the alarm/event information will be uploaded continuously unit the status is set to "inactive"-->
 </eventState>
 <eventDescription><!--req, xs:string, alarm/event description--></eventDescription>
 <...><!--opt, for different alarm/event types, the nodes are different, see the message examples in different applications--></...>
</EventNotificationAlert>
```

A.3.18 XML_IntercomProtocolType

Message about the parameters for switching between the private video intercom protocol and the standard SIP in XML format.

```
<IntercomProtocolType version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
 <protocolType><!--required, xs:string, protocol type: "Private"-private video intercom protocol, "StandardSIP"-standard SIP protocol--></protocolType>
</IntercomProtocolType>
```

A.3.19 XML_IpViewDevAbility

XML message about the video intercom device capability

```
<IpViewDevAbility version="2.0">
 <!--video intercom device capability-->
 <SipServerLogin><!--required, registration capability of SIP server-->
  <AutoLogin><!--required, whether to support automatic registration-->
   <enable></enable>
  </AutoLogin>
  <loginStatus opt="registered,unregistered"/><!--required, registration status-->
  <sipLoginNameLen min="1" max="32"/><!--required, length of registered user name-->
  <sipLoginPasswordLen min="1" max="16"/><!--required, length of registered password-->
  <displayNameLen min="1" max="128"/><!--required, length of displayed device name-->
  <localNumber min="1" max="32"/><!--required, length of local station number-->
  <loginCycle min="1" max="99"/><!--required, registration period, unit: minute-->
  <serverSipPort  min="1024" max="65535"/>
  <localPort min="1024" max="65535"><!--optional, local port No.-->
  <isNotSupportLocalPort></isNotSupportLocalPort><!--optional, whether the local port is not supported: true-yes (the local port is not supported), this field is not supported-no (the local port is supported)-->
  <domainNameLen min="", max=""/><!--optional, the meaning of this field is the same as that of the field <sipServerDomain>. For new devices, both fields need to be returned-->
  <addressType opt="IP/IPV6, domain"/><!--supported address type-->
  <mutexAbility opt="gbt28181"/><!--required, mutex ability, gbt28181-->
  <notSupportCharacter opt=""><!--whether the domain name contains unsupported characters: true-yes (the domain name contains unsupported characters)-->
  <isNotSupportSipServerIP></isNotSupportSipServerIP><!--whether the IP address of the SIP server is not supported: true-yes (the IP address of the SIP server is not supported), this field is not supported-no (the IP address of the SIP server is supported). For old devices, this field will not be returned; this field is used to check whether new devices support the IP address of the SIP server-->
```

```
  <sipServerDomain min="" max=""/><!--optional, domain name of the SIP server. The meaning of this field is the same as that of the field <domainNameLen>. For new devices, both fields need to be returned-->
  <stunServerIP min="" max=""/><!--optional, IP address of the STUN server-->
  <stunServerDomain min="" max=""/><!--optional, domain name of the STUN server-->
  <stunServerPort min="" max=""/><!--optional, port No. of the STUN server-->
  <proxyServerIP min="" max=""/><!--optional, IP address of the proxy server-->
  <proxyServerDomain min="" max=""/><!--optional, domain name of the proxy server-->
  <proxyServerPort min="" max=""/><!--optional, port No. of the proxy server-->
  <netWork opt="0,1,2,3"/><!--optional, network type: 0-invalid, 1-wired network 1, 2-wired network 2, 3-wireless network-->
  <CalledTargetName min="" max=""/><!--optional, user name length of the called person-->
 </SipServerLogin>
 <LocalAbility><!--required, basic capability of video intercom extension-->
  <defaultRing min="1" max="6"/><!--required, options of the default local ringtone-->
  <ringVolume min="1" max="9"/><!--required, range of the local ringtone volume-->
  <inputVolume min="0" max="6"/><!--required, input volume options-->
  <outputVolume min="0" max="9"/><!--required, output volume options-->
  <audioEncPriNum><!--required, supported number of audio encoding levels--></audioEncPriNum>
  <delayPreview min="0" max="30"/><!--required, range of live view delay, unit: second-->
  <AudioEncEntry><!--required, supported audio encoding type, multiple types can be set for the same node-->
   <index></index>
   <name></name>
   <packetLen opt="160,320"/>
  </AudioEncEntry>
  <CallAbility><!--required, calling capability-->
   <AutoResponse><!--required, support automatic response-->
    <autoResponse min="0" max="30"/><!--required, range of automatic response duration, unit: second-->
   </AutoResponse>
   <callNumber><!--required, supported number of calling numbers--></callNumber>
   <callNumberLen min="0" max="32"/>
  </CallAbility>
 </LocalAbIlity>
 <VideoIntercom>
  <!--video intercom, indoor station, door station, master station, door station (V series), doorphone-->
  <monitorChannelNo min="1" max="100"/><!--required, camera channel of master station or indoor station-->
  <DeviceID><!--device No.-->
   <enabled>true</enabled><!--supported device No. configuration-->
   <supportUnitType opt=""/><!--required, supported video intercom type-->
   <InDoorDevice><!--required, indoor station-->
    <floorNum min="1" max="16"/><!--required, floor No.-->
    <roomNum min="1" max="16"/><!--required, room No.-->
    <devIndex min="0" max="10"/><!--required, indoor station No.-->
   </InDoorDevice>
   <OutDoorDevice><!--required, door station/intelligent access control device-->
    <period min="1" max="16"/><!--required, community No.-->
    <buildingNum min="1" max="16"/><!--required, building No.-->
    <unitNum min="0" max="10"/><!--required, unit No.-->
    <floorNum min="1" max="16"/><!--required, floor No.-->
    <devIndex min="0" max="10"/><!--required, door station No.-->
   </OutDoorDevice>
   <ManageUnitDevice><!--required, master station-->
    <period min="1" max="16"/><!--required, community No.-->
```

```
      <devIndex min="0" max="10"/><!--required, master station No.-->
     </ManageUnitDevice>
     <OutDoorFenceDevice><!--required, outer door station-->
      <period min="1" max="16"/><!--required, community No.-->
      <devIndex min="0" max="10"/><!--required, outer door station No.-->
     </OutDoorFenceDevice>
     <VillaOutDoorDevice><!--required, door station (V series)-->
      <period min="1" max="16"/><!--required, community No.-->
      <buildingNum min="1" max="16"/><!--required, building No.-->
      <unitNum min="0" max="10"/><!--required, unit No.-->
      <floorNum min="1" max="16"/><!--required, floor No.-->
      <devIndex min="0" max="10"/><!--required, door station No.-->
     </VillaOutDoorDevice>
     <AgainDevice><!--required, doorphone--></AgainDevice>
    </DeviceID>
    <PrivilegePwd><!--permission password-->
     <pwdType opt="engineering,setupAlarm,householderUnlock,antiHijacking"/><!--password type: "engineering"-
project password, "setupAlarm"-arming and disarming password, "householderUnlock"-resident unlocking password,
"antiHijacking"-duress password-->
     <pwdLen min="6" max="16"/><!--password length-->
    </PrivilegePwd>
    <OperationTime><!--operation time-->
     <monitoringTime min="10" max="60"/><!--maximum live view duration, unit: second-->
     <ringTime min="15" max="60"/><!--maximum ringing duration, unit: second-->
     <messageTime min="30" max="60"/><!--maximum messaging duration, unit: second-->
     <talkTime min="90" max="120"/><!--maximum calling duration, unit: second-->
     <callForwardingTime min="0" max="20"/><!--calling forwarding timeout, unit: second-->
     <dwRingDurationTime min="30" max="60"/><!--ringing time duration, unit: second-->
    </OperationTime>
    <RelateDevice>
     <outdoorUnitIP opt="ipv4,ipv6"/><!--IP address of main door station-->
     <manageUnitIP opt="ipv4,ipv6"/><!--IP address of master station-->
     <sipServerIP opt="ipv4,ipv6"/><!--IP address of SIP server-->
     <centerIP opt="ipv4,ipv6"/><!--center IP address-->
     <centerPort min="" max=""/><!--center port No.-->
     <indoorUnitIP opt="ipv4,ipv6"/><!--IP address of indoor station-->
     <notSupportAgainUnitIP opt="true,false"/><!--required, whether configuring doorphone IP address is not
supported: "true"-yes, "false"-no-->
     <againUnitIP opt="ipv4,ipv6"/><!--doorphone IP address-->
     <notSupportOutDoorType opt="true,false"/><!--required, whether configuring main door station type is not
supported: "true"-yes, "false"-no-->
     <outDoorType opt="unitOutdoor,villaOutDoor"/><!--main door station type: "unitOutdoor"-main door station (D
series), "villaOutDoor"-main door station (V series)-->
     <outInConnectMode opt="sameLan,diffLan"/><!--networ mode of door station and sub indoor station: "sameLan"-
in the same LAN, "diffLan"-in different LANs-->
     <indoorConnectMode opt="wireless,wired"/><!--network mode of main indoor station and sub indoor station:
"wireless"-wireless NIC, "wired"-wired NIC-->
    </RelateDevice>
    <NoticeData>
     <noticeTime><!--notice time--></noticeTime>
     <noticeNumberLen min="0" max="32"/><!--notice No.-->
     <noticeThemeLen min="0" max="64"/><!--notice theme-->
```

```
    <noticeDetailLen min="0" max="1024"/><!--notice details-->
    <noticeLevel opt="advertisement,propertyMgmt,alarm,notification"/><!--notice level: "advertisement"-
advertisement information, "propertyMgmt"-property information, "alarm"-alarm information, "notification"-notice
information-->
    <maxPicNum><!--number of pictures--></maxPicNum>
    <maxSinglePicSize><!--maximum size of a picture, unit: KB--></maxSinglePicSize>
   </NoticeData>
   <ControlGateway><!--unlock remotely-->
    <gatewayIndex min="1"/><!--access control No.-->
    <command opt="close,open"/><!--control command-->
    <controlSrc>true</controlSrc><!--operation source information-->
    <controlType opt="monitor,calling"/><!--unlocking scene type: "monitor", "calling"-->
    <lockType opt="normal,smartLock"/><!--lock type-->
    <lockID min="" max=""/><!--lock ID-->
    <password min="" max=""/><!--password length of the smart lock-->
   </ControlGateway>
   <Zone><!--zone configuration-->
    <ZoneConfig>
     <enabled><!--whether to support configuring zone parameters (alarm input parameters)--></enabled>
     <delayInParam><!--method of setting delay: "true"-the client sets the delay time by dwParam in the structure
NET_DVR_ALARMIN_PARAM (this method is used by power and environment monitoring system and ATM security
control panel), "false"-the client sets the delay time by wEnterDelay and wExitDelay in the structure
NET_DVR_ALARMSUBSYSTEMPARAM--></delayInParam>
     <detectorType
opt="panicButton,magneticContact,smokeDetector,activeInfraredDetector,passiveInfraredDetector,glassBreakDetector,
vibrationDetector,dualTechnologyPirDetector,tripleTechnologyPirDetector,humidityDetector,temperatureDetector,com
bustibleGasDetecto,dynamicSwitch,controlSwitch,smartLock,waterDetector,otherDetector"/><!--detector type
supported by the device: "panicButton"-panic switch, "magneticContact"-magnetic contact, "smokeDetector"-smoke
detector, "activeInfraredDetector"-active infrared detector, "passiveInfraredDetector"-passive infrared detector,
"glassBreakDetector"-glass break detector, "vibrationDetector"-vibration detector, "dualTechnologyPirDetector"-dual
technology motion detector, "tripleTechnologyPirDetector"-triple technology detector, "humidityDetector"-humidity
detector, "temperatureDetector"-temperature detector, "combustibleGasDetecto"-gas detector, "dynamicSwitch"-
follow-up switch, "controlSwitch"-control switch, "smartLock"-smart lock, "waterDetector"-water detector,
"otherDetector"-other detector type-->
     <zoneType opt="instantZone,
24hourAudibleZone,delayZone,interiorWithDelayZone,keyswitchZone,supervisedFireZone,perimeterZone,
24hourSlientZone,disable"/><!--zone type supported by the device-->
     <LimitedDetectorType>
      <Detector>
       <name>smokeDetector</name>
       <zoneType opt="24hourAudibleZone"/>
      </Detector>
      <Detector>
       <name>glassBreakDetector</name>
       <zoneType opt="24hourAudibleZone"/>
      </Detector>
     </LimitedDetectorType>
     <uploadAlarmRecoveryReport>true</uploadAlarmRecoveryReport><!--whether to support report configuration
of uploading alarm recovery-->
     <zoneDelayTime min="" max=""/><!--delayed zone delay-->
     <sensitivity opt="10ms,250ms,500ms,750ms"/><!--sensitivity-->
     <arrayBypass>true</arrayBypass><!--whether to support zone group bypass configuration-->
```

```
      <moduleStatus attri="readonly" opt="offline,online"/><!--module status-->
      <moduleAddress min="" max=""/><!--module address-->
      <moduleChannel>true</moduleChannel><!--module channel-->
      <moduleType opt="localZone, 1zoneExpander,2zoneExpander,8ZoneExpander,8sensorZoneExpander,
1Zone&Trigger"/><!--supported zone type-->
      <zoneNo attri="readonly" min="" max=""/><!--zone No. which can only be obtained-->
      <subsystemNo attri="readonly"><!--No. of the partition that the zone belongs to, it can only be obtained--></
subsystemNo>
      <alarmType opt="open,close" default="open"/><!--required, alarm device type: "open"-remain open, "close"-
remain closed-->
      <InDelayTime min="0" max="255"/><!--required, entrance delay, unit: second-->
      <OutDelayTime min="0" max="255"/><!--required, exiting delay, unit: second-->
    </ZoneConfig>
    <GetZoneList>
      <enabled><!--whether to support getting zone list--></enabled>
    </GetZoneList>
    <ZoneArmDisarm>
      <enabled><!--whether to support arming and disarming the zone--></enabled>
    </ZoneArmDisarm>
    <ZoneGroupBypass>
      <enabled><!--whethe to support zone group bypass--></enabled>
    </ZoneGroupBypass>
  </Zone>
  <IOIn>
    <IOInNo attri="readonly" min="" max=""/><!--IP input No. which can only be obtained-->
    <useType opt="disabled,openDoorBtn,doorStatus,custom"/><!--purpose: "disabled", "openDoorBtn"-door exit
button, "doorStatus"-door status, "custom"-->
  </IOIN>
  <IOOut>
    <IOOutNo attri="readonly" min="" max=""/><!--IO output No. which can only be obtained-->
    <useType opt="disabled,electricLock,custom"/><!--purpose: "disabled", "electricLock"-electric lock, "custom"-->
  </IOOut>
  <ElevatorControl>
    <elevatorNo attri="readonly" min="" max=""/> <!--ro, elevator No.-->
    <interfaceType opt="RS485,network"/><!--interface type: "RS485"-RS-485, "network"-->
    <RS485Protocol opt="private,custom"/><!--RS-485 protocol type: "private", "custom"-->
    <networkProtocol opt="private,custom"/><!--NIC protocol type: "private", "custom"-->
    <serverIP opt="ipv4,ipv6"/><!--optional, IP address of the elevator control server-->
    <serverPort min="" max=""/><!--optional, port No. of the elevator control server-->
  </ElevatorControl>
  <RS485Config>
    <!--RS-485 configuration, this node will not be returned if RS-485 configuration is not supported-->
    <deviceNameLength min="0" max="32"/><!--RS-485 name-->
    <deviceType>true</DeviceType><!--whether to support configuring device type-->
    <deviceProtocol>true</deviceProtocol><!--whether to support configuring device protocol-->
  </RS485Config>
  <supportDevInfo opt="true,false"/><!--required, whether to support getting complete indoor station No. This node
will not be returned if this function is not supported, and this node is valid only when the device is an indoor station-->
  <supportRegisterInfo opt="true,false"/><!--required, whether the door station supports getting registration
information. This node will not be returned if this function is not supported, and this node is valid only when the
device is an door station-->
  <CallRoomConfig><!--configuration of calling resident by pressing button for door station (V series)-->
```

```
    <keyNo min="" max=""/><!--villa button No.-->
    <floorNo min="" max=""/><!--villa floor No.-->
    <roomNo min="" max=""/><!--villa room No.-->
    <callManageCenter opt="true,false"/><!--required, whether to set it to the calling management center-->
    <calledName min="" max=""/><!--optional, user name to be called, it supports letters, digits, @, and dot. This
node is valid in standard SIP mode-->
  </CallRoomConfig>
  <VideoCall><!--optional, video intercom capability-->
    <enabled opt="true,false"/><!--required, whether supports starting video intercom-->
    <supportCmd opt="callRequest,Cancel,Answer,Decline,Timeout,Bye,deviceCalling,clientCalling"/>
    <!--required, command type supported by the device: "callRequest"-request for call, "Cancel"-cancel the call,
"Answer"-answer, "Decline"-decline, "Timeout"-time out, "Bye"-end the call, "deviceCalling"-device is in call,
"clientCalling"-client is in call-->
  </VideoCall>
  <CallerDevice><!--required, calling device information-->
    <period min="1" max="16"/><!--required, community No.-->
    <buildingNum min="1" max="16"/><!--required, building No.-->
    <unitNum min="0" max="10"/><!--required, unit No.-->
    <floorNum min="1" max="16"/><!--required, floor No.-->
    <devIndex min="0" max="10"/><!--required, device No.-->
    <devType min="1" max="9"/><!--required, device type-->
  </CallerDevice>
  <CallStatus><!--required, calling status-->
    <callingStatus min="1" max="3"/><!--required, calling status-->
  </CallStatus>
  <EzvizDeviceInfo><!--required, EZVIZ device information-->
    <deviceNum min="1" max="32"/><!--required, number of devices-->
    <DeviceCfg size="16"/>
    <deviceNameLen min="1" max="32"/><!--required, device name-->
    <deviceType min="1" max="3"/><!--required, device type-->
    <deviceID min="0" max="10"/><!--required, device No.-->
  </EzvizDeviceInfo>
  <VideoCallParam><!--required, signal interaction command (non-persistent connection)-->
    <cmdType min="0" max="4"/><!--required, command-->
  </VideoCallParam>
  <VideoIntercomStream><!--optional, video source-->
    <sourceType opt="IPC,DVR/DVS/NVR,OutDoorDevice,OutDoorFenceDevice,AgainDevice"/><!--optional, video
source type-->
    <againDeviceNumber min="" max=""/><!--optional, number of doorphones-->
  </VideoIntercomStream>
  <indoorDevChangeEnabled opt="true,false"/><!--required, whether to support switching between main indoor
station and sub indoor station-->
  <indoorDevChangeReboot opt="true,false"/><!--required, whether the device will reboot after switching devices of
main indoor station and sub indoor station-->
 </VideoIntercom>
 <UploadAlarmCfg><!--configuration of video intercom alarm, it corresponds to the structure
NET_DVR_VIDEO_INTERCOM_ALARM_CFG-->
  <UploadDoorNotCloseAlarm opt="true,false"/><!--required, whether to upload alarms of unlocking the door-->
 </UploadAlarmCfg>
 <ZoneList size="64"><!--special zone configuration, which is the capability of distinguishing different zones according
to the zone ID-->
  <ZoneConfig>
```

```
    <zoneID min="" max=""><!--zone ID--></zoneID>
    <enabled><!--whether to support configuring zone parameters (alarm input parameters)--></enabled>
    <delayInParam><!--method of setting delay: "true"-the client sets the delay time by dwParam in the structure
NET_DVR_ALARMIN_PARAM (this method is used by power and environment monitoring system and ATM security
control panel), "false"-the client sets the delay time by wEnterDelay and wExitDelay in the structure
NET_DVR_ALARMSUBSYSTEMPARAM--></delayInParam>
    <detectorType
opt="panicButton,magneticContact,smokeDetector,activeInfraredDetector,passiveInfraredDetector,glassBreakDetector,
vibrationDetector,dualTechnologyPirDetector,tripleTechnologyPirDetector,humidityDetector,temperatureDetector,com
bustibleGasDetecto,dynamicSwitch,controlSwitch,smartLock,waterDetector,otherDetector"/><!--detector type
supported by the device: "panicButton"-panic switch, "magneticContact"-magnetic contact, "smokeDetector"-smoke
detector, "activeInfraredDetector"-active infrared detector, "passiveInfraredDetector"-passive infrared detector,
"glassBreakDetector"-glass break detector, "vibrationDetector"-vibration detector, "dualTechnologyPirDetector"-dual
technology motion detector, "tripleTechnologyPirDetector"-triple technology detector, "humidityDetector"-humidity
detector, "temperatureDetector"-temperature detector, "combustibleGasDetecto"-gas detector, "dynamicSwitch"-
follow-up switch, "controlSwitch"-control switch, "smartLock"-smart lock, "waterDetector"-water detector,
"otherDetector"-other detector type-->
    <zoneType opt="instantZone,
24hourAudibleZone,delayZone,interiorWithDelayZone,keyswitchZone,supervisedFireZone,perimeterZone,
24hourSlientZone,disable"/><!--zone type supported by the device-->
    <uploadAlarmRecoveryReport><!--whether to support report configuration of uploading alarm recovery--></
uploadAlarmRecoveryReport>
    <zoneDelayTime min="" max=""/><!--delayed zone delay-->
    <sensitivity opt="10ms,250ms,500ms,750ms"/><!--sensitivity-->
    <arrayBypass><!--whether to support zone group bypass configuration--></arrayBypass>
    <moduleStatus attri="readonly" opt="offline,online"/><!--module status-->
    <moduleAddress min="" max=""/><!--module aqddress-->
    <moduleChannel><!--module channel--></moduleChannel>
    <moduleType opt="localZone, 1zoneExpander,2zoneExpander,8ZoneExpander,8sensorZoneExpander,
1ZoneAndTrigger"/><!--supported zone type-->
    <zoneNo attri="readonly" min="" max=""/><!--zone No. which can only be obtained-->
    <subsystemNo attri="readonly"><!--No. of the partition that the zone belongs to, it can only be obtained--></
subsystemNo>
    <alarmType opt="open,close" default="open"/><!--required, alarm device type: "open"-remain open, "close"-
remain closed-->
    <InDelayTime min="0" max="255"/><!--required, entrance delay, unit: second-->
    <OutDelayTime min="0" max="255"/><!--required, exiting delay, unit: second-->
   </ZoneConfig>
   <GetZoneList>
    <enabled><!--whether the device supports getting zone list--></enabled>
   </GetZoneList>
   <ZoneArmDisarm>
    <enabled><!--whether the device supports arming and disarming the zone--></enabled>
   </ZoneArmDisarm>
   <ZoneGroupBypass>
    <enabled><!--whether the device supports zone group bypass--></enabled>
   </ZoneGroupBypass>
  </ZoneList>
  <NoNeedReboot>
   <videoResolutionChange opt="true,false"/><!--optional, whether the device does not reboot after changing video
resolution-->
   <videoFrameRateChange opt="true,false"/><!--optional, whether the device does not reboot after changing video
```

```
frame rate-->
  </NoNeedReboot>
</IpViewDevAbility >
```

A.3.20 XML_KeyCfg

Message about the parameters of pressing the button to call of a specific button in XML format.

```
<KeyCfg version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:integer, button No. which corresponds to the <ID> in the request URI /ISAPI/VideoIntercom/
keyCfg/<ID>--></id>
  <module><!--optional, xs:string, module to be configured: "main"-main module (default), "sub"-sub module-->    </
module>
  <moduleId><!--dependent, xs:integer, sub module ID, this node is valid when <module> is "sub" and is used to
specify that the button information of which module will be configured--></moduleId>
  <callNumber><!--dependent, xs:string, called number, this field is valid when <callMethod> is "callNumber"--></
callNumber>
  <enableCallCenter><!--optional, xs:boolean, whether to call the management center--></enableCallCenter>
  <callMethod><!--optional, xs:string, calling method: "callNumber"-call by number, "manageCenter"-call the
management center, "app"-call APP. The empty string is invalid, and if this node is configured, the node
<enableCallCenter> is not required--></callMethod>
</KeyCfg>
```

A.3.21 XML_KeyCfgList

Message about the parameters of pressing the button to call of multiple buttons in XML format

```
<KeyCfgList version ="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <KeyCfg/><!--optional, see details in XML_KeyCfg-->
</KeyCfgList>
```

See Also

XML_KeyCfg

A.3.22 XML_PasswordAuthenticationCfg

XML message about the video intercom encryption parameters

```
<PasswordAuthenticationCfg version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <password>
    <!--required, xs: string, password-->
  </password>
</PasswordAuthenticationCfg>
```

A.3.23 XML_ResponseStatus

XML message about response status

```
<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>
</ResponseStatus>
```

Note

See ***Response Codes of Text Protocol*** for details about sub status codes and corresponding error codes.

A.3.24 XML_SceneNowMode

XML message about scene parameters of video intercom

```
<SceneNowMode version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
 <nowMode>
  <!--required, xs: string, current scene mode: "atHome", "goOut", "goToBed", "custom"-->
 </nowMode>
</SceneNowMode>
```

A.3.25 XML_VideoIntercomCap

XML message about the video intercom capability

```
<VideoIntercomCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
 <isSupportDeviceId><!--optional, xs:boolean--></isSupportDeviceId>
 <isSupportDeviceId><!--optional, xs:boolean--></isSupportDeviceId>
 <isSupportOperationTime><!--optional, xs:boolean--></isSupportOperationTime>
 <isSupportCallerInfo><!--optional, xs:boolean--></isSupportCallerInfo>
 <isSupportCallSignal><!--optional, xs:boolean--></isSupportCallSignal>
 <isSupportCallStatus><!--optional, xs:boolean--></isSupportCallStatus>
 <isSupportRingManage><!--optional, xs: boolean, whether it supports ring management--></isSupportRingManage>
```

```
<isSupportPasswordAuthentication>
  <!--required, xs: boolean, whether it supports password verification of video intercom-->
 </isSupportPasswordAuthentication>
 <isSupportRelatedDeviceAdress><!--optional, xs:boolean--></isSupportRelatedDeviceAdress>
 <isSupportCardSectorCheck>
  <!--required, xs: boolean, whether it supports section encryption verification of card-->
 </isSupportCardSectorCheck>
 <isSupportKeyCfg><!--optional, xs:boolean, whether it supports the configuration of pressing the button to call the
resident--></isSupportKeyCfg>
 <isSupportWorkModeCfg><!--optional, xs:boolean--></isSupportWorkModeCfg>
 <isSupportPrivilegePasswordStatus><!--req,xs:boolean, whether it supports getting the status of the device
permission password--></isSupportPrivilegePasswordStatus>
 <isSupportElevatorControlCfg><!--optional, xs:boolean--></isSupportElevatorControlCfg>
 <isSupportPrivilegePassword><!--optional, xs:boolean--></isSupportPrivilegePassword>
 <isSupportAlarmControlByPhone>
  <!--required, xs: boolean, whether it supports arming and control via mobile phone--></
isSupportAlarmControlByPhone>
 <isSupportSceneManage><!--optional, xs: boolean, whether it supports scene management--></
isSupportSceneManage>
 <isSupportZoneCfgByScene>
  <!--optional, xs: boolean, whether it supports configuring zone parameters by scene-->
 </isSupportZoneCfgByScene>
 <isSupportCallElevator><!--required, xs: boolean, whether it supports calling elevator--></isSupportCallElevator>
 <isSupportGetSmartLockParam>
  <!--required, xs: boolean, whether it supports getting smart lock information-->
 </isSupportGetSmartLockParam>
 <isSupportAppKeyConfiguration>
  <!--required, xs: boolean, whether it supports configuring APPKey-->
 </isSupportAppKeyConfiguration>
 <isSupportDeviceLanguageCfg>
  <!--required, xs:boolean, whether it supports switching languages-->
 </isSupportDeviceLanguageCfg>
 <isSupportSubModules><!--optional, xs: boolean, whether it supports sub module management--></
isSupportSubModules>
 <isSupportCallCfg>
  <!--optional, xs: boolean, whether it supports configuring keypad of sub module-->
 </isSupportCallCfg>
 <isSupportSubModulesCfg>
  <!--optional, xs: boolean, whether it supports configuring sub module parameters-->
 </isSupportSubModulesCfg>
 <isSupportIssueCards><!--optional, xs:boolean, whether it supports configuration the function of issuing main card--
></isSupportIssueCards>
 <isSupportSendCardCfg><!--optional, xs:boolean, whether the door station supports configuring card issuing
function--></isSupportSendCardCfg>
 <isSupportRingBackTone><!--optional, xs:boolean, whether it supports importing the ringtone for calling back--></
isSupportRingBackTone>
 <isSupportCallPriority> <!--optional,  xs:boolean--> </isSupportCallPriority>
 <isSupportPhoneCfg><!--optional, xs:boolean, whether it supports configuring the resident's phone--></
isSupportPhoneCfg>
 <isSupportDisplayRegion> <!--optional,  xs:boolean --> </isSupportDisplayRegion>
 <isSupportWatchPoint><!--optional, xs:boolean, related URI: /ISAPI/VideoIntercom/WatchPoint--></
isSupportWatchPoint>
```

```
 <isSupportVoiceConfiguration><!--optional, xs:boolean, whether it supports enabling speech recognition and audio prompt--></isSupportVoiceConfiguration>
 <isSupportUpdateData> <!--optional, xs:boolean, whether it supports importing the upgrading file--> </isSupportUpdateData>
 <isSupportUpdateStrategy> <!--optional, xs:boolean, whether it supports upgrading configuration--> </isSupportUpdateStrategy>
 <isSupportMetaInfo> <!--optional, xs:boolean, whether it supports getting the upgrading file information--></isSupportMetaInfo>
 <platform opt="R0,R1,..."><!--optional, string, platform type. The actually supported platform types will be returned--></platform>
 <isSupportSecurityMode><!--optional, xs:boolean, whether it supports switching the security protocol version--></isSupportSecurityMode>
 <isSupportDeviceCommunication><!--optional, xs:boolean, whether it supports configuring communication parameters of the device network--></isSupportDeviceCommunication>
 <isSupportIncomingCallLinkage><!--optional, xs:boolean, whether it supports configuring the calling linkage--></isSupportIncomingCallLinkage>
</VideoIntercomCap>
```

A.3.26 XML_WorkStatus

XML message about the working status of the door station

```
<WorkStatus version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
 <lockStatusList><!--for the lock of the door station, it is always online, so this node may not be returned; for the lock of the external module, this node is required-->
  <node>
   <lockID>
    <!--required, xs: integer, lock ID whose value may be 0 or 1-->
   </lockID>
   <lockStatus>
    <!--required, xs: string, lock status: "online", "offline"-->
   </lockStatus>
  </node>
 </lockStatusList>
</WorkStatus>
```

A.4 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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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 doest 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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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).

Error Name	Error Code	Error Description
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.dl.l
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.
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 wirting permission.
NET_DVR_IP_CONFLICT	174	IP conflicted.
NET_DVR_INSUFFICIENT_ STORAGEPOOL_SPACE	175	Storage pool space is full.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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_ CONFILICT	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 rull(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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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
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.

Error Name	Error Code	Error Description
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.
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_DISCONNEC	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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
NET_DVR_ERR_DETECTION_LINE	840	Rule line configuration error. Please try again and make sure the line is within the red region.
NET_DVR_EXCEED_FACE_IMAGES_ ERROR	843	No more face pictures can be added.
NET_DVR_ANALYSIS_FACE_IMAGES_ ERROR	844	Picture recognition failed.
NET_ERR_ALARM_INPUT_OCCUPIED	845	A<-1 alarm number is used for triggering vehicle capture.
NET_DVR_FACELIB_DATABASE_ERROR	846	Database version in face picture library mismatched.
NET_DVR_FACELIB_DATA_ERROR	847	Face picture library data error.
NET_DVR_FACE_DATA_ID_ERROR	848	Invalid face data PID.
NET_DVR_FACELIB_ID_ERROR	849	Invalid face picture library ID.
NET_DVR_EXCEED_FACE_LIBARY_ ERROR	850	No more face picture libraries can be established..
NET_DVR_PIC_ANALYSIS_NO_ TARGET_ERROR	851	No target recognized in the picture.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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).

Error Name	Error Code	Error Description
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
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

Error Name	Error Code	Error Description
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)
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.

Error Name	Error Code	Error Description
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
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.

Error Name	Error Code	Error Description
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
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

Error Name	Error Code	Error Description
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
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.

Error Name	Error Code	Error Description
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 ablity.
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.
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.

Error Name	Error Code	Error Description
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.
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	Slave 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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
NET_ERR_VQD_CHAN_MAX	1503	The total number of VQD plans exceeds the max limit.
NET_ERR_VQD_TASK_MAX	1504	The total number of VQD tasks exceeds the max limit.
NET_DVR_ERR_EXCEED_MAX_ CAPTURE_TIMES	1600	Capture times exceed 2 in flash mode.
NET_DVR_ERR_REDAR_TYPE_ CONFLICT	1601	Radar type conflict.
NET_DVR_ERR_LICENSE_PLATE_NULL	1602	The license plate is null.
NET_DVR_ERR_WRITE_DATABASE	1603	Failed to write data into the database.
NET_DVR_ERR_LICENSE_EFFECTIVE_ TIME	1604	The effective time of license plate error.
NET_DVR_ERR_PRERECORDED_ STARTTIME_LONG	1605	The pre recorded start time is greater than the number of illegal capture.
NET_DVR_ERR_TRIGGER_RULE_LINE	1606	Trigger rule line error.
NET_DVR_ERR_LEFTRIGHT_ TRIGGERLINE_NOTVERTICAL	1607	Left and right trigger line is not vertical.
NET_DVR_ERR_FLASH_LAMP_MODE	1608	Flash lamp mode error.
NET_DVR_ERR_ILLEGAL_SNAPSHOT_ NUM	1609	Illegal capture number error.
NET_DVR_ERR_ILLEGAL_DETECTION_ TYPE	1610	Illegal detection type error.
NET_DVR_ERR_POSITIVEBACK_ TRIGGERLINE_HIGH	1611	Positive back to trigger line height error.
NET_DVR_ERR_MIXEDMODE_ CAPTYPE_ALLTARGETS	1612	Mixed mode only supports capture type all targets.

Error Name	Error Code	Error Description
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.
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

Error Name	Error Code	Error Description
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
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.

Error Name	Error Code	Error Description
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
NET_DVR_ERR_ROOMNO_ UNDEFINED	1951	Room number is not set
NET_DVR_ERR_NO_CALLING	1952	No call
NET_DVR_ERR_RINGING	1953	Ringing
NET_DVR_ERR_IS_CALLING_NOW	1954	Call in progress
NET_DVR_ERR_LOCK_PASSWORD_ WRONG	1955	Incorrect smart lock password
NET_DVR_ERR_CONTROL_LOCK_ FAILURE	1956	Lock control failure
NET_DVR_ERR_CONTROL_LOCK_ OVERTIME	1957	Lock control timed out
NET_DVR_ERR_LOCK_DEVICE_BUSY	1958	Smart lock device busy
NET_DVR_ERR_UNOPEN_REMOTE_ LOCK_FUNCTION	1959	Remote lock control not enabled
NET_DVR_ERR_FILE_NOT_COMPLETE	2100	Downloaded file is incomplete
NET_DVR_ERR_IPC_EXIST	2101	The camera already exists
NET_DVR_ERR_ADD_IPC	2102	Camera has been added to the channel
NET_DVR_ERR_OUT_OF_RES	2103	Not enough network bandwidth
NET_DVR_ERR_CONFLICT_TO_ LOCALIP	2104	IP address of camera conflicts with that of DVR
NET_DVR_ERR_IP_SET	2105	Invalid IP address
NET_DVR_ERR_PORT_SET	2106	Invalid port number
NET_ERR_WAN_NOTSUPPORT	2107	Not in the same LAN, cannot set security question or export GUID file
NET_ERR_MUTEX_FUNCTION	2108	Mutually exclusive function
NET_ERR_QUESTION_CONFIGNUM	2109	Error in number of security question configurations

Error Name	Error Code	Error Description
NET_ERR_FACECHAN_NORESOURCE	2110	All the face VCA channels are occupied.
NET_ERR_DATA_CALLBACK	2111	Data is calling back.
NET_ERR_ATM_VCA_CHAN_IS_ RELATED	2112	The VCA channel is already linked.
NET_ERR_ATM_VCA_CHAN_IS_ OVERLAPED	2113	The VCA channel is already overlayed.
NET_ERR _FACE_CHAN_UNOVERLAP_ EACH_OTHER	2114	The face channels cannot be overlayed.
NET_DVR_SMD_ENCODING_ NORESOURSE	2116	Insufficient SMD encoding resource
NET_DVR_SMD_DECODING_ NORESOURSE	2117	Insufficient SMD decoding resource
NET_DVR_FACELIB_DATA_ PROCESSING	2118	Face picture library data is in processing
NET_DVR_ERR_LARGE_TIME_ DIFFRENCE	2119	There is a great time difference between device and server.
NET_DVR_NO_SUPPORT_WITH_ PLAYBACK	2120	It is not supported. Playback is enabled.
NET_DVR_CHANNEL_NO_SUPPORT_ WITH_SMD	2121	It is not supported. SMD of channel is enabled.
NET_DVR_CHANNEL_NO_SUPPORT_ WITH_FD	2122	It is not supported. Face capture of channel is enabled.
NET_DVR_ILLEGAL_PHONE_NUMBER	2123	Invalid telephone number
NET_DVR_ILLEGAL_CERITIFICATE_ NUMBER	2124	Invalid ID No.
NET_DVR_ERR_CHANNEL_ RESOLUTION_NO_SUPPORT	2125	The channel resolution is not supported
NET_DVR_ERR_CHANNEL_ COMPRESSION_NO_SUPPORT	2126	The channel encoding format is not supported
NET_DVR_ERR_CLUSTER_DEVICE_ TOO_LESS	2127	Deleting is not allowed. The number of devices is not enough

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
NET_DVR_ERR_SAFETY_HELMET_NO_RESOURCE	2143	No enough hard hat detection resource.
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

Error Name	Error Code	Error Description
NET_DVR_INVALID_EZVIZ_SECRET_KEY	2189	Invalid verification code
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)

Error Name	Error Code	Error Description
NET_DVR_ERR_PANORAMIC_LIMIT_OPERATED	3004	Panoramic map and limit cannot be operated at same time
NET_DVR_ERR_SMARTH264_ROI_OPERATED	3005	SmartH264 and ROI cannot be enabled at the same time.
NET_DVR_ERR_RULENUM_LIMIT	3006	No more rules can be added.
NET_DVR_ERR_LASER_DEICING_OPERATED	3007	Laser and deicing function cannot be enabled at the same time.
NET_DVR_ERR_OFFDIGITALZOOM_OR_MINZOOMLIMIT	3008	Please disable the digital zoom function or set the zoom limit to the minimum value. Otherwise, when enabling smoke and fire detection, behavior analysis, ship detection, defective point correction, temperature measurement, smoke and fire shielding function, this error code will be prompted.
NET_DVR_SYNCHRONIZEFOV_ERROR	3010	Field of view synchronization failed.
NET_DVR_RULE_SHIELDMASK_CONFLICT_ERROR	3013	The rule region conflicts with the shielded area.
NET_DVR_ERR_NO_SAFETY_HELMET_REGION	3501	The hard hat detection area is not configured.
NET_DVR_ERR_UNCLOSED_SAFETY_HELMET	3502	The hard hat detection is enabled.
NET_DVR_UPLOAD_HBDLIBID_ERROR	3504	Incorrect ID of human body picture library (incorrect HBDID or customHBDID)

RTSP Communication Library Related Errors

Error Name	Error Code	Error Description
NET_DVR_RTSP_ERROR_NOENOUGHPRI	401	Authentication failed: if server returns 401, it will change to this error code
NET_DVR_RTSP_ERROR_ALLOC_RESOURCE	402	Failed to allocate the resource
NET_DVR_RTSP_ERROR_PARAMETER	403	Parameter error

Error Name	Error Code	Error Description
NET_DVR_RTSP_ERROR_NO_URL	404	The assigned URL does not exist: when the server returns 404, SDK turns to this error code. E.g. the channel is not available, or the channel does not support sub stream
NET_DVR_RTSP_ERROR_FORCE_STOP	406	The user forces to exit midway
NET_DVR_RTSP_GETPORTFAILED	407	RTSP port getting error.
NET_DVR_RTSP_DESCRIBERROR	410	RTSP DECRIBE communicate error
NET_DVR_RTSP_ DESCRIBESENDTIMEOUT	411	Sending "RTSP DECRIBE" is timeout.
NET_DVR_RTSP_DESCRIBESENDERROR	412	Failed to send "RTSP DECRIBE".
NET_DVR_RTSP_ DESCRIBERECVTIMEOUT	413	Receiving "RTSP DECRIBE" is timeout.
NET_DVR_RTSP_ DESCRIBERECVDATALOST	414	Receiving data of "RTSP DECRIBE" error.
NET_DVR_RTSP_DESCRIBERECVERROR	415	Failed to receive "RTSP DECRIBE".
NET_DVR_RTSP_DESCRIBESERVERERR	416	"RTSP DECRIBE, the device returns the error code: 501 (failed to allocate the resource in the device)
NET_DVR_RTSP_SETUPERROR	420	(or 419), RTSP SETUP interaction error. Generally, it is that the address(URL) returned by the device is not accessible, or it is rejected by the server
NET_DVR_RTSP_SETUPSENDTIMEOUT	421	Sending "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPSENDERROR	422	Sending "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRECVTIMEOUT	423	Receiving "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPRECVDATALOST	424	Receiving data of "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRECVERROR	425	Failed to receive "RTSP SETUP".
NET_DVR_RTSP_OVER_MAX_CHAN	426	"RTSP SETUP" device returns the error that values 401 or 501. It

Error Name	Error Code	Error Description
		exceeds the max connection number.
NET_DVR_RTSP_PLAYERROR	430	RTSP PLAY interaction error.
NET_DVR_RTSP_PLAYSENDTIMEOUT	431	Sending "RTSP PLAY" is timeout.
NET_DVR_RTSP_PLAYSENDERROR	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_ TEARDOWNSENDERROR	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.
NET_PLAYM4_DEC_VIDEO_ERROR	504	Failed to decode video data.

Error Name	Error Code	Error Description
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.
NET_PLAYM4_UPDATE_ERROR	523	Failed to update overlay surface

Error Name	Error Code	Error Description
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
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

Error Name	Error Code	Error Description
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.

Error Name	Error Code	Error Description
NET_ERR_NPQ_NOTSUPPORT	8005	NPQ library: This function is not supported.
NET_ERR_NPQ_NOTCALLBACK	8100	No data is called back.
NET_ERR_NPQ_LOADLIB	8101	Loading NPQ library failed.
NET_ERR_NPQ_STEAM_CLOSE	8104	The NPQ function of this stream is not enabled.
NET_ERR_NPQ_MAX_LINK	8110	No more streaming channel's NPQ function can be enabled.
NET_ERR_NPQ_STREAM_CFG_ CONFLICT	8111	The configured encoding parameters conflicted.

A.5 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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
SSDFileSystemIsError	0x30001013	SSD file system error (Error occurs when it is non-Ext4 file system)
insufficientSSDCapacityForFPD	0x30001014	Insufficient SSD space for person frequency detection
wifiException	0x3000100A	Wi-Fi big data server exception
structException	0x3000100D	Video parameters structure server exception.
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.
synchronizationError	0x40001003	Synchronization failed.
synchronizing	0x40001004	Synchronizing.
importError	0x40001005	Importing failed.
importing	0x40001006	Importing.
fileAlreadyExists	0x40001007	The file already exists.
invalidID	0x40001008	Invalid ID.
backupnodeNotAlloweLog	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.

Sub Status Code	Error Code	Description
exportItemsExceedLimit	0x40001010	No more items can be exported.
noFiles	0x40001011	The file does not exist.
beingExportedByAnotherUser	0x40001012	Being exported by others.
needReAuthentication	0x40001013	Authentication is needed after upgrade.
unitAddNotOnline	0x40001015	The added data analysis server is offline.
unitControl	0x40001016	The data analysis server is already added.
analysis unitFull	0x40001017	No more data analysis server can be added.
unitIDError	0x40001018	The data analysis server ID does not exist.
unitExit	0x40001019	The data analysis server already exists in the list.
unitSearch	0x4000101A	Searching data analysis server in the list failed.
unitNotOnline	0x4000101B	The data analysis server is offline.
unitInfoEror	0x4000101C	Getting data analysis server information failed.
unitGetNodeInfoError	0x4000101D	Getting node information failed.
unitGetNetworkInfoError	0x4000101E	Getting the network information of data analysis server failed
unitSetNetworkInfoError	0x4000101F	Setting the network information of data analysis server failed
setSmartNodeInfoError	0x40001020	Setting node information failed.
setUnitNetworkInfoError	0x40001021	Setting data analysis server network information failed.
unitRestartCloseError	0x40001022	Rebooting or shutting down data analysis server failed.
virtualIPnotAllowed	0x40001023	Adding virtual IP address is not allowed.
unitInstalled	0x40001024	The data analysis server is already installed.
badSubnetMask	0x40001025	Invalid subnet mask.
uintVersionMismatched	0x40001026	Data analysis server version mismatches.

Sub Status Code	Error Code	Description
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	Submiting task failed.
taskDelError	0x40001031	Deleting task failed.
taskPauseError	0x40001032	Pausing task failed.
taskContinueError	0x40001033	Starting task failed.
taskSeverNoCfg	0x40001035	Full-text search server is not configured.
taskPicSeverNoCfg	0x40001036	The picture server is not configured.
taskStreamError	0x40001037	Streaming information exception.
taskRecSDK	0x40001038	History recording is not supported.
taskCasaError	0x4000103A	Cascading is not supported.
taskVCARuleError	0x4000103B	Invalid VCA rule.
taskNoRun	0x4000103C	The task is not executed.
unitLinksNoStorageNode	0x4000103D	No node is linked with the data analysis server. Configure the node first.
searchFailed	0x4000103E	Searching video files failed.
searchNull	0x4000103F	No video clip.
userScheOffline	0x40001040	The task scheduler service is offline.
updateTypeUnmatched	0x40001041	The upgrade package type mismatches.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
TDAConnectFailed	0x40001064	Connecting to traffic data access server failed. Checking the server status.
86sdkConnectFailed	0x40001065	Connecting to listening port of iVMS-8600 System failed. Check the parameters.
nameExist	0x40001066	Duplicated server name.
batchProcessFailed	0x40001067	Processing in batch failed.
IDNotExist	0x40001068	The server ID does not exist.
serviceNumberReachesLimit	0x40001069	No more service can be added.
invalidServiceType.	0x4000106A	Invalid service type.
clusterGetInfo	0x4000106B	Getting cluster group information failed.
clusterDelNode	0x4000106C	Deletion node failed.
clusterAddNode	0x4000106D	Adding node failed.
clusterInstalling	0x4000106E	Creating cluster…Do not operate.
clusterUninstall	0x4000106F	Reseting cluster…Do not operate.
clusterInstall	0x40001070	Creating cluster failed.

Sub Status Code	Error Code	Description
clusterIpError	0x40001071	Invalid IP address of task scheduler server.
clusterNotSameSeg	0x40001072	The master node and slave node must be in the same network segment.
clusterVirIpError	0x40001073	Automatically getting virtual IP address failed. Enter manually.
clusterNodeUnadd	0x40001074	The specified master(slave) 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 master and slave nodes.
addNodeNetFailed	0x40001077	Adding node failed. The network disconnected.
needTwoMgmtNode	0x40001078	Two management nodes are required when adding master and slave nodes.
ipConflict	0x40001079	The virtual IP address and data analysis server's IP address conflicted.
ipUsed	0x4000107A	The virtual IP address has been occupied.
cloudAlalyseOnline	0x4000107B	The cloud analytic server is online.
virIP&mainIPnotSame NetSegment	0x4000107C	The virtual IP address is not in the same network segment with the IP address of master/slave node.
getNodeDispatchInfoFailed	0x4000107D	Getting node scheduler information failed.
unableModifyManagementNetworkIP	0x4000107E	Editing management network interface failed. The analysis board is in the cluster.
notSpecifyVirtualIP	0x4000107F	Virtual IP address should be specified for master and slave 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.

Sub Status Code	Error Code	Description
ArmingTimeError	0x40001088	Invalid arming time period.
ArmingSearchTimeError	0x40001089	Invalid search time period.
armingRelationshipReachesLimit	0x4000108A	No more relation can be created.
duplicateAarmingName	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.
searchByPictureTimedOut	0x40001094	Searching picture by picture timeout. Search again.
comparisonTimeRangeError	0x40001095	Comparison time period error.
selectMonitorNumberUpperLimit	0x40001096	No more monitoring point ID can be filtered.
noMoreComparisonTasksAdded	0x40001097	No more comparison task can be executed at the same time.
GetComparisonResultFailed	0x40001098	Getting comparison result failed.
comparisonTypeError	0x40001099	Comparison type error.
comparisonUnfinished	0x4000109A	The comparison is not completed.
facePictureModelInvalid	0x4000109B	Invalid face model.

Sub Status Code	Error Code	Description
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.
licenseExeedLimit	0x400010A3	Dongle limited.
licenseExpired	0x400010A4	Dongle expired.
licenseDisabled	0x400010A5	Unavailable dongle.
licenseNotExist	0x400010A6	The dongle does not exist.
SessionExpired	0x400010A7	Session expired .
beyondConcurrentLimit	0x400010A8	Out of concurrent limit.
stopSync	0x400010A9	Synchronization stopped.
getProgressFaild	0x400010AA	Getting progress failed.
uploadExtraCaps	0x400010AB	No more files can be uploaded.
timeRangeError	0x400010AC	Time period error.
dataPortNotConnected	0x400010AD	The data port is not connected.
addClusterNodeFailed	0x400010AE	Adding to the cluster failed. The device is already added to other cluster.
taskNotExist	0x400010AF	The task does not exist.
taskQueryFailed	0x400010B0	Searching task failed.
modifyTimeRuleFailed	0x400010B2	The task already exists. Editing time rule is not allowed.
modifySmartRuleFailed	0x400010B3	The task already exists. Editing VAC rule is not allowed.
queryHistoryVideoFaile d	0x400010B4	Searching history video failed.

Sub Status Code	Error Code	Description
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 storaged server failed.
storageServiceIPNotExist	0x400010BD	The storage server IP address does not exist.
syncBackupDatabaseFailed	0x400010BE	Synchronizing slave database failed. Try again.
syncBackupNTPTimeFailed	0x400010BF	Synchronizing NTP time of slave server failed.
clusterNotSelectLoopbackAddress	0x400010C0	Loopbacl address is not supported by the master or slave cluster.
addFaceRecordFailed	0x400010C1	Adding face record failed.
deleteFaceRecordFailed	0x400010C2	Deleting face record failed.
modifyFaceRecordFailed	0x400010C3	Editing face record failed.
queryFaceRecordFailed	0x400010C4	Searching face record failed.
faceDetectFailed	0x400010C5	Detecting face failed.
libraryNotExist	0x400010C6	The library does not exist.
blackListQueryExporting	0x400010C7	Exporting matched blacklists.
blackListQueryExported	0x400010C8	The matched blacklists are exported.
blackListQueryStopExporting	0x400010C9	Exporting matched blacklists is stopped.

Sub Status Code	Error Code	Description
blackListAlarmQueryExporting	0x400010CA	Exporting matched blacklist alarms.
blackListAlarmQueryExported	0x400010CB	The matched blacklists alarms are exported.
blackListAlarmQueryStopExporting	0x400010CC	Exporting matched blacklist alarms is stopped.
getBigDataCloudAnalysisFailed	0x400010CD	Getting big data cloud analytic information failed.
setBigDataCloudAnalysisFailed	0x400010CE	Configuring big data cloud analytic failed.
submitMapSearchFailed	0x400010CF	Submitting search by picture task failed.
controlRelationshipNotExist	0x400010D0	The relation does not exist.
getHistoryAlarmInfoFailed	0x400010D1	Getting history alarm information failed.
getFlowReportFailed	0x400010D2	Getting people counting report failed.
addGuardFailed	0x400010D3	Adding arming configuration failed.
deleteGuardFailed	0x400010D4	Deleting arming configuration failed.
modifyGuardFailed	0x400010D5	Editing arming configuration failed.
queryGuardFailed	0x400010D6	Searching arming configurations failed.
uploadUserSuperCaps	0x400010D7	No more user information can be uploaded.
bigDataServerConnectFailed	0x400010D8	Connecting to big data server failed.
microVideoCloudRequestInfoBuildFailed	0x400010D9	Adding response information of micro video cloud failed.
microVideoCloudResponseInfoBuildFailed	0x400010DA	Parsing response information of micro video cloud failed.
transcodingServerRequestInfoBuildFailed	0x400010DB	Adding response information of transcoding server failed.
transcodingServerResponseInfoParseFailed	0x400010DC	Parsing response information of transcoding server failed.

Sub Status Code	Error Code	Description
transcodingServerOffline	0x400010DD	Transcoding server is offline.
microVideoCloudOffline	0x400010DE	Micro video cloud is offline.
UPSServerOffline	0x400010DF	UPS monitor server is offline.
statisticReportRequestInfoBuildFailed	0x400010E0	Adding response information of statistics report failed.
statisticReportResponseInfoParseFailed	0x400010E1	Parsing response information of statistics report failed.
DisplayConfigInfoBuildFailed	0x400010E2	Adding display configuration information failed.
DisplayConfigInfoParseFailed	0x400010E3	Parsing display configuration information failed.
DisplayConfigInfoSaveFailed	0x400010E4	Saving display configuration information failed.
notSupportDisplayConfigType	0x400010E5	The display configuration type is not supported.
passError	0x400010E7	Incorrect password.
upgradePackageLarge	0x400010EB	Too large upgrade package.
sesssionUserReachesLimit	0x400010EC	No more user can log in via session.
ISO 8601TimeFormatError	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.

Sub Status Code	Error Code	Description
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.
forwardingServer ReturnFailed	0x400010FA	No response is returned by forwarding service.
storageServer ReturnFailed	0x400010FB	No response is returned by storage service.
cloudAnalysisServerReturnFailed	0x400010FC	No response is returned by cloud analytic service.
modelEmpty	0x400010FD	No model is obtained.
mainAndBackupNodeCannotModifyManagementNetworkInterfaceIP	0x400010FE	Editing the management interface IP address of master node and backup node is not allowed.
IDTooLong	0x400010FF	The ID is too long.
pictureCheckFailed	0x40001100	Detecting picture failed.
pictureModelingFailed	0x40001101	Modeling picture failed.
setCloudAnalsisDefaultProvinceFailed	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.

Sub Status Code	Error Code	Description
storageServiceIPEmpty	0x40001107	The storage server IP address is required.
bigDataServerRequestInfoBuildFail	0x40001108	Creating request message of big data service failed.
analysiTimedOut	0x40001109	Analysis time out.
high-performanceModeDisabled.	0x4000110A	Please enable high-performance mode.
configuringUPSMonitoringServerTimedOut	0x4000110B	Configurating the UPS monitoring server time out. Check IP address.
cloudAnalysisRequestInformationBuildFailed	0x4000110C	Creating request message of cloud analytic service failed.
cloudAnalysisResponseInformationParseFailed	0x4000110D	Parsing response message of cloud analytic service failed.
allCloudAnalysisInterfaceFailed	0x4000110E	Calling API for cloud analytic service failed.
cloudAnalysisModelCompareFailed	0x4000110F	Model comparison of cloud analytic service failed.
cloudAnalysisFacePictureQualityRatingFailed	0x40001110	Getting face quality grading of cloud analytic service failed.
cloudAnalysisExtractFeaturePointsFailed	0x40001111	Extracting feature of cloud analytic service failed.
cloudAnalysisExtractPropertyFailed	0x40001112	Extracting property of cloud analytic service failed.
getAddedNodeInformationFailed	0x40001113	Getting the added nodes information of data analysis server failed.
noMoreAnalysisUnitsAdded	0x40001114	No more data analysis servers can be added.
detectionAreaInvalid	0x40001115	Invalid detection region.
shieldAreaInvalid	0x40001116	Invalid shield region.
noMoreShieldAreasAdded	0x40001117	No more shield region can be drawn.
onlyAreaOfRectangleShapeAllowed	0x40001118	Only drawing rectangle is allowed in detection area.

Sub Status Code	Error Code	Description
numberReachedLlimit	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 master and backup cluster requires at least two devices.
updatePackageFileNotUploaded	0x4000111E	Upgrade package is not uploaded.
highPerformanceTasksNotSupportDrawingDetectionRegions	0x4000111F	Drawing detection area is not allowed under high-performance mode.
controlCenterIDDoesNotExist	0x40001120	The control center ID does not exist.
regionIDDoesNotExist	0x40001121	The area ID does not exist.
licensePlateFormatError	0x40001122	Invalid license plate format.
managementNodeDoesNotSupportThisOperation	0x40001123	The operation is not supported.
searchByPictureResourceNotConfiged	0x40001124	The conditions for searching picture by picture are not configured.
videoFileEncapsulationFormatNotSupported	0x40001125	The video container format is not supported.
videoPackageFailure	0x40001126	Converting video container format failed.
videoCodingFormatNotSupported	0x40001127	Video coding format is not supported.
monitorOfDeviceArmingdeleteArmingInfo	0x40001129	The camera is armed. Disarm it and try again.
getVideoSourceTypeFailed	0x4000112A	Getting video source type failed.
smartRulesBuildFailed	0x4000112B	Creating VAC rule failed.
smartRulesParseFailed	0x4000112C	Parsing VAC rule failed.

Sub Status Code	Error Code	Description
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 master node and backup node failed. Check the node status.
theSearchIdDoesNotExist	0x40001133	The search ID does not exist.
theSynchronizationIdDoesNotExist	0x40001134	The synchronization ID does not exist.
theUserIdDoesNotExist	0x40001136	The user ID does not exist.
theIndexCodeDoesNotExist	0x40001138	The index code does not exist.
theControlCenterIdDoesNotExist	0x40001139	The control center ID does not exist.
theAreaIdDoesNotExist	0x4000113A	The area ID does not exist.
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.

Sub Status Code	Error Code	Description
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 serice.
duplicatedSchedule	0x40001145	Schedule period cannot be overlapped.
continuousScheduleWithSameAlgorithmTypeShouldBeMerged	0x40001146	The continuous schedule periods with same algorithm type should be merged.
invalidStreamingTimeRange	0x40001147	Invalid streaming time period.
invalidListLibraryType	0x40001148	Invalid list library type.
theNumberOfMatchedResultsShouldBeLargerThan0	0x40001149	The number of search results should be larger than 0.
invalidValueRangeOfSimilarity	0x4000114A	Invalid similarity range.
invalidSortingType	0x4000114B	Invalid sorting type.
noMoreListLibraryCanBeLinkedToTheDevice	0x4000114C	No more lists can be added to one device.
InvalidRecipientAddressFormat	0x4000114D	Invalid address format of result receiver.
creatingClusterFailedTheDongleIsNotPluggedIn	0x4000114E	Insert the dongle before creating cluster.
theURLIsTooLong	0x4000114F	No schedule configured for the task.
noScheduleIsConfiguredForTheTask	0x40001150	No schedule configured for the task.
theDongleIsExpiried	0x40001151	Dongle has expired.

Sub Status Code	Error Code	Description
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 serice.
theCloudAnalyticsIsInterrupted	0x4000115D	The operation of cloud analytic serice is interrupted.
theServiceIsNotReady	0x4000115E	The service is not ready.
searchingForExternalApiFailed	0x4000115F	Searching external interfaces failed.
noOnlineNode	0x40001160	No node is online.
noNodeAllocated	0x40001161	No allocated node.
noMatchedList	0x40001162	No matched list.
allocatingFailedTooManyFacePictureLists	0x40001163	Allocation failed. Too many lists of big data service.
searchIsNotCompletedSearchAgain	0x40001164	Current searching is not completed. Search again.
allocatingListIsNotCompleted	0x40001165	Allocating list is not completed.
searchingForCloudAnalyticsResultsFailed	0x40001166	Searching cloud analytic serice overtime.

Sub Status Code	Error Code	Description
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.
noAvailableDataSlicingVersionInformationArmFirstAndSliceTheData	0x40001169	Invalid standard version information.
duplicatedOperationDataSlicingIsExecuting	0x4000116A	Slicing failed. Duplicated operation.
slicinDataFailedNoArmedFacePictureLibrary	0x4000116B	Slicing failed. No arming information in the face big data.
GenerateBenchmarkFileFailedSlicingAgain	0x4000116C	Generating sliced file failed. Slice again.
NonprimaryNodeIsProhibitedFromSlcingData	0x4000116D	Slicing is not allowed by the backup node.
NoReadyNodeToClusterServers	0x4000116E	Creating the cluster failed. No ready node.
NodeManagementServiceIsOffline	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-	0x40001174	Getting the search result details of high frequency alarms failed.

Sub Status Code	Error Code	Description
frequencyPeopleDetectionLogsFailed.		
theArmedCamerasAlreadyExistInTheControlCenter	0x40001175	Some cameras in control center are already armed.
disarmingFailedTheCameraIsNotArmed	0x40001177	Disarming failed. The camera is not armed.
noDataReturned	0x40001178	No response is returned by the big data service.
preallocFailure	0x40001179	Pre-allocating algorithm resource failed.
overDogLimit	0x4000117A	Configuration failed. No more resources can be pre-allocated.
analysisServicesDoNotSupport	0x4000117B	Not supported.
commandAndDispatchServiceError	0x4000117C	Scheduling service of cloud analytic serice error.
engineModuleError	0x4000117D	Engine module of cloud analytic serice error.
streamingServiceError	0x4000117E	Streaming component of cloud analytic serice error.
faceAnalysisModuleError	0x4000117F	Face analysis module of cloud analytic serice error.
vehicleAnalysisModuleError	0x40001180	Vehicle pictures analytic module of cloud analytic serice error.
videoStructuralAnalysisModuleError	0x40001181	Video structuring module of cloud analytic serice error.
postprocessingModuleError	0x40001182	Post-processing module of cloud analytic serice error.
frequentlyAppearedPersonAlarmIsAlreadyConfiguredForListLibrary	0x40001183	High frequency alarm is already armed for blacklist library.
creatingListLibraryFaile d	0x40001184	Creating list library failed.
invalidIdentiryKeyOfListLibrary	0x40001185	Invalid identity key of list library.

Sub Status Code	Error Code	Description
noMoreDevicesCanBeArmed	0x40001186	No more camera can be added.
settingAlgorithmTypeForDeviceFailed	0x40001187	Allocating task resource failed.
gettingHighFrequencyPersonDetectionAlarmInformationFailed	0x40001188	Setting high frequency alarm failed.
invalidSearchConfition	0x40001189	Invalid result.
theTaskIsNotCompleted	0x4000118B	The task is not completed.
resourceOverRemainLimit	0x4000118C	No more resource can be pre-allocated.
frequentlyAppearedPersonAlarmIsAlreadyConfiguredForTheCameraDisarmFirstAndTryAgain	0x4000118D	The high frequency alarm of this camera is configured. Delete the arming information and try again.
switchtimedifflesslimit	0x4000123b	Time difference between power on and off should be less than 10 minutes.
associatedFaceLibNumOverLimit	0x40001279	Maximum number of linked face picture libraries reached.
noMorePeopleNumChangeRulesAdded	0x4000128A	Maximum number of people number changing rules reached.
noMoreViolentMotionRulesAdded	0x4000128D	Maximum number of violent motion rules reached.
noMoreLeavePositionRulesAdded	0x4000128E	Maximum number of leaving position rules reached.
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.

Sub Status Code	Error Code	Description
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.
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.
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.
invalLicense	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.
audioIsPlayingPleaseWait	0x40002067	Audio is playing. Please wait.
twoWayAudioInProgressPleaseWait	0x40002068	Two-way audio in progress. Please wait.
calibrationPointNumFull	0x40002069	The maximum number of calibration points reached.
completeTheLevelCalibrationFirst	0x4000206A	The level calibration is not set.
completeTheRadarCameraCalibrationFirst	0x4000206B	The radar-camera calibration is not set.
pointsOnStraightLine	0x4000209C	Calibrating failed. The calibration points cannot be one the same line.
TValueLessThanOrEqualZero	0x4000209D	Calibration failed. The T value of the calibration points should be larger than 0.

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.
incorrectConsolePassword	0x40002106	Saving failed. Incorrect console command.
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.

Sub Status Code	Error Code	Description
zoneFault	0x4000800E	The zone is in fault status.
pwdConflict	0x4000800F	Password conflicted.
audioTestEntryFailed	0x40008010	Enabling audio test mode failed.
audioTestRecoveryFailed	0x40008011	Disabling audio test mode failed.
addCardMode	0x40008012	Adding card mode.
searchMode	0x40008013	Search mode.
addRemoterMode	0x40008014	Adding keyfob mode.
registerMode	0x40008015	Registration mode.
exDevNotExist	0x40008016	The peripheral does not exist.
theNumberOfExDevLimited	0x40008017	No peripheral can be added.
sirenConfigFailed	0x40008018	Setting siren failed.
chanCannotRepeatedBinded	0x40008019	This channel is already linked by the zone.
inProgramMode	0x4000801B	The keypad is in programming mode.
inPaceTest	0x4000801C	In pacing mode.
arming	0x4000801D	Arming.
masterSlaveIsEnable	0x4000802c	The master-slave relationship has taken effect, the slave 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?

Sub Status Code	Error Code	Description
radarMoudleConnectFail	0x40008033	Radar module communication failed.
importCfgFilePasswordErr	0x40008034	Incorrect password for importing configuration files.
overAudioFileNumLimit	0x40008038	The number of audio files exceeds the limit.
audioFileNameIsLong	0x40008039	The audio file name is too long.
audioFormatIsWrong	0x4000803a	The audio file format is invalid.
audioFileIsLarge	0x4000803b	The size of the audio file exceeds the limit.
pircamCapTimeOut	0x4000803c	Capturing of pircam timed out.
pircamCapFail	0x4000803d	Capturing of pircam failed.
pircamIsCaping	0x4000803e	The pircam is capturing.
audioFileHasExisted	0x4000803f	The audio file already exists.
subscribeTypeErr	0x4000a016	This metadata type is not supported to be subscribed.
startAppFail	/	Starting running application program failed.
yuvconflict	/	The raw video stream conflicted.
overMaxAppNum	/	No more application program can be uploaded.
noFlash	/	Insufficient flash.
noFlash	/	The platform mismatches.

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.

Sub Status Code	Error Code	Description
badIPv6Address	0x60000005	Invalid IPv6 address.
conflictIPv4Address	0x60000006	IPv4 address conflicted.
conflictIPv6Address	0x60000007	IPv6 address conflicted.
badDomainName	0x60000008	Invalid domain name.
connectSreverFail	0x60000009	Connecting to server failed.
conflictDomainName	0x6000000A	Domain name conflicted.
badPort	0x6000000B	Port number conflicted.
portError	0x6000000C	Port error.
exportErrorData	0x6000000D	Importing data failed.
badNetMask	0x6000000E	Invalid sub-net mask.
badVersion	0x6000000F	Version mismatches.
badDevType	0x60000010	Device type mismatches.
badLanguage	0x60000011	Language mismatches.
incorrentUserNameOrPassword	0x600000012	Incorrect user name or password.
invalidStoragePoolOfCloudServer	0x600000013	Invalid storage pool. The storage pool is not configured or incorrect ID.
noFreeSpaceOfStoragePool	0x600000014	Storage pool is full.
riskPassword	0x600000015	Risky password.
UnSupportCapture	0x600000016	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.

Sub Status Code	Error Code	Description
beyondARGSRangeLimit	0x60000027	The parameter value is out of limit.
DetectionLineOutofDetectionRegion	0x60000085	The rule line is out of region.
DetectionRegionError	0x60000086	Rule region error. Make sure the rule region is convex polygon.
DetectionRegionOutOfCountingRegion	0x60000087	The rule region must be marked as red frame.
PedalAreaError	0x60000088	The pedal area must be in the rule region.
DetectionAreaABError	0x60000089	The detection region A and B must be in the a rule frame.
ABRegionCannotIntersect	0x6000008a	Region A and B cannot be overlapped.
customHBPIDError	0x6000008b	Incorrect ID of custom human body picture library
customHBPIDRepeat	0x6000008c	Duplicated ID of custom human body picture library
dataVersionsInHBDLibMismatches	0x6000008d	Database versions mismatches of human body picture library
invalidHBPID	0x6000008e	Invalid human body picture PID
invalidHBDID	0x6000008f	Invalid ID of human body picture library
humanLibraryError	0x60000090	Error of human body picture library
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.

Sub Status Code	Error Code	Description
analysisEnginesUsageExcced	0x60001001	The engine usage is overloaded.
PicAnalysisNoResourceError	0x60001002	No analysis engine provided for picture secondary recognition.
analysisEnginesLoadingError	0x60001003	Initializing analysis engine.
analysisEnginesAbnormaError	0x60001004	Analysis engine exception.
analysisEnginesFacelibImportin g	0x60001005	Importing pictures to face picture library. Failed to edit analysis engine parameters.
analysisEnginesAssociatedChan nel	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.
noSupportWithPlaybackAbstra ct	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.

Sub Status Code	Error Code	Description
noSupportWithPersonDensityDetect	0x60001013	This function is not supported. The people density detection is enabled.
ipcResolutionOverflow	0x60001014	The configured resolution of network camera is invalid.
ipcBitrateOverflow	0x60001015	The configured bit rate of network camera is invalid.
tooGreatTimeDifference	0x60001016	Too large time difference between device and server.
noSupportWithPlayback	0x60001017	This function is not supported. Playback is enabled.
channelNoSupportWithSMD	0x60001018	This function is not supported. Motion detection is enabled.
channelNoSupportWithFD	0x60001019	This function is not supported. Face capture is enabled.
illegalPhoneNumber	0x6000101a	Invalid phone number.
illegalCertificateNumber	0x6000101b	Invalid certificate No.
linkedCameraOutLimit	0x6000101c	Connecting camera timed out.
achieveMaxChannelLimit	0x6000101e	No more channels are allowed.
humanMisInfoFilterEnabledChanNumError	0x6000101f	No more channels are allowed to enable preventing false alarm.
humanEnginesNoResource	0x60001020	Insufficient human body analysis engine resources.
taskNumberOverflow	0x60001021	No more tasks can be added.
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

Sub Status Code	Error Code	Description
noDoubleVerificationUser	0x60001043	No double verification user
timeSpanNumOverLimit	0x60001044	Max. number of time buckets reached
channelNumOverLimit	0x60001045	Max. number of channels reached
noSearchIDResource	0x60001046	Insufficient searchID resources
noSupportDeleteStrangerLib	0x60001051	Deleting stranger library is not supported
noSupportCreateStrangerLib	0x60001052	Creating stranger library is not supported
behaviorAnalysisRuleInfoError	0x60001053	Behavior analysis rule parameters error.
safetyHelmetParamError	0x60001054	Hard hat parameters error.
OneChannelOnlyCanBindOneEngine	0x60001077	No more engines can be bound.
engineTypeMismatch	0x60001079	Engine type mismatched.
badUpgradePackage	0x6000107A	Invalid upgrade package.
AudioFileNameDuplicate	0x60001135	Duplicated audio file name.
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-adptive 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.

StatusCode=7

SubStatusCode	Error Code	Description
rebootRequired	0x70000001	Reboot to take effect.