



Device Network SDK (Person Arming)

Developer Guide

Legal Information

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

Contents

Chapter 1 Overview	1
1.1 Introduction	1
1.2 Update History	1
Chapter 2 Typical Applications	2
2.1 Create Human Body Picture Library	2
2.2 Manage Records in Human Body Picture Library	3
2.3 Configure Auto Person Arming	7
2.4 Configure Manual Person Arming	10
2.5 Receive Event or Alarm Details	11
2.5.1 Receive Alarm/Event in Arming Mode	11
2.5.2 Receive Alarm/Event in Listening Mode	14
2.6 Integrate by Transmitting Text Protocol	17
Chapter 3 API Reference	19
3.1 NET_DVR_Cleanup	19
3.2 NET_DVR_CloseAlarmChan_V30	19
3.3 NET_DVR_GetDVRConfig	20
3.4 NET_DVR_GetErrorMsg	21
3.5 NET_DVR_GetLastError	21
3.6 NET_DVR_GetUploadResult	21
3.7 NET_DVR_GetUploadState	22
3.8 NET_DVR_Init	24
3.9 NET_DVR_Login_V40	24
3.10 NET_DVR_Logout	25
3.11 NET_DVR_SDKChannelToISAPI	26
3.12 NET_DVR_SetDVRConfig	26
3.13 NET_DVR_SetDVRMessageCallBack_V50	27

3.14 NET_DVR_SetSDKInitCfg	35
3.15 NET_DVR_SetupAlarmChan_V50	37
3.16 NET_DVR_StartListen_V30	37
3.17 NET_DVR_STDXMLConfig	38
3.18 NET_DVR_StopListen_V30	39
3.19 NET_DVR_UploadClose	40
3.20 NET_DVR_UploadFile_V40	40
3.21 NET_DVR_UploadSend	41
3.22 Callback Function	42
3.22.1 fLoginResultCallBack	42
3.22.2 MSGCallBack	42
Appendix A. Data Structure	44
A.1 NET_ALARM_CVR_SUBINFO_UNION	44
A.2 NET_ALARM_RECORDFILE_LOSS	44
A.3 NET_ALARM_RECORD_EXCEPTION	45
A.4 NET_ALARM_RESOURCE_USAGE	45
A.5 NET_ALARM_STREAM_EXCEPTION	46
A.6 NET_DVR_ALARM_FIXED_HEADER	46
A.7 NET_DVR_ALARM_ISAPI_INFO	50
A.8 NET_DVR_ALARM_ISAPI_PICDATA	51
A.9 NET_DVR_ALARMINFO_DEV	51
A.10 NET_DVR_ALARMINFO_DEV_V40	52
A.11 NET_DVR_ALARMINFO_V30	53
A.12 NET_DVR_ALARMINFO_V40	54
A.13 NET_DVR_ALARMER	55
A.14 NET_DVR_DEVICEINFO_V30	56
A.15 NET_DVR_DEVICEINFO_V40	60
A.16 NET_DVR_ETHERNET_V30	63

A.17 NET_DVR_HBDLIB_COND	64
A.18 NET_DVR_INIT_CFG_ABILITY	64
A.19 NET_DVR_IPADDR_UNION	65
A.20 NET_DVR_LOCAL_SDK_PATH	65
A.21 NET_DVR_MIME_UNIT	66
A.22 NET_DVR_NETCFG_V50	67
A.23 NET_DVR_PPPOECFG	68
A.24 NET_DVR_SEND_PARAM_IN	69
A.25 NET_DVR_SETUPALARM_PARAM_V50	70
A.26 NET_DVR_TIME	73
A.27 NET_DVR_TIME_EX	73
A.28 NET_DVR_USER_LOGIN_INFO	74
A.29 NET_DVR_XML_CONFIG_INPUT	75
A.30 NET_DVR_XML_CONFIG_OUTPUT	76
Appendix B. Enumeration	77
B.1 NET_SDK_UPLOAD_TYPE	77
Appendix C. Device Network SDK Errors	80
Appendix D. Request URIs	124
D.1 http://<ipAddress>:<portNo>/<url>	126
D.2 /ISAPI/Event/notification/alertStream	126
D.3 /ISAPI/Event/notification/httpHosts	127
D.4 /ISAPI/Event/notification/httpHosts/<ID>/test	129
D.5 /ISAPI/Event/notification/httpHosts/capabilities	129
D.6 /ISAPI/Event/schedules/faceLib/<ID>/<FDID>	130
D.7 /ISAPI/Event/schedules/HBDLib-<ID>/<HBDID>	131
D.8 /ISAPI/Event/triggers/faceLib-<channelID>/<FDID>	131
D.9 /ISAPI/Event/triggers/HBDLib-<ID>/<HBDID>	133
D.10 /ISAPI/Event/triggersCap	133

D.11 /ISAPI/Intelligent/channels/<ID>/capabilities	134
D.12 /ISAPI/Intelligent/channels/<ID>/manualPersonArmingTrack/capabilities?format=json	134
D.13 /ISAPI/Intelligent/channels/<ID>/manualPersonArmingTrack?format=json	135
D.14 /ISAPI/Intelligent/channels/<ID>/personArmingTrack/advanceConfiguration/capabilities?format=json	135
D.15 /ISAPI/Intelligent/channels/<ID>/personArmingTrack/advanceConfiguration?format=json	136
D.16 /ISAPI/Intelligent/channels/<ID>/personArmingTrack/capabilities?format=json	137
D.17 /ISAPI/Intelligent/channels/<ID>/personArmingTrack?format=json	137
D.18 /ISAPI/Intelligent/channels/<ID>/restoreAlgLibParam/<NAME>	138
D.19 /ISAPI/Intelligent/HBDLib/<HBDID>/capacity?format=json	139
D.20 /ISAPI/Intelligent/HBDLib/<HBDID>/HBP/<HBPID>?format=json	139
D.21 /ISAPI/Intelligent/HBDLib/<HBDID>?format=json	141
D.22 /ISAPI/Intelligent/HBDLib/capabilities?format=json	143
D.23 /ISAPI/Intelligent/HBDLib/extend?format=json	143
D.24 /ISAPI/Intelligent/HBDLib/HBDSearch?format=json	144
D.25 /ISAPI/Intelligent/HBDLib/HBP/delete?format=json	145
D.26 /ISAPI/Intelligent/HBDLib/HBPUploadData?format=json	145
D.27 /ISAPI/Intelligent/HBDLib/manualModeling/<HBDID>/HBP/<HBPID>?format=json	146
D.28 /ISAPI/Intelligent/HBDLib/manualModeling/<HBDID>?format=json	147
D.29 /ISAPI/Intelligent/HBDLib/manualModeling/progressState/<taskID>?format=json	147
D.30 /ISAPI/Intelligent/HBDLib/manualModelingRangeTask?format=json	148
D.31 /ISAPI/Intelligent/HBDLib/modelingStatus/<HBDID>/HBP/<HBPID>?format=json	148
D.32 /ISAPI/Intelligent/HBDLib/modelingStatus?format=json	149
D.33 /ISAPI/Intelligent/HBDLib?format=json	149
D.34 /ISAPI/System/capabilities	151
Appendix E. Request and Response Messages	152
E.1 JSON_AdvanceConfiguration	152

E.2 JSON_AdvanceConfigurationCap	153
E.3 JSON_CreateHBDLibList	154
E.4 JSON_EventNotificationAlert_Alarm/EventInfo	155
E.5 JSON_EventNotificationAlert_manualPersonArmingTrackMsg	156
E.6 JSON_EventNotificationAlert_personArmingTrackMsg	163
E.7 JSON_HBDInfo	174
E.8 JSON_HBDLibBaseCfg	174
E.9 JSON_HBDLibBaseCfgList	175
E.10 JSON_HBDLibCap	175
E.11 JSON_HBDLibInfoList	179
E.12 JSON_HBDModelingStatusSearchDescription	179
E.13 JSON_HBDModelingStatusSearchResult	180
E.14 JSON_HBDSearchDescription	181
E.15 JSON_HBDSearchResult	182
E.16 JSON_HBPCapacity	183
E.17 JSON_HBPUploadData	183
E.18 JSON_HBPUploadResult	185
E.19 JSON_HumanAppendData	185
E.20 JSON_ManualModeling	186
E.21 JSON_ManualModeling_TaskInfo	186
E.22 JSON_ManualPersonArmingTrack	186
E.23 JSON_ManualPersonArmingTrackCap	187
E.24 JSON_ModelingRangeList	188
E.25 JSON_ModelingRangeTaskState	188
E.26 JSON_ModelingStatus	188
E.27 JSON_PersonArmingTrack	189
E.28 JSON_PersonArmingTrackCap	190
E.29 JSON_PersonInfoExtend	191

E.30 JSON_ResponseStatus	191
E.31 XML_DeviceCap	192
E.32 XML_EventNotificationAlert_AlarmEventInfo	204
E.33 XML_EventNotificationAlert_HeartbeatInfo	205
E.34 XML_EventTrigger	206
E.35 XML_EventTriggerCapType	207
E.36 XML_EventTriggerNotification	208
E.37 XML_EventTriggerNotificationList	209
E.38 XML_EventTriggersCap	210
E.39 XML_HttpHostNotification	213
E.40 XML_HttpHostNotificationCap	215
E.41 XML_HttpHostNotificationList	216
E.42 XML_HttpHostTestResult	217
E.43 XML_IntelliCap	218
E.44 XML_Schedule	222
E.45 XML_ResponseStatus	223
Appendix F. Response Codes of Text Protocol	225

Chapter 1 Overview

1.1 Introduction

When the device detects the armed face or human body in the monitoring scene, alarm is triggered, device tracks and captures the target. During the tracking, the target location and other related information are uploaded. To realize the auto tracking of a target person, you should complete the settings of multi-target-type detection, human body picture library, face picture library, person arming rules, linkage actions, and so on.



Note

There are two person arming types: device auto person arming and manual person arming. If both are supported, the manual person arming has a higher priority, that is, you can switch to manual person arming during ongoing auto person arming.

1.2 Update History

Summary of Changes in Version 6.1.4.20_May, 2020

New document.

Chapter 2 Typical Applications

2.1 Create Human Body Picture Library

Human body picture library is used to store modeled human body pictures and information for management and comparison.

Steps

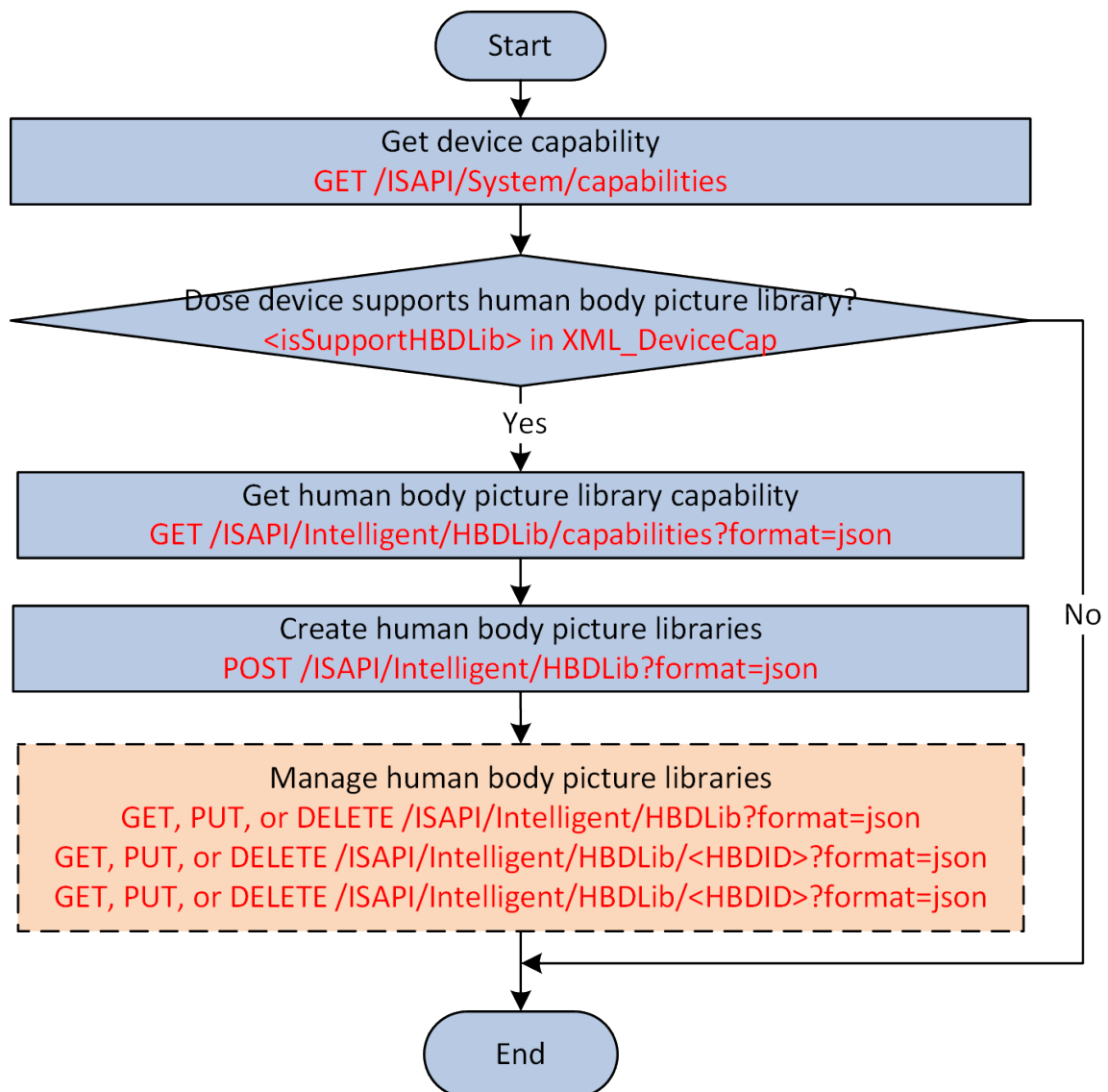


Figure 2-1 API Calling Flow of Creating Human Body Picture Library

Note

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

1. Call **/ISAPI/System/capabilities** by GET method to get the device capability to check whether the device supports configuration of human body picture library.
 - If the node **<isSupportHBDLib>** exists in the returned device capability **XML_DeviceCap**, and its value is true, perform the following steps to continue.
 - Otherwise, end this task as the human body picture library is not supported.
2. Call **/ISAPI/Intelligent/HBDLib/capabilities?format=json** by GET method to get the configuration capability of human body picture library.

The configuration capability is returned in the message **JSON_HBDLibCap**.
3. Call **/ISAPI/Intelligent/HBDLib?format=json** by POST method to create human body picture libraries.
4. **Optional:** Perform the following operation(s) after creating human body library.

Get, Set, or Delete Human Body Picture Libraries in a Batch

GET, PUT, or DELETE **/ISAPI/Intelligent/HBDLib?format=json**

Get, Set, or Delete a Human Body Picture Library

GET, PUT, or DELETE **/ISAPI/Intelligent/HBDLib/<HBDID>?format=json**

Get or Set Custom Information of Human Body Picture Library

GET or PUT **/ISAPI/Intelligent/HBDLib/extend?format=json**

2.2 Manage Records in Human Body Picture Library

You can upload human body pictures with linked information to the human body picture library and manage the existing human body pictures.

Before You Start

- Make sure you have called **NET_DVR_Init** to initialize the development environment.
- Make sure you have called **NET_DVR_Login_V40** to log in to the device.
- Make sure you have created a human body picture library. Refer to *Create Human Body Picture Library* for details.

Steps

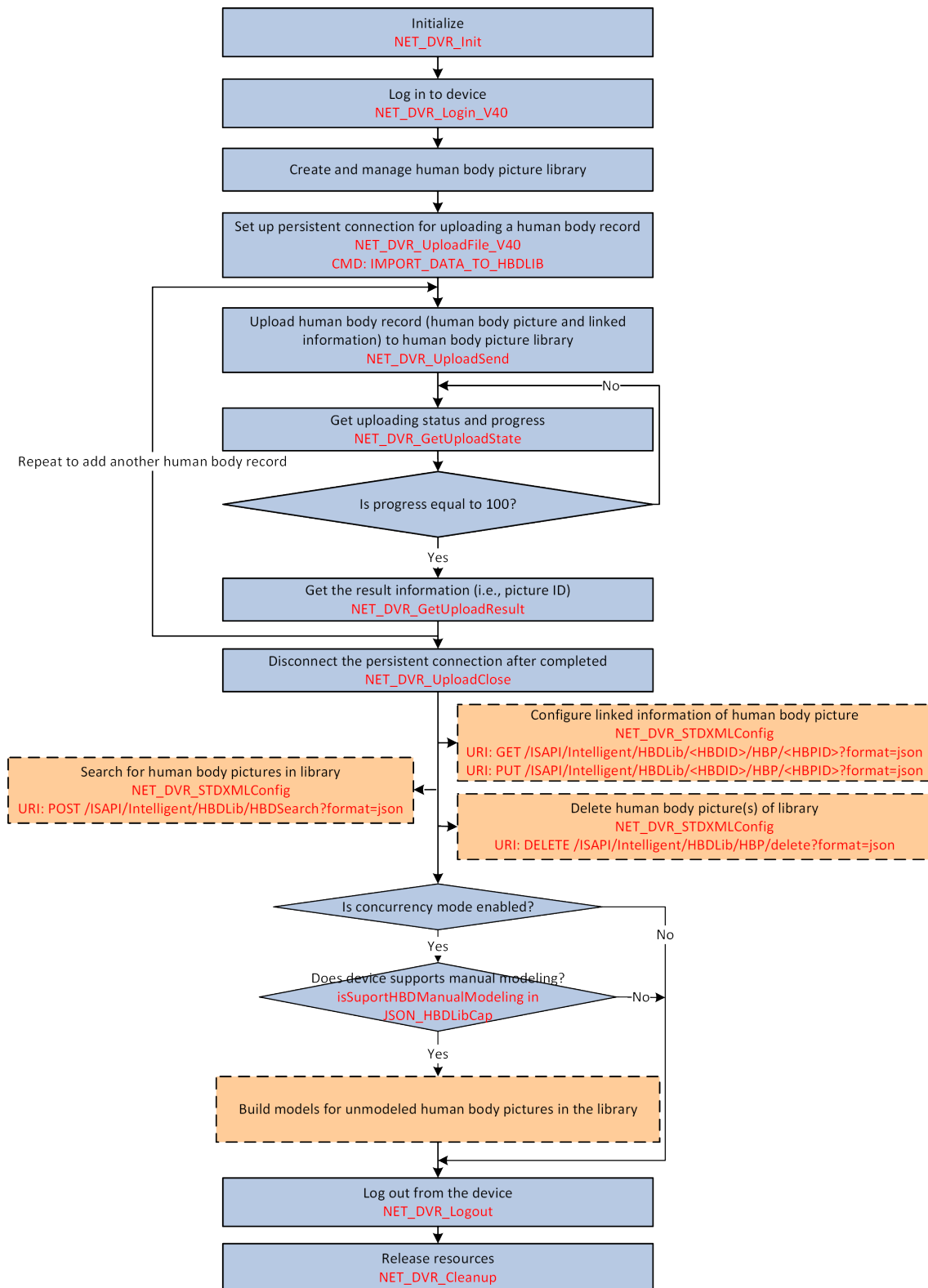


Figure 2-2 API Calling Flow of Managing Records in Human Body Picture Library

1. Call **NET_DVR_UploadFile_V40**, set **dwUploadType** to "IMPORT_DATA_TO_HBDLIB" (value: 60), and set **lpInBuffer** to the structure **NET_DVR_HBDLIB_COND** for setting up a persistent connection.

Note

- If the concurrency mode is enabled, when importing pictures to the library, the pictures will only be stored without modeling, which can fasten the importing speed. So that the pictures must be modeled manually. Enabling concurrency mode is only supported by front-end devices.
- If you want to model the imported human body pictures manually, you can set the value of **byConcurrent** in structure **NET_DVR_HBDLIB_COND** to 1 (enable).
- Only the pictures with JPEG format can be imported and the maximum picture size is 300 KB.

2. Add a human body record to human body picture library.

- 1) Call **NET_DVR_UploadSend** for starting uploading a human body record to human body picture library.
- 2) Call **NET_DVR_GetUploadState** for getting the uploading status and progress to check if the human body record is uploaded.

Note

Only when the returned **pProgress** equals to "1", it indicates uploading completed, and you can continue the following steps.

- 3) Call **NET_DVR_GetUploadResult** for getting the uploading result information.

The result information, i.e., uploaded picture ID (**PID**) is returned by **lpOutBuffer**.

Note

- You can repeat this step to add other human body records.
- Before adding human body records, you can call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Intelligent/HBDLib/<HBDID>/capacity?format=json** for getting the capacity of a human body picture library.

3. Call **NET_DVR_UploadClose** to disconnect the persistent connection after uploading completed.
4. **Optional:** Perform the following operation(s) for managing human body records in human body picture library.

Delete Human Body Picture(s) of Library

- Delete multiple human body pictures:
Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Intelligent/HBDLib/HBP/delete?format=json**
- Delete a single human body picture:
Call **NET_DVR_STDXMLConfig** to transmit the request URI: DELETE **/ISAPI/Intelligent/HBDLib/<HBDID>/HBP/<HBPID>?format=json**

Configure Linked Information of Human Body Picture

- Get linked information of a human body picture:
Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Intelligent/HBDLib/<HBDID>/HBP/<HBPID>?format=json**
- Edit linked information of a human body picture:

Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT /ISAPI/Intelligent/HBDLib/<HBDID>/HBP/<HBPID>?format=json

Search for Human Body Pictures in Library

Call **NET_DVR_STDXMLConfig** to transmit the request URI: POST /ISAPI/Intelligent/HBDLib/HBDSearch?format=json

5. Check whether the concurrent mode is enabled in step 1.
 - If the value of **byConcurrent** in structure **NET_DVR_HBDLIB_COND** is set to 1, perform the following steps to continue.
 - If the value of **byConcurrent** in structure **NET_DVR_HBDLIB_COND** is set to 0, the modeling will complete automatically and you can end the task.
6. Check whether the device supports manual modeling.
 - If the node **isSupportHBDManualModeling** exists in the configuration capability of human body picture library **JSON_HBDLibCap**, and its value is true, perform the following step.
 - Otherwise, end the task as manual modeling is not supported.
7. **Optional:** Perform the following operation(s) to build models manually if the concurrency mode is enabled.

Build Models in a Batch for Specified Human Body Picture Library

Call **NET_DVR_STDXMLConfig** to transmit the request URI: POST /ISAPI/Intelligent/HBDLib/manualModeling/<HBDID>?format=json

Build Models for Specified Human Body Picture

Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET /ISAPI/Intelligent/HBDLib/manualModeling/<HBDID>/HBP/<HBPID>?format=json

Build Models for Human Body Pictures of Different Libraries

Call **NET_DVR_STDXMLConfig** to transmit the request URI: POST /ISAPI/Intelligent/HBDLib/manualModelingRangeTask?format=json

Searching for Modeling Task Progress

Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET /ISAPI/Intelligent/HBDLib/manualModeling/progressState/<taskID>?format=json

Search for All Modeling Statuses

Call **NET_DVR_STDXMLConfig** to transmit the request URI: POST /ISAPI/Intelligent/HBDLib/modelingStatus?format=json

Get Modeling Status of a Human Body Picture

Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET /ISAPI/Intelligent/HBDLib/modelingStatus/<HBDID>/HBP/<HBPID>?format=json

What to do next

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

2.3 Configure Auto Person Arming

The auto person arming is to enable device to track and capture targets automatically according to the configured rules. You should configure face picture libraries, human body picture libraries, auto person arming rules, and linkage actions to realize device auto person arming.

Before You Start

- Make sure you have created and configure face picture libraries. Refer to *Configure Face Picture Library*.
- Make sure you have created and configured human body picture libraries. See *Create Human Body Picture Library* and *Manage Records in Human Body Picture Library* for details.

Steps

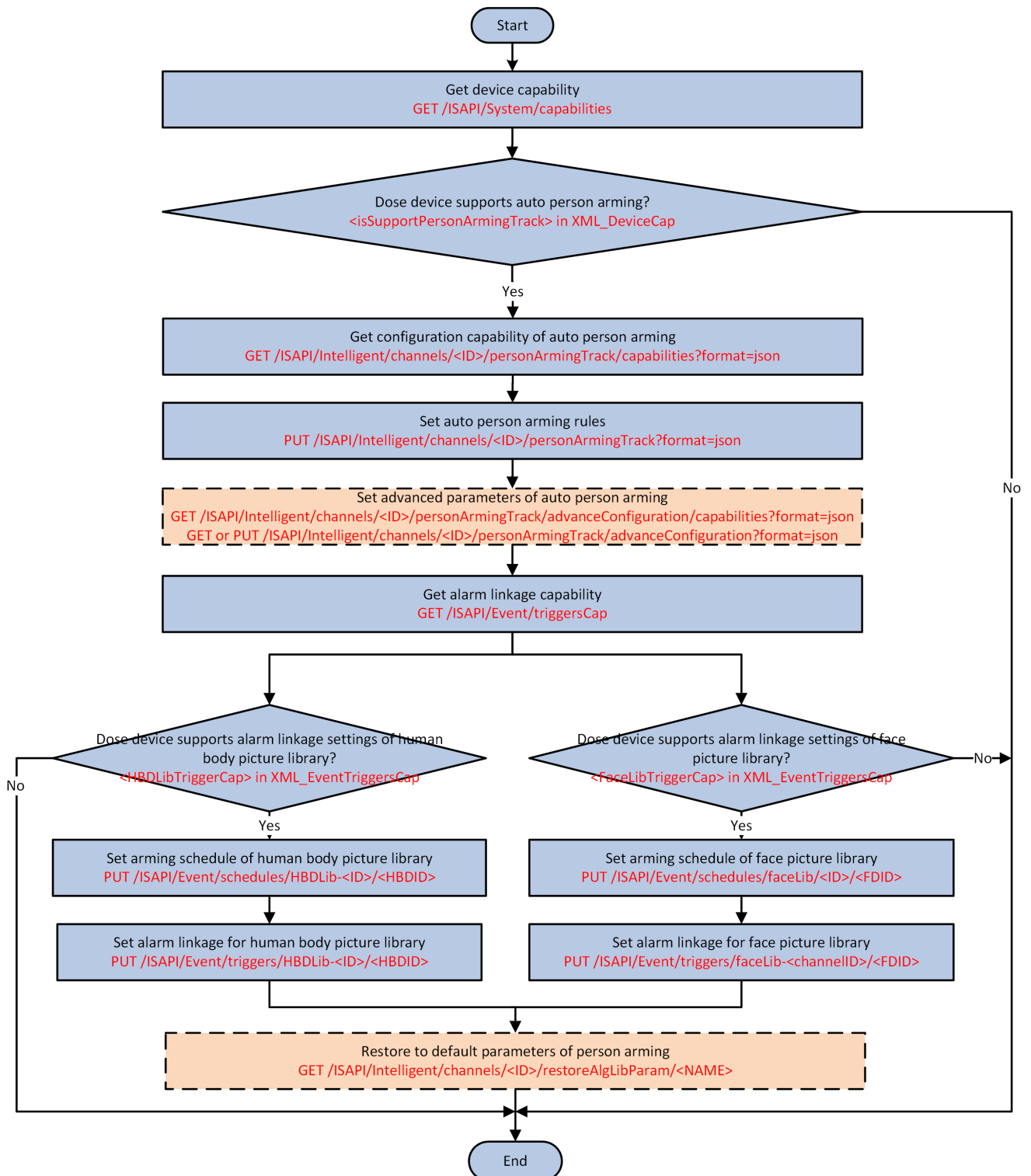


Figure 2-3 API Calling Flow of Configuring Auto Person Arming

Note

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

1. Call **/ISAPI/System/capabilities** by GET method to get the device capability to check whether the device supports auto person arming.
 - If the node **<isSupportPersonArmingTrack>** exists in the returned device capability **XML_DeviceCap**, and its value is true, perform the following steps to continue.
 - Otherwise, end this task as the auto person arming is not supported.
 2. Call **/ISAPI/Intelligent/channels/<ID>/personArmingTrack/capabilities?format=json** by GET method to get configuration capability of auto person arming.

The configuration capability is returned in the message **JSON_PersonArmingTrackCap**.
 3. Call **/ISAPI/Intelligent/channels/<ID>/personArmingTrack?format=json** by PUT method to set the auto person arming rules.
-

Note

Before setting the auto person arming rules, you'd better call above URI by GET method to get the existing configurations for reference.

4. **Optional:** Configure the advanced parameters of auto person arming.
-

Note

Before configuring advanced parameters, you should check whether the device supports the function. If the node **isSupportAdvanceConfiguration** exists in the capability **JSON_PersonArmingTrackCap** and its value is true, you can perform following steps.

- 1) Call **/ISAPI/Intelligent/channels/<ID>/personArmingTrack/advanceConfiguration/capabilities?format=json** by GET method to get the configuration capability of person arming advanced parameters.

The configuration capability is returned in the message **JSON_AdvanceConfigurationCap**.
 - 2) Call **/ISAPI/Intelligent/channels/<ID>/personArmingTrack/advanceConfiguration?format=json** by PUT method to set the person arming advanced parameters.
-

Note

Before setting, you'd better call the URI by GET method to get the existing configuration for reference.

5. Call **/ISAPI/Event/triggersCap** by GET method to get the linkage action capability to check whether human body picture library and face picture library support setting linkage action.
 - If the capability **XML_EventTriggersCap** is returned, and the nodes **<HBDLibTriggerCap>** and **<FaceLibTriggerCap>** exist in the capability, perform the following steps.
 - If getting capability **XML_EventTriggersCap** failed, or the nodes **<HBDLibTriggerCap>** and **<FaceLibTriggerCap>** do not exist in the capability, end the task as setting linkage actions for human body picture library and face picture library is not supported.
-

6. Perform the following operations to set arming schedule and linkage actions for human body picture library and face picture library.

Set Arming Schedule of Human Body Picture Library	PUT <u>/ISAPI/Event/schedules/HBDLib-<ID>/<HBDID></u>
Set Arming Schedule of Face Picture Library	PUT <u>/ISAPI/Event/schedules/faceLib/<ID>/<FDID></u>
Set Linkage Actions of Human Body Picture Library	PUT <u>/ISAPI/Event/triggers/HBDLib-<ID>/<HBDID></u>
Set Linkage Actions of Face Picture Library	PUT <u>/ISAPI/Event/triggers/faceLib-<channelID>/<FDID></u>

7. Optional: Call [/ISAPI/Intelligent/channels/<ID>/restoreAlgLibParam/<NAME>](#) by GET method to restore the algorithm library parameters to default.

2.4 Configure Manual Person Arming

During manual person arming, the device tracks and captures the manually selected target, and uploads the related data.



Note

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

Function	Request URI
Get manual person arming capability	GET <u>/ISAPI/Intelligent/channels/<ID>/manualPersonArmingTrack/capabilities?format=json</u>
Start manual person arming	PUT <u>/ISAPI/Intelligent/channels/<ID>/manualPersonArmingTrack?format=json</u>



Note

If device supports manual person arming, the node **[<isSupportManualPersonArmingTrack>](#)** exists in the device capability **[XML_DeviceCap](#)** (call [/ISAPI/System/capabilities](#) by GET method) and its value is true.

2.5 Receive Event or Alarm Details

Before receiving event or alarm details in arming mode or listening mode, you should configure the corresponding event or alarm parameters, and set the corresponding event type (**ICommand**) in message callback function.

Note

See **Receive Alarm/Event in Arming Mode** and **Receive Alarm/Event in Listening Mode** for receiving event or alarm details in arming or listening mode.

ICommand	Type Value	Description	Details
COMM_VCA_ALARM	0x4993	Auto person arming event	<u>JSON_EventNotificationAlert_personArmingTrackMsg</u>
		Manual person arming event	<u>JSON_EventNotificationAlert_manualPersonArmingTrackMsg</u>

2.5.1 Receive Alarm/Event in Arming Mode

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

Before You Start

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

Steps

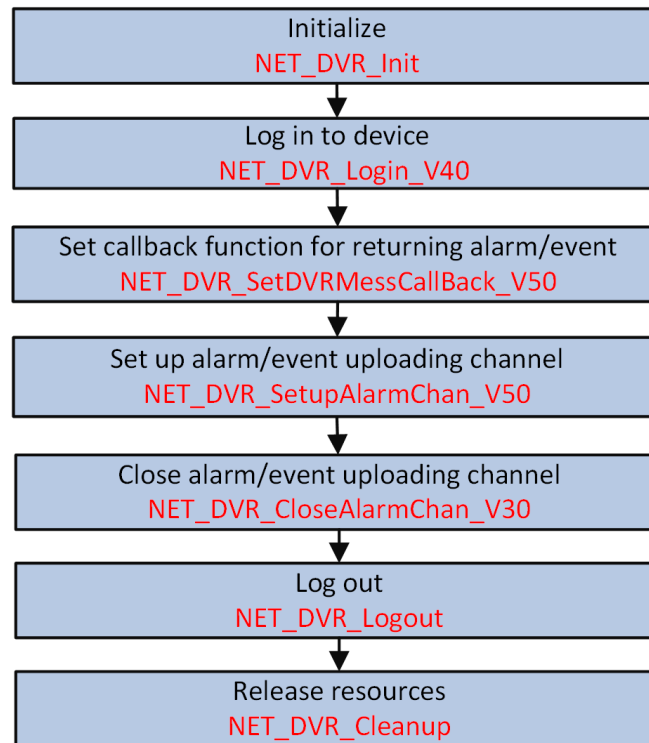


Figure 2-4 Programming Flow of Receiving Alarm/Event in Arming Mode

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



Note

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

2. Call **NET_DVR_SetupAlarmChan_V50** to set up uploading channel.
3. Call **NET_DVR_CloseAlarmChan_V30** to close uploading channel and stop receiving alarm or event information.

Example

Sample Code of Receiving Alarm or Event in Arming Mode

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

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

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

    //Enable arming
    NET_DVR_SETUPALARM_PARAM_V50 struSetupParamV50={0};
    struSetupParamV50.dwSize=sizeof(NET_DVR_SETUPALARM_PARAM_V50);
    //Alarm category to be uploaded
    struSetupParamV50.byAlarmInfoType=1;
    //Arming level
    struSetupParamV50.byLevel=1;

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

```
{
    //Arm
    lHandle = NET_DVR_SetupAlarmChan_V50(lUserID, &struSetupParamV50, NULL,
strlen(szSubscribe));
}
else
{
    //Subscribe
    LlHandle = NET_DVR_SetupAlarmChan_V50(lUserID, &struSetupParamV50,
szSubscribe, strlen(szSubscribe));
}

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

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

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

What to do next

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

2.5.2 Receive Alarm/Event in Listening Mode

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

Before You Start

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

Steps

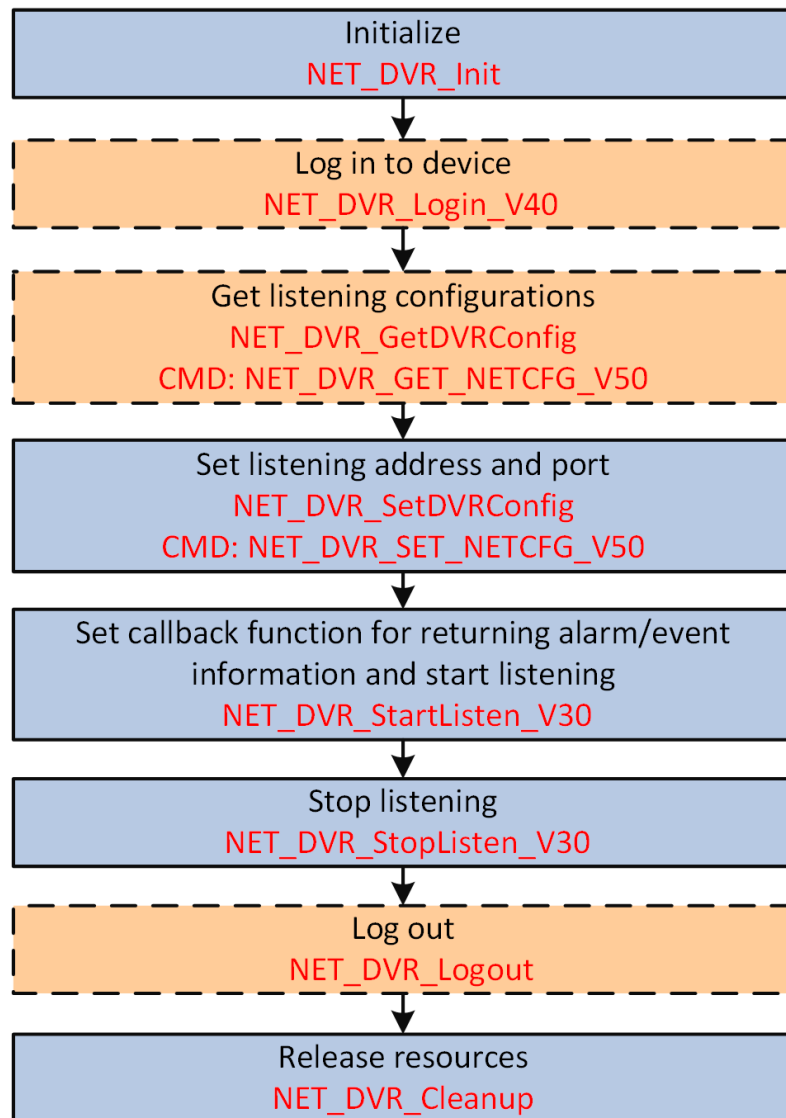


Figure 2-5 Programming Flow of Receiving Alarm/Event in Listening Mode

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.

2.6 Integrate by Transmitting Text Protocol

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

Before You Start

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

Steps

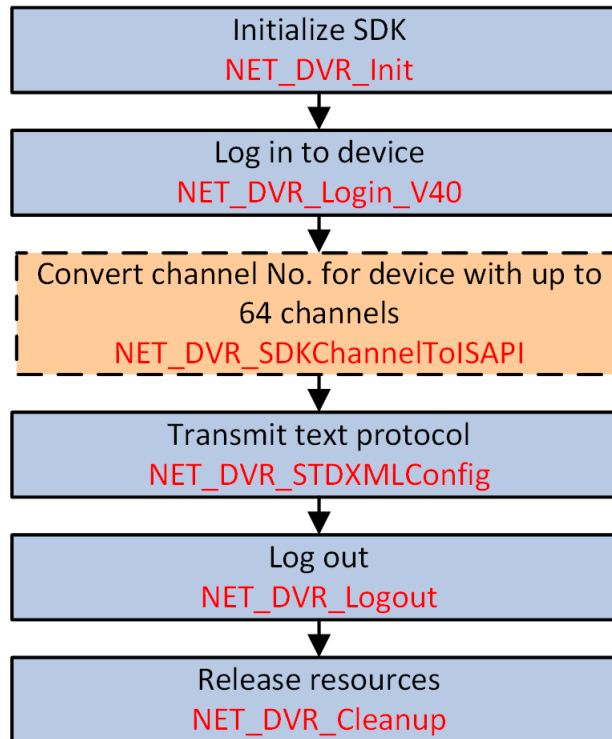


Figure 2-6 API Calling Flow of Integrating by Transmitting Text Protocol

1. **Optional:** Call **NET_DVR_SDKChannelToISAPI** to convert the device channel No. when integrating based on Device Network SDK and text protocol transmission.

Note

- This step is only available for rear-end devices with up to 64 network channels.
 - For the integration based on text protocol transmission, the channel No. starts from 1, for the integration based on Device Network SDK, the channel No. of device with up to 64 network channels starts from 33, so when the SDK's API is called by the platform or system for transmitting text protocol to device, the channel No. returned by device starts from 1, but the start channel No. of platform or system should starts from 33, this may cause the problem.
-
2. Call **NET_DVR_STDXMLConfig** to transmit text protocol, including operation methods, request URIs, query parameters, and request or response messages, for realizing the corresponding applications.

What to do next

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

Chapter 3 API Reference

3.1 NET_DVR_Cleanup

Release the resources after the program is ended.

API Definition

```
BOOL NET_DVR_Cleanup(  
);
```

Return Values

Returns *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.

3.2 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** .

3.3 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**](#)

3.4 NET_DVR_GetErrorMsg

Return the error information of the last operation.

API Definition

```
char *NET_DVR_GetErrorMsg(  
    LONG    *pErrorNo  
) ;
```

Parameters

pErrorNo

[OUT] Error code pointer.

Return Values

The return values are the pointers of error information, see [**Device Network SDK Errors**](#) for details.

Remarks

You can call [**NET_DVR_GetLastError**](#) to get the error codes.

3.5 NET_DVR_GetLastError

Return the error code of the last operation.

API Definition

```
DWORD NET_DVR_GetLastError(  
) ;
```

Return Values

The return values are error codes, see [**Device Network SDK Errors**](#) for details.

Remarks

You can also call [**NET_DVR_GetErrorMsg**](#) to directly get the error information.

3.6 NET_DVR_GetUploadResult

Get the uploaded data information.

API Definition

```
BOOL NET_DVR_GetUploadResult(  
    LONG        lUploadHandle,  
    LPVOID      lpOutBuffer,  
    DWORD       dwOutBufferSize  
);
```

Parameters

lUploadHandle

[IN] Handling for uploading files, which is returned by [NET_DVR_UploadFile_V40](#).

lpOutBuffer

[OUT] Buffer of uploaded data information, different uploading types (**dwUploadType**) correspond to different information.

dwOutBufferSize

[OUT] Buffer size

Return Value

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

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

3.7 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 3-1 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.
34	Incorrect security verification key.
35	Downloading picture via URL has not started.
36	Duplicate custom ID of different persons.

Return Value	Description
37	Person ID error (The ID is saved in customHumanID of FaceAppendData).
38	Modeling failed. Device inner error.
39	Modeling failed. Face modeling error.
40	Modeling failed. Face score error.
41	Modeling failed. Feature collection error.
42	Modeling failed. Attribute collection error.
43	Picture data error.
44	Picture additional information error.
45	Certificate has already existed.

3.8 NET_DVR_Init

Initialize the programming environment before calling other APIs.

API Definition

```
BOOL NET_DVR_Init(
);
```

Return Values

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

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

The available error codes of this API are 0, 41, and 53. See details in **Device Network SDK Errors** .

Remarks

Before initializing, you can call **NET_DVR_SetSDKInitCfg** to set the initialization parameters, such as supported capabilities, loading path of component libraries (only supported by Linux system), and so on.

See Also

NET_DVR_Cleanup

3.9 NET_DVR_Login_V40

Log in to the device (supports asynchronous login).

API Definition

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

Parameters

pLoginInfo

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

lpDeviceInfo

[OUT] Device information. See details in the structure **NET_DVR_DEVICEINFO_V40**.

Return Values

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

Remarks

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

See Also

NET_DVR_Logout

3.10 NET_DVR_Logout

Log out from devices.

API Definitions

```
BOOL NET_DVR_Logout (
    LONG    lUserID
);
```

Parameters

lUserID

[IN] User ID, which is returned by [***NET_DVR_Login_V40***](#) .

Return Values

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

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

The available error codes may be returned by this API are 0, 3, 7, 8, 9, 10, 14, 17, 41, 44, 47, 72, and 73. See details in [***Device Network SDK Errors***](#) .

3.11 NET_DVR_SDKChannelToISAPI

Convert channel No. between the private protocol and a text protocol.

API Definition

```
BOOL NET_DVR_SDKChannelToISAPI(  
    LONG    lUserID,  
    LONG    lInChannel,  
    BOOL    bSDKToISAPI  
);
```

Parameters

lUserID

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

lInChannel

[IN] Channel No.

bSDKToISAPI

[OUT] Channel No. conversion type: "TRUE"-convert channel No. of private protocol to that of text protocol, "FALSE"-convert channel No. of text protocol to that of private protocol.

Return Values

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

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

3.12 NET_DVR_SetDVRConfig

Set the device parameters.

API Definition

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

Parameters

lUserID

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

dwCommand

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

lChannel

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

lpInBuffer

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

dwInBufferSize

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

Return Values

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

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

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

See Also

[NET_DVR_GetDVRConfig](#)

3.13 NET_DVR_SetDVRMessageCallBack_V50

Set callback functions for getting the video data.

API Definition

```
BOOL NET_DVR_SetDVRMessageCallBack_V50(  
    int            iIndex,
```

```
MSGCallback    fMessageCallback,  
void           *pUser  
);
```

Parameters

iIndex

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

fMessageCallback

[IN] Callback function, see details in [***MSGCallback***](#).

pUser

[IN] User data.

Return Values

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

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

Remarks

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

Example

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

```
#include <stdio.h>  
#include <iostream>  
#include "Windows.h"  
#include "HCNetSDK.h"  
using namespace std;  
  
int iNum=0;  
void CALLBACK MessageCallbackNo1(LONG lCommand, NET_DVR_ALARMER *pAlarmer, char  
*pAlarmInfo, DWORD dwBufLen, void* pUser)  
{  
    int i=0;  
    char filename[100];  
    FILE *fSnapPic=NULL;  
    FILE *fSnapPicPlate=NULL;  
  
    //This sample code is for reference only. Actually, it is not recommended  
    to process the data and save file in the callback function directly.  
    //You'd better process the data in the message response function via message  
    mode (PostMessage).
```

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

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

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

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

```
(struITSPlateResult.struPicInfo[i].byType == 2))
{
    sprintf(filename, "testITSpic%d_%d.jpg", iNum, i);
    fSnapPic=fopen(filename, "wb");
    fwrite(struITSPlateResult.struPicInfo[i].pBuffer,
struITSPlateResult.struPicInfo[i].dwDataLen, 1, fSnapPic);
    iNum++;
    fclose(fSnapPic);
}
//License plate thumbnails
if ((struITSPlateResult.struPicInfo[i].dwDataLen !=
0)&&(struITSPlateResult.struPicInfo[i].byType == 0))
{
    sprintf(filename, "testPicPlate%d_%d.jpg", iNum, i);
    fSnapPicPlate=fopen(filename, "wb");
    fwrite(struITSPlateResult.struPicInfo[i].pBuffer,
struITSPlateResult.struPicInfo[i].dwDataLen, 1, \ fSnapPicPlate);
    iNum++;
    fclose(fSnapPicPlate);
}
//Processing other data...
}
break;
}
default:
    break;
}
}

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

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

    switch(lCommand)
    {
        case COMM_ALARM:
        {
            NET_DVR_ALARMINFO struAlarmInfo;
            memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_DVR_ALARMINFO));
            switch (struAlarmInfo.dwAlarmType)
            {
                case 3: //Motion detection alarm
                    for (i=0; i<16; i++) //define MAX_CHANNUM 16 //The
```

```
maximum number of channel
{
    if (struAlarmInfo.dwChannel[i] == 1)
    {
        printf("Channel No. with Motion Detection Alarm %d\n", i+1);
    }
    break;
default:
    break;
}
break;
}
case COMM_UPLOAD_PLATE_RESULT:
{
    NET_DVR_PLATE_RESULT struPlateResult={0};
    memcpy(&struPlateResult, pAlarmInfo, sizeof(struPlateResult));
    printf("License Plate Number: %s\n",
struPlateResult.struPlateInfo.sLicense); //License plate number

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



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

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

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

void main() {

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

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

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

    //Enable arming
    NET_DVR_SETUPALARM_PARAM struSetupParam={0};
    struSetupParam.dwSize=sizeof(NET_DVR_SETUPALARM_PARAM);

    //Alarm information type to upload: 0-History Alarm (NET_DVR_PLATE_RESULT), 1-
Real-Time Alarm (NET_ITS_PLATE_RESULT)
    struSetupParam.byAlarmInfoType=1;
    //Arming Level: Level-2 arming (for traffic device)
```

```
struSetupParam.byLevel=1;

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

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

//User logout
NET_DVR_Logout(lUserID);
//Release SDK resource
NET_DVR_Cleanup();
return;
}
```

See Also

NET_DVR_SetupAlarmChan_V50

3.14 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 3-2 NET_SDK_INIT_CFG_TYPE

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

IpInBuff

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

Return Values

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

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

Remarks

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

3.15 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**.

3.16 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  
MSGCallback  
void  
) ;  
  
wLocalPort,  
DataCallback,  
*pUserData
```

Parameters

sLocalIP

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

wLocalPort

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

DataCallback

[IN] Alarm/event information callback function, see details in [***MSGCallback***](#) .

pUserData

[IN] User data.

Return Values

Return -1 for failure, and return other values for the handle parameters of

[***NET_DVR_StopListen_V30***](#) .

If -1 is returned, you can call [***NET_DVR_GetLastError***](#) to get the error code.

The available error codes of this API are 0, 3, 6, 12, 17, 41, 44, 47, 72, and 75. See details in the [***Device Network SDK Errors***](#) .

Remarks

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

3.17 NET_DVR_STDXMLConfig

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

API Definition

```
BOOL NET_DVR_STDXMLConfig(  
    LONG  
    const NET_DVR_XML_CONFIG_INPUT  
  
    lUserID,  
    *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 <u>NET_DVR_XML_CONFIG_INPUT</u>)	Method+URL E.g., GET /ISAPI/System/capabilities
	lpInBuffer (see in structure <u>NET_DVR_XML_CONFIG_INPUT</u>)	Request Message
lpOutputParam	lpOutBuffer (see in structure <u>NET_DVR_XML_CONFIG_OUTPUT</u>)	Response Message
	lpStatusBuffer (see in structure <u>NET_DVR_XML_CONFIG_OUTPUT</u>)	Response Message

3.18 NET_DVR_StopListen_V30

Stop listening (supports multiple threads).

API Definition

```
BOOL NET_DVR_StopListen_V30(  
    LONG    lListenHandle  
) ;
```

Parameters

lListenHandle

Listening handle, which is returned by [NET_DVR_StartListen_V30](#) .

Return Values

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

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

The available error codes of this API are 0, 3, 12, and 17. See details in the [Device Network SDK Errors](#) .

3.19 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.

3.20 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

IUserID

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

dwUploadType

[IN] Uploading commands, which specify the file type to upload, see details in the enumeration [NET_SDK_UPLOAD_TYPE](#) .

lpInBuffer

[IN] Input parameters, which are different according to different uploading commands.

dwInBufferSize

[IN] Input buffer size.

sFileName

[IN] Name of the file to be uploaded. For the complete file path (including the file name), the maximum size is 128 bytes, and the maximum size of the file name is 32 bytes.

lpOutBuffer

[OUT] Output parameters, which are different according to different uploading commands.

dwOutBufferSize

[OUT] Output buffer size.

Return Values

Return -1 for failure, and return other values as the parameter of [NET_DVR_UploadClose](#) and [NET_DVR_GetUploadState](#) .

If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

3.21 NET_DVR_UploadSend

Send data.

API Definition

```
LONG NET_DVR_UploadSend(  
    LONG                lUploadHandle,  
    LPNET_DVR_SEND_PARAM_IN pstruSendParamIN,
```

```
void                                     *lpOutBuffer
);
```

Parameters

IUploadHandle

[IN] File uploading handle, returned by **NET_DVR_UploadFile_V40**.

pstruSendParamIN

[IN] Uploaded data. See the structure **NET_DVR_SEND_PARAM_IN** for details.

lpOutBuffer

[OUT] Output parameter, reserved, set to NULL

Return Value

Return 0 for success, and return -1 for failure.

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

3.22 Callback Function

3.22.1 fLoginResultCallBack

Login Status Callback Function

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

3.22.2 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.

Appendix A. Data Structure

A.1 NET_ALARM_CVR_SUBINFO_UNION

Union about CVR Alarm Information

Member	Data Type	Description
byLen	BYTE[]	Union size, the maximum array length is 492 bytes.
struRecordLost	<u>NET_ALARM_RECORD_FILE_LOSS</u>	Video loss alarm information, the value of dwAlarmType in <u>NET_DVR_ALARMINFO_DEV_V40</u> is 8.
struStreamException	<u>NET_ALARM_STREAM_EXCEPTION</u>	Streaming exception alarm information, the value of dwAlarmType in <u>NET_DVR_ALARMINFO_DEV_V40</u> is 9.
struResourceUsage	<u>NET_ALARM_RESOURCE_USAGE</u>	Resource usage alarm information, the value of dwAlarmType in <u>NET_DVR_ALARMINFO_DEV_V40</u> is 10.
struRecordException	<u>NET_ALARM_RECORD_EXCEPTION</u>	Recording exception alarm information, the value of dwAlarmType in <u>NET_DVR_ALARMINFO_DEV_V40</u> is 12.

A.2 NET_ALARM_RECORDFILE_LOSS

Structure about Video Loss Alarm Information

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

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

A.3 NET_ALARM_RECORD_EXCEPTION

Structure about Recording Exception Alarm Information

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

A.4 NET_ALARM_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.5 NET_ALARM_STREAM_EXCEPTION

Structure about Video Exception Alarm Information

Member	Data Type	Description
struIP	<u>NET_DVR_IPADDR_UNION</u>	IP address of video exception channel.
dwChanNo	DWORD	Channel No.
dwIDIndex	DWORD	Encoder ID.
sName	BYTE[]	Encoder name, the maximum array length is "STREAM_ID_LEN" (32 bytes).
byExceptionCase	BYTE	Exception reason: 0-data writing exception, 1-network exception.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 307 bytes.

A.6 NET_DVR_ALARM_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,

Member	Data Type	Description
		8-invalid login, 9-video exception, 10-recording exception, 11-scene change, 12-RAID exception, 13-resolution mismatched, 15-VCA detection, 16- PoE power supply exception, 17-education sharing system alarm, 18-two-way audio request alarm, 23-pulse alarm, 24-face picture library HDD exception, 25-face picture library changed, 26-picture of face picture library changed, 27-POC exception, 28-camera FOV exception, 30-no SD card, 31-supply voltage exception, 32-PTZ locked
struAlarmTime	<u>NET_DVR_TIME_EX</u>	Alarm time
uStruAlarm	Union (<u>Table 4-1</u>)	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-1 Union about Alarm Information Structures (uStruAlarm)

Member	Data Type	Description
byUnionLen	Array of BYTE	Union size, which is 116 bytes.
struIOAlarm	Struct (<u>Table 4-2</u>)	Structure about alarm input parameters
struAlarmChannel	Struct (<u>Table 4-3</u>)	Structure about alarm channel parameters
struAlarmHardDisk	Struct (<u>Table 4-4</u>)	Structure about HDD alarm parameters
struRecordingHost	Struct (<u>Table 4-5</u>)	Structure about alarm parameters of education sharing system

Member	Data Type	Description
struVoltageInstable	Struct (Table 4-6)	Structure about alarm parameters of supply voltage exception
struPTLocking	Struct (Table 4-7)	Structure about parameters of PTZ locked alarm

Table A-2 Structure about Alarm Input Parameters (struIOAlarm)

Member	Data Type	Description
dwAlarmInputNo	DWORD	Alarm input No.
dwTrigerAlarmOutNum	DWORD	The number of triggered alarm outputs. It is used for calculating the number of all triggered alarm outputs by pAlarmData in NET_DVR_ALARMINFO_V40 , each alarm output is represented by 4 bytes.
dwTrigerRecordChanNum	DWORD	The number of triggered recording channels. It is used for calculating the number of all triggered recording channels by pAlarmData of NET_DVR_ALARMINFO_V40 , each channel is represented by 4 bytes.

Table A-3 Structure about Alarm Channel Parameters (struAlarmChannel)

Member	Data Type	Description
dwAlarmChanNum	DWORD	The number of alarm channels. It is used for calculating the number of all alarm channels by pAlarmData of 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

Member	Data Type	Description
		operating system and 64-bit Linux operating system.

Table A-4 Structure about HDD Alarm Parameters (struAlarmHardDisk)

Member	Data Type	Description
dwAlarmHardDiskNum	DWORD	The number of alarm HDD. It is used for calculating the number of all alarm HDDs by pAlarmData of <i>NET DVR ALARMINFO V40</i> , each alarm HDD is represented by 4 bytes.

Table A-5 Structure about Alarm Parameters of Education Sharing System (struRecordingHost)

Member	Data Type	Description
bySubAlarmType	BYTE	Alarm minor type: 1-one-touch post-record
byRes1	Array of BYTE	Reserved, set to 0. The maximum size is 3 bytes.
struRecordEndTime	<i>NET DVR TIME EX</i>	Recording end time.

Table A-6 Structure about Alarm Parameters of Supply Voltage Exception (struVoltageInstable)

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

Table A-7 Structure about Parameters of PTZ Locked Alarm (struPTLocking)

Member	Data Type	Description
fTemperature	float	Sensor temperature, which is accurate to one decimal place.
dwCustomInfoLength	DWORD	Custom information length.
pCustomInfo	BYTE*	Custom information.

Member	Data Type	Description
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.7 NET_DVR_ALARM_ISAPI_INFO

Structure about Alarm Information Transmitted Based on Text Protocol

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

Member	Data Type	Description
pPicPackData	void*	Alarm picture structure, see <u>NET_DVR_ALARM_ISAPI_PICDATA</u> for details.
byRes	Array of BYTE	Reserved. The maximum size is 32 bytes.

Remarks

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

A.8 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.9 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

Memeber	Data Type	Description
		recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel.
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.10 NET_DVR_ALARMINFO_DEV_V40

Structure about CVR Alarm Information

Member	Data Type	Description
dwAlarmType	DWORD	Alarm categories: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel, 8-video loss alarm, 9-real-time health monitoring alarm, 10-usage alarm, 11-CVR exception recovered, 12-recording exception.
struTime	<u>NET_DVR_TIME</u>	Alarm time
uSubAlarmInfo	<u>NET_ALARM_CVR_SUB_ALARM_INFO_UNION</u>	CVR alarm information structure, and it is valid when the alarm type is 8, 9, 10, and 12.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 256 bytes.

Member	Data Type	Description
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.11 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. dwAlarmRelateChannel[0]==1 indicates that the channel No.1 is triggered to record.

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

Remarks

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

A.12 NET_DVR_ALARMINFO_V40

Structure About Uploaded Alarm Information

Member	Data Type	Description
struAlarmFixedHeader	<u>NET_DVR_ALRAM_FIX ED_HEADER</u>	Constant content in alarm information, see details in the structure .
pAlarmData	DWORD*	Variable content in alarm information

Remarks

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

Table A-8 Relations Between pAlarmData and dwAlarmType

dwAlarmType	Description	pAlarmData
0, 23	Alarm input alarm, pulse alarm	dwTrigerAlarmOutNum*(DWOR D) 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.13 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

Member	Data Type	Description
IUserID	LONG	Value returned by <i>NET DVR Login V40</i> , 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.14 NET_DVR_DEVICEINFO_V30

Device parameter structure (V30).

Device Parameter Structure (V30)

Member	Data Type	Description
sSerialNumber	BYTE	Device serial No.
byAlarmInPortNum	BYTE	Number of analog alarm inputs
byAlarmOutPortNum	BYTE	Number of analog alarm outputs
byDiskNum	BYTE	Number of HDDs
byDVRType	BYTE	Device type
byChanNum	BYTE	Number of analog channels
byStartChan	BYTE	Start No. of analog channel, which starts from 1.
byAudioChanNum	BYTE	Number of two-way audio channels
byIPChanNum	BYTE	Number of digital channels, low 8-bit.

Member	Data Type	Description
byZeroChanNum	BYTE	Number of channel-zero
byMainProto	BYTE	Transmission protocol type of main stream: 0-private protocol (default), 1-RTSP, 2-private protocol+RTSP
bySubProto	BYTE	Transmission protocol type of sub-stream: 0-private protocol (default), 1-RTSP, 2-private protocol+RTSP
bySupport	BYTE	<p>Capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> 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	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> bySupport1&0x1: whether supports SNMP with version 30. bySupport1&0x2: whether supports playback and downloading video files. bySupport1&0x4: whether supports setting the arming priority. bySupport1&0x8: whether supports extending the arming time period.

Member	Data Type	Description
		<ul style="list-style-type: none"> bySupport1&0x10: whether supports multiple HDDs (more than 33). bySupport1&0x20: whether supports RTP over RTSP. bySupport1&0x80: whether supports license plate recognition alarm.
bySupport2	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> bySupport2&0x1: whether supports getting stream via URL. bySupport2&0x2: whether supports FTP with version 40. bySupport2&0x4: whether supports ANR. bySupport2&0x20: whether supports getting device status. bySupport2&0x40: whether supports encrypting stream.
wDevType	WORD	Device model
bySupport3	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, while, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> bySupport3&0x1: whether supports multi-stream. bySupport3&0x4: whether supports configuring by group (e.g., image, alarm input, alarm output, user, device status, JPEG picture capture, continuous and scheduled capture, .HDD group management, and so on). bySupport3&0x20: whether supports getting stream via DDNS.
byMultiStreamProto	BYTE	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.

Member	Data Type	Description
		<ul style="list-style-type: none"> byMultiStreamProto&0x1: whether supports third-stream. byMultiStreamProto&0x2: whether supports fourth-stream. byMultiStreamProto&0x40: whether supports main stream. byMultiStreamProto&0x80: whether supports sub-stream.
byStartDChan	BYTE	Start No. of digital channel, 0-no digital channel (e.g., DVR, network camera).
byStartDTalkChan	BYTE	Start No. of two-way audio channel, 0-no two-way audio channel.
byHighDChanNum	BYTE	Number of digital channels, high 8-bit.
bySupport4	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> bySupport4&0x01: whether all stream types support RTSP and private protocol. bySupport4&0x02: whether the device supports transmitting form format data via API (NET_DVR_STDXMLConfig). bySupport4&0x10: whether supports loading network disk by domain name.
byLanguageType	BYTE	<p>Supported language types, if the result of bitwise operation is 0, it refers to not support, if the result is 1, it refers to support.</p> <ul style="list-style-type: none"> byLanguageType ==0: this field is not supported by device. byLanguageType&0x1: whether supports Chinese. byLanguageType&0x2: whether supports English.
byVoiceInChanNum	BYTE	Number of audio input channels
byStartVoiceInChanNo	BYTE	Start No. of audio input channel, 0-invalid.
byRes3	Array of BYTE	Reserved, set to 0.

Member	Data Type	Description
byMirrorChanNum	BYTE	Number of mirror channels
wStartMirrorChanNo	WORD	Start No. of mirror channel
byRes2	Array of BYTE	Reserved, set to 0.

Remarks

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

See Also

NET_DVR_DEVICEINFO_V40

A.15 NET_DVR_DEVICEINFO_V40

Device Parameter Structure (V40)

Member	Data Type	Description
struDeviceV30	<u>NET_DVR_DEVICEINFO_V30</u>	Device parameters
bySupportLock	BYTE	Whether supports locking function: 1-support.
byRetryLoginTime	BYTE	Remaining login attempts, it is valid when the user name or password is incorrect and the bySupportLock is 1.
byPasswordLevel	BYTE	Password strength: 0-invalid, 1-default password, 2-valid password, 3-risky password. For default password or risky password, the users are reminded to change password.
byProxyType	BYTE	Proxy type: 0-no proxy, 1-standard proxy, 2-EHome proxy.
dwSurplusLockTime	DWORD	Remaining locking time, unit: second. It is valid only when bySupportLock is 1. During the locking time, if the user try to log in to again, the

Member	Data Type	Description
		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 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

Member	Data Type	Description
		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.16 NET_DVR_ETHERNET_V30

Ethernet Configuration Structure

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

A.17 NET_DVR_HBDLIB_COND

Structure about Condition of Importing Human Body Picture Data

Member	Data Type	Description
dwSize	DWORD	Structure size
szHBDID	char	ID of human body picture library, the maximum size is 256 bytes (macro definition value: "NET_SDK_MAX_HBDID_LEN")
byConcurrent	BYTE	Concurrent processing: 0-Disable (automatic modeling is available), 1-Enable (automatic modeling is not available)
byCover	BYTE	Whether to support overwriting when importing (when the human body picture library is full, the earliest picture will be overwritten): 0-No, 1-Yes
byCustomHBDID	Array of BYTE	Whether ID of human body picture library is custom: 0-No, 1-Yes
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 125 bytes.

A.18 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: <div>enum _INIT_CFG_MAX_NUM_{ INIT_CFG_NUM_2048 = 2048,</div>

Member	Data Type	Description
		<pre>INIT_CFG_NUM_5120 = 5120, INIT_CFG_NUM_10240 = 10240, INIT_CFG_NUM_15360 = 15360, INIT_CFG_NUM_20480 = 20480 }INIT_CFG_MAX_NUM</pre>
enumMaxAlarmNum	INIT_CFG_MAX_NUM	<p>Maximum number of alarm channels, see details below:</p> <pre>enum _INIT_CFG_MAX_NUM_{ INIT_CFG_NUM_2048 = 2048, INIT_CFG_NUM_5120 = 5120, INIT_CFG_NUM_10240 = 10240, INIT_CFG_NUM_15360 = 15360, INIT_CFG_NUM_20480 = 20480 }INIT_CFG_MAX_NUM</pre>
byRes	Array of BYTE	Reserved, set to 0.

Remarks

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

See Also

NET_DVR_SetSDKInitCfg

A.19 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.20 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.21 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.22 NET_DVR_NETCFG_V50

Network Configuration Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
struEtherNet	Array of <u>NET_DVR_ETHERNET_V30</u>	Ethernet interface
struRes1	Array of	Reserved, set to 0.
struAlarmHostIpAddr	<u>NET_DVR_IPADDR_UNION</u>	Listening service IP address
byRes2	Array of BYTE	Reserved, set as 0
wAlarmHostIpPort	WORD	Listening service port No.
byUseDhcp	BYTE	Whether to enable DHCP: 0xff- invalid; 0-disable, 1-enable
byIPv6Mode	BYTE	Allocation mode of IPv6 address: 0-by router advertisement, 1-by manual setting, 2-by enabling DHCP allocation.
struDnsServer1IpAddr	<u>NET_DVR_IPADDR_UNION</u>	IP address of domain name server 1
struDnsServer2IpAddr	<u>NET_DVR_IPADDR_UNION</u>	IP address of domain name server 2
byIpResolver	Array of BYTE	IP resolver domain name or IP address (if the port No. of device is 8000, the domain name is not supported).
wIpResolverPort	WORD	IP resolver port No.
wHttpPortNo	WORD	HTTP port No.
struMulticastIpAddr	<u>NET_DVR_IPADDR_UNION</u>	Multicast group address
struGatewayIpAddr	<u>NET_DVR_IPADDR_UNION</u>	Gateway address
struPPPoE	<u>NET_DVR_PPPOECFG</u>	PPPoE parameters

Member	Data Type	Description
byEnablePrivateMulticastDiscovery	BYTE	Private multicast search (SADP): 0-default, 1-enable, 2-disable
byEnableOnvifMulticastDiscovery	BYTE	Onvif multicast search (SADP): 0-default, 1-enable, 2-disable
wAlarmHost2IpPort	WORD	Port No. of listening host 2.
struAlarmHost2IpAddr	<u>NET_DVR_IPADDR_UNION</u>	IP address of listening host 2
byEnableDNS	BYTE	DNS address setting mode: 0-automatically get, 1-manually set.
byRes	Array of BYTE	Reserved, set to 0

Remarks

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

A.23 NET_DVR_PPPOECFG

PPPoE Configuration Structure

Member	Data Type	Description
dwPPPOE	DWORD	Whether to enable PPPoE: 0-no, 1-yes.
sPPPoEUser	Array of BYTE	PPPoE user name.
sPPPoEPassword	Array of char	PPPoE password.
struPPPoEIP	<u>NET_DVR_IPADDR_UNION</u>	PPPoE IP address

A.24 NET_DVR_SEND_PARAM_IN

Structure about input parameters for sending data.

Structure Definition

```
struct{
    BYTE                *pSendData;
    DWORD              dwSendDataLen;
    NET_DVR_TIME_V30   struTime;
    BYTE               byPicType;
    BYTE               byPicURL;
    BYTE               byRes1[2];
    DWORD              dwPicMangeNo;
    BYTE               sPicName[NAME_LEN/*32*/];
    DWORD              dwPicDisplayTime;
    BYTE               *pSendAppendData;
    DWORD              dwSendAppendDataLen;
    BYTE               byRes[192];
}NET_DVR_SEND_PARAM_IN, *LPNET_DVR_SEND_PARAM_IN;
```

Members

ipSendData

Picture buffer

dwSendDataLen

Picture data size

struTime

Time displayed on the picture

byPicType

Picture format: 1-JPEG, 2-BMP, 3-PNG

byPicURL

Picture data mode: 0-binary, 1-URL

byRes1

Reserved, set to 0

dwPicMangeNo

Picture management No.

sPicName

Picture name

dwPicDisplayTime

Picture display time, unit: second

pSendAppendData

Additional information buffer of sent picture, set it to NULL if not supported

dwSendAppendDataLen

Additional information data length of sent picture, set it to 0 if not supported

byRes

Reserved

A.25 NET_DVR_SETUPALARM_PARAM_V50

Arming Parameter Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
byLevel	BYTE	Arming priority: 0-high, 1-medium, 2-low.
byAlarmInfoType	BYTE	Intelligent traffic alarm information type: 0-old (NET_DVR_PLATE_RESULT),1-new (NET_ITS_PLATE_RESULT).
byRetAlarmTypeV40	BYTE	0-the motion detection, video loss, video tampering, and alarm input alarm information is uploaded in normal mode (alarm type: COMM_ALARM_V30, alarm information structure: <u>NET_DVR_ALARMINFO_V30</u>); 1-alarm information is uploaded in variable size (alarm type: COMM_ALARM_V40, alarm information structure: <u>NET_DVR_ALARMINFO_V40</u>).
byRetDevInfoVersion	BYTE	Alarm types of CVR: 0-COMM_ALARM_DEVICE (alarm information structure: <u>NET_DVR_ALARMINFO_DEV</u>), 1-COMM_ALARM_DEVICE_V40 (alarm information structure: <u>NET_DVR_ALARMINFO_DEV_V40</u>).
byRetVQDAlarmType	BYTE	VQD alarm types: 0-COMM_ALARM_VQD (alarm information structure: NET_DVR_VQD_DIAGNOSE_INFO), 1-COMM_ALARM_VQD_EX (alarm information structure: NET_DVR_VQD_ALARM, including camera information and captured pictures)

Member	Data Type	Description
byFaceAlarmDetection	BYTE	Face detection alarm types: 1-face detection alarm (alarm type: COMM_ALARM_FACE_DETECTION, alarm information structure: NET_DVR_FACE_DETECTION), 0-face capture alarm (alarm type: COMM_UPLOAD_FACESNAP_RESULT, alarm information structure: NET_VCA_FACESNAP_RESULT).
bySupport	BYTE	Capabilities, which is represented by bit: <ul style="list-style-type: none"> • bit0-whether to upload picture: 0-yes, 1-no • bit1-whether to enable ANR: 0-no, 1-yes • 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: <ul style="list-style-type: none"> • bit0-whether to enable ANR for ANPR: 0-no, 1-yes. • bit1-whether to enable ANR for people counting: 0-no, 1-yes. • bit2-whether to enable ANR for heat map: 0-no, 1-yes. • bit3-whether to enable ANR for face capture: 0-no, 1-yes. • bit4-whether to enable ANR for face picture comparison: 0-no, 1-yes. • bit5-whether to enable ANR for JSON message transmission: 0-no, 1-yes. • bit6: whether to enable ANR for uploading heat map data by dwell time duration and by people quantity: 0-no, 1-yes. • bit7: whether to enable ANR for uploading intersection analysis result: 0-no, 1-yes.
wTaskNo	BYTE	Task No.

Member	Data Type	Description
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.
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. <ul style="list-style-type: none"> bit0-type of captured face pictures: 0-binary data, 1-URL bit1-type of picture uploaded in message: 0-binary, 1-URL bit2-type of picture uploaded for face picture comparison: 0-binary, 1-URL
byCustomCtrl	BYTE	Custom control type, which is represented by bit, bit0-whether to upload the face thumbnail of the front passenger: 0-no, 1-yes
byRes4	Array [BYTE]	Reserved, set to 0. The maximum size is 128 bytes.

Remarks

- The parameters **byLevel** and **byAlarmInfoType** are available for traffic cameras. Up to 1 cameras can be armed in the priority of level 0, up to 3 cameras can be armed in the priority of level 1,

and up to 5 cameras can be armed in the priority of level 3, the alarm/event information from the camera in highest priority will be uploaded first.

- For arming via client software, only supports arming one channel, and supports uploading the alarm/event when device is offline; for real-time arming, up to four channels can be armed at same time, but uploading alarm/event when device is offline is not supported.
- The parameter **wTaskNo** is used to distinguish different arming connections. If the value of this parameter in different arming connections is same, error will be returned.

A.26 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.27 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.28 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	<u>fLoginResultCallback</u>	Callback function used to return login status, it is valid only when bUseAsynLogin is "1".
pUser	void*	User data.
bUseAsynLogin	BOOL	Whether to enable asynchronous login: 0-no, 1-yes.
byProxyType	BYTE	Proxy server type: 0-no proxy, 1-standard proxy, 2-EHome proxy.
byUseUTCTime	BYTE	0-not convert (default), 1-input or output UTC time, 2-input or output local time.
byLoginMode	BYTE	Login mode: 0-login by private protocol, 1-login by text protocol, 2-self-adaptive (it is available when the protocol type supported by device is unknown, and this mode does not support asynchronous login).
byHttps	BYTE	Whether to enable TLS for login (by private protocol or by text protocol): 0-no, 1-yes, 2-self-adaptive (which is usually used when the protocol type supported by device is unknown. Both HTTP and HTTPS requests will be sent).
iProxyID	LONG	Proxy server No.
byVerifyMode	BYTE	Whether to enable verification mode: 0-no, 1-bidirectional verification (currently not

Member	Data Type	Description
		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 <code>NET_DVR_SetSDKLocalCfg</code> to load CA certificate, and the enumeration value is "NET_SDK_LOCAL_CFG_CERTIFICATION".
byRes3	BYTE[]	Reserved, the maximum length is 119 bytes.

A.29 NET_DVR_XML_CONFIG_INPUT

Input Parameter Structure of Message Transmission API (NET_DVR_STDXMLConfig)

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

Related API

NET_DVR_STDXMLConfig

A.30 NET_DVR_XML_CONFIG_OUTPUT

Output Parameter Structure of Message Transmission API (NET_DVR_STDXMLConfig)

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

Related API

NET_DVR_STDXMLConfig

Appendix B. Enumeration

B.1 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_BLOCKALLOWLST_FILE	13	Vehicle blocklist file.
UPLOAD_PICTURE_TO_CLOUD	15	Picture file to be uploaded to cloud storage.
UPLOAD_VIDEO_FILE	16	Video file.
UPLOAD_SCREEN_FILE	17	Screen server file.
UPLOAD_PUBLISH_MATERIAL	18	Local material file of information release system.

Enumeration Type	Macro Definition Value	Description
UPLOAD_PUBLISH_UPGRADE_FILE	19	Upgrade file of information release system.
UPLOAD_RING_FILE	20	Ringtone file.
UPLOAD_ENCRYPT_CERT	21	Encryption certificate.
UPLOAD_THERMOMETRIC_FILE	22	Calibration file for temperature measurement.
UPLOAD_SUBBRAND_FILE	23	Vehicle sub brand file.
UPLOAD_LED_CHECK_FILE	24	LED correction file.
BATCH_UPLOAD_PICTURE_FILE	25	Picture files for uploading in batch.
UPLOAD_EDID_CFG_FILE	26	EDID configuration file.
UPLOAD_PANORAMIC_STITCH	27	Panorama stitching configuration file.
UPLOAD_BINOCULAR_COUNTING	28	Binocular counting correction sheet.
UPLOAD_AUDIO_FILE	29	Audio file.
UPLOAD_PUBLISH_THIRD_PARTY_FILE	30	Third-party file.
UPLOAD_DEEPEYES_BINOCULAR	31	TX1 binocular correction sheet.
UPLOAD_CERTIFICATE_BLOCKLIST	32	ID card blocklist.
UPLOAD_HD_CAMERA_CORRECT_TABLE	33	HD camera correction sheet (CAL format).
UPLOAD_FD_DATA	35	Face data file to be imported to face picture library.
UPLOAD_FACE_DATA	36	Face picture file to be imported to face picture library.
UPLOAD_FACE_ANALYSIS_DATA	37	Picture file to be imported to picture recognition target.
UPLOAD_FILEVOLUME_DATA	38	File volume file
IMPORT_DATA_TO_FACELIB	39	Face data (face picture and picture additional information) to be imported to face picture library of device.

Enumeration Type	Macro Definition Value	Description
UPLOAD_LEFTEYE_4K_CALIBFILE	40	Camera calibration parameter file.
UPLOAD_SECURITY_CFG_FILE	41	Configuration file to be securely imported.
UPLOAD_RIGHT_CONTROLLER_AUDIO	42	Audio file of main controller.
UPLOAD_MODBUS_CFG_FILE	43	Configuration file of Modbus protocol.
UPLOAD_NOTICE_VIDEO_DATA	44	Bulletin video file.
UPLOAD_RS485_PROTOCOL_DLL_FILE	45	Dynamic library file of RS485 protocol.
UPLOAD_PIC_BY_BUF	46	Picture file for importing by picture cache.
UPLOAD_CLIENT_CALIBFILE	47	User calibration file (PTO format).
UPLOAD_HD_CAMERA_CORRECT_TABLE_3200W	48	HD camera correction sheet (CAL format).
UPLOAD_DOOR_CONTENT	49	Contact information of the door at the building unit.
UPLOAD_ASR_CONTROL_FILE	50	Speech recognition control file.
UPLOAD_APP_FILE	51	Application program file.
UPLOAD_AI_ALGORITHM_MODEL	52	Algorithm model in binary format.
UPLOAD_AI_BASE_PICTURE	55	Reference pictures in binary format for AI target comparison.
UPLOAD_OFFLINE_CAPTURE_INFO	56	User list of offline collection to be imported.
IMPORT_DATA_TO_HBDLIB	60	Import human body picture with linked information to library.
UPLOAD_SCENE_FILE	61	Scene file to be imported.
UPLOAD_RATIOSTITCHING_FILE	62	Ratio stitching file to be imported.
UPLOAD_LENS_PARAM_FILE	63	Lens parameters file to be imported.

Appendix C. Device Network SDK Errors

The errors that may occur during the device network SDK integration are listed here for reference. You can search for the error descriptions according to the error codes or names returned by a specific API (NET_DVR_GetLastError or NET_DVR_GetErrorMsg).

General Errors

Error Name	Error Code	Error Description
NET_DVR_NOERROR	0	No error.
NET_DVR_PASSWORD_ERROR	1	Incorrect user name or password.
NET_DVR_NOENOUGHPRI	2	No permission.
NET_DVR_NOINIT	3	Uninitialized.
NET_DVR_CHANNEL_ERROR	4	Incorrect channel No.
NET_DVR_OVER_MAXLINK	5	No more device can be connected.
NET_DVR_VERSIONNOMATCH	6	Version mismatches.
NET_DVR_NETWORK_FAIL_CONNECT	7	Connecting to device failed. The device is offline or network connection timed out.
NET_DVR_NETWORK_SEND_ERROR	8	Sending data to device failed.
NET_DVR_NETWORK_RECV_ERROR	9	Receiving data from device failed.
NET_DVR_NETWORK_RECV_TIMEOUT	10	Receiving data from device timed out.
NET_DVR_NETWORK_ERRORDATA	11	The data sent to the device is illegal, or the data received from the device error. E.g. The input data is not supported by the device for remote configuration.
NET_DVR_ORDER_ERROR	12	API calling order error.
NET_DVR_OPERNOPERMIT	13	No permission for this operation.
NET_DVR_COMMANDTIMEOUT	14	Executing device command timed out.
NET_DVR_ERRORSERIALPORT	15	Incorrect serial port No. The specified serial port does not exist.

Error Name	Error Code	Error Description
NET_DVR_ERRORALARMPORT	16	Alarm port No. error. The alarm input or output port of the specified device does not exist.
NET_DVR_PARAMETER_ERROR	17	Incorrect parameter. The input or output parameters of the SDK API is empty, or the parameter value or format is invalid.
NET_DVR_CHAN_EXCEPTION	18	Device channel is in exception status.
NET_DVR_NODISK	19	No HDD in the device.
NET_DVR_ERRORDISKNUM	20	Incorrect HDD No.
NET_DVR_DISK_FULL	21	HDD full.
NET_DVR_DISK_ERROR	22	HDD error.
NET_DVR_NOSUPPORT	23	Device does not support this function.
NET_DVR_BUSY	24	Device is busy.
NET_DVR_MODIFY_FAIL	25	Failed to edit device parameters.
NET_DVR_PASSWORD_FORMAT_ERROR	26	Invalid password format.
NET_DVR_DISK_FORMATING	27	HDD is formatting. Failed to startup.
NET_DVR_DVRNORESOURCE	28	Insufficient device resources.
NET_DVR_DVROPRATEFAILED	29	Device operation failed.
NET_DVR_OPENHOSTSOUND_FAIL	30	Failed to collect local audio data or open audio output during two-way audio and broadcast.
NET_DVR_DVRVOICEOPENED	31	Two-way audio channel is occupied.
NET_DVR_TIMEINPUTERROR	32	Incorrect time input.
NET_DVR_NOSPECFILE	33	No video file for playback.
NET_DVR_CREATEFILE_ERROR	34	Failed to create a file during local recording, saving picture, getting configuration file or downloading video file remotely.
NET_DVR_FILEOPENFAIL	35	Failed to open a file. The file does not exist or directory error.

Error Name	Error Code	Error Description
NET_DVR_OPERNOTFINISH	36	Operation conflicted.
NET_DVR_GETPLAYTIMEFAIL	37	Failed to get the current played time.
NET_DVR_PLAYFAIL	38	Failed to play.
NET_DVR_FILEFORMAT_ERROR	39	Invalid file format.
NET_DVR_DIR_ERROR	40	File directory error.
NET_DVR_ALLOC_RESOURCE_ERROR	41	Allocating resources failed.
NET_DVR_AUDIO_MODE_ERROR	42	Invalid sound card mode error. The opened sound play mode and configured mode mismatched.
NET_DVR_NOENOUGH_BUF	43	Insufficient buffer for receiving data or saving picture.
NET_DVR_CREATESOCKET_ERROR	44	Failed to create SOCKET.
NET_DVR_SETSOCKET_ERROR	45	Failed to set SOCKET.
NET_DVR_MAX_NUM	46	No more registrations and live views can be connected.
NET_DVR_USERNOTEXIST	47	The user does not exist. The user ID is logged out or unavailable.
NET_DVR_WRITEFLASHERROR	48	Writing FLASH error during device upgrade.
NET_DVR_UPGRADEFAIL	49	Failed to upgrade device. Network problem or language mismatches.
NET_DVR_CARDHAVEINIT	50	The decoding card is already initialized.
NET_DVR_PLAYERFAILED	51	Failed to call the function of player SDK.
NET_DVR_MAX_USERNUM	52	No more users can log in to.
NET_DVR_GETLOCALIPANDMACFAIL	53	Failed to get the IP address or physical address of local PC.
NET_DVR_NOENCODEING	54	The decoding function of this channel is not enabled.
NET_DVR_IPMISMATCH	55	IP address mismatches.

Error Name	Error Code	Error Description
NET_DVR_MACMISMATCH	56	MAC address mismatches.
NET_DVR_UPGRADELANGMISMATCH	57	The language of upgrade file mismatches.
NET_DVR_MAX_PLAYERPORT	58	No more channels can be started to play.
NET_DVR_NOSPACEBACKUP	59	Insufficient space to back up file.
NET_DVR_NODEVICEBACKUP	60	No backup device found.
NET_DVR_PICTURE_BITS_ERROR	61	Picture pixel bit mismatches. Only 24 bits are allowed.
NET_DVR_PICTURE_DIMENSION_ERROR	62	Too large picture. The height*width should be less than 128x256.
NET_DVR_PICTURE_SIZ_ERROR	63	Too large picture. The picture size should be smaller than 100K.
NET_DVR_LOADPLAYERSDKFAILED	64	Failed to load the player(PlayCtrl.dll, SuperRender.dll, AudioRender.dll) to the current directory.
NET_DVR_LOADPLAYERSDKPROC_ERROR	65	Failed to find the function in player SDK.
NET_DVR_LOADDSSDKFAILED	66	Failed to load the DS SDK to the current directory.
NET_DVR_LOADDSSDKPROC_ERROR	67	Failed to find the function in the DS SDK.
NET_DVR_DSSDK_ERROR	68	Failed to call the API in the hardware decoding library.
NET_DVR_VOICEMONOPOLIZE	69	The sound card is exclusive.
NET_DVR_JOINMULTICASTFAILED	70	Failed to join to multicast group.
NET_DVR_CREATEDIR_ERROR	71	Failed to create log file directory.
NET_DVR_BINDSOCKET_ERROR	72	Failed to bind socket.
NET_DVR_SOCKETCLOSE_ERROR	73	Socket disconnected. Network disconnected or the destination is unreachable.

Error Name	Error Code	Error Description
NET_DVR_USERID_ISUSING	74	Operation is executing. Failed to log out.
NET_DVR_SOCKETLISTEN_ERROR	75	Failed to listen.
NET_DVR_PROGRAM_EXCEPTION	76	Program exception.
NET_DVR_WRITEFILE_FAILED	77	Failed to write file during local recording, downloading file remotely or saving picture.
NET_DVR_FORMAT_READONLY	78	The HDD is read-only. Formatting is forbidden.
NET_DVR_WITHSAMEUSERNAME	79	The user name already exists.
NET_DVR_DEVICETYPE_ERROR	80	Device model mismatches when importing parameters.
NET_DVR_LANGUAGE_ERROR	81	Language mismatches when importing parameters.
NET_DVR_PARAVERSION_ERROR	82	Software version mismatches when importing parameters.
NET_DVR_IPCHAN_NOTALIVE	83	The external IP channel is offline live view.
NET_DVR_RTSP_SDK_ERROR	84	Failed to load StreamTransClient.dll.
NET_DVR_CONVERT_SDK_ERROR	85	Failed to load SystemTransform.dll.
NET_DVR_IPC_COUNT_OVERFLOW	86	No more IP channels can access to.
NET_DVR_MAX_ADD_NUM	87	No more video tags can be added.
NET_DVR_PARAMMODE_ERROR	88	Invalid parameter mode of image enhancement.
NET_DVR_CODESPITTER_OFFLINE	89	Code distributer is offline.
NET_DVR_BACKUP_COPYING	90	Device is backing up.
NET_DVR_CHAN_NOTSUPPORT	91	This operation is not supported by the channel.
NET_DVR_CALLINEINVALID	92	The height line is too concentrated, or the length line is not inclined enough.

Error Name	Error Code	Error Description
NET_DVR_CALCANCELCONFLICT	93	Cancel calibration conflict, if the rule and global actual size filter are configured.
NET_DVR_CALPOINTOUTRANGE	94	The calibration point is out of limitation.
NET_DVR_FILTERRECTINVALID	95	The size filter does not meet the requirement.
NET_DVR_DDNS_DEVOFFLINE	96	Device has not registered to DDNS.
NET_DVR_DDNS_INTER_ERROR	97	DDNS internal error.
NET_DVR_FUNCTION_NOT_SUPPORT_OS	98	This function is not supported by this Operating system.
NET_DVR_DEC_CHAN_REBIND	99	Decoding channel binding display output is limited.
NET_DVR_INTERCOM_SDK_ERROR	100	Failed to load the two-way audio SDK of the current directory.
NET_DVR_NO_CURRENT_UPDATEFILE	101	No correct upgrade packet.
NET_DVR_USER_NOT_SUCC_LOGIN	102	Login failed.
NET_DVR_USE_LOG_SWITCH_FILE	103	The log switch file is under using.
NET_DVR_POOL_PORT_EXHAUST	104	No port can be bound in the port pool.
NET_DVR_PACKET_TYPE_NOT_SUPPORT	105	Incorrect stream packaging format.
NET_DVR_IPPARA_IPID_ERROR	106	Incorrect IPID for IP access configuration.
NET_DVR_LOAD_HCPREVIEW_SDK_ERROR	107	Failed to load the live view component.
NET_DVR_LOAD_HCVOICETALK_SDK_ERROR	108	Failed to load the audio component.
NET_DVR_LOAD_HCALARM_SDK_ERROR	109	Failed to load the alarm component.
NET_DVR_LOAD_HCPLAYBACK_SDK_ERROR	110	Failed to load the playback component.

Error Name	Error Code	Error Description
NET_DVR_LOAD_HCDISPLAY_SDK_ERROR	111	Failed to load the display component.
NET_DVR_LOAD_HCINDUSTRY_SDK_ERROR	112	Failed to load application component.
NET_DVR_LOAD_HCGENERALCFGMGR_SDK_ERROR	113	Failed to load the general configuration management component.
NET_DVR_CORE_VER_MISMATCH	121	Component version and core version mismatched when loading the component singly.
NET_DVR_CORE_VER_MISMATCH_HCPREVIEW	122	Live view component version and core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCVOICETALK	123	Audio component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCALARM	124	Alarm component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCPLAYBACK	125	Playback component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCDISPLAY	126	Display component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCINDUSTRY	127	Application component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCGENERALCFGMGR	128	General configuration management component version and the core version mismatched.
NET_DVR_COM_VER_MISMATCH_HCPREVIEW	136	Live view component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCVOICETALKy	137	Audio component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCALARM	138	Alarm component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCPLAYBACK	139	Playback component version and SDK version mismatched.

Error Name	Error Code	Error Description
NET_DVR_COM_VER_MISMATCH_HCDISPLAY	140	Display component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCINDUSTRY	141	Application component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCGENERALCFGMGR	142	General configuration management component version and SDK version mismatched.
NET_DVR_ALIAS_DUPLICATE	150	Duplicated alias(for HiDDNS configuration).
NET_DVR_USERNAME_NOT_EXIST	152	User name does not exist (error code of network camera and network speed dome with version from 5.1.7 to 5.3.1).
NET_ERR_USERNAME_LOCKED	153	The user name is locked.
NET_DVR_INVALID_USERID	154	Invalid user ID.
NET_DVR_LOW_LOGIN_VERSION	155	The version is too low.
NET_DVR_LOAD_LIBEAY32_DLL_ERROR	156	Failed to load libeay32.dll.
NET_DVR_LOAD_SSLEAY32_DLL_ERROR	157	Failed to load ssleay32.dll.
NET_ERR_LOAD_LIBICONV	158	Failed to load libiconv.dll.
NET_ERR_SSL_CONNECT_FAILED	159	Connecting to SSL failed.
NET_DVR_TEST_SERVER_FAIL_CONNECT	165	Failed to connect to test server.
NET_DVR_NAS_SERVER_INVALID_DIR	166	Failed to load NAS server to the directory, Invalid directory, or incorrect user name and password.
NET_DVR_NAS_SERVER_NOENOUGH_PRI	167	Failed to load NAS server th the directory. No permission.
NET_DVR_EMAIL_SERVER_NOT_CONFIG_DNS	168	The server uses domain name without configuring DNS, the domain name may be invalid.

Error Name	Error Code	Error Description
NET_DVR_EMAIL_SERVER_NOT_CONFIG_GATEWAY	169	No gateway configured. Sending email may be failed.
NET_DVR_TEST_SERVER_PASSWORD_ERROR	170	Incorrect user name or password of test server.
NET_DVR_EMAIL_SERVER_CONNECT_EXCEPTION_WITH_SMTP	171	Interaction exception between device and SMTP server.
NET_DVR_FTP_SERVER_FAIL_CREATE_DIR	172	FTP server creating directory failed.
NET_DVR_FTP_SERVER_NO_WRITE_PIR	173	FTP server has no writing permission.
NET_DVR_IP_CONFLICT	174	IP conflicted.
NET_DVR_INSUFFICIENT_STORAGEPOOL_SPACE	175	Storage pool space is full.
NET_DVR_STORAGEPOOL_INVALID	176	Invalid cloud storage pool. No storage pool configured or incorrect storage pool ID.
NET_DVR_EFFECTIVENESS_REBOOT	177	Restart to take effect.
NET_ERR_ANR_ARMING_EXIST	178	The ANR arming connection already exists(the error will be returned when arming with ANR function if the private SDK protocol arming connection is established).
NET_ERR_UPLOADLINK_EXIST	179	The ANR uploading connection already exists(the error will be returned when EHome protocol and private SDK protocol do not support ANR at the same time).
NET_ERR_INCORRECT_FILE_FORMAT	180	The imported file format is incorrect.
NET_ERR_INCORRECT_FILE_CONTENT	181	The imported file content is incorrect.
NET_ERR_MAX_HRUDP_LINK	182	No more HRUDP can be connected to device.
NET_ERR_MAX_PORT_MULTIPLEX	183	Maximum number of multiplexed ports reaches.
NET_ERR_CREATE_PORT_MULTIPLEX	184	Creating port multiplier failed.

Error Name	Error Code	Error Description
NET_DVR_NONBLOCKING_CAPTURE_NOTSUPPORT	185	Non-blocking picture capture is not supported.
NET_SDK_ERR_FUNCTION_INVALID	186	Invalid function. The asynchronous mode is enabled.
NET_SDK_ERR_MAX_PORT_MULTIPLEX	187	Maximum number of multiplex ports reached.
NET_DVR_INVALID_LINK	188	Link has not been created or the link is invalid.
NET_DVR_NAME_NOT_ONLY	200	This name already exists.
NET_DVR_OVER_MAX_ARRAY	201	The number of RAID reaches the upper-limit.
NET_DVR_OVER_MAX_VD	202	The number of virtual disk reaches the upper-limit.
NET_DVR_VD_SLOT_EXCEED	203	The virtual disk slots are full.
NET_DVR_PD_STATUS_INVALID	204	The physical disk for rebuilding RAID is error.
NET_DVR_PD_BE_DEDICATE_SPARE	205	The physical disk for rebuilding RAID is specified as hot spare.
NET_DVR_PD_NOT_FREE	206	The physical disk for rebuilding RAID is busy.
NET_DVR_CANNOT_MIG2NEWMODE	207	Failed to migrate the current RAID type to the new type.
NET_DVR_MIG_PAUSE	208	Migration is paused.
NET_DVR_MIG_ABOUTED	209	Migration is cancelled.
NET_DVR_EXIST_VD	210	Failed to delete RAID. Virtual disk exists in the RAID.
NET_DVR_TARGET_IN_LD_FUNCTIONAL	211	Target physical disk is a part of the virtual disk and it is working normally.
NET_DVR_HD_IS_ASSIGNED_ALREADY	212	The specified physical disk is allocated as virtual disk.
NET_DVR_INVALID_HD_COUNT	213	The number of physical disks and specified RAID level mismatched.

Error Name	Error Code	Error Description
NET_DVR_LD_IS_FUNCTIONAL	214	The RAID is normal. Failed to rebuild.
NET_DVR_BGA_RUNNING	215	Background task is executing.
NET_DVR_LD_NO_ATAPI	216	Failed to create virtual disk by ATAPI disk.
NET_DVR_MIGRATION_NOT_NEED	217	There is no need to migrate the RAID.
NET_DVR_HD_TYPE_MISMATCH	218	The physical disk type is not allowed.
NET_DVR_NO_LD_IN_DG	219	No virtual disk. Operation failed.
NET_DVR_NO_ROOM_FOR_SPARE	220	Insufficient disk space. Failed to allocate the disk as hot spare.
NET_DVR_SPARE_IS_IN_MULTI_DG	221	The disk is already allocated as the hot spare of one RAID.
NET_DVR_DG_HAS_MISSING_PD	222	No disk in the RAID.
NET_DVR_NAME_EMPTY	223	The name is empty.
NET_DVR_INPUT_PARAM	224	Incorrect input parameters.
NET_DVR_PD_NOT_AVAILABLE	225	The physical disk is not available.
NET_DVR_ARRAY_NOT_AVAILABLE	226	The RAID is not available.
NET_DVR_PD_COUNT	227	Incorrect number of physical disks.
NET_DVR_VD_SMALL	228	Insufficient virtual disk space.
NET_DVR_NO_EXIST	229	Not exist.
NET_DVR_NOT_SUPPORT	230	This operation is not supported.
NET_DVR_NOT_FUNCTIONAL	231	The RAID status is exception.
NET_DVR_DEV_NODE_NOT_FOUND	232	The device node of virtual disk does not exist.
NET_DVR_SLOT_EXCEED	233	No more slots are allowed.
NET_DVR_NO_VD_IN_ARRAY	234	No virtual disk exists in the RAID.
NET_DVR_VD_SLOT_INVALID	235	Invalid virtual disk slot.
NET_DVR_PD_NO_ENOUGH_SPACE	236	Insufficient physical disk space.
NET_DVR_ARRAY_NONFUNCTION	237	Only the RAID in normal status supports to be migrated.

Error Name	Error Code	Error Description
NET_DVR_ARRAY_NO_ENOUGH_SPACE	238	Insufficient RAID space.
NET_DVR_STOPPING_SCANNING_ARRAY	239	Pulling disk out safely or rescanning.
NET_DVR_NOT_SUPPORT_16T	240	Creating RAID with size larger than 16T is not supported.
NET_DVR_ERROR_DEVICE_NOT_ACTIVATED	250	The device is not activated (login failed.)
NET_DVR_ERROR_RISK_PASSWORD	251	Risky password.
NET_DVR_ERROR_DEVICE_HAS_ACTIVATED	252	The device is already activated.
NET_DVR_ID_ERROR	300	The configured ID is invalid.
NET_DVR_POLYGON_ERROR	301	Invalid polygon shape.
NET_DVR_RULE_PARAM_ERROR	302	Invalid rule parameters.
NET_DVR_RULE_CFG_CONFLICT	303	Configured information conflicted.
NET_DVR_CALIBRATE_NOT_READY	304	No calibration information.
NET_DVR_CAMERA_DATA_ERROR	305	Invalid camera parameters.
NET_DVR_CALIBRATE_DATA_UNFIT	306	Invalid inclination angle for calibration.
NET_DVR_CALIBRATE_DATA_CONFLICT	307	Calibration error.
NET_DVR_CALIBRATE_CALC_FAIL	308	Failed to calculate calibration parameter values of camera.
NET_DVR_CALIBRATE_LINE_OUT_RECT	309	The inputted calibration line exceeds the external sample rectangle.
NET_DVR_ENTER_RULE_NOT_READY	310	No region entrance is configured.
NET_DVR_AID_RULE_NO_INCLUDE_LANE	311	No lane configured in the traffic event rule (especially for traffic jam or driving against the traffic).
NET_DVR_LANE_NOT_READY	312	Lane not configured.
NET_DVR_RULE_INCLUDE_TWO_WAY	313	Two different directions are contained in event rule.

Error Name	Error Code	Error Description
NET_DVR_LANE_TPS_RULE_CONFLICT	314	Lane and data rule conflicted.
NET_DVR_NOT_SUPPORT_EVENT_TYPE	315	This event type is not supported.
NET_DVR_LANE_NO_WAY	316	The lane has no direction.
NET_DVR_SIZE_FILTER_ERROR	317	Invalid size of filter frame.
NET_DVR_LIB_FFL_NO_FACE	318	No face picture exists in the image inputted when positioning feature point.
NET_DVR_LIB_FFL_IMG_TOO_SMALL	319	The inputted image is too small when positioning feature point.
NET_DVR_LIB_FD_IMG_NO_FACE	320	No face picture exists in the image inputted when detecting single face picture.
NET_DVR_LIB_FACE_TOO_SMALL	321	Face picture is too small when building model.
NET_DVR_LIB_FACE_QUALITY_TOO_BAD	322	The face picture quality is too poor when building model.
NET_DVR_KEY_PARAM_ERR	323	The configured advanced parameter is incorrect.
NET_DVR_CALIBRATE_DATA_ERR	324	Calibration sample number error, or data value error, or the sample points are beyond the horizontal line.
NET_DVR_CALIBRATE_DISABLE_FAIL	325	Canceling calibration is not allowed for configured rules.
NET_DVR_VCA_LIB_FD_SCALE_OUTRANGE	326	The minimum width and height of maximum filter frame are twice or more larger than the maximum width and height of minimum filter frame.
NET_DVR_LIB_FD_REGION_TOO_LARGE	327	Too large detection region. The maximum region should be 2/3 of the image.
NET_DVR_TRIAL_OVERDUE	328	Trial period is ended.
NET_DVR_CONFIG_FILE_CONFLICT	329	Device type and configuration file conflicted.

Error Name	Error Code	Error Description
NET_DVR_FR_FPL_FAIL	330	Failed to positioning face feature points.
NET_DVR_FR_IQA_FAIL	331	Failed to test face picture quality.
NET_DVR_FR_FEM_FAIL	332	Failed to extract the face feature points.
NET_DVR_FPL_DT_CONF_TOO_LOW	333	The face detection validity is too low when positioning face feature points.
NET_DVR_FPL_CONF_TOO_LOW	334	The validity of feature points positionong is too low.
NET_DVR_E_DATA_SIZE	335	Data size mismatches.
NET_DVR_FR_MODEL_VERSION_ERR	336	Incorrect model version in face model library.
NET_DVR_FR_FD_FAIL	337	Failed to detect face in the face recognition library.
NET_DVR_FA_NORMALIZE_ERR	338	Failed to normalize face attribute.
NET_DVR_DOG_PUSTREAM_NOT_MATCH	339	Dongle type and camera type mismatched.
NET_DVR_DEV_PUSTREAM_NOT_MATCH	340	Camera version mismatches.
NET_DVR_PUSTREAM_ALREADY_EXISTS	341	This camera is already added to other channels of devices.
NET_DVR_SEARCH_CONNECT_FAILED	342	Failed to connect to face retrieval server.
NET_DVR_INSUFFICIENT_DISK_SPACE	343	Insufficient storage space.
NET_DVR_DATABASE_CONNECTION_FAILED	344	Failed to connect to database.
NET_DVR_DATABASE_ADM_PW_ERROR	345	Incorrect database user name and password.
NET_DVR_DECODE_YUV	346	Decoding failed.
NET_DVR_IMAGE_RESOLUTION_ERROR	347	Invalid picture resolution

Error Name	Error Code	Error Description
NET_DVR_CHAN_WORKMODE_ERROR	348	Invalid channel working mode.
NET_ERROR_TRUNK_LINE	711	Sub system is configured as the trunk line.
NET_ERROR_MIXED_JOINT	712	Mixed joint is not supported.
NET_ERROR_DISPLAY_SWITCH	713	Switch of display channel is not supported.
NET_ERROR_USED_BY_BIG_SCREEN	714	Decoded resource is occupied by the big screen.
NET_ERROR_USE_OTHER_DEC_RESOURCE	715	Using resources of other sub system is not allowed.
NET_ERROR_SCENE_USING	717	The scene is being used.
NET_ERR_NO_ENOUGH_DEC_RESOURCE	718	Insufficient resources for decoding.
NET_ERR_NO_ENOUGH_FREE_SHOW_RESOURCE	719	Insufficient resources for display.
NET_ERR_NO_ENOUGH_VIDEO_MEMORY	720	Insufficient video storage resources.
NET_ERR_MAX_VIDEO_NUM	721	Insufficient resources for multiple channels.
NET_ERR_WINDOW_COVER_FREE_SHOW_AND_NORMAL	722	Windows cover free display output channel and normal output channel.
NET_ERR_FREE_SHOW_WINDOW_SPLIT	723	Window division is not supported for free display windows.
NET_ERR_INAPPROPRIATE_WINDOW_FREE_SHOW	724	For the windows whose number is not integral multiple of the number of output channels, free display is not supported.
NET_DVR_TRANSPARENT_WINDOW_NOT_SUPPORT_SPLIT	725	For windows whose transparency configuration is enabled, window division is not supported.
NET_DVR_SPLIT_WINDOW_NOT_SUPPORT_TRANSPARENT	726	For windows whose window division is enabled, transparency configuration is not supported.

Error Name	Error Code	Error Description
NET_ERR_TERMINAL_BUSY	780	The terminal busy.
NET_DVR_FUNCTION_RESOURCE_USAGE_ERROR	791	Failed to enable this function. The resources is occupied by other functions.
NET_DVR_DEV_NET_OVERFLOW	800	Network traffic is out of the limitation.
NET_DVR_STATUS_RECORDFILE_WRITING_NOT_LOCK	801	Failed to lock. The video file is recording.
NET_DVR_STATUS_CANT_FORMAT_LITTLE_DISK	802	Failed to format HDD. The HDD space is too small.
NET_SDK_ERR_REMOTE_DISCONNECT	803	Failed to connect to the remote terminal.
NET_SDK_ERR_RD_ADD_RD	804	Spare server cannot be added to spare server.
NET_SDK_ERR_BACKUP_DISK_EXCEPT	805	Backup disk exception.
NET_SDK_ERR_RD_LIMIT	806	No more spare server can be added.
NET_SDK_ERR_ADDED_RD_IS_WD	807	The added spare server is a working server.
NET_SDK_ERR_ADD_ORDER_WRONG	808	Adding flow error.
NET_SDK_ERR_WD_ADD_WD	809	Working server cannot be added to working server.
NET_SDK_ERR_WD_SERVICE_EXCETP	810	CVR service exception (For N+1 mode, it refers to CVR working server exception).
NET_SDK_ERR_RD_SERVICE_EXCETP	811	Spare CVR server exception.
NET_SDK_ERR_ADDED_WD_IS_RD	812	The added working server is spare server.
NET_SDK_ERR_PERFORMANCE_LIMIT	813	The performance reaches the upper-limit.
NET_SDK_ERR_ADDED_DEVICE_EXIST	814	This device already exists.
NET_SDK_ERR_INQUEST_RESUMING	815	Inquest resuming.
NET_SDK_ERR_RECORD_BACKUPING	816	Inquest video backing up.

Error Name	Error Code	Error Description
NET_SDK_ERR_DISK_PLAYING	817	Playing.
NET_SDK_ERR_INQUEST_STARTED	818	Inquest started.
NET_SDK_ERR_LOCAL_OPERATING	819	Locally operating.
NET_SDK_ERR_INQUEST_NOT_START	820	Inquest is not started.
NET_SDK_ERR_CHAN_AUDIO_BIND	821	The channel is not bound or binding two-way audio failed.
NET_DVR_N_PLUS_ONE_MODE	822	Ddevice is in N+1 mode. Cloud storage is not supported.
NET_DVR_CLOUD_STORAGE_OPENED	823	Cloud storage mode is enbaled.
NET_DVR_ERR_OPER_NOT_ALLOWED	824	Operation failed. The device is in N+0 taken over status.
NET_DVR_ERR_NEED_RELOCATE	825	The device is in N+0 taken over status. Get re-positioning information and try again.
NET_SDK_ERR_IR_PORT_ERROR	830	IR output error.
NET_SDK_ERR_IR_CMD_ERROR	831	IR output port command number error
NET_SDK_ERR_NOT_INQUESTING	832	Device is not in inquest status.
NET_SDK_ERR_INQUEST_NOT_PAUSED	833	Device is not in paused status.
NET_DVR_CHECK_PASSWORD_MISTAKE_ERROR	834	Incorrect verification code.
NET_DVR_CHECK_PASSWORD_NULL_ERROR	835	Verification code is required.
NET_DVR_UNABLE_CALIB_ERROR	836	Failed to calibrate.
NET_DVR_PLEASE_CALIB_ERROR	837	Calibration first.
NET_DVR_ERR_PANORAMIC_CAL_EMPTY	838	Panoramic calibration is empty in Flash.
NET_DVR_ERR_CALIB_FAIL_PLEASEAGAIN	839	Calibration failed, please try again.

Error Name	Error Code	Error Description
NET_DVR_ERR_DETECTION_LINE	840	Rule line configuration error. Please try again and make sure the line is within the red region.
NET_DVR_EXCEED_FACE_IMAGES_ERROR	843	No more face pictures can be added.
NET_DVR_ANALYSIS_FACE_IMAGES_ERROR	844	Picture recognition failed.
NET_ERR_ALARM_INPUT_OCCUPIED	845	A<-1 alarm number is used for triggering vehicle capture.
NET_DVR_FACELIB_DATABASE_ERROR	846	Database version in face picture library mismatched.
NET_DVR_FACELIB_DATA_ERROR	847	Face picture library data error.
NET_DVR_FACE_DATA_ID_ERROR	848	Invalid face data PID.
NET_DVR_FACELIB_ID_ERROR	849	Invalid face picture library ID.
NET_DVR_EXCEED_FACE_LIBRARY_ERROR	850	No more face picture libraries can be established..
NET_DVR_PIC_ANALYSIS_NO_TARGET_ERROR	851	No target recognized in the picture.
NET_DVR_SUBPIC_ANALYSIS_MODELING_ERROR	852	Sub picture modeling failed.
NET_DVR_PIC_ANALYSIS_NO_RESOURCE_ERROR	853	No VCA engine supports picture secondary recognition.
NET_DVR_ANALYSIS_ENGINES_NO_RESOURCE_ERROR	854	No VCA engine.
NET_DVR_ANALYSIS_ENGINES_USAGE_EXCEED_ERROR	855	Overload. The engine CPU reached 100%.
NET_DVR_EXCEED_HUMANMISINFO_FILTER_ENABLED_ERROR	856	No more false alarm channel can be enabled.
NET_DVR_NAME_ERROR	857	Name error.
NET_DVR_NAME_EXIST_ERROR	858	The name already exists.
NET_DVR_FACELIB_PIC_IMPORTING_ERROR	859	The pictures is importing to face picture library.

Error Name	Error Code	Error Description
NET_DVR_PIC_FORMAT_ERROR	864	Invalid picture format.
NET_DVR_PIC_RESOLUTION_INVALID_ERROR	865	Invalid picture resolution.
NET_DVR_PIC_SIZE_EXCEED_ERROR	866	The picture size is too large.
NET_DVR_PIC_ANALYSIS_TARGRT_NUM_EXCEED_ERROR	867	Too many targets in the picture.
NET_DVR_ANALYSIS_ENGINES_LOADING_ERROR	868	Initializing analysis engine.
NET_DVR_ANALYSIS_ENGINES_ABNORMA_ERROR	869	Analysis engine exception.
NET_DVR_ANALYSIS_ENGINES_FACELIB_IMPORTING	870	Analysis engine is importing pictures to face picture library.
NET_DVR_NO_DATA_FOR_MODELING_ERROR	871	No data for modeling.
NET_DVR_FACE_DATA_MODELING_ERROR	872	Device is modeling picture. Concurrent processing is not supported.
NET_ERR_FACELIBDATA_OVERLIMIT	873	No more face picture can be added to the device (the data of imported face picture library)
NET_DVR_ANALYSIS_ENGINES_ASSOCIATED_CHANNEL	874	Channel is linked to the analysis engine.
NET_DVR_ERR_CUSTOMID_LEN	875	The minimum length of upper layer custom ID is 32 bytes.
NET_DVR_ERR_CUSTOMFACELIBID_REPEAT	876	The applied custom face picture library ID is duplicated
NET_DVR_ERR_CUSTOMHUMANID_REPEAT	877	The applied custom person ID is duplicated.
NET_DVR_ERR_URL_DOWNLOAD_FAIL	878	URL download failed.
NET_DVR_ERR_URL_DOWNLOAD_NOTSTART	879	URL download has not started.

Error Name	Error Code	Error Description
NET_DVR_CFG_FILE_SECRETKEY_ERROR	880	The security verification key of configuration file is error.
NET_DVR_THERMOMETRY_REGION_OVERSTEP_ERROR	883	Invalid thermometry region
NET_DVR_ERR_TOO_SHORT_CALIBRATING_TIME	894	Too short time for calibration.
NET_DVR_ERR_AUTO_CALIBRATE_FAILED	895	Auto calibration failed.
NET_DVR_ERR_VERIFICATION_FAILED	896	Verification failed.
NET_DVR_NO_TEMP_SENSOR_ERROR	897	No temperature sensor.
NET_DVR_PUPIL_DISTANCE_OVERSIZE_ERROR	898	The pupil distance is too large.
NET_ERR_WINCHAN_IDX	901	Window channel index error.
NET_ERR_WIN_LAYER	902	Window layer number error(the count of window layers on a single screen exceeds the max number).
NET_ERR_WIN_BLK_NUM	903	Window block number error(the count of screens that single window overlays exceeds the max number).
NET_ERR_OUTPUT_RESOLUTION	904	The output resolution error.
NET_ERR_LAYOUT	905	Layout index error.
NET_ERR_INPUT_RESOLUTION	906	The input resolution is not supported.
NET_ERR_SUBDEVICE_OFFLINE	907	The sub-device is off-line.
NET_ERR_NO_DECODE_CHAN	908	There is no free decoding channel.
NET_ERR_MAX_WINDOW_ABILITY	909	The upper limit of window number.
NET_ERR_ORDER_ERROR	910	Calling order error.
NET_ERR_PLAYING_PLAN	911	Be playing plan.
NET_ERR_DECODER_USED	912	Decoder board is being used.
NET_ERR_OUTPUT_BOARD_DATA_OVERFLOW	913	Output board data overflow
NET_ERR_SAME_USER_NAME	914	Duplicate user name

Error Name	Error Code	Error Description
NET_ERR_INVALID_USER_NAME	915	Invalid user name
NET_ERR_MATRIX_USING	916	Input matrix is in use.
NET_ERR_DIFFERENT_CHAN_TYPE	917	Different channel type (the type of matrix output channel mismatches that of the controller input channel)
NET_ERR_INPUT_CHAN_BINDED	918	Input channel has been bound by other matrix
NET_ERR_BINDED_OUTPUT_CHAN_OVERFLOW	919	The matrix output channels in use exceeded the number bound by matrix and controller
NET_ERR_MAX_SIGNAL_NUM	920	Number of input signals reached upper limit
NET_ERR_INPUT_CHAN_USING	921	Input channel is in use
NET_ERR_MANAGER_LOGON	922	Administrator has logged in, operation failed
NET_ERR_USERALREADY_LOGON	923	The user has logged in, operation failed
NET_ERR_LAYOUT_INIT	924	Scene is initializing, operation failed
NET_ERR_BASEMAP_SIZE_NOT_MATCH	925	Base image size does not match
NET_ERR_WINDOW_OPERATING	926	Window is in other operation, operation failed
NET_ERR_SIGNAL_UPLIMIT	927	Number of signal source window reached upper limit
NET_ERR_WINDOW_SIZE_OVERLIMIT	943	The window size exceeds the limit.
NET_ERR_MAX_WIN_OVERLAP	951	The number of windows overlap has reached the maximum limit.
NET_ERR_STREAMID_CHAN_BOTH_VALID	952	stream ID and channel number are both valid.
NET_ERR_NO_ZERO_CHAN	953	The device has no zero channel.
NEED_RECONNECT	955	Need redirection (for transcoding system)

Error Name	Error Code	Error Description
NET_ERR_NO_STREAM_ID	956	The stream ID does not exist.
NET_DVR_TRANS_NOT_START	957	The transcoding has not been started.
NET_ERR_MAXNUM_STREAM_ID	958	The number of stream ID has reached the maximum limit.
NET_ERR_WORKMODE_MISMATCH	959	The work mode does not match with the requirement.
NET_ERR_MODE_IS_USING	960	It Has been working in current mode.
NET_ERR_DEV_PROGRESSING	961	The device is in processing
NET_ERR_PASSIVE_TRANSCODING	962	It is in transcoding.
NET_DVR_ERR_WINDOW_SIZE_PLACE	975	Wrong window position.
NET_DVR_ERR_RGIONAL_RESTRICTIONS	976	Screen distance exceeds the limit.
NET_DVR_ERR_CLOSE_WINDOWS	984	Operation failed. Close the window first.
NET_DVR_ERR_MATRIX_LOOP_ABILITY	985	Beyond the cycle decoding capacity.
NET_DVR_ERR_MATRIX_LOOP_TIME	986	Invalid cycle decoding time.
NET_DVR_ERR_LINKED_OUT_ABILITY	987	No more linked camera can be added.
NET_ERR_RESOLUTION_NOT_SUPPORT_ODD_VOUT	990	The resolution is not supported (odd No.).
NET_ERR_RESOLUTION_NOT_SUPPORT_EVEN_VOUT	991	The resolution is not supported (even No.).
NET_ERR_UnitConfig_Failed	998	Unit configuration failed.
XML_ABILITY_NOTSUPPORT	1000	Getting capability node is not supported
XML_ANALYZE_NOENOUGH_BUF	1001	Not enough output memory
XML_ANALYZE_FIND_LOCALXML_ERROR	1002	Failed to find related local xml
XML_ANALYZE_LOAD_LOCALXML_ERROR	1003	Loading local xml error

Error Name	Error Code	Error Description
XML_NANLYZE_DVR_DATA_FORMAT_ERROR	1004	Device capability data format error
XML_ANALYZE_TYPE_ERROR	1005	Capability set type error
XML_ANALYZE_XML_NODE_ERROR	1006	XML capability node format error
XML_INPUT_PARAM_ERROR	1007	Input capability XML node value error
XML_VERSION_MISMATCH	1008	XML version does not match
NET_ERR_TRANS_CHAN_START	1101	Transparent channel has been open, operation failed
NET_ERR_DEV_UPGRADING	1102	Device is upgrading
NET_ERR_MISMATCH_UPGRADE_PACK_TYPE	1103	Upgrade pack type does not match
NET_ERR_DEV_FORMATTING	1104	Device is formatting
NET_ERR_MISMATCH_UPGRADE_PACK_VERSION	1105	Upgrade pack version does not match
NET_ERR_PT_LOCKED	1106	PT is locked.
NET_DVR_ERR_ILLEGAL_VERIFICATION_CODE	1111	Illegal verification code. Change the verification code.
NET_DVR_ERR_LACK_VERIFICATION_CODE	1112	No verification code. Enter the verification code.
NET_DVR_ERR_FORBIDDEN_IP	1113	The IP address cannot be configured.
NET_DVR_ERR_HTTP_BKN_EXCEED_ONE	1125	Up to one channel's ANR function can be enabled.
NET_DVR_ERR_FORMATTING_FAILED	1131	Formatting HDD failed.
NET_DVR_ERR_ENCRYPTED_FORMATTING_FAILED	1132	Formatting encrypted HDD failed.
NET_DVR_ERR_WRONG_PASSWORD	1133	Verifying password of SD card failed. Incorrect password.
NET_ERR_SEARCHING_MODULE	1201	Searching peripherals.
NET_ERR_REGISTERING_MODULE	1202	Registering external module
NET_ERR_GETTING_ZONES	1203	Getting arming region parameter
NET_ERR_GETTING_TRIGGERS	1204	Getting trigger

Error Name	Error Code	Error Description
NET_ERR_ARMED_STATUS	1205	System is in arming status
NET_ERR_PROGRAM_MODE_STATUS	1206	System is in programming mode
NET_ERR_WALK_TEST_MODE_STATUS	1207	System is in pacing measuring mode
NET_ERR_BYPASS_STATUS	1208	Bypass status
NET_ERR_DISABLED_MODULE_STATUS	1209	Function not enabled
NET_ERR_NOT_SUPPORT_OPERATE_ZONE	1210	Operation is not supported by arming region
NET_ERR_NOT_SUPPORT_MOD_MODULE_ADDR	1211	Module address cannot be modified
NET_ERR_UNREGISTERED_MODULE	1212	Module is not registered
NET_ERR_PUBLIC_SUBSYSTEM_ASSOCIATE_SELF	1213	Public sub system associate with its self
NET_ERR_EXCEEDS_ASSOCIATE_SUBSYSTEM_NUM	1214	Number of associated public sub system reached upper limit
NET_ERR_BE_ASSOCIATED_BY_PUBLIC_SUBSYSTEM	1215	Sub system is associated by other public sub system
NET_ERR_ZONE_FAULT_STATUS	1216	Arming region is in failure status
NET_ERR_SAME_EVENT_TYPE	1217	Same event type exists in enable event trigger alarm output and disable event trigger alarm output
NET_ERR_ZONE_ALARM_STATUS	1218	Arming region is in alarm status
NET_ERR_EXPANSION_BUS_SHORT_CIRCUIT	1219	Extension bus short-circuit
NET_ERR_PWD_CONFLICT	1220	Password conflict, e.g., lock password is identical with duress password
NET_ERR_DETECTOR_GISTERED_BY_OTHER_ZONE	1221	Detector has been registered by other arming regions
NET_ERR_DETECTOR_GISTERED_BY_OTHER_PU	1222	Detector has been registered by other hosts
NET_ERR_DETECTOR_DISCONNECT	1223	Detector offline
NET_ERR_CALL_BUSY	1224	Device in call

Error Name	Error Code	Error Description
NET_ERR_FILE_NAME	1357	File name error, empty or invalid
NET_ERR_BROADCAST_BUSY	1358	Device in broadcast
NET_DVR_ERR_LANENUM_EXCEED	1400	Over the number of lanes.
NET_DVR_ERR_PRAREA_EXCEED	1401	Recognition area is too large.
NET_DVR_ERR_LIGHT_PARAM	1402	Signal lamp access parameters error.
NET_DVR_ERR_LANE_LINE_INVALID	1403	Lane configuration error.
NET_DVR_ERR_STOP_LINE_INVALID	1404	Stop line configuration error.
NET_DVR_ERR_LEFTORRIGHT_LINE_INVALID	1405	Turn left / right boundary configuration error.
NET_DVR_ERR_LANE_NO_REPEAT	1406	Overlay lane number repetition.
NET_DVR_ERR_PRAREA_INVALID	1407	The polygon does not meet the requirements.
NET_DVR_ERR_LIGHT_NUM_EXCEED	1408	Video detection of traffic light signal exceeds the maximum number of.
NET_DVR_ERR_SUBLIGHT_NUM_INVALID	1409	Video detection of traffic signal lamp lights are not legitimate
NET_DVR_ERR_LIGHT_AREASIZE_INVALID	1410	The size of the video detection of traffic light input signal lamp is not valid.
NET_DVR_ERR_LIGHT_COLOR_INVALID	1411	The color of the video detection of traffic light input signal lamp color is not legitimate.
NET_DVR_ERR_LIGHT_DIRECTION_INVALID	1412	The direction property of the video detection of traffic light input light is not valid.
NET_DVR_ERR_LACK_IOABILITY	1413	Lack of IO ability.
NET_DVR_ERR_FTP_PORT	1414	FTP port error.
NET_DVR_ERR_FTP_CATALOGUE	1415	FTP catalogue error.
NET_DVR_ERR_FTP_UPLOAD_TYPE	1416	FTP upload type error.
NET_DVR_ERR_FLASH_PARAM_WRITE	1417	Setting param flash write error.

Error Name	Error Code	Error Description
NET_DVR_ERR_FLASH_PARAM_READ	1418	Getting param flash read error.
NET_DVR_ERR_PICNAME_DELIMITER	1419	Pic name delimiter error.
NET_DVR_ERR_PICNAME_ITEM	1420	Pic name item error.
NET_DVR_ERR_PLATE_RECOGNIZE_TYPE	1421	Plate recognize type error.
NET_DVR_ERR_CAPTURE_TIMES	1422	Capture times error.
NET_DVR_ERR_LOOP_DISTANCE	1423	Loop distance error.
NET_DVR_ERR_LOOP_INPUT_STATUS	1424	Loop input status error.
NET_DVR_ERR_RELATE_IO_CONFLICT	1425	Related IO conflict.
NET_DVR_ERR_INTERVAL_TIME	1426	Interval time error.
NET_DVR_ERR_SIGN_SPEED	1427	Sign speed error.
NET_DVR_ERR_PIC_FLIP	1428	Flip is used.
NET_DVR_ERR_RELATE_LANE_NUMBER	1429	Related lane number error.
NET_DVR_ERR_TRIGGER_MODE	1430	Trigger mode error.
NET_DVR_ERR_DELAY_TIME	1431	Delay time error.
NET_DVR_ERR_EXCEED_RS485_COUNT	1432	Exceed RS485 count.
NET_DVR_ERR_RADAR_TYPE	1433	Radar type error.
NET_DVR_ERR_RADAR_ANGLE	1434	Radar angle error.
NET_DVR_ERR_RADAR_SPEED_VALID_TIME	1435	Radar speed valid time error.
NET_DVR_ERR_RADAR_LINE_CORRECT	1436	Radar line correct error.
NET_DVR_ERR_RADAR_CONST_CORRECT	1437	Radar const correct error.
NET_DVR_ERR_RECORD_PARAM	1438	Record param error.
NET_DVR_ERR_LIGHT_WITHOUT_COLOR_AND_DIRECTION	1439	Light number and other param error.

Error Name	Error Code	Error Description
NET_DVR_ERR_LIGHT_WITHOUT_DETECTION_REGION	1440	Light number and detection region error.
NET_DVR_ERR_RECOGNIZE_PROVINCE_PARAM	1441	Plate recognize Province param error.
NET_DVR_ERR_SPEED_TIMEOUT	1442	IO Speed TimeOut Param error.
NET_DVR_ERR_NTP_TIMEZONE	1443	NTP TimeZone Param error.
NET_DVR_ERR_NTP_INTERVAL_TIME	1444	NTP Interval Time error.
NET_DVR_ERR_NETWORK_CARD_NUM	1445	Network Card Num error.
NET_DVR_ERR_DEFAULT_ROUTE	1446	Default Route error.
NET_DVR_ERR_BONDING_WORK_MODE	1447	Banding Work Mode error.
NET_DVR_ERR_SLAVE_CARD	1448	Sub-Card error.
NET_DVR_ERR_PRIMARY_CARD	1449	Primary Card error.
NET_DVR_ERR_DHCP_PPOE_WORK	1450	DHCP and PPOE not Meanwhile start.
NET_DVR_ERR_NET_INTERFACE	1451	Net Interface invalid.
NET_DVR_ERR_MTU	1452	Invalid MTU parameters.
NET_DVR_ERR_NETMASK	1453	Netmask address invalid.
NET_DVR_ERR_IP_INVALID	1454	IP address invalid.
NET_DVR_ERR_MULTICAST_IP_INVALID	1455	Multicast IP address invalid.
NET_DVR_ERR_GATEWAY_INVALID	1456	Gateway address invalid.
NET_DVR_ERR_DNS_INVALID	1457	DNS Param invalid.
NET_DVR_ERR_ALARMHOST_IP_INVALID	1458	AlarmHost IP invalid.
NET_DVR_ERR_IP_CONFLICT	1459	IP address Conflict.
NET_DVR_ERR_NETWORK_SEGMENT	1460	IP not support Multi Network segment.
NET_DVR_ERR_NETPORT	1461	NetPort error.
NET_DVR_ERR_PPPOE_NOSUPPORT	1462	PPPoE is not supported.

Error Name	Error Code	Error Description
NET_DVR_ERR_DOMAINNAME_NOSUPPORT	1463	Not Support Domain Name.
NET_DVR_ERR_NO_SPEED	1464	Speed Not Enabled.
NET_DVR_ERR_IOSTATUS_INVALID	1465	IO Status invalid.
NET_DVR_ERR_BURST_INTERVAL_INVALID	1466	Burst Interval invalid.
NET_DVR_ERR_RESERVE_MODE	1467	Reserve Mode invalid.
NET_DVR_ERR_LANE_NO	1468	Lane No error.
NET_DVR_ERR_COIL_AREA_TYPE	1469	Coil Area Type error.
NET_DVR_ERR_TRIGGER_AREA_PARAM	1470	Trigger Area Param error.
NET_DVR_ERR_SPEED_LIMIT_PARAM	1471	Speed Limit Param error.
NET_DVR_ERR_LANE_PROTOCOL_TYPE	1472	Lane Protocol Type error.
NET_DVR_ERR_INTERVAL_TYPE	1473	Capture Interval Type error.
NET_DVR_ERR_INTERVAL_DISTANCE	1474	Capture Interval Distance error.
NET_DVR_ERR_RS485_ASSOCIATE_DEVTYPE	1475	Rs485 Associate DevType error.
NET_DVR_ERR_RS485_ASSOCIATE_LANENO	1476	Rs485 Associate LaneNo error.
NET_DVR_ERR_LANENO_ASSOCIATE_MULTIRS485	1477	LaneNo Associate MulitRs485 error.
NET_DVR_ERR_LIGHT_DETECTION_REGION	1478	Light Detection Region error.
NET_DVR_ERR_DN2D_NOSUPPORT	1479	UnSupport Capture Frame 2D Noise Reduction.
NET_DVR_ERR_IRISMODE_NOSUPPORT	1480	UnSupport scene Mode.
NET_DVR_ERR_WB_NOSUPPORT	1481	UnSupport White Balance Mode.
NET_DVR_ERR_IO_EFFECTIVENESS	1482	IO Effectiveness invalid.

Error Name	Error Code	Error Description
NET_DVR_ERR_LIGHTNO_MAX	1483	Access Detector Lights Red / Yellow Overrun.
NET_DVR_ERR_LIGHTNO_CONFLICT	1484	Access Detector Lights Red / Yellow Conflict.
NET_DVR_ERR_CANCEL_LINE	1485	Trigger straight line error.
NET_DVR_ERR_STOP_LINE	1486	Subject line area stop line error.
NET_DVR_ERR_RUSH_REDLIGHT_LINE	1487	Red light trigger lines error.
NET_DVR_ERR_IOOUTNO_MAX	1488	IO out port error.
NET_DVR_ERR_IOOUTNO_AHEADTIME_MAX	1489	IO out ahead time error.
NET_DVR_ERR_IOOUTNO_IOWORKTIME	1490	IO out inwork time error.
NET_DVR_ERR_IOOUTNO_FREQMULTI	1491	IO out frequency multiplication error.
NET_DVR_ERR_IOOUTNO_DUTYRATE	1492	IO out duty rate error.
NET_DVR_ERR_VIDEO_WITH_EXPOSURE	1493	IO out work mode error.
NET_DVR_ERR_PLATE_BRIGHTNESS_WITHOUT_FLASHDET	1494	Plate enable in plate compensate mode on.
NET_DVR_ERR_RECOGNIZE_TYPE_PARAM	1495	Recognize Type error.
NET_DVR_ERR_PALTE_RECOGNIZE_AREA_PARAM	1496	Plate Recognize Area Param error.
NET_DVR_ERR_PORT_CONFLICT	1497	Port Conflict.
NET_DVR_ERR_LOOP_IP	1498	IP cannot be the loopback address.
NET_DVR_ERR_DRIVELINE_SENSITIVE	1499	Driveline sensitivity error.
NET_ERR_VQD_TIME_CONFLICT	1500	The time period conflict.
NET_ERR_VQD_PLAN_NO_EXIST	1501	The diagnostic plan of VQD dese not exist.
NET_ERR_VQD_CHAN_NO_EXIST	1502	The channel dese not exist.

Error Name	Error Code	Error Description
NET_ERR_VQD_CHAN_MAX	1503	The total number of VQD plans exceeds the max limit.
NET_ERR_VQD_TASK_MAX	1504	The total number of VQD tasks exceeds the max limit.
NET_DVR_ERR_EXCEED_MAX_CAPTURE_TIMES	1600	Capture times exceed 2 in flash mode.
NET_DVR_ERR_REDAR_TYPE_CONFLICT	1601	Radar type conflict.
NET_DVR_ERR_LICENSE_PLATE_NULL	1602	The license plate is null.
NET_DVR_ERR_WRITE_DATABASE	1603	Failed to write data into the database.
NET_DVR_ERR_LICENSE_EFFECTIVE_TIME	1604	The effective time of license plate error.
NET_DVR_ERR_PRERECORDED_STARTTIME_LONG	1605	The pre recorded start time is greater than the number of illegal capture.
NET_DVR_ERR_TRIGGER_RULE_LINE	1606	Trigger rule line error.
NET_DVR_ERR_LEFTRIGHT_TRIGGERLINE_NOTVERTICAL	1607	Left and right trigger line is not vertical.
NET_DVR_ERR_FLASH_LAMP_MODE	1608	Flash lamp mode error.
NET_DVR_ERR_ILLEGAL_SNAPSHOT_NUM	1609	Illegal capture number error.
NET_DVR_ERR_ILLEGAL_DETECTION_TYPE	1610	Illegal detection type error.
NET_DVR_ERR_POSITIVEBACK_TRIGGERLINE_HIGH	1611	Positive back to trigger line height error.
NET_DVR_ERR_MIXEDMODE_CAPTYPE_ALLTARGETS	1612	Mixed mode only supports capture type all targets.
NET_DVR_ERR_CARSIGNSPEED_GREATERTHAN_LIMITSPEED	1613	Car sign speed greater than speed limit value.
NET_DVR_ERR_BIGCARSIGNSPEED_GREATERTHAN_LIMITSPEED	1614	Big car sign speed limit greater than speed limit value.
NET_DVR_ERR_BIGCARSIGNSPEED_GREATERTHAN_CARSIGNSPEED	1615	Big car sign speed limit is greater than the car sign speed limit value.

Error Name	Error Code	Error Description
NET_DVR_ERR_BIGCARLIMITSPEED_GREATERTHAN_CARLIMITSPEED	1616	Big car speed limit value is greater than the car speed limit value.
NET_DVR_ERR_BIGCARLOWSPEEDLIMIT_GREATERTHAN_CARLOWSPEEDLIMIT	1617	Big car low speed limit value is greater than the car low speed limit value.
NET_DVR_ERR_CARLIMITSPEED_GREATERTHAN_EXCEPHIGHSPEED	1618	Car speed limit greater than exception high speed value.
NET_DVR_ERR_BIGCARLIMITSPEED_GREATERTHAN_EXCEPHIGHSPEED	1619	Big car speed limit greater than exception high speed value.
NET_DVR_ERR_STOPLINE_MORETHAN_TRIGGERLINE	1620	Stopping more than straight lines trigger lines.
NET_ERR_TIME_OVERLAP	1900	Time periods overlap
NET_ERR_HOLIDAY_PLAN_OVERLAP	1901	Holiday plan overlap
NET_ERR_CARDNO_NOT_SORT	1902	Card number is not sorted
NET_ERR_CARDNO_NOT_EXIST	1903	Card number does not exist
NET_ERR_ILLEGAL_CARDNO	1904	Card number error
NET_ERR_ZONE_ALARM	1905	Arming region is in arming status (parameter cannot be modified)
NET_ERR_ZONE_OPERATION_NOT_SUPPORT	1906	Arming region does not support the operation
NET_ERR_INTERLOCK_ANTI_CONFLICT	1907	Interlock and anti-passback configuration conflict
NET_ERR_DEVICE_CARD_FULL	1908	Card full (return after card reached 10,000)
NET_ERR_HOLIDAY_GROUP_DOWNLOAD	1909	Failed to download holiday group
NET_ERR_LOCAL_CONTROL_OFF	1910	Distributed access controller offline
NET_ERR_LOCAL_CONTROL_DISADD	1911	Distributed access controller is not added
NET_ERR_LOCAL_CONTROL_HASADD	1912	Distributed access controller is added
NET_ERR_LOCAL_CONTROL_DOORNO_CONFLICT	1913	Conflict with added distributed access controller

Error Name	Error Code	Error Description
NET_ERR_LOCAL_CONTROL_COMMUNICATION_FAIL	1914	Distributed access controller communication failed
NET_ERR_OPERAND_INEXISTENCE	1915	Operation object does not exist (operation to door, alarm output, alarm input, return when the object is not added)
NET_ERR_LOCAL_CONTROL_OVER_LIMIT	1916	Distributed access controller exceeded device capability upper limit
NET_ERR_DOOR_OVER_LIMIT	1917	Door exceeded device capability upper limit
NET_ERR_ALARM_OVER_LIMIT	1918	Alarm input and output exceeded device capability upper limit
NET_ERR_LOCAL_CONTROL_ADDRESS_INCONFORMITY_TYPE	1919	Distributed access controller address does not match with type
NET_ERR_NOT_SUPPORT_ONE_MORE_CARD	1920	not support one person multi-card
NET_ERR_DELETE_NO_EXISTENCE_FACE	1921	The face picture does not exist.
NET_ERR_DOOR_SPECIAL_PASSWORD_REPEAT	1922	Repeated door door duress code, the super password, or the dismiss code.
NET_ERR_AUTH_CODE_REPEAT	1923	Repeated device authentication code
NET_ERR_DEPLOY_EXCEED_MAX	1924	No more devices can be armed.
NET_ERR_NOT_SUPPORT_DEL_FP_BY_ID	1925	The fingerprint module does not support deleting fingerprint by finger ID.
NET_ERR_TIME_RANGE	1926	Invalid range of the effective period.
NET_ERR_CAPTURE_TIMEOUT	1927	Collection timed out.
NET_ERR_LOW_SCORE	1928	Low quality of collected data.
NET_ERR_OFFLINE_CAPTURING	1929	The device is collecting data offline and cannot respond.
NET_DVR_ERR_OUTDOOR_COMMUNICATION	1950	Communication exception with outdoor terminal

Error Name	Error Code	Error Description
NET_DVR_ERR_ROOMNO_UNDEFINED	1951	Room number is not set
NET_DVR_ERR_NO_CALLING	1952	No call
NET_DVR_ERR_RINGING	1953	Ringling
NET_DVR_ERR_IS_CALLING_NOW	1954	Call in progress
NET_DVR_ERR_LOCK_PASSWORD_WRONG	1955	Incorrect smart lock password
NET_DVR_ERR_CONTROL_LOCK_FAILURE	1956	Lock control failure
NET_DVR_ERR_CONTROL_LOCK_OVERTIME	1957	Lock control timed out
NET_DVR_ERR_LOCK_DEVICE_BUSY	1958	Smart lock device busy
NET_DVR_ERR_UNOPEN_REMOTE_LOCK_FUNCTION	1959	Remote lock control not enabled
NET_DVR_ERR_FILE_NOT_COMPLETE	2100	Downloaded file is incomplete
NET_DVR_ERR_IPC_EXIST	2101	The camera already exists
NET_DVR_ERR_ADD_IPC	2102	Camera has been added to the channel
NET_DVR_ERR_OUT_OF_RES	2103	Not enough network bandwidth
NET_DVR_ERR_CONFLICT_TO_LOCALIP	2104	IP address of camera conflicts with that of DVR
NET_DVR_ERR_IP_SET	2105	Invalid IP address
NET_DVR_ERR_PORT_SET	2106	Invalid port number
NET_ERR_WAN_NOTSUPPORT	2107	Not in the same LAN, cannot set security question or export GUID file
NET_ERR_MUTEX_FUNCTION	2108	Mutually exclusive function
NET_ERR_QUESTION_CONFIGNUM	2109	Error in number of security question configurations
NET_ERR_FACECHAN_NORESOURCE	2110	All the face VCA channels are occupied.
NET_ERR_DATA_CALLBACK	2111	Data is calling back.

Error Name	Error Code	Error Description
NET_ERR_ATM_VCA_CHAN_IS_RELATED	2112	The VCA channel is already linked.
NET_ERR_ATM_VCA_CHAN_IS_OVERLAPED	2113	The VCA channel is already overlayed.
NET_ERR_FACE_CHAN_UNOVERLAP_EACH_OTHER	2114	The face channels cannot be overlayed.
NET_DVR_SMD_ENCODING_NORESOURCE	2116	Insufficient SMD encoding resource
NET_DVR_SMD_DECODING_NORESOURCE	2117	Insufficient SMD decoding resource
NET_DVR_FACELIB_DATA_PROCESSING	2118	Face picture library data is in processing
NET_DVR_ERR_LARGE_TIME_DIFFERENCE	2119	There is a great time difference between device and server.
NET_DVR_NO_SUPPORT_WITH_PLAYBACK	2120	It is not supported. Playback is enabled.
NET_DVR_CHANNEL_NO_SUPPORT_WITH_SMD	2121	It is not supported. SMD of channel is enabled.
NET_DVR_CHANNEL_NO_SUPPORT_WITH_FD	2122	It is not supported. Face capture of channel is enabled.
NET_DVR_ILLEGAL_PHONE_NUMBER	2123	Invalid telephone number
NET_DVR_ILLEGAL_CERTIFICATE_NUMBER	2124	Invalid ID No.
NET_DVR_ERR_CHANNEL_RESOLUTION_NO_SUPPORT	2125	The channel resolution is not supported
NET_DVR_ERR_CHANNEL_COMPRESSION_NO_SUPPORT	2126	The channel encoding format is not supported
NET_DVR_ERR_CLUSTER_DEVICE_TOO_LESS	2127	Deleting is not allowed. The number of devices is not enough
NET_DVR_ERR_CLUSTER_DEL_DEVICE_CM_PLAYLOAD	2128	Deleting is not allowed. The device is cluster host.
NET_DVR_ERR_CLUSTER_DEVNUM_OVER_UPPER_LIMIT	2129	No more devices can be added.

Error Name	Error Code	Error Description
NET_DVR_ERR_CLUSTER_DEVICE_TYPE_INCONFORMITY	2130	Device type mismatched.
NET_DVR_ERR_CLUSTER_DEVICE_VERSION_INCONFORMITY	2131	Device version mismatched.
NET_DVR_ERR_CLUSTER_IP_CONFLICT	2132	Cluster system IP address conflict: ipv4 address conflict, invalid ipv6.
NET_DVR_ERR_CLUSTER_IP_INVALID	2133	Invalid cluster system IP address: invalid ipv4, invalid ipv6.
NET_DVR_ERR_CLUSTER_PORT_CONFLICT	2134	Cluster system port conflict
NET_DVR_ERR_CLUSTER_PORT_INVALID	2135	Invalid cluster system port
NET_DVR_ERR_CLUSTER_USERNAEM_OR_PASSWORD_INVALID	2136	Invalid user name or password
NET_DVR_ERR_CLUSTER_DEVICE_ALREADY_EXIST	2137	The device already exists.
NET_DVR_ERR_CLUSTER_DEVICE_NOT_EXIST	2138	The device does not exist.
NET_DVR_ERR_CLUSTER_NON_CLUSTER_MODE	2139	The device working mode is not the cluster mode .
NET_DVR_ERR_CLUSTER_IP_NOT_SAME_LAN	2140	IP addresses are in different LAN. Building cluster or extending capacity for NVRs in different LAN is not allowed.
NET_DVR_ERR_IDENTITY_KEY	2147	Incorrect interaction password
NET_DVR_MISSING_IDENTITY_KEY	2148	Interaction password is missing
NET_DVR_ERR_CAPTURE_PACKAGE_FAILED	2141	Capturing packets failed.
NET_DVR_ERR_CAPTURE_PACKAGE_PROCESSING	2142	Capturing packet.
NET_DVR_ERR_SAFETY_HELMET_NO_RESOURCE	2143	No enough hard hat detection resource.

Error Name	Error Code	Error Description
NET_DVR_NO_SUPPORT_WITH_ABSTRACT	2144	This function is not supported. Video synopsis is already enabled.
NET_DVR_INSUFFICIENT_DEEP_LEARNING_RESOURCES	2146	No more deep learning resources can be added.
NET_DVR_NO_SUPPORT_WITH_PERSON_DENSITY_DETECT	2149	People gathering density is enabled, it is not supported
NET_DVR_IPC_RESOLUTION_OVERFLOW	2150	The network camera resolution is too large
NET_DVR_IPC_BITRATE_OVERFLOW	2151	The network camera bitrate is too large
NET_DVR_ERR_INVALID_TASKID	2152	Invalid taskID
NET_DVR_PANEL_MODE_NOT_CONFIG	2153	The ATM panel mode is not configured.
NET_DVR_NO_HUMAN_ENGINES_RESOURCE	2154	No enough engine resource
NET_DVR_ERR_TASK_NUMBER_OVERFLOW	2155	No more task data is allowed
NET_DVR_ERR_COLLISION_TIME_OVERFLOW	2156	Collision time is over the limit
NET_DVR_ERR_EVENT_NOTSUPPORT	2159	Subscribing alarm/event is not supported.
NET_DVR_IPC_NUM_REACHES_LIMIT	2184	The max. number of network camera channels reached.
NET_DVR_IOT_NUM_REACHES_LIMIT	2185	The max. number of IoT channels reached
NET_DVR_IOT_CHANNEL_DEVICE_EXIST	2186	Device of the IoT channel already exists.
NET_DVR_IOT_CHANNEL_DEVICE_NOT_EXIST	2187	Device of the IoT channel does not exist.
NET_DVR_INVALID_IOT_PROTOCOL_TYPE	2188	Invalid IoT protocol type
NET_DVR_INVALID_EZVIZ_SECRET_KEY	2189	Invalid verification code

Error Name	Error Code	Error Description
NET_DVR_DUPLICATE_IOT_DEVICE	2190	Duplicated IoT device
NET_DVR_ERROR_NEED_DOUBLE_VERIFICATION	2206	Double verification is required
NET_DVR_NO_DOUBLE_VERIFICATION_USER	2207	No double verification user
NET_DVR_TIMESPAN_NUM_OVER_LIMIT	2209	Max. number of time buckets reached
NET_DVR_CHANNEL_NUM_OVER_LIMIT	2210	Max. number of channels reached
NET_DVR_NO_SEARCH_ID_RESOURCE	2211	Insufficient searchID resources
NET_DVR_SWITCH_TIMEDIFF_LESS_LIMIT	2249	Time difference between power on and off should be less than 10 minutes.
NET_DVR_NO_SUPPORT_DELETE_STRANGER_LIB	2262	Deleting stranger library is not supported
NET_DVR_NO_SUPPORT_CREATE_STRANGER_LIB	2263	Creating stranger library is not supported
NET_DVR_SSD_FILE_SYSTEM_ERROR	2266	SSD file system error
NET_DVR_INSUFFICIENT_SSD__FOR_FPD	2267	Insufficient SSD space for person frequency detection
NET_DVR_SMRDISK_NOT_SUPPORT_RAID	2269	SMR disk does not support RAID.
NET_DVR_ERR_NOTSUPPORT_DEICING	3001	Device does not support deicing function under current status.(Deicing function is only supported under the power status of POE+, AC24V, and DC12V).
NET_DVR_ERR_THERMENABLE_CLOSE	3002	Temperature measurement function is not enabled. (The enable function in NET_DVR_THERMOMETRY_BASICPARAM is not turned on)
NET_DVR_ERR_PANORAMIC_LIMIT_OPERATED	3004	Panoramic map and limit cannot be operated at same time

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

RTSP Communication Library Related Errors

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

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

Error Name	Error Code	Error Description
		exceeds the max connection number.
NET_DVR_RTSP_PLAYERROR	430	RTSP PLAY interaction error.
NET_DVR_RTSP_PLAYSENDTIMEOUT	431	Sending "RTSP PLAY" is timeout.
NET_DVR_RTSP_PLAYSENDERERROR	432	Sending "RTSP PLAY" error.
NET_DVR_RTSP_PLAYRECVTIMEOUT	433	Receiving "RTSP PLAY" is timeout.
NET_DVR_RTSP_PLAYRECVDATALOST	434	Receiving data of "RTSP PLAY" error.
NET_DVR_RTSP_PLAYRECVERROR	435	Failed to receive "RTSP PLAY".
NET_DVR_RTSP_PLAYSERVERERR	436	"RTSP PLAY" device returns the error that values 401 or 501.
NET_DVR_RTSP_TEARDOWNERROR	440	RTSP TEARDOWN interaction error.
NET_DVR_RTSP_TEARDOWNSENDTIMEOUT	441	Sending "RTSP TEARDOWN" is timeout.
NET_DVR_RTSP_TEARDOWNSENDERERROR	442	Sending "RTSP TEARDOWN" error.
NET_DVR_RTSP_TEARDOWNRECVTIMEOUT	443	Receiving "RTSP TEARDOWN" is timeout.
NET_DVR_RTSP_TEARDOWNRECVDATALOST	444	Receiving data of "RTSP TEARDOWN" error.
NET_DVR_RTSP_TEARDOWNRECVERROR	445	Failed to receive "RTSP TEARDOWN".
NET_DVR_RTSP_TEARDOWNSERVERERR	446	"RTSP TEARDOWN" device returns the error that values 401 or 501.

Software Decoding Library Related Errors

Error Name	Error Code	Error Description
NET_PLAYM4_NOERROR	500	No error.
NET_PLAYM4_PARA_OVER	501	Input parameter is invalid.
NET_PLAYM4_ORDER_ERROR	502	API calling order error.
NET_PLAYM4_TIMER_ERROR	503	Failed to create multimedia clock.

Error Name	Error Code	Error Description
NET_PLAYM4_DEC_VIDEO_ERROR	504	Failed to decode video data.
NET_PLAYM4_DEC_AUDIO_ERROR	505	Failed to decode audio data.
NET_PLAYM4_ALLOC_MEMORY_ERROR	506	Failed to allocate memory.
NET_PLAYM4_OPEN_FILE_ERROR	507	Failed to open the file.
NET_PLAYM4_CREATE_OBJ_ERROR	508	Failed to create thread event.
NET_PLAYM4_CREATE_DDRAW_ERROR	509	Failed to create DirectDraw object.
NET_PLAYM4_CREATE_OFFSCREEN_ERROR	510	Failed to create backstage cache for OFFSCREEN mode.
NET_PLAYM4_BUF_OVER	511	Buffer overflow, failed to input stream.
NET_PLAYM4_CREATE_SOUND_ERROR	512	Failed to create audio equipment.
NET_PLAYM4_SET_VOLUME_ERROR	513	Failed to set the volume.
NET_PLAYM4_SUPPORT_FILE_ONLY	514	This API can be called only for file playback mode.
NET_PLAYM4_SUPPORT_STREAM_ONLY	515	This API can be called only when playing stream.
NET_PLAYM4_SYS_NOT_SUPPORT	516	Not support by the system. Decoder can only work on the system above Pentium 3.
NET_PLAYM4_FILEHEADER_UNKNOWN	517	There is no file header.
NET_PLAYM4_VERSION_INCORRECT	518	The version mismatch between decoder and encoder.
NET_PLAYM4_INIT_DECODER_ERROR	519	Failed to initialize the decoder.
NET_PLAYM4_CHECK_FILE_ERROR	520	The file is too short, or the stream data is unknown.
NET_PLAYM4_INIT_TIMER_ERROR	521	Failed to initialize multimedia clock.
NET_PLAYM4_BLT_ERROR	522	BLT failure.

Error Name	Error Code	Error Description
NET_PLAYM4_UPDATE_ERROR	523	Failed to update overlay surface
NET_PLAYM4_OPEN_FILE_ERROR_MULTI	524	Failed to open video & audio stream file.
NET_PLAYM4_OPEN_FILE_ERROR_VIDEO	525	Failed to open video stream file.
NET_PLAYM4_JPEG_COMPRESS_ERROR	526	JPEG compression error.
NET_PLAYM4_EXTRACT_NOT_SUPPORT	527	Don't support the version of this file.
NET_PLAYM4_EXTRACT_DATA_ERROR	528	Extract video data failed.

Container Format Conversion Library Related Errors

Error Name	Error Code	Error Description
NET_CONVERT_ERROR_NOT_SUPPORT	581	This container format is not supported.

Two Way Audio Library Related Errors

Error Name	Error Code	Error Description
NET_AUDIOINTERCOM_OK	600	No error.
NET_AUDIOINTECOM_ERR_NOTSUPORT	601	Not support.
NET_AUDIOINTECOM_ERR_ALLOC_MEMERY	602	Memory allocation error.
NET_AUDIOINTECOM_ERR_PARAMETER	603	Parameter error.
NET_AUDIOINTECOM_ERR_CALL_ORDER	604	API calling order error.
NET_AUDIOINTECOM_ERR_FIND_DEVICE	605	No audio device
NET_AUDIOINTECOM_ERR_OPEN_DEVICE	606	Failed to open the audio device
NET_AUDIOINTECOM_ERR_NO_CONTEXT	607	Context error.
NET_AUDIOINTECOM_ERR_NO_WAVFILE	608	WAV file error.
NET_AUDIOINTECOM_ERR_INVALID_TYPE	609	The type of WAV parameter is invalid

Error Name	Error Code	Error Description
NET_AUDIOINTECOM_ERR_ENCODE_FAIL	610	Failed to encode data
NET_AUDIOINTECOM_ERR_DECODE_FAIL	611	Failed to decode data
NET_AUDIOINTECOM_ERR_NO_PLAYBACK	612	Failed to play audio
NET_AUDIOINTECOM_ERR_DENOISE_FAIL	613	Failed to denoise
NET_AUDIOINTECOM_ERR_UNKOWN	619	Unknown

QoS Stream Control Library Related Errors

Error Name	Error Code	Error Description
NET_QOS_ERR_SCHEDPARAMS_BAD_MINIMUM_INTERVAL	678	Incorrect predefined minimum interval.
NET_QOS_ERR_SCHEDPARAMS_BAD_FRACTION	679	Incorrect predefined score.
NET_QOS_ERR_SCHEDPARAMS_INVALID_BANDWIDTH	680	Invalid predefined bandwidth.
NET_QOS_ERR_PACKET_TOO_BIG	687	The packet size is too large.
NET_QOS_ERR_PACKET_LENGTH	688	Invalid packet size.
NET_QOS_ERR_PACKET_VERSION	689	Incorrect packet versio information.
NET_QOS_ERR_PACKET_UNKNOW	690	Unknown packet.
NET_QOS_ERR_OUTOFMEM	695	Out of memory.
NET_QOS_ERR_LIB_NOT_INITIALIZED	696	The library is not initialized.
NET_QOS_ERR_SESSION_NOT_FOUND	697	No session found.
NET_QOS_ERR_INVALID_ARGUMENTS	698	Invalid parameters.
NET_QOS_ERROR	699	QoS Stream Control Library error.
NET_QOS_OK	700	No error.

NPQ (Network Protocol Quality) Related Error

Error Name	Error Code	Error Description
NET_ERR_NPQ_PARAM	8001	NPQ library: Incorrect parameter.
NET_ERR_NPQ_SYSTEM	8002	NPQ library: Operating system error.
NET_ERR_NPQ_GENRAL	8003	NPQ library: Internal error.
NET_ERR_NPQ_PRECONDITION	8004	NPQ library: Calling sequence error.
NET_ERR_NPQ_NOTSUPPORT	8005	NPQ library: This function is not supported.
NET_ERR_NPQ_NOTCALLBACK	8100	No data is called back.
NET_ERR_NPQ_LOADLIB	8101	Loading NPQ library failed.
NET_ERR_NPQ_STREAM_CLOSE	8104	The NPQ function of this stream is not enabled.
NET_ERR_NPQ_MAX_LINK	8110	No more streaming channel's NPQ function can be enabled.
NET_ERR_NPQ_STREAM_CFG_CONFLICT	8111	The configured encoding parameters conflicted.

Appendix D. Request URIs

Description	URI	Method	Request and Response Message
Get device information.	/ISAPI/System/deviceInfo	GET	XML_DeviceInfo XML_ResponseStatus
Edit device information.	/ISAPI/System/deviceInfo	PUT	-
Control PTZ.	/ISAPI/PTZCtrl/channels/<ID>/continuous	PUT	XML_ResponseStatus
Get preset list.	/ISAPI/PTZCtrl/channels/<ID>/presets	GET	XML_PTZPresetList XML_ResponseStatus
Manage all configured presets.	/ISAPI/PTZCtrl/channels/<ID>/presets	POST	-
Delete all presets.	/ISAPI/PTZCtrl/channels/<ID>/presets	DELETE	-
Add a preset.	/ISAPI/PTZCtrl/channels/<ID>/presets/<ID>	PUT	XML_ResponseStatus
Delete a preset.	/ISAPI/PTZCtrl/channels/<ID>/presets/<ID>	DELETE	XML_ResponseStatus
Get a preset.	/ISAPI/PTZCtrl/channels/<ID>/presets/<ID>	GET	-
Call a preset.	/ISAPI/PTZCtrl/channels/<ID>/presets/<ID>/goto	PUT	XML_ResponseStatus
Get partition status.	/ISAPI/SecurityCP/status/subSystems?format=json	GET	JSON_SubSysList JSON_ResponseStatus
Arm a partition.	/ISAPI/SecurityCP/control/arm/<ID>?ways=<string>&format=json	PUT	JSON_ResponseStatus
Disarm a partition.	/ISAPI/SecurityCP/control/disarm/<ID>?format=json	PUT	JSON_ResponseStatus
Clear partition alarms.	/ISAPI/SecurityCP/control/clearAlarm/<ID>?format=json	PUT	JSON_ResponseStatus
Get zone status	/ISAPI/SecurityCP/status/zones?format=json	GET	JSON_ZoneList JSON_ResponseStatus

Search partition status according to conditions.	/ISAPI/SecurityCP/status/zones?format=json	POST	-
Zone bypass.	/ISAPI/SecurityCP/control/bypass?format=json	PUT	JSON_ResponseStatus
Recover bypass of multiple zones.	/ISAPI/SecurityCP/control/bypassRecover?format=json	PUT	JSON_ResponseStatus
Get relay status by specific conditions.	/ISAPI/SecurityCP/status/outputStatus?format=json	POST	JSON_OutputSearch JSON_ResponseStatus
Control relay in batch.	/ISAPI/SecurityCP/control/outputs?format=json	POST	JSON_ResponseStatus
Get the information of all I/O output ports.	/ISAPI/System/IO/outputs	GET	XML_IOOutputPortList XML_ResponseStatus
Get status of a specific alarm output.	/ISAPI/System/IO/outputs/<ID>/status	GET	XML_IOPortStatus XML_ResponseStatus
Manually trigger a specific alarm output.	/ISAPI/System/IO/outputs/<ID>/trigger	PUT	XML_ResponseStatus
Get device time zone.	/ISAPI/System/time	GET	XML_TimeData XML_ResponseStatus
Get or set device time parameters.	/ISAPI/System/time	PUT	-
Operations about management of all digital channels.	/ISAPI/ContentMgmt/InputProxy/channels	GET	XML_InputProxyChannelList XML_ResponseStatus
Configure operations about management of all digital channels.	/ISAPI/ContentMgmt/InputProxy/channels	PUT	-
Create digital channels	/ISAPI/ContentMgmt/InputProxy/channels	POST	-

Get status of all digital channels.	/ISAPI/ContentMgmt/InputProxy/channels/status	GET	XML_ InputProxyChannelStatusList XML_ResponseStatus
Refresh the video mode manually before playback.	/ISAPI/ContentMgmt/record/control/manualRefresh/channels/<ID>	PUT	XML_ResponseStatus
Search for access control events.	/ISAPI/AccessControl/AcsEvent?format=json	POST	JSON_AcsEvent XML_ResponseStatus
Search for person information.	/ISAPI/AccessControl/UserInfo/Search?format=json	POST	JSON_UserInfoSearch XML_ResponseStatus

D.1 http://<ipAddress>:<portNo>/<url>

HTTP listening sever sends alarm information to alarm center.

Request URL Definition

Table C-1 POST http://<ipAddress>:<portNo>/<url>

Method	POST
Description	HTTP listening sever sends alarm information to alarm center.
Query	None
Request	None
Response	Succeeded: <u>XML_EventNotificationAlert_AlarmEventInfo</u> or <u>JSON_EventNotificationAlert_Alarm/EventInfo</u> Failed: <u>XML_ResponseStatus</u>

Remarks


- The <ipAddress> in the request URL refers to the IP address or domain name of HTTP listening server, the <portNo> is the port No. of HTTP listening server, and the <url> represents the streaming URL, which is configured via the HTTP listening server.
- The default port No. is 80, so the request URL without port No. is also valid.

D.2 /ISAPI/Event/notification/alertStream

Get the uploaded heartbeat or alarm/event information.

Request URI Definition

Table C-2 GET /ISAPI/Event/notification/alertStream

Method	GET
Description	Get the heartbeat or uploaded alarm/event information.
Query	None.
Request	None.
Response	<p>Option 1: <u><i>XML_EventNotificationAlert_AlarmEventInfo</i></u> or <u><i>XML_EventNotificationAlert_HeartbeatInfo</i></u></p> <p>Option 2: <u><i>JSON_EventNotificationAlert_Alarm/EventInfo</i></u></p> <p> Note</p> <p>The messages here only show the format of alarm/event information to be uploaded. For details, refer to the corresponding alarm/event configuration chapters.</p>

Remarks

- After calling this URI, a persistent connection is set up between the device and the platform, and the alarm or event information will be uploaded from device continuously once the alarm is triggered or event occurred.
- You can check if the XML response message is the heartbeat information according to the nodes `<eventType>` and `<eventState>`. If the values of these two node are "videoloss" and "inactive", respectively, the returned message is the heartbeat information.

D.3 /ISAPI/Event/notification/httpHosts

Get or set parameters of all HTTP listening servers, add a HTTP listening server, and delete all HTTP listening servers.

Request URI Definition

Table C-3 GET /ISAPI/Event/notification/httpHosts

Method	GET
Description	Get parameters of all HTTP listening servers.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is

	1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode.
Request	None
Response	Succeeded: <u><i>XML_HttpHostNotificationList</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Table C-4 PUT /ISAPI/Event/notification/httpHosts

Method	PUT
Description	Set parameters of all HTTP listening servers.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode.
Request	<u><i>XML_HttpHostNotificationList</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

Table C-5 POST /ISAPI/Event/notification/httpHosts

Method	POST
Description	Add a HTTP listening server.
Query	security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode.
Request	<u><i>XML_HttpHostNotification</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

Table C-6 DELETE /ISAPI/Event/notification/httpHosts

Method	DELETE
Description	Delete all HTTP listening servers.
Query	None
Request	None
Response	<u><i>XML_ResponseStatus</i></u>

D.4 /ISAPI/Event/notification/httpHosts/<ID>/test

Check whether the HTTP listening server is working normally.

Request URI Definition

Table C-7 POST /ISAPI/Event/notification/httpHosts/<ID>/test

Method	POST
Description	Check whether the HTTP listening server is working normally.
Query	None
Request	<u><i>XML_HttpHostNotification</i></u>
Response	Succeeded: <u><i>XML_HttpHostTestResult</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the HTTP listening server ID.

D.5 /ISAPI/Event/notification/httpHosts/capabilities

Get the configuration capabilities of all HTTP listening servers.

Request URI Definition

Table C-8 GET /ISAPI/Event/notification/httpHosts/capabilities

Method	GET
Description	Get the configuration capabilities of all HTTP listening servers.

Query	None
Request	None
Response	Succeeded: <u><i>XML_HttpHostNotificationCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

D.6 /ISAPI/Event/schedules/faceLib/<ID>/<FDID>

Get or set arming schedule of a face picture comparison alarm.

Request URI Definition

Table C-9 GET /ISAPI/Event/schedules/faceLib/<ID>/<FDID>

Method	GET
Description	Get the arming schedule of a face picture comparison alarm.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_Schedule</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Table C-10 PUT /ISAPI/Event/schedules/faceLib/<ID>/<FDID>

Method	PUT
Description	Set the arming schedule of a face picture comparison alarm
Query	None.
Request	<u><i>XML_Schedule</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

Remarks

- The <ID> in the URI is the video channel ID.
- The <FDID> in the URI is the face picture library ID, which is returned by device. But the user can also define the library ID and apply the custom ID to the device, and then the device will link the <FDID> with the custom ID, so you can configure and manage the library via the custom ID. If the face picture library ID is customized, the corresponding URI should be */ISAPI/Event/schedules/faceLib/<ID>/<FDID>?FDType=custom*.

D.7 /ISAPI/Event/schedules/HBDLib-<ID>/<HBDID>

Get or set the arming schedule of a specified human body picture library.

Request URI Definition

Table C-11 GET /ISAPI/Event/schedules/HBDLib-<ID>/<HBDID>

Method	GET
Description	Get the arming schedule of a specified human body picture library.
Query	HBDType (optional): determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library according to customHBDID .
Request	None.
Response	Succeeded: <u><i>XML_Schedule</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Table C-12 PUT /ISAPI/Event/schedules/HBDLib-<ID>/<HBDID>

Method	PUT
Description	Set the arming schedule of a specified human body picture library.
Query	HBDType (optional): determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library according to customHBDID .
Request	<u><i>XML_Schedule</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

Remarks

- The <ID> in the URI refers to channel No.
- The <HBDID> in the URI refers to human body picture library ID, when the **HBDType**=custom is applied, it refers to **customHBDID** (ID of custom human body picture library).

D.8 /ISAPI/Event/triggers/faceLib-<channelID>/<FDID>

Get, set, or delete the linkage action parameters of specified face picture library.

Request URI Definition

Table C-13 GET /ISAPI/Event/triggers/faceLib-<channelID>/<FDID>

Method	GET
Description	Get the linkage action parameters of a specified face picture library.
Query	FDType (optional)
Request	None.
Response	Succeeded: <u><i>XML_EventTrigger</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Table C-14 PUT /ISAPI/Event/triggers/faceLib-<channelID>/<FDID>

Method	PUT
Description	Set the linkage action of a specified face picture library.
Query	FDType (optional)
Request	<u><i>XML_EventTrigger</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

Table C-15 DELETE /ISAPI/Event/triggers/faceLib-<channelID>/<FDID>

Method	DELETE
Description	Delete the linkage action of a specified face picture library.
Query	FDType (optional)
Request	None
Response	<u><i>XML_ResponseStatus</i></u>

Remarks

- The <channelID> in the URI refers to channel No.
- The <FDID> in the URI refers to face picture library ID, which is returned by the device. But the user can also define the library ID and apply the custom ID to the device, and then the device will link the <FDID> with the custom ID, so you can configure and manage the library via the custom ID. If the face picture library ID is customized, the corresponding URI should be */ISAPI/Event/triggers/faceLib-<channelID>/<FDID>?FDType=custom*.

D.9 /ISAPI/Event/triggers/HBDLib-<ID>/<HBDID>

Get or set the linkage configuration parameters of a human body picture library.

Request URI Definition

Table C-16 GET /ISAPI/Event/triggers/HBDLib-<ID>/<HBDID>

Method	GET
Description	Get the linkage configuration parameters of a human body picture library.
Query	HBDType (optional): determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library according to customHBDID .
Request	None.
Response	Succeeded: <u><i>XML_EventTrigger</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Table C-17 PUT /ISAPI/Event/triggers/HBDLib-<ID>/<HBDID>

Method	PUT
Description	Set the linkage configuration parameters of a human body picture library.
Query	HBDType (optional): determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library according to customHBDID .
Request	<u><i>XML_EventTrigger</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

Remarks

- The <ID> in the URI refers to channel No.
- The <HBDID> in the URI refers to human body picture library ID, when the **HBDType**=custom is applied, it refers to **customHBDID** (ID of custom human body picture library).

D.10 /ISAPI/Event/triggersCap

Get alarm linkage capability.

Request URI Definition

Table C-18 GET /ISAPI/Event/triggersCap

Method	GET
Description	Get alarm linkage capability.
Query	None
Request	None
Response	Succeeded: <u><i>XML_EventTriggersCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

D.11 /ISAPI/Intelligent/channels/<ID>/capabilities

Get the intelligent capability of device.

Request URI Definition

Table C-19 GET /ISAPI/Intelligent/channels/<ID>/capabilities

Method	GET
Description	Get the intelligent capability of device.
Query	None
Request	None.
Response	Succeeded: <u><i>XML_IntelliCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Remarks

The <ID> in the URI refers to the intelligent channel ID.

D.12 /ISAPI/Intelligent/channels/<ID>/manualPersonArmingTrack/capabilities?format=json

Get the capability of manual person arming.

Request URI Definition

Table C-20 GET /ISAPI/Intelligent/channels/<ID>/manualPersonArmingTrack/capabilities?format=json

Method	GET
Description	Get the capability of manual person arming.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_ManualPersonArmingTrackCap</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

Remarks

The <ID> in the URI refers to the channel No.

D.13 /ISAPI/Intelligent/channels/<ID>/manualPersonArmingTrack?format=json

Start manual person arming.

Request URI Definition

Table C-21 PUT /ISAPI/Intelligent/channels/<ID>/manualPersonArmingTrack?format=json

Method	PUT
Description	Start manual person arming.
Query	format: determine the format of request or response message.
Request	<i><u>JSON_ManualPersonArmingTrack</u></i>
Response	<i><u>JSON_ResponseStatus</u></i>

Remarks

The <ID> in the URI refers to the channel No.

D.14 /ISAPI/Intelligent/channels/<ID>/personArmingTrack/advanceConfiguration/capabilities?format=json

Get the configuration capability of person arming advanced parameters.

Request URI Definition

Table C-22 GET /ISAPI/Intelligent/channels/<ID>/personArmingTrack/advanceConfiguration/capabilities?format=json

Method	GET
Description	Get the configuration capability of person arming advanced parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_AdvanceConfigurationCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the URI refers to the channel No.

D.15 /ISAPI/Intelligent/channels/<ID>/personArmingTrack/advanceConfiguration?format=json

Get or set the advanced parameters of person arming.

Request URI Definition

Table C-23 GET /ISAPI/Intelligent/channels/<ID>/personArmingTrack/advanceConfiguration?format=json

Method	GET
Description	Get the advanced parameters of person arming.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_AdvanceConfiguration</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table C-24 PUT /ISAPI/Intelligent/channels/<ID>/personArmingTrack/advanceConfiguration?format=json

Method	PUT
Description	Set the advanced parameters of person arming.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_AdvanceConfiguration</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the URI refers to the channel No.

D.16 /ISAPI/Intelligent/channels/<ID>/personArmingTrack/capabilities?format=json

Get configuration capability of auto person arming.

Request URI Definition

Table C-25 GET /ISAPI/Intelligent/channels/<ID>/personArmingTrack/capabilities?format=json

Method	GET
Description	Get configuration capability of auto person arming.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_PersonArmingTrackCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the URI refers to the channel No.

D.17 /ISAPI/Intelligent/channels/<ID>/personArmingTrack?format=json

Get or set the auto person arming parameters.

Request URI Definition

Table C-26 GET /ISAPI/Intelligent/channels/<ID>/personArmingTrack?format=json

Method	GET
Description	Get the auto person arming parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON PersonArmingTrack</u></i> Failed: <i><u>JSON ResponseStatus</u></i>

Table C-27 PUT /ISAPI/Intelligent/channels/<ID>/personArmingTrack?format=json

Method	PUT
Description	Set the auto person arming parameters.
Query	format: determine the format of request or response message.
Request	<i><u>JSON PersonArmingTrack</u></i>
Response	<i><u>JSON ResponseStatus</u></i>

Remarks

The <ID> in the URI refers to the channel No.

D.18 /ISAPI/Intelligent/channels/<ID>/restoreAlgLibParam/<NAME>

Restore the algorithm library parameters to default according to algorithm library name.

Request URI Definition

Table C-28 GET /ISAPI/Intelligent/channels/<ID>/restoreAlgLibParam/<NAME>

Method	GET
Description	Restore the algorithm library parameters to default according to algorithm library name.
Query	format: determine the format of request or response message.
Request	None
Response	<i><u>JSON ResponseStatus</u></i>

Remarks

- The <ID> in the URI refers to the channel No.
- The <NAME> in the URI refers to the algorithm library name: personArming-person arming.
- You can get the intelligent function capability and check if this function is supported via the request URI: GET /ISAPI/Intelligent/channels/<ID>/capabilities , when the node: <RestoreAlgLibParam> exist, the function is supported.

D.19 /ISAPI/Intelligent/HBDLib/<HBDID>/capacity?format=json

Get the capacity of a specified human body picture library.

Request URI Definition

Table C-29 GET /ISAPI/Intelligent/HBDLib/<HBDID>/capacity?format=json

Method	GET
Description	Get the capacity of a specified human body picture library.
Query	format : determine the format of request or response message. HBDType (optional): determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library.
Request	None
Response	Succeeded: <u>JSON_HBPCapacity</u> Failed: <u>JSON_ResponseStatus</u>

Remarks

The <HBDID> in the URI refers to human body picture library ID, when the value of **HBDType** is custom, it refers to **customHBDID** (ID of custom human body picture library).

D.20 /ISAPI/Intelligent/HBDLib/<HBDID>/HBP/<HBPID>?format=json

Get, edit, or delete the information related to a record in human body picture library.

Request URI Definition

Table C-30 GET /ISAPI/Intelligent/HBDLib/<HBDID>/HBP/<HBPID>?format=json

Method	GET
Description	Get the information related to a record in human body picture library.
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2. HBDType (optional): determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library.
Request	None.
Response	Succeeded: <u><i>JSON_HumanAppendData</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table C-31 PUT /ISAPI/Intelligent/HBDLib/<HBDID>/HBP/<HBPID>?format=json

Method	PUT
Description	Edit the information related to a record in human body picture library.
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2. HBDType (optional): determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library.

Request	<u><i>JSON_HumanAppendData</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Table C-32 DELETE /ISAPI/Intelligent/HBDLib/<HBDID>/HBP/<HBPID>?format=json

Method	DELETE
Description	Delete a record in human body picture library.
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2. HBDType (optional): determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library.
Request	None
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <HBDID> in the URI indicates the ID of human body picture library.

D.21 /ISAPI/Intelligent/HBDLib/<HBDID>?format=json

Manage a specified human body picture library.

Request URI Definition

Table C-33 GET /ISAPI/Intelligent/HBDLib/<HBDID>?format=json

Method	GET
Description	Get the basic information of a specified human body picture library.
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is

	<p>1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode.</p> <p>iv: the initialization vector, and it is required when security is 1 or 2.</p> <p>HBDType (optional): determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library.</p>
Request	None
Response	<p>Succeeded: <u><i>JSON_HBDLibBaseCfg</i></u></p> <p>Failed: <u><i>JSON_ResponseStatus</i></u></p>

Table C-34 PUT /ISAPI/Intelligent/HBDLib/<HBDID>?format=json

Method	PUT
Description	Edit the basic information of a specified human body picture library.
Query	<p>format: determine the format of request or response message.</p> <p>security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode.</p> <p>iv: the initialization vector, and it is required when security is 1 or 2.</p> <p>HBDType (optional): determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library.</p>
Request	<u><i>JSON_HBDLibBaseCfg</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Table C-35 DELETE /ISAPI/Intelligent/HBDLib/<HBDID>?format=json

Method	DELETE
Description	Delete a specified human body picture library.
Query	format : determine the format of request or response message.
Request	None
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <**HBDID**> in the URI refers to human body picture library ID, when the value of **HBDType** is custom, it refers to **customHBDID** (ID of custom human body picture library).

D.22 /ISAPI/Intelligent/HBDLib/capabilities?format=json

Get the configuration capability of human body picture library.

Request URI Definition

Table C-36 GET /ISAPI/Intelligent/HBDLib/capabilities?format=json

Method	GET
Description	Get the configuration capability of human body picture library.
Query	format : determine the format of request or response message.
Request	None
Response	Succeeded: <i><u>JSON_HBDLibCap</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

D.23 /ISAPI/Intelligent/HBDLib/extend?format=json

Get or edit the custom information of human body picture library.

Request URI Definition

Table C-37 GET /ISAPI/Intelligent/HBDLib/extend?format=json

Method	GET
Description	Get the custom information of human body picture library.
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.

Request	None.
Response	Succeeded: <u><i>JSON_PersonInfoExtend</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table C-38 PUT /ISAPI/Intelligent/HBDLib/extend?format=json

Method	PUT
Description	Edit the custom information of human body picture library.
Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	<u><i>JSON_PersonInfoExtend</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

D.24 /ISAPI/Intelligent/HBDLib/HBDSearch?format=json

Search for human body picture(s).

Request URI Definition

Table C-39 POST /ISAPI/Intelligent/HBDLib/HBDSearch?format=json

Method	POST
Description	Search for human body picture(s).
Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.

Request	<u><i>JSON_HBDSearchDescription</i></u>
Response	Succeeded: <u><i>JSON_HBDSearchResult</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

D.25 /ISAPI/Intelligent/HBDLib/HBP/delete?format=json

Delete picture(s) of human body picture library.

Request URI Definition

Table C-40 PUT /ISAPI/Intelligent/HBDLib/HBP/delete?format=json

Method	PUT
Description	Delete picture(s) of human body picture library.
Query	format : determine the format of request or response message.
Request	<u><i>JSON_HBDInfo</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

D.26 /ISAPI/Intelligent/HBDLib/HBPUploadData?format=json

Import a human body picture with linked information to human body picture library.

Request URI Definition

Table C-41 POST /ISAPI/Intelligent/HBDLib/HBPUploadData?format=json

Method	POST
Description	Import a human body picture with linked information to human body picture library
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.

	type (optional): processing mode, when the type is concurrent, it indicates only importing picture without modeling. mode (optional): import mode, when the mode is cover, it indicates overwriting earliest picture with imported picture.
Request	<u><i>JSON_HBPUploadData</i></u>
Response	Succeeded: <u><i>JSON_HBPUploadResult</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

D.27 /ISAPI/Intelligent/HBDLib/manualModeling/<HBDID>/HBP/<HBPID>?format=json

Build a model for a specified human body picture.

Request URI Definition

Table C-42 GET /ISAPI/Intelligent/HBDLib/manualModeling/<HBDID>/HBP/<HBPID>?format=json

Method	GET
Description	Build a model for a specified human body picture.
Query	format : determine the format of request or response message. HBDType (optional): determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library.
Request	None.
Response	Succeeded: <u><i>JSON_ModelingStatus</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

- The <HBDID> in the URI refers to human body picture library ID, when the value of **HBDType** is custom, it refers to **customHBDID** (ID of custom human body picture library).
- The <HBPID> in the URI refers to human body picture ID, when the value of **HBDType** is custom, it refers to **customHBPID** (custom human body picture ID).

D.28 /ISAPI/Intelligent/HBDLib/manualModeling/<HBDID>?format=json

Build models for multiple pictures of specified human body picture library in a batch.

Request URI Definition

Table C-43 POST /ISAPI/Intelligent/HBDLib/manualModeling/<HBDID>?format=json

Method	POST
Description	Build models for multiple pictures of specified human body picture library in a batch.
Query	format : determine the format of request or response message. HBDType : determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library according to customHBDID .
Request	<i><u>JSON_ManualModeling</u></i>
Response	Succeeded: <i><u>JSON_ManualModeling_TaskInfo</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

Remarks

The <HBDID> in the URI refers to human body picture library ID, when the **HBDType**=custom is applied, it refers to **customHBDID** (ID of custom human body picture library).

D.29 /ISAPI/Intelligent/HBDLib/manualModeling/progressState/<taskID>?format=json

Search for modeling task progress according to task ID.

Request URI Definition

Table C-44 GET /ISAPI/Intelligent/HBDLib/manualModeling/progressState/<taskID>?format=json

Method	GET
Description	Search for modeling task progress according to task ID.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_ModelingRangeTaskState</u></i>

	Failed: <i><u>JSON_ResponseStatus</u></i>
--	---

Remarks

The <taskID> in the URI refers to modeling task ID, which is returned by the URI */ISAPI/Intelligent/HBDLib/manualModeling/<HBDID>?format=json* or */ISAPI/Intelligent/HBDLib/manualModelingRangeTask?format=json* .

D.30 /ISAPI/Intelligent/HBDLib/manualModelingRangeTask?format=json

Build models for multiple human body pictures of different human body picture libraries.

Request URI Definition

Table C-45 POST /ISAPI/Intelligent/HBDLib/manualModelingRangeTask?format=json

Method	POST
Description	Build models for multiple human body pictures of different human body picture libraries.
Query	format: determine the format of request or response message.
Request	<i><u>JSON_ModelingRangeList</u></i>
Response	Succeeded: <i><u>JSON_ManualModeling_TaskInfo</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

D.31 /ISAPI/Intelligent/HBDLib/modelingStatus/<HBDID>/HBP/<HBPID>?format=json

Search for modeling status of a specified human body picture.

Request URI Definition

Table C-46 GET /ISAPI/Intelligent/HBDLib/modelingStatus/<HBDID>/HBP/<HBPID>?format=json

Method	GET
Description	Search for modeling status of a specified human body picture.
Query	format: determine the format of request or response message.

	HBDType (optional): determine human body picture library type, when HBDType is custom, it indicates operations of a custom human body picture library according to customHBDID .
Request	None.
Response	Succeeded: <u><i>JSON_ModelingStatus</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

- The <**HBDID**> in the URI refers to human body picture library ID, when the value of **HBDType** is custom, it refers to **customHBDID** (ID of custom human body picture library).
- The <**HBPID**> in the URI refers to human body picture ID, when the value of **HBDType** is custom, it refers to **customHBPID** (custom human body picture ID).

D.32 /ISAPI/Intelligent/HBDLib/modelingStatus?format=json

Search for modeling status.

Request URI Definition

Table C-47 POST /ISAPI/Intelligent/HBDLib/modelingStatus?format=json

Method	POST
Description	Search for modeling status.
Query	format : determine the format of request or response message.
Request	<u><i>JSON_HBDModelingStatusSearchDescription</i></u>
Response	Succeeded: <u><i>JSON_HBDModelingStatusSearchResult</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

D.33 /ISAPI/Intelligent/HBDLib?format=json

Manage all human body picture libraries.

Request URI Definition

Table C-48 GET /ISAPI/Intelligent/HBDLib?format=json

Method	GET
Description	Get the basic information of all human body picture libraries.
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	None
Response	Succeeded: <u><i>JSON_HBDLibBaseCfgList</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table C-49 PUT /ISAPI/Intelligent/HBDLib?format=json

Method	PUT
Description	Edit the basic information of all human body picture libraries.
Query	format : determine the format of request or response message. security : the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv : the initialization vector, and it is required when security is 1 or 2.
Request	<u><i>JSON_HBDLibBaseCfgList</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Table C-50 POST /ISAPI/Intelligent/HBDLib?format=json

Method	POST
Description	Create multiple human body picture libraries.

Query	format: determine the format of request or response message. security: the version No. of encryption scheme. When security does not exist, it indicates that the data is not encrypted; when security is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when security is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. iv: the initialization vector, and it is required when security is 1 or 2.
Request	<u>JSON_CreateHBDLibList</u>
Response	Succeeded: <u>JSON_HBDLibInfoList</u> Failed: <u>JSON_ResponseStatus</u>

Table C-51 DELETE /ISAPI/Intelligent/HBDLib?format=json

Method	DELETE
Description	Delete all human body picture libraries.
Query	format: determine the format of request or response message.
Request	None
Response	<u>JSON_ResponseStatus</u>

D.34 /ISAPI/System/capabilities

Get device capability.

Request URI Definition

Table C-52 GET /ISAPI/System/capabilities

Method	GET
Description	Get device capability.
Query	None
Request	None.
Response	Succeeded: <u>XML_DeviceCap</u> Failed: <u>XML_ResponseStatus</u>

Appendix E. Request and Response Messages

E.1 JSON_AdvanceConfiguration

JSON message about person arming advanced parameters

```
{
  "AdvanceConfiguration":{
    "intervalTime": ,
    /*optional, int, tracking information upload interval,range: [100, 1000], unit:
    millisecond, default value: 1000ms*/
    "waitingTime": ,
    /*optional, int, dwell time after tracking failed, during the dwell time, the
    PTZ channels stays at the position where the target tracking fails, range:
    [1,60], unit: second, default value: 30s*/
    "lowConfidenceStopInterval": ,
    /*optional, int, wait to stop tracking if low validity, range: [15,60], default
    value: 18, unit: number of algorithm processed frames; during tracking, if the
    number of frames with low validity exceeds lowConfidenceStopInterval, the
    device will stop tracking*/
    "lowConfidenceDropInterval": ,
    /*optional, int, wait to lower tracking speed if low validity, range: [0,60],
    default value: 30, unit: number of algorithm processed frames; during tracking,
    if the number of frames with low validity exceeds lowConfidenceDropInterval,
    the device will lower the tracking speed*/
    "displayTarget": ,
    /*optional, boolean, whether to display target information during DSP live
    view; default value: false*/
    "trackingRatio": ,
    /*optional, dependent, int, tracking zoom ratio, range: [15,50], default value:
    50; it is valid when the value of ratioCtrlMode is "targetFrame"*/
    "ratioCtrlMode": "",
    /*optional, string, zooming control modes: "targetFrame"-by target frame,
    "verticalAngle"-by tilt angle; default value: "targetFrame"*/
    "panoramaTargetFrameEnabled": ,
    /*optional, boolean, whether to enable displaying a frame around the target on
    Panorama; default value: false*/
    "trackSnapEnabled": ,
    /*optional, boolean, whether to enable tracking capture*/
    "faceSnapThreshold": ,
    /*optional, int, face capture threshold, range: [0,100], default value: 20*/
    "bodySnapThreshold": ,
    /*optional, int, human body capture threshold, range: [0,100], default value:
    20*/
    " maxTrackTime":
    /*optional, int, the tracking duration, range: [0,300], default value: 300*/
  }
}
```


E.2 JSON_AdvanceConfigurationCap

JSON message about configuration capability of person arming advanced parameters

```
{
  "AdvanceConfigurationCap":{
    /*required, configuration capability of person arming advanced parameters*/
    "intervalTime":{
      /*optional, int, tracking information upload interval,range: [100, 1000], unit:
      millisecond, default value: 1000ms*/
      "@min":100,
      "@max":1000,
      "@def":1000
    },
    "waitingTime":{
      /*optional, int, dwell time after tracking failed, during the dwell time, the
      PTZ channels stays at the position where the target tracking fails, range:
      [1,60], unit: second, default value: 30s*/
      "@min":1,
      "@max":60,
      "@def":30
    },
    "lowConfidenceStopInterval":{
      /*optional, int, wait to stop tracking if low validity, range: [15,60], default
      value: 18, unit: number of algorithm processed frames; during tracking, if the
      number of frames with low validity exceeds lowConfidenceStopInterval, the
      device will stop tracking*/
      "@min":15,
      "@max":60,
      "@def":18
    },
    "lowConfidenceDropInterval":{
      /*optional, int, wait to lower tracking speed if low validity, range: [0,60],
      default value: 30, unit: number of algorithm processed frames; during tracking,
      if the number of frames with low validity exceeds lowConfidenceDropInterval,
      the device will lower the tracking speed*/
      "@min":0,
      "@max":60,
      "@def":30
    },
    "displayTarget":{
      /*optional, boolean, whether to display target information during DSP live
      view; default value: false*/
      "@opt": [true, false]
    },
    "trackingRatio":{
      /*optional, dependent, int, tracking zoom ratio, range: [15,50], default value:
      50; it is valid when the value of ratioCtrlMode is "tgtFrame"*/
      "@min":15,
      "@max":50,
      "@def":50
    }
  }
}
```

```
    },
    "ratioCtrlMode":{
/*optional, string, zooming control modes: "targetFrame"-by target frame,
"vertgicalAngle"-by tilt angle; default value: "targetFrame"*/
    "@opt": ["targetFrame", "vertgicalAngle"],
    "@def": "targetFrame"
    },
    "panoramaTargetFrameEnabled":{
/*optional, boolean, whether to enable displaying a frame around the target on
Panorama; default value: false*/
    "@opt": [true, false]
    },
    "trackSnapEnabled":{
/*optional, boolean, whether to enable tracking capture*/
    "@opt": [true, false]
    },
    "faceSnapThreshold":{
/*optional, int, face capture threshold, range: [0,100], default value: 20*/
    "@min":0,
    "@max":100,
    "@def":20
    },
    "bodySnapThreshold":{
/*optional, int, human body capture threshold, range: [0,100], default value:
20*/
    "@min":0,
    "@max":100,
    "@def":20
    },
    "maxTrackTime":{
/*optional, int, the tracking duration, range: [0,300], default value: 300*/
    "@min":0,
    "@max":300,
    "@def":300
    }
  }
}
```

E.3 JSON_CreateHBDLibList

JSON message about information of creating human body picture library

```
{
  "CreateHBDLibList": [
/*required, information of creating human body picture library*/
    {
      "id": ,
/*required, sub item No. in the list, starts from 1*/
      "HBDLibName": "",
/*required, string, name of human body picture library, the maximum length is
```

```
64*/
    "HBDLibThreshold": ,
/*required, integer32, similarity threshold for human body picture comparison,
range: [0,100]*/
    "customHBDID": "",
/*optional, string, ID of custom human body picture library, the maximum length
is 128*/
    "customInfo": ""
/*optional, string, additional information of human body picture library, the
maximum length is 32*/
    }
]
}
```

E.4 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*/
}
```

E.5 JSON_EventNotificationAlert_manualPersonArmingTrackMsg

JSON message about alarm details of manual person arming

Content-Type: multipart/form-data;boundary=<frontier>

--<frontier>

Content-Disposition: form-data; name="dailyFaceTracking"

Content-Type: application/json

Content-Length: text_length

```
{
  "ipAddress": "",
  /*required, device IPv4 address , string, the maximum size is 32*/
  "ipv6Address": "",
  /*optional, device IPv6 address, string, the maximum size is 128*/
  "portNo": ,
  /*optional, device port No., integer32*/
  "protocolType": "",
  /*optional, string, protocol type: "HTTP", "HTTPS", "EHome"-ISUPSDK protocol,
the maximum size is 32*/
  "macAddress": "",
  /*optional, string, MAC address, the maximum length is 32 bytes*/
  "channelID": ,
  /*optional, integer32, device channel No. that triggers alarm*/
  "dateTime": "",
  /*required, string, alarm triggering time based on ISO8601 time format, the
maximum length is 32*/
  "activePostCount": ,
  /*required, integer32, number of the same uploaded alarm*/
  "eventType": "",
  /*required, string, event type: "manualPersonArmingTrack"-manual person arming,
the maximum length is 128 bytes*/
  "eventState": "",
  /*required, string, event triggering status: "active"-triggered, "inactive"-not
triggered (refers to heartbeat data), the maximum length is 32 bytes*/
  "eventDescription": "",
  /*required, string, event description: manual person arming track*/
  "channelName": "",
  /*required, channel name (camera name), which should be the same as the
<channelName> in message XML_StreamingChannel (related URI: /ISAPI/Streaming/
channels/<ID>)*
  "deviceID": "",
  /*optional, device ID (PUID), which should be returned during integration of
ISUPSDK and should be the same as the <deviceID> in XML_Ehome (related URI: /
```

```
ISAPI/System/Network/Ehome)*/
    "persistentEventStatus": "",
    /*dependent, event status: "started", "ongoing", "ended"; it is valid when the
    value of eventType "personArmingTrack"*/
    "uid": "",
    /*optional, string, event uploading ID, which can consists of time (accurate to
    millisecond) and random digits, the maximum value size is 64 bytes*/
    "ManualPersonArmingTrack":{
    /*required, information of manual person arming tracking*/
    "targetUUID": "",
    /*optional, string, UUID of person arming target, it is unique*/
    "TargetGPSInfo":{
    /*required, target position information*/
    "longitudeType": "",
    /*required, string, longitude type: "E"-east, "W"-west*/
    "latitudeType": "",
    /*required, string, latitude type: "S"-south, "N"-north*/
    "Longitude":{
    /*required, longitude value*/
    "degree": ,
    /*required, int, degree: [0,180]*/
    "minute": ,
    /*required, int, minute: [0,59]*/
    "sec":
    /*required, float, second: [0.000000,60.000000], corrects to 6 decimal places*/
    },
    "Latitude":{
    /*required, latitude value*/
    "degree": ,
    /*required, int, degree: [0,180]*/
    "minute": ,
    /*required, int, minute: [0,59]*/
    "sec":
    /*required, float, second: [0.000000,60.000000], corrects to 6 decimal places*/
    }
    }
    }

    /*****Person
    Information*****/
    "PersonInfo": {
    /*optional, target person information; when the value of persistentEventStatus
    is "started", the PersonInfo will be uploaded; when the value of
persistentEventStatus is "ended", the PersonInfo will not be uploaded unless
    the tracking capture has enabled; when the value of persistentEventStatus is
    "ongoing", the PersonInfo will not be uploaded*/
    "armingType": "",
    /*optional, string, arming type: "face"-upload face information, "human"-upload
    human body information, "face_human"-upload face and human body information*/
    "Face": {
    /*optional, face information*/
    "FaceCaptureResult": {
    /*optional, face capture result*/
```

```

/*****Face Rectangle Frame
Information*****/
    "Rect": {
/*optional, face thumbnail rectangle frame*/
        "height": ,
/*optional, float, frame height, range: [0.000,1.000]*/
        "width": ,
/*optional, float, frame width, range: [0.000,1.000]*/
        "x": ,
/*required, float, X-coordinate of upper-left corner, range: [0.000,1.000],
corrects to three decimal place*/
        "y":
/*required, float, Y-coordinate of upper-left corner, range: [0.000,1.000],
corrects to three decimal place*/
    },
    "FacePictureRect": {
/*optional, face rectangle frame*/
        "height": ,
/*optional, float, frame height, range: [0.000,1.000]*/
        "width": ,
/*optional, float, frame width, range: [0.000,1.000]*/
        "x": ,
/*required, float, X-coordinate of upper-left corner, range: [0.000,1.000],
corrects to three decimal place*/
        "y":
/*required, float, Y-coordinate of upper-left corner, range: [0.000,1.000],
corrects to three decimal place*/
    },
/*****Face Attributes*****/
    "age": {
/*optional, age information*/
        "value": ,
/*optional, integer32, age*/
        "ageGroup": ""
/*optional, string, age range; the maximum size is 32 bytes*/
    },
    "gender":{
/*optional, gender*/
        "value":""
/*optional, string, gender: "male", "female", "unknown"; the maximum length is
32*/
    },
    "glass":{
/*optional, wear glasses or not*/
        "value":""
/*optional, string, wearing glasses or not: "no", "yes", "sunglasses"-wearing
sunglasses; the maximum length is 32*/
    },
    "smile":{
/*optional, smiling or not*/
        "value":""
/*optional, string, smiling or not: "no", "yes", "unknown"; the maximum length

```

```
is 32*/
    },
    "mask": {
/*optional, wearing mask or not*/
        "value": ""
/*optional, string, wearing mask or not: "no", "yes", "unknown"; the maximum
length is 32*/
    },
    "faceExpression":{
/*optional, face expression*/
        "value": ""
/*optional, string, face expression; the maximum size is 32*/
    },
    "hat": {
/*optional, whether the person is wearing a hat*/
        "value": ""
/*optional, string, whether the person is wearing a hat: "no", "yes",
"unknown"; the maximum length is 32*/
    },
    /*****Face Picture
Information*****/
    "FaceImage":{
/*optional, face thumbnail, resource transmission module; when the returned
alarm message consists additional resource (e.g., picture), this node should be
returned*/
        "resourcesContentType": "",
/*dependent, string, resource transmission type: "url", "binary"; when the
returned alarm message consists additional resource (e.g., picture), this node
should be returned*/
        "resourcesContent": ""
/*dependent, string, resource ID; when the returned alarm message consists
additional resource (e.g., picture), this node should be returned; when the
value of resourcesContentType is "binary", it must be the same as Content-ID of
picture, when the value of resourcesContentType is url, its value is the
picture URL*/
    },
    "FaceBackgroundImage":{
/*optional, face background picture, resource transmission module; when the
returned alarm message consists additional resource (e.g., picture), this node
should be returned*/
        "resourcesContentType": "",
/*dependent, string, resource transmission type: "url", "binary"; when the
returned alarm message consists additional resource (e.g., picture), this node
should be returned*/
        "resourcesContent": ""
/*dependent, string, resource ID; when the returned alarm message consists
additional resource (e.g., picture), this node should be returned; when the
value of resourcesContentType is "binary", it must be the same as Content-ID of
picture, when the value of resourcesContentType is url, its value is the
picture URL*/
    }
}
```

```
    },
    /*******Human Body
    Information******/
    "Human":{
        "HumanCaptureResult":{
            /*optional, human body capture result*/
            /*******Human Body Rectangle Frame
            Information******/
            "Rect":{
                /*optional, rectangle frame of human body thumbnail*/
                "height": ,
                /*optional, float, frame height, range: [0.000,1.000]*/
                "width": ,
                /*optional, float, frame width, range: [0.000,1.000]*/
                "x": ,
                /*required, float, X-coordinate of upper-left corner, range: [0.000,1.000],
                corrects to three decimal place*/
                "y":
                /*required, float, Y-coordinate of upper-left corner, range: [0.000,1.000],
                corrects to three decimal place*/
            },
            /*******Human Body
            Attributes******/
            "jacketColor":{
                /*optional, top's color*/
                "value": ""
                /*optional, string, top's color, the maximum length is 32*/
            },
            "trousersColor":{
                /*optional, trousers' color*/
                "value": ""
                /*optional, string, trousers' color, the maximum length is 32*/
            },
            "glasses":{
                /*optional, wear glasses or not*/
                "value":""
                /*optional, string, wearing glasses or not: "no", "yes", "sunglasses"-wearing
                sunglasses; the maximum length is 32*/
            },
            "age": {
                /*optional, age information*/
                "value": ,
                /*optional, integer32, age*/
                "ageGroup": ""
                /*optional, string, age range; the maximum size is 32 bytes*/
            },
            "ride":{
                /*optional, whether the person is riding*/
                "value": ""
                /*optional, string, whether the person is riding: "no", "yes", "unknown"; the
                maximum length is 32*/
            },
        },
    },
}
```



```
        "gender":{
/*optional, gender*/
    "value":""
/*optional, string, gender: "male", "female", "unknown"; the maximum length is
32*/
    },
    "bag":{
/*optional, whether the person is carrying a bag*/
    "value": ""
/*optional, string, whether the person is carrying a bag: "no", "yes",
"unknown"; the maximum length is 32*/
    },
    "things":{
/*optional, whether the person is carrying things*/
    "value": ""
/*optional, string, whether the person is carrying things: "no", "yes",
"unknown"; the maximum length is 32*/
    },
    "hat": {
/*optional, whether the person is wearing a hat*/
    "value": ""
/*optional, string, whether the person is wearing a hat: "no", "yes",
"unknown"; the maximum length is 32*/
    },
    "jacketType":{
/*optional, top's type*/
    "value":"","
/*optional, string, top's type; the maximum length is 32*/
    },
    "trousersType":{
/*optional, trousers' type*/
    "value": ""
/*optional, string, trousers' type; the maximum length is 32*/
    },
    "mask": {
/*optional, whether the person is wearing mask*/
    "value": ""
/*optional, string, whether the person is wearing mask: "no", "yes", "unknown";
the maximum length is 32*/
    },
    "hairStyle":{
/*optional, hair style*/
    "value": ""
/*optional, string, hair style; the maximum length is 32*/
    },
    "score":{
/*optional, human body grading*/
    "value":
/*optional, int, human body grading, range: [0,100]*/
    },
    "motionStatus":{
/*optional, motion status*/
```

```

    "value": ""
  /*optional, string, motion: "enter", "leave", "unknown"; the maximum length is
  32*/
  },
  /******Human Body Picture
  Information******/
  "HumanImage":{
  /*optional, human body thumbnail, resource transmission module; when the
  returned alarm message consists additional resource (e.g., picture), this node
  should be returned*/
    "resourcesContentType": "",
  /*dependent, string, resource transmission type: "url", "binary"; when the
  returned alarm message consists additional resource (e.g., picture), this node
  should be returned*/
    "resourcesContent": ""
  /*dependent, string, resource ID; when the returned alarm message consists
  additional resource (e.g., picture), this node should be returned; when the
  value of resourcesContentType is "binary", it must be the same as Content-ID of
  picture, when the value of resourcesContentType is url, its value is the
  picture URL*/
  },
  "HumanBackgroundImage":{
  /*optional, human body background picture, resource transmission module; when
  the returned alarm message consists additional resource (e.g., picture), this
  node should be returned*/
    "resourcesContentType": "",
  /*dependent, string, resource transmission type: "url", "binary"; when the
  returned alarm message consists additional resource (e.g., picture), this node
  should be returned*/
    "resourcesContent": ""
  /*dependent, string, resource ID; when the returned alarm message consists
  additional resource (e.g., picture), this node should be returned; when the
  value of resourcesContentType is "binary", it must be the same as Content-ID of
  picture, when the value of resourcesContentType is url, its value is the
  picture URL*/
  }
  }
  }
  }
}
--<frontier>
Content-Disposition: form-data; name="faceImage"; filename="faceImage.jpg"
Content-Type: image/jpeg
Content-Length: image_length
Content-ID: faceImage

[faceImage]
--<frontier>
Content-Disposition: form-data; name="faceBackgroundImage";
filename="faceBackgroundImage.jpg"
Content-Type: image/jpeg

```

```
Content-Length: image_length
Content-ID: faceBackgroundImage

[faceBackgroundImage]
--<frontier>
Content-Disposition: form-data; name="FDLibImage"; filename="FDLibImage.jpg"
Content-Type: image/jpeg
Content-Length: image_length
Content-ID: FDLibImage

[FDLibImage]
--<frontier>
Content-Disposition: form-data; name="humanImage"; filename="humanImage.jpg"
Content-Type: image/jpeg
Content-Length: image_length
Content-ID: humanImage

[humanImage]
--<frontier>
Content-Disposition: form-data; name="humanBackgroundImage";
filename="humanBackgroundImage.jpg"
Content-Type: image/jpeg
Content-Length: image_length
Content-ID: humanBackgroundImage

[humanBackgroundImage]
--<frontier>
Content-Disposition: form-data; name="HBDLibImage"; filename="HBDLibImage.jpg"
Content-Type: image/jpeg
Content-Length: image_length
Content-ID: HBDLibImage

[HBDLibImage]
--<frontier>--
```

E.6 JSON_EventNotificationAlert_personArmingTrackMsg

JSON message about alarm details of auto person arming

```
Content-Type: Content-Type: multipart/form-data;boundary=<frontier>

--<frontier>
Content-Disposition: form-data; name="personArmingTrack"
Content-Type: application/json
Content-Length: text_length

{
  "ipAddress": "",
  /*required, device IPv4 address , string, the maximum size is 32*/
  "ipv6Address": "",
```

```
/*optional, device IPv6 address, string, the maximum size is 128*/
  "portNo": ,
/*optional, device port No., integer32*/
  "protocol": "",
/*optional, string, protocol type: "HTTP", "HTTPS", "EHome"-ISUPSDK protocol,
the maximum size is 32*/
  "macAddress": "",
/*optional, string, MAC address, the maximum length is 32 bytes*/
  "channelID": ,
/*optional, integer32, device channel No. that triggers alarm*/
  "dateTime": "",
/*required, string, alarm triggering time based on ISO8601 time format, the
maximum length is 32*/
  "activePostCount": ,
/*required, integer32, number of the same uploaded alarm*/
  "eventType": "",
/*required, string, event type: "personArmingTrack"-auto person arming, the
maximum length is 128 bytes*/
  "eventState": "",
/*required, string, event triggering status: "active"-triggered, "inactive"-not
triggered (refers to heartbeat data), the maximum length is 32 bytes*/
  "eventDescription": "",
/*required, string, event description: person arming track, the maximum length
is 128 bytes*/
  "channelName": "",
/*required, channel name (camera name), which should be the same as the
<channelName> in message XML_StreamingChannel (related URI: /ISAPI/Streaming/
channels/<ID>)*/*
  "deviceID": "",
/*optional, device ID (PUID), which should be returned during integration of
ISUPSDK and should be the same as the <deviceID> in XML_Ehome (related URI: /
ISAPI/System/Network/Ehome)*/*
  "persistentEventStatus": "",
/*dependent, event status: "started", "ongoing", "ended"; it is valid when the
value of eventType "personArmingTrack"*/
  "uid": "",
/*optional, string, event uploading ID, which can consists of time (accurate to
millisecond) and random digits, the maximum value size is 64 bytes*/
  "PersonArmingTrackInfo":{
/*required, person arming tracking information*/
/*****Person
Information*****/
  "PersonInfo": {
/*optional, target person information; when the value of persistentEventStatus
is "started", the PersonInfo will be uploaded; when the value of
persistentEventStatus is "ended", the PersonInfo will not be uploaded unless
the tracking capture has enabled; when the value of persistentEventStatus is
"ongoing", the PersonInfo will not be uploaded*/
    "armingType": "",
/*optional, string, arming type: "face"-upload face information, "human"-upload
human body information, "face_human"-upload face and human body information*/
    "Face": {
```

```
/*optional, face information*/
    "FaceCaptureResult": {
/*optional, face capture result*/
/*****Face Rectangle Frame
Information*****/
        "Rect": {
/*optional, face thumbnail rectangle frame*/
            "height": ,
/*optional, float, frame height, range: [0.000,1.000]*/
            "width": ,
/*optional, float, frame width, range: [0.000,1.000]*/
            "x": ,
/*required, float, X-coordinate of upper-left corner, range: [0.000,1.000],
corrects to three decimal place*/
            "y":
/*required, float, Y-coordinate of upper-left corner, range: [0.000,1.000],
corrects to three decimal place*/
        },
        "FacePictureRect": {
/*optional, face rectangle frame*/
            "height": ,
/*optional, float, frame height, range: [0.000,1.000]*/
            "width": ,
/*optional, float, frame width, range: [0.000,1.000]*/
            "x": ,
/*required, float, X-coordinate of upper-left corner, range: [0.000,1.000],
corrects to three decimal place*/
            "y":
/*required, float, Y-coordinate of upper-left corner, range: [0.000,1.000],
corrects to three decimal place*/
        },
/*****Face Attributes*****/
        "age": {
/*optional, age information*/
            "value": ,
/*optional, integer32, age*/
            "ageGroup": ""
/*optional, string, age range; the maximum size is 32 bytes*/
        },
        "gender":{
/*optional, gender*/
            "value":""
/*optional, string, gender: "male", "female", "unknown"; the maximum length is
32*/
        },
        "glass":{
/*optional, wear glasses or not*/
            "value":""
/*optional, string, wearing glasses or not: "no", "yes", "sunglasses"-wearing
sunglasses; the maximum length is 32*/
        },
        "smile":{
```

```
/*optional, smiling or not*/
    "value":""
/*optional, string, smiling or not: "no", "yes", "unknown"; the maximum length
is 32*/
    },
    "mask": {
/*optional, wearing mask or not*/
    "value": ""
/*optional, string, wearing mask or not: "no", "yes", "unknown"; the maximum
length is 32*/
    },
    "faceExpression":{
/*optional, face expression*/
    "value": ""
/*optional, string, face expression; the maximum size is 32*/
    },
    "hat": {
/*optional, whether the person is wearing a hat*/
    "value": ""
/*optional, string, whether the person is wearing a hat: "no", "yes",
"unknown"; the maximum length is 32*/
    },
/*****Face Picture
Information*****/
    "FaceImage":{
/*optional, face thumbnail, resource transmission module; when the returned
alarm message consists additional resource (e.g., picture), this node should be
returned*/
    "resourcesContentType": "",
/*dependent, string, resource transmission type: "url", "binary"; when the
returned alarm message consists additional resource (e.g., picture), this node
should be returned*/
    "resourcesContent": ""
/*dependent, string, resource ID; when the returned alarm message consists
additional resource (e.g., picture), this node should be returned; when the
value of resourcesContentType is "binary", it must be the same as Content-ID of
picture, when the value of resourcesContentType is url, its value is the
picture URL*/
    },
    "FaceBackgroundImage":{
/*optional, face background picture, resource transmission module; when the
returned alarm message consists additional resource (e.g., picture), this node
should be returned*/
    "resourcesContentType": "",
/*dependent, string, resource transmission type: "url", "binary"; when the
returned alarm message consists additional resource (e.g., picture), this node
should be returned*/
    "resourcesContent": ""
/*dependent, string, resource ID; when the returned alarm message consists
additional resource (e.g., picture), this node should be returned; when the
value of resourcesContentType is "binary", it must be the same as Content-ID of
picture, when the value of resourcesContentType is url, its value is the
```

```
picture URL*/
    }
},
/*****Face Comparison
Result*****/
    "FaceContrastResult":{
        "similarity": ,
/*optional, float, the maximum similarity of face comparison results, range:
[0,1], corrects to six decimal places*/
/*****Face Picture Library
Information*****/
        "FDID": "",
/*optional, string, face picture library ID, the maximum length is 64*/
        "FPID": "",
/*optional, string, face picture ID, the maximum length is 64*/
        "FDLibName": "",
/*optional, face picture library name*/
        "customFaceLibID": "",
/*optional, string, ID of custom face picture library, the maximum length is
64*/
        "customHumanID": "",
/*optional, string, custom person ID, the maximum length is 64*/
        "FDLibThreshold": ,
/*optional, face grading threshold for detection, range: [0,100]*/
        "FaceAppendData": {
/*optional, additional information of person in face picture*/
            "name": "",
/*required, string, name, the maximum length is 64*/
            "gender": "",
/*optional, string, gender: "male", "female", "unknown", the maximum length is
10*/
            "bornTime": "",
/*required, string, date of birth, ISO8601 time format, the maximum length is
32*/
            "province": "",
/*optional, string, province of birth, the maximum length is 32*/
            "city": "",
/*optional, string, city of birth, the maximum length is 32*/
            "certificateType": "",
/*optional, string, certificate type: "officerID"-officer ID, "ID"-identity
card, "other"-other; the maximum length is 10*/
            "certificateNumber": "",
/*optional, string, certificate No., the maximum length is 32*/
            "ageGroup": "",
/*optional, string, age group, the maximum length is 32*/
            "PersonInfoExtend":[{
/*optional, custom extension information of person*/
                "id": ,
/*optional, integer32, serial No. of person custom extension information,
starts from 1*/
                "enable": ,
/*optional, boolean, whether to enable person extension information*/
```

```
        "name": " ",
/*optional, string, tag name of person extension information*/
        "value": ""
/*optional, string, extension information content*/
    }
},
    "modelData": "",
/*optional, string, the modeling data of face picture base*/
/*****Face Picture
Information*****/
    "FDLibImage": {
/*optional, face picture base, resource transmission module; when the returned
alarm message consists additional resource (e.g., picture), this node should be
returned*/
        "resourcesContent": ""
/*dependent, string, resource transmission type: "url", "binary"; when the
returned alarm message consists additional resource (e.g., picture), this node
should be returned*/
        "resourcesContent": ""
/*dependent, string, resource ID; when the returned alarm message consists
additional resource (e.g., picture), this node should be returned; when the
value of resourcesContentType is "binary", it must be the same as Content-ID of
picture, when the value of resourcesContentType is url, its value is the
picture URL*/
    }
},
/*****Human Body
Information*****/
    "Human": {
        "HumanCaptureResult": {
/*optional, human body capture result*/
/*****Human Body Rectangle Frame
Information*****/
            "Rect": {
/*optional, rectangle frame of human body thumbnail*/
                "height": ,
/*optional, float, frame height, range: [0.000,1.000]*/
                "width": ,
/*optional, float, frame width, range: [0.000,1.000]*/
                "x": ,
/*required, float, X-coordinate of upper-left corner, range: [0.000,1.000],
corrects to three decimal place*/
                "y": ,
/*required, float, Y-coordinate of upper-left corner, range: [0.000,1.000],
corrects to three decimal place*/
            },
/*****Human Body Attributes
Information*****/
            "jacketColor": {
/*optional, top's color*/
                "value": ""
            }
        }
    }
}
```



```
/*optional, string, top's color, the maximum length is 32*/
    },
    "trousersColor":{
/*optional, trousers' color*/
    "value": ""
/*optional, string, trousers' color, the maximum length is 32*/
    },
    "glass":{
/*optional, wear glasses or not*/
    "value":""
/*optional, string, wearing glasses or not: "no", "yes", "sunglasses"-wearing
sunglasses; the maximum length is 32*/
    },
    "age": {
/*optional, age information*/
    "value": ,
/*optional, integer32, age*/
    "ageGroup": ""
/*optional, string, age range; the maximum size is 32 bytes*/
    },
    "ride":{
/*optional, whether the person is riding*/
    "value": ""
/*optional, string, whether the person is riding: "no", "yes", "unknown"; the
maximum length is 32*/
    },
    "gender":{
/*optional, gender*/
    "value":""
/*optional, string, gender: "male", "female", "unknown"; the maximum length is
32*/
    },
    "bag":{
/*optional, whether the person is carrying a bag*/
    "value": ""
/*optional, string, whether the person is carrying a bag: "no", "yes",
"unknown"; the maximum length is 32*/
    },
    "things":{
/*optional, whether the person is carrying things*/
    "value": ""
/*optional, string, whether the person is carrying things: "no", "yes",
"unknown"; the maximum length is 32*/
    },
    "hat": {
/*optional, whether the person is wearing a hat*/
    "value": ""
/*optional, string, whether the person is wearing a hat: "no", "yes",
"unknown"; the maximum length is 32*/
    },
    "jacketType":{
/*optional, top's type*/
```

```
        "value": "",
/*optional, string, top's type; the maximum length is 32*/
    },
    "trousersType": {
/*optional, trousers' type*/
        "value": ""
/*optional, string, trousers' type; the maximum length is 32*/
    },
    "mask": {
/*optional, whether the person is wearing mask*/
        "value": ""
/*optional, string, whether the person is wearing mask: "no", "yes", "unknown";
the maximum length is 32*/
    },
    "hairStyle": {
/*optional, hair style*/
        "value": ""
/*optional, string, hair style; the maximum length is 32*/
    },
    "score": {
/*optional, human body grading*/
        "value": ""
/*optional, int, human body grading, range: [0,100]*/
    },
    "motionStatus": {
/*optional, motion status*/
        "value": ""
/*optional, string, motion: "enter", "leave", "unknown"; the maximum length is
32*/
    },
    /*****Human Body Picture
    Information*****/
    "HumanImage": {
/*optional, human body thumbnail, resource transmission module; when the
returned alarm message consists additional resource (e.g., picture), this node
should be returned*/
        "resourcesContentType": "",
/*dependent, string, resource transmission type: "url", "binary"; when the
returned alarm message consists additional resource (e.g., picture), this node
should be returned*/
        "resourcesContent": ""
/*dependent, string, resource ID; when the returned alarm message consists
additional resource (e.g., picture), this node should be returned; when the
value of resourcesContentType is "binary", it must be the same as Content-ID of
picture, when the value of resourcesContentType is url, its value is the
picture URL*/
    },
    "HumanBackgroundImage": {
/*optional, human body background picture, resource transmission module; when
the returned alarm message consists additional resource (e.g., picture), this
node should be returned*/
        "resourcesContentType": "",
```

```
/*dependent, string, resource transmission type: "url", "binary"; when the
returned alarm message consists additional resource (e.g., picture), this node
should be returned*/
    "resourcesContent": ""
/*dependent, string, resource ID; when the returned alarm message consists
additional resource (e.g., picture), this node should be returned; when the
value of resourcesContentType is "binary", it must be the same as Content-ID of
picture, when the value of resourcesContentType is url, its value is the
picture URL*/
    }
},
/*****Human Body Picture Comparison
Result*****/
    "HumanContrastResult":{
/*optional,result of human body picture comparison*/
        "similarity": ,
/*optional, float, the maximum similarity of human body picture comparison,
range: [0,1], corrects to six decimal places*/
        "HBDID": "",
/*optional, string, ID of human body picture library, the maximum length is
64*/
        "HBPID": "",
/*optional, string, human body picture ID; the maximum length is 64*/
        "HBDLibName": "",
/*optional, string, name of human body picture library, the maximum length is
64*/
        "customHBDID": "",
/*optional, string, ID of custom human body picture library, the maximum length
is 128*/
        "customHBPID": "",
/*optional, string, ID of custom human body picture, the maximum length is 64*/
        "HBDLibThreshold": ,
/*required, integer32, similarity threshold for human body picture comparison,
the higher the threshold, the more accurate the comparison is; range: [0,100]*/
        "HumanAppendData":{
/*optional, additional information of person in human body picture*/
            "name": "",
/*required, string, person name, the maximum length is 64*/
            "gender": "",
/*optional, string, gender: "male", "female", "unknown"*/
            "bornTime": "",
/*optional, string, date of birth, ISO8601 time format, the maximum length is
32*/
            "province": "",
/*optional, string, province of birth, the maximum length is 32*/
            "city": "",
/*optional, string, city of birth, the maximum length is 32*/
            "certificateType": "",
/*optional, string, certificate type: "officerID"-officer ID, "ID"-identity
card, "other"-other; the maximum length is 10*/
            "certificateNumber": "",
/*optional, string, certificate No., the maximum length is 32*/
```

```
        "PersonInfoExtend": [{
/*optional, custom extension information of person*/
            "id": ,
/*optional, integer32, serial No. of person custom extension information,
starts from 1*/
            "enable": ,
/*optional, boolean, whether to enable person custom extension information*/
            "name": "",
/*optional, string, tag name of person tag extension information*/
            "value": ""
/*optional, string, extension information content*/
        }],
        "modelData": "",
/*optional, string, the modeling data of human body picture base*/
/******Human Body Picture
Information*****/
        "HBDLibImage": {
/*optional, human body base picture, resource transmission module; when the
returned alarm message consists additional resource (e.g., picture), this node
should be returned*/
            "resourcesContentType": "",
/*dependent, string, resource transmission type: "url", "binary"; when the
returned alarm message consists additional resource (e.g., picture), this node
should be returned*/
            "resourcesContent": ""
/*dependent, string, resource ID; when the returned alarm message consists
additional resource (e.g., picture), this node should be returned; when the
value of resourcesContentType is "binary", it must be the same as Content-ID of
picture, when the value of resourcesContentType is url, its value is the
picture URL*/
        }
    },
/******Person Tracking
Information*****/
    "TrackInfo": {
/*optional, tracking information of person arming; it should be uploaded when
the value of persistentEventStatus is "ongoing" or "ended"*/
        "targetUUID": "",
/*optional, string, UUID of person arming target, it is unique*/
        "TargetGPSInfo": {
/*required, target position information*/
            "longitudeType": "",
/*required, string, longitude type: "E"-east, "W"-west*/
            "latitudeType": "",
/*required, string, latitude type: "S"-south, "N"-north*/
            "Longitude": {
/*required, longitude value*/
                "degree": ,
/*required, int, degree: [0,180]*/
```

```

        "minute": ,
/*required, int, minute: [0,59]*/
        "sec":
/*required, float, second: [0.000000,60.000000], corrects to 6 decimal places*/
    },
    "Latitude":{
/*required, latitude value*/
        "degree": ,
/*required, int, degree: [0,180]*/
        "minute": ,
/*required, int, minute: [0,59]*/
        "sec":
/*required, float, second: [0.000000,60.000000], corrects to 6 decimal places*/
    }
}
}
}
}
--<frontier>
Content-Disposition: form-data; name="faceImage"; filename="faceImage.jpg"
Content-Type: image/jpeg
Content-Length: image_length
Content-ID: faceImage

[faceImage]
--<frontier>
Content-Disposition: form-data; name="faceBackgroundImage";
filename="faceBackgroundImage.jpg"
Content-Type: image/jpeg
Content-Length: image_length
Content-ID: faceBackgroundImage

[faceBackgroundImage]
--<frontier>
Content-Disposition: form-data; name="FDLibImage"; filename="FDLibImage.jpg"
Content-Type: image/jpeg
Content-Length: image_length
Content-ID: FDLibImage

[FDLibImage]
--<frontier>
Content-Disposition: form-data; name="humanImage"; filename="humanImage.jpg"
Content-Type: image/jpeg
Content-Length: image_length
Content-ID: humanImage

[humanImage]
--<frontier>
Content-Disposition: form-data; name="humanBackgroundImage";
filename="humanBackgroundImage.jpg"
Content-Type:image/jpeg
Content-Length: image length

```

```
Content-ID: humanBackgroundImage
```

```
[humanBackgroundImage]
--<frontier>
Content-Disposition: form-data; name="HBDLibImage"; filename="HBDLibImage.jpg"
Content-Type: image/jpeg
Content-Length: image_length
Content-ID: HBDLibImage

[HBDLibImage]
--<frontier>--
```

E.7 JSON_HBDInfo

JSON message about information of human body picture(s) to be deleted

```
{
  "HBDInfo":{
    "HBDID": "",
    /*optional, string, ID of human body picture library*/
    "customHBDID": "",
    /*optional, string, ID of custom human body picture library, the maximum length
    is 128*/
    "HBPList":[{
    /*optional, list of human body pictures. When this node exists, it indicates
    that device supports deleting specified multiple pictures in a batch*/
      "HBPID": "",
      /*optional, string, human body picture ID; the maximum length is 64*/
      "customHBPID": ""
    /*optional, string, custom human body picture ID; the maximum length is 64*/
    }
    ]
  }
}
```

E.8 JSON_HBDLibBaseCfg

JSON message about basic information of a human body picture library

```
{
  "HBDLibBaseCfg":{
    /*required, basic information of a human body picture library*/
    "id": ,
    /*required, sub item No. in the list, starts from 1*/
    "HBDID": "",
    /*required, string, ID of human body picture library, the maximum length is 64;
    for cameras, only the default HBDID will be returned*/
    "HBDLibName": "",
    /*required, string, name of human body picture library, the maximum length is
    64*/
  }
}
```

```
    "HBDLibThreshold": ,
    /*required, integer32, similarity threshold for human body picture comparison,
    range: [0,100]*/
    "customHBDID": "",
    /*optional, string, ID of custom human body picture library, the maximum length
    is 128*/
    "customInfo": ""
    /*optional, string, additional information of human body picture library, the
    maximum length is 32*/
  }
}
```

E.9 JSON_HBDLibBaseCfgList

JSON message about basic information of all human body picture libraries

```
{
  "HBDLibBaseCfgList":[
    /*required, basic information of all human body picture libraries*/
    {
      "id":,
      /*required, sub item No. in the list, starts from 1*/
      "HBDID": "",
      /*required, string, ID of human body picture library, the maximum length is 64;
      for cameras, only the default HBDID will be returned*/
      "HBDLibName": "",
      /*required, string, name of human body picture library, the maximum length is
      64*/
      "HBDLibThreshold":,
      /*required, integer32, similarity threshold for human body picture comparison,
      range: [0,100]*/
      "customHBDID": "",
      /*optional, string, ID of custom human body picture library, the maximum length
      is 128*/
      "customInfo": ""
      /*optional, string, additional information of human body picture library, the
      maximum length is 32*/
    }
  ]
}
```

E.10 JSON_HBDLibCap

JSON message about configuration capability of human body picture library

```
{
  "HBDLibCap":{
    "CreateHBDLib":{
      /*required, capability of creating human body picture library*/
    }
  }
}
```

```
"HBDMaxSize": ,
/*required, integer32, the maximum number of supported human body picture
libraries*/
"HBDID":{
/*required, string, ID of human body picture library, the maximum length is 64;
for cameras, only the default HBDID will be returned*/
    "@min":"1",
    "@max":"64"
},
"HBDLibName":{
/*required, string, name of human body picture library, the maximum length is
64*/
    "@min":"1",
    "@max":"64"
},
"HBDLibThreshold":{
/*required, integer32, similarity threshold for human body picture comparison,
the higher the threshold, the more accurate the comparison is; range: [0,100]*/
    "@min":0,
    "@max":100
},
"HBPMMaxSize":{
/*required, integer32, the maximum size of pictures in human body picture
library*/
    "@max":15000
},
"customHBDID":{
/*optional, string, ID of custom human body picture library, the maximum length
is 128*/
    "@min":"0",
    "@max":"128"
},
"customInfo":{
/*optional, string, additional information of human body picture library, the
maximum length is 32*/
    "@min":"0",
    "@max":"32"
}
},
"HumanAppendData":{
/*optional, additional information of person in human body picture*/
    "name":{
/*required, string, name, the maximum length is 64*/
        "@min":"0",
        "@max":"64"
    },
    "gender":{
/*optional, string, gender: "male", "female", "unknown", the maximum length is
10*/
        "@opt":["male", "female", "unknown"]
    },
    "bornTime":"","
```



```
/*required, string, date of birth, ISO8601 time format, the maximum length is 32*/
"province":{
/*optional, string, province of birth, the maximum length is 32*/
"@min":"0",
"@max":"32"
},
"city":{
/*optional, string, city of birth, the maximum length is 32*/
"@min":"0",
"@max":"32"
},
"certificateType ":{
/*optional, string, certificate type: "officerID"-officer ID, "ID"-identity card, "other"-other; the maximum length is 10*/
"@opt":["officerID", "ID", "other"]
},
"certificateNumber":{
/*optional, string, certificate No., the maximum length is 32*/
"@min":"0",
"@max":"32"
},
"PersonInfoExtend":{
/*optional, custom extension information of person*/
"MaxSize": ,
/*required, integer32, the maximum size of supported custom extension information*/
"id":{
/*optional, integer32, serial No. of person custom extension information, starts from 1*/
"@min":1,
"@max":4
},
"enable":{
/*optional, boolean, whether to enable person custom extension information*/
"@opt":[false, true]
},
"name":{
/*optional, string, name of person extension information tag*/
"@min":"",
"@max":""
},
"value":{
/*optional, string, extension information details*/
"@min":"",
"@max":""
}
}
},
"searchID":{
/*optional, string, search ID, which is used to check whether the current search requester is the same; if they are the same, the search record will be
```

```
stored in the device to speed up the next search*/
    "@min": "1",
    "@max": "64"
  },
  "searchResultPosition": ,
  /*required, integer32, the end position of search result in result list; in a
  single search, if you cannot get all the records in the result list, you can
  mark the end position and get the following records after the marked position
  in the next search*/
  "maxResults": ,
  /*required, integer32, maximum number of search results obtained this time, up
  to 500 records can be obtained per search; if the value of maxResults reaches
  the device supported limit, the device will return the maximum number of search
  results supported by the device*/
  "isSuportHBDSearch": ,
  /*optional, boolean, whether the device supports searching for human body
  picture of human body picture library*/
  "ModelingStatus": {
  /*optional, modeling related information*/
    "status": {
  /*optional, string, modeling status: "modeling"-modeled, "unmodeled",
  "modelingFailed"-modeling failed*/
      "@opt": ["modeling", "unmodeled", "modelingFailed"]
    },
    "reason": {
  /*optional, string, modeling failure reason, it is valid when the value of
status is "modelingFailed": "noHuman"-no human body, "humanSizeSmall"-picture
size is too small, "shadeHuman"-human body is tempered, "unknow"-unknown error,
"imageURLDownloadFailed"-picture download failed, "noJpgPicture"-the picture is
not in JPEG format, "errorPictureResolution"-invalid resolution, "humanSizeBig"-
picture size is too large, "diskError"-disk error, "diskFull"-disk is full,
"analysisFailed"-analysis failed, "multipleHumans"-there are multiple human
bodies*/
      "@opt": ["noHuman", "humanSizeSmall", "shadeHuman",
"unknow", "imageURLDownloadFailed", "noJpgPicture", "errorPictureResolution", "human
SizeBig", "diskError", "diskFull", "analysisFailed", "multipleHumans"]
    }
  },
  "isSuportHBDMannualModeling": ,
  /*optional, boolean, whether the device supports manual modeling of human body
  picture library*/
  "isSupportCapacity":
  /*optional, boolean, whether the device supports getting storage capacity of
  human body picture library (related URI: /ISAPI/Intelligent/HBDLib/<HBDID>/
  capacity?format=json)*/
  }
}
```

E.11 JSON_HBDLibInfoList

JSON message about result of creating human body picture libraries

```
{
  "HBDLibInfoList":[
    {
      "id": ,
      /*required, sub item No. in the list, starts from 1*/
      "HBDID": "",
      /*required, string, ID of human body picture library, the maximum length is 64;
      for cameras, only the default HBDID will be returned*/
      "HBDLibName": "",
      /*required, string, name of human body picture library, the maximum length is
      64*/
      "statusCode": ,
      /*optional, read-only, it is returned when creating human body picture library
      failed*/
      "subStatusCode": ""
      /*optional, read-only, it is returned when creating human body picture library
      failed*/
    }
  ]
}
```

E.12 JSON_HBDModelingStatusSearchDescription

JSON message about condition of searching for modeling status

```
{
  "HBDModelingStatusSearchDescription":{
    "searchID": ,
    /*required, string, search ID, which is used to check whether the current
    search requester is the same as the previous one. If they are the same, the
    search record will be stored in the device to speed up the next search*/
    "searchResultPosition": ,
    /*required, int, the end position of search result in result list; it is 0 for
    first search*/
    "maxResults": ,
    /*required, int, the maximum number of records in this search*/
    "modelStatus": "",
    /*optional, string, modeling status: "modeling"-modeled, "unmodeled",
    "modelingFailed"-modeling failed*/
    "HBDIDList":[{
      /*optional, search by human body picture library ID; HBDIDList and taskID are
      mutually exclusive*/
      "HBDID": "",
      /*optional, string, ID of human body picture library, the maximum length is 64*/
      "customHBDID": ""
    }
  ]
}
```

```
/*optional, string, ID of custom human body picture library, the maximum length
is 128*/
    },
    "taskID": ""
/*optional, string, search by task ID, the maximum length is 64; taskID and
HBDIDList are mutually exclusive*/
    }
}
```

E.13 JSON_HBDModelingStatusSearchResult

JSON message about modeling status search result

```
{
  "HBDModelingStatusSearchResult":{
    "searchID": ,
/*required, string, search ID, which is used to check whether the current
search requester is the same as the previous one. If they are the same, the
search record will be stored in the device to speed up the next search*/
    "responseStatus": ,
/*required, string, search status: "true"-succeeded, "false"-failed*/
    "responseStatusStrg": "",
/*required, string, search status description: "OK"-search ended, "MORE"-
searching, "NO_MATCHES"-no data found*/
    "numOfMatches": ,
/*required, string, number of matched results that returned for once*/
    "totalMatches": ,
/*required, string, total number of matched results*/
    "ModelingStatusList":[{
/*optional, detailed information of matched results*/
      "HBDID": "",
/*optional, string, ID of human body picture library, the maximum length is
64*/
      "HBPID": "",
/*optional, string, human body picture ID; the maximum length is 64*/
      "customHBDID": "",
/*optional, string, ID of custom human body picture library, the maximum length
is 128*/
      "customHBPID": "",
/*optional, string, ID of custom human body picture, the maximum length is 64*/
      "modelStatus": "",
/*required, string, modeling status: "modeling"-modeled, "unmodeled",
"modelingFailed"-modeling failed*/
      "reason": "",
/*dependent, string, reason of modeling failure, it is valid when the value of
modelStatus is "modelingFailed": "noHuman"-no human body, "humanSizeSmall"-
picture size is too small, "shadeHuman"-human body is tempered, "unknow"-
unknown error, "imageURLDownloadFailed"-picture download failed, "noJpgPicture"-
the picture is not in JPEG format, "errorPictureResolution"-invalid resolution,
"humanSizeBig"-picture size is too large, "diskError"-disk error, "diskFull"-
```

```
disk is full, "analysisFailed"-analysis failed, "multipleHumans"-there are
multiple human bodies*/
    "name": ""
/*required, string, person name,the maximum length is 64*/
  ]]
}
}
```

E.14 JSON_HBDSearchDescription

JSON message about condition of searching for human body picture

```
{
  "HBDSearchDescription":{
    "searchID": ,
/*required, string, search ID, which is used to check whether the current
search requester is the same as the previous one. If they are the same, the
search record will be stored in the device to speed up the next search*/
    "searchResultPosition": ,
/*required, int, the end position of search result in result list; it is 0 for
first search*/
    "maxResults": ,
/*required, int, the maximum number of records in this search*/
    "HBDID": "",
/*required, string, ID of human body picture library, the maximum length is 64*/
    "customHBDID": "",
/*optional, string, ID of custom human body picture library, the maximum length
is 128*/
    "startTime": "",
/*optional, string, ISO 8601 format, search start time; the maximum length is
32*/
    "endTime": "",
/*optional, string, ISO 8601 format, search end time; the maximum length is
32*/
    "name": "",
/*optional, string, person name, the maximum length is 64*/
    "gender": "",
/*optional, string, gender: "male", "female", "unknown"*/
    "province": "",
/*optional, string, province of birth, the maximum length is 32*/
    "city": "",
/*optional, string, city of birth, the maximum length is 32*/
    "certificateType ": "",
/*optional, string, certificate type: "officerID"-officer ID, "ID"-identity
card, "other"-other; the maximum length is 10*/
    "certificateNumber": ""
/*optional, string, certificate No., the maximum length is 32*/
    "modelStatus": ""
/*optional, string, modeling status: "modeling"-modeled, "unmodeled",
"modelingFailed"-modeling failed*/
```

```
}  
}
```

E.15 JSON_HBDSearchResult

JSON message about result of searching for human body picture

```
{  
  "HBDSearchResult":{  
    "searchID": ,  
    /*required, string, search ID, which is used to check whether the current  
    search requester is the same as the previous one. If they are the same, the  
    search record will be stored in the device to speed up the next search*/  
    "responseStatus": ,  
    /*required, string, search status: "true"-succeeded, "false"-failed*/  
    "responseStatusStrg": "",  
    /*required, string, search status description: "OK"-search ended, "MORE"-  
    searching, "NO_MATCHES"-no data found*/  
    "numOfMatches": ,  
    /*required, string, number of matched results that returned for once*/  
    "totalMatches": ,  
    /*required, string, total number of matched results*/  
    "MatchList":[{  
    /*optional, detailed information of matched results*/  
      "HBDID": "",  
      /*required, string, ID of human body picture library, the maximum length is  
      64*/  
      "HBPID": "",  
      /*required, string, human body picture ID; the maximum length is 64*/  
      "HumanAppendData":{  
      /*optional, additional information of person in human body picture*/  
        "name": "",  
        /*required, string, person name, the maximum length is 64*/  
        "gender": "",  
        /*optional, string, gender: "male", "female", "unknown"*/  
        "bornTime": "",  
        /*optional, string, date of birth, ISO8601 time format, the maximum length is  
        32*/  
        "province": "",  
        /*optional, string, province of birth, the maximum length is 32*/  
        "city": "",  
        /*optional, string, city of birth, the maximum length is 32*/  
        "certificateType ": "",  
        /*optional, string, certificate type: "officerID"-officer ID, "ID"-identity  
        card, "other"-other; the maximum length is 10*/  
        "certificateNumber": "",  
        /*optional, string, certificate No., the maximum length is 32*/  
        "PersonInfoExtend":[{  
        /*optional, custom extension information of person*/  
          "id": ,
```

```
/*optional, integer32, serial No. of person custom extension information,
starts from 1*/
    "enable": ,
/*optional, boolean, whether to enable person custom extension information*/
    "name": "",
/*optional, string, tag name of person tag extension information*/
    "value": ""
/*optional, string, extension information content*/
    }]
},
"ModelingStatus":{
/*optional, modeling status information*/
    "status": "",
/*optional, string, modeling status: "modeling"-modeled, "unmodeled",
"modelingFailed"-modeling failed*/
    "reason": ""
/*optional, string, reason of modeling failure, it is valid when the value of
status is "modelingFailed": "noHuman"-no human body, "humanSizeSmall"-picture
size is too small, "shadeHuman"-human body is tempered, "unknow"-unknown error,
"imageURLDownloadFailed"-picture download failed, "noJpgPicture"-the picture is
not in JPEG format, "errorPictureResolution"-invalid resolution, "humanSizeBig"-
picture size is too large, "diskError"-disk error, "diskFull"-disk is full,
"analysisFailed"-analysis failed, "multipleHumans"-there are multiple human
bodies*/
    },
    "picURL":""
/*optional, string, picture URL*/
    }]
}
}
```

E.16 JSON_HBPCapacity

JSON message about capacity of human body picture library

```
{
  "HBPCapacity": {
/*required, capacity of human body picture library*/
    "surplusPicNums":
/*required, the remaining number of pictures can be stored in the human body
picture library*/
    }
}
```

E.17 JSON_HBPUploadData

JSON message about information of imported human body picture

```
Content-Type: multipart/form-data;
boundary=-----7e13971310878
-----7e13971310878
Content-Disposition: form-data; name="HBPUploadData";
Content-Type: text/json
{
  "HBPUploadData":{
    "HBDID": "",
    /*required, string, ID of human body picture library, the maximum length is
    64*/
    "customHBDID": "",
    /*optional, string, ID of custom human body picture library, the maximum length
    is 128*/
    "HumanAppendData":{
    /*optional, additional information of person in human body picture*/
      "name": "",
      /*required, string, name, the maximum length is 64*/
      "gender": "",
      /*optional, string, gender: "male", "female", "unknown", the maximum length is
      10*/
      "bornTime": "",
      /*required, string, date of birth, ISO8601 time format, the maximum length is
      32*/
      "province": "",
      /*optional, string, province of birth, the maximum length is 32*/
      "city": "",
      /*optional, string, city of birth, the maximum length is 32*/
      "certificateType": "",
      /*optional, string, certificate type: "officerID"-officer ID, "ID"-identity
      card, "other"-other; the maximum length is 10*/
      "certificateNumber": "",
      /*optional, string, certificate No., the maximum length is 32*/
      "PersonInfoExtend":[
    /*optional, custom extension information of person*/
      {
        "id": ,
        /*optional, integer32, serial No. of person custom extension information,
        starts from 1*/
        "value": ""
        /*optional, string, extension information details*/
      }
    ]
      },
      "customHBPID": "",
      /*optional, string, ID of custom human body picture, the maximum length is 64*/
      "picURL":""
    /*optional, string, picture URL*/
  }
}
-----7e13971310878
Content-Disposition: form-data; name="importImage";
Content-Type: image/jpeg
```


Opaque Data, picture data in binary format
-----7e13971310878--

E.18 JSON_HBPUploadResult

JSON message about result of importing human body picture to human body picture library

```
{
  "HBPUploadResult": {
    /*optional, result of importing human body picture to human body picture
    library*/
    "HBPID": "",
    /*required, string, ID of imported human body picture, the maximum length is
    64*/
  }
}
```

E.19 JSON_HumanAppendData

JSON message about additional information of person in human body picture.

```
{
  "HumanAppendData":{
    "name": "",
    /*required, string, person name, the maximum length is 64*/
    "gender": "",
    /*optional, string, gender: "male", "female", "unknown", the maximum length is
    10*/
    "bornTime": "",
    /*required, string, date of birth, ISO8601 time format, the maximum length is
    32*/
    "province": "",
    /*optional, string, province of birth, the maximum length is 32*/
    "city": "",
    /*optional, string, city of birth, the maximum length is 32*/
    "certificateType ": "",
    /*optional, string, certificate type: "officerID"-officer ID, "ID"-identity
    card, "other"-other; the maximum length is 10*/
    "certificateNumber": ""
    /*optional, string, certificate No., the maximum length is 32*/
    "PersonInfoExtend": [{
    /*optional, custom extension information of person*/
      "id": ,
      /*optional, integer32, serial No. of person custom extension information,
      starts from 1*/
      "value": ""
    /*optional, string, extension information content*/
    }]
  }
```

```
}  
}
```

E.20 JSON_ManualModeling

JSON message about modeling information of multiple human body pictures

```
{  
  "ManualModeling": {  
    /*required, build models for multiple human body pictures in a batch of  
    specified human body picture library*/  
    "modelStatus": "",  
    /*required, string, modeling task type: "all"-build models for all pictures,  
    when device upgraded (algorithm upgraded), all pictures of device should be  
    remodeled, "unmodeled"-build models for unmodeled pictures in human body  
    picture library*/  
  }  
}
```

E.21 JSON_ManualModeling_TaskInfo

JSON message about modeling task information

```
{  
  "TaskInfo": {  
    /*required, task information*/  
    "taskID": ""  
    /*required, string, task ID, the maximum length is 64; task ID should be  
    unique*/  
  }  
}
```

E.22 JSON_ManualPersonArmingTrack

JSON message about manual person arming parameters

```
{  
  "ManualPersonArmingTrack": {  
    /*required, manual person arming parameters*/  
    "TargetRect": {  
      "StartPoint": {  
        /*required, coordinates of rectangular frame start point*/  
        "positionX": ,  
        /*required, X-coordiante of rectangular frame start point*/  
        "positionY":  
        /*required, Y-coordiante of rectangular frame start point*/  
      },  
      "EndPoint": {  
        /*required, coordinates of rectangular frame end point*/  
        "positionX": ,  
        /*required, X-coordiante of rectangular frame end point*/  
        "positionY":  
        /*required, Y-coordiante of rectangular frame end point*/  
      }  
    }  
  }  
}
```

```
    "EndPoint":{
/*required, coordinates of rectangular frame end point*/
    "positionX": ,
/*required, X-coordiante of rectangular frame end point*/
    "positionY":
/*required, Y-coordiante of rectangular frame end point*/
    }
  }
}
```

E.23 JSON_ManualPersonArmingTrackCap

JSON message about capability of manual person arming

```
{
  "ManualPersonArmingTrackCap":{
/*required, manual person arming capability*/
    "TargetRect":{
      "StartPoint":{
/*required, coordinates of rectangular frame start point*/
        "positionX":{
/*required, X-coordiante of rectangular frame start point*/
          "@min":0,
          "@max":1
        },
        "positionY":{
/*required, Y-coordiante of rectangular frame start point*/
          "@min":0,
          "@max":1
        }
      },
      "EndPoint":{
/*required, coordinates of rectangular frame end point*/
        "positionX":{
/*required, X-coordiante of rectangular frame end point*/
          "@min":0,
          "@max":1
        },
        "positionY":{
/*required, Y-coordiante of rectangular frame end point*/
          "@min":0,
          "@max":1
        }
      }
    }
  }
}
```

E.24 JSON_ModelingRangeList

JSON message about modeling information of multiple human body pictures

```
{
  "ModelingRangeList": [{
    "HBDID": "",
    /*optional, string, ID of human body picture library, the maximum length is 64*/
    "customHBDID": "",
    /*optional, string, ID of custom human body picture library, the maximum length
    is 128*/
    "HBPID": [""],
    /*optional, string, human body picture ID, the maximum length is 64*/
    "customHBPID": [""],
    /*optional, string, custom human body picture ID, the maximum length is 64*/
  }]
}
```

E.25 JSON_ModelingRangeTaskState

JSON message about modeling task progress

```
{
  "ModelingRangeTaskState":{
    "taskTotalNumber": ,
    /*required, integer32, total number of modelings in the task*/
    "completeNumber": ,
    /*required, integer32, number of completed modelings*/
    "failedNumber":
    /*optional, integer32, number of failed modelings, if all modelings are
    succeeded, this node will not be returned*/
  }
}
```

E.26 JSON_ModelingStatus

JSON message about modeling result of specified human body picture

```
{
  "ModelingStatus":{
    /*required, modeling result*/
    "HBDID": "",
    /*optional, string, ID of human body picture library, the maximum length is
    64*/
    "HBPID": "",
    /*optional, string, human body picture ID, the maximum length is 64*/
    "customHBDID": "",
  }
}
```

```
/*optional, string, ID of custom human body picture library, the maximum length
is 128*/
    "customHBPID": "",
/*optional, string, custom human body picture ID, the maximum length is 64*/
    "modelStatus": "",
/*required, string, modeling status: "modeling"-modeled, "unmodeled",
"modelingFailed"-modeling failed*/
    "reason": "",
/*dependent, string, reason of modeling failure, it is valid when the value of
modelStatus is "modelingFailed": "noHuman"-no human body, "humanSizeSmall"-
picture size is too small, "shadeHuman"-human body is tempered, "unknow"-
unknown error, "imageURLDownloadFailed"-picture download failed, "noJpgPicture"-
the picture is not in JPEG format, "errorPictureResolution"-invalid resolution,
"humanSizeBig"-picture size is too large, "diskError"-disk error, "diskFull"-
disk is full, "analysisFailed"-analysis failed, "multipleHumans"-there are
multiple human bodies*/
    "name": ""
/*required, string, person name,the maximum length is 64*/
}
}
```

E.27 JSON_PersonArmingTrack

JSON message about person arming parameters

```
{
    "PersonArmingTrack":{
        "enabled": ,
/*required, boolean, whether to enable person arming*/
        "faceThreshold": ,
/*optional, int, face grading threshold for detection, range: [0,100], default
value: 70; when a face with its grading is larger than or equal to configured
threshold, comparison with the pictures in the face picture library will be
triggered*/
        "humanThreshold": ,
/*optional, int, human body grading threshold for detection, range: [0,100],
default value: 70; when a human body with its grading is larger than or equal
to configured threshold, comparison with the pictures in the human body picture
library will be triggered*/
        "faceDetectionTime": ,
/*optional, int, maximum detection time of face, range: [0,300], unit: second,
default value: 10s; when face grading does not reach the configured
faceThreshold during the faceDetectionTime, the device stops grading the target
face*/
        "humanDetectionTime":
/*optional, int, maximum detection time of human body, range: [0,300], unit:
second, default value: 10s; when human body grading does not reach the
configured humanThreshold during the humanDetectionTime, the device stops
grading the target human body*/
```

```
}  
}
```

E.28 JSON_PersonArmingTrackCap

JSON message about person arming configuraiton capability

```
{  
  "PersonArmingTrackCap":{  
    "enabled":{  
/*required, boolean, whether to enable person arming*/  
      "@opt": [true, false]  
    },  
    "faceThreshold":{  
/*optional, int, face grading threshold for detection, range: [0,100], default  
value: 70; when a face with its grading is larger than or equal to configured  
threshold, comparison with the pictures in the face picture library will be  
triggered*/  
      "@min":0,  
      "@max":100,  
      "@def":70  
    },  
    "humanThreshold":{  
/*optional, int, human body grading threshold for detection, range: [0,100],  
default value: 70; when a human body with its grading is larger than or equal  
to configured threshold, comparison with the pictures in the human body picture  
library will be triggered*/  
      "@min":0,  
      "@max":100,  
      "@def":70  
    },  
    "faceDetectionTime":{  
/*optional, int, maximum detection time of face, range: [0,300], unit: second,  
default value: 10s; when face grading does not reach the configured  
faceThreshold during the faceDetectionTime, the device stops grading the target  
face*/  
      "@min":0,  
      "@max":300,  
      "@def":10  
    },  
    "humanDetectionTime":{  
/*optional, int, maximum detection time of human body, range: [0,300], unit:  
second, default value: 10s; when human body grading does not reach the  
configured humanThreshold during the humanDetectionTime, the device stops  
grading the target human body*/  
      "@min":0,  
      "@max":300,  
      "@def":10  
    },  
    "isSupportAdvanceConfiguration":
```

```
/*optional, boolean, whether the device supports advanced parameters of person  
arming (related URI: /ISAPI/Intelligent/channels/<ID>/personArmingTrack/  
advanceConfiguration?format=json)*/  
}  
}
```

E.29 JSON_PersonInfoExtend

JSON message about custom information of human body picture library

```
{  
  "PersonInfoExtend": [{  
/*required, custom extension information*/  
    "id": ,  
/*optional, integer32, serial No. of person custom extension information,  
starts from 1*/  
    "enable": ,  
/*optional, boolean, whether to enable person custom extension information*/  
    "name": ""  
/*optional, string, tag name of person extension information*/  
  }]  
}
```

E.30 JSON_ResponseStatus

JSON message about response status

```
{  
  "requestURL": "",  
/*optional, string, request URL*/  
  "statusCode": ,  
/*optional, int, status code*/  
  "statusString": "",  
/*optional, string, status description*/  
  "subStatusCode": "",  
/*optional, string, sub status code*/  
  "errorCode": ,  
/*required, int, error code, which corresponds to subStatusCode, this field is  
required when statusCode is not 1. The returned value is the transformed  
decimal number*/  
  "errorMsg": "",  
/*required, string, error details, this field is required when statusCode is  
not 1*/  
  "MErrCode": "0xFFFFFFFF",  
/*optional, string, error code categorized by functional modules*/  
  "MErrDevSelfEx": "0xFFFFFFFF"  
/*optional, string, extension of MErrCode. It is used to define the custom
```

```
error code, which is categorized by functional modules*/  
}
```

E.31 XML_DeviceCap

XML message about device capability

```
<DeviceCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  
  <SysCap><!--optional-->  
    <isSupportDst><!--optional, xs: boolean, whether it supports daylight  
saving time--></isSupportDst>  
    <NetworkCap><!--optional, xs: boolean, network capability-->  
    <IOCap><!--optional, IO capability-->  
    <SerialCap><!--optional, serial port capability-->  
    <VideoCap><!--optional, video capability, see details in the message of  
XML_VideoCap-->  
    <AudioCap><!--optional, audio capability-->  
    <isSupportHolidy><!--optional, xs:boolean--></isSupportHolidy>  
    <RebootConfigurationCap>  
      <Genetec><!--optional, xs:boolean--></Genetec>  
      <ONVIF><!--optional, xs:boolean--></ONVIF>  
      <RTSP><!--optional, xs:boolean--></RTSP>  
      <HTTP><!--optional, xs:boolean--></HTTP>  
      <SADP>  
        <ISDiscoveryMode><!--optional, xs:boolean--></ISDiscoveryMode>  
        <PcapMode><!--optional, xs:boolean--></PcapMode>  
      </SADP>  
      <IPCAddStatus><!--optional, xs:boolean--></IPCAddStatus>  
    </RebootConfigurationCap>  
    <isSupportExternalDevice><!--optional, xs:boolean--></  
isSupportExternalDevice>  
    <isSupportChangedUpload>  
      <!--optional, xs: boolean, whether it supports uploading status changes-->  
    </isSupportChangedUpload>  
    <isSupportGettingWorkingStatus>  
      <!--optional, xs:boolean, whether it supports getting device status-->  
    </isSupportGettingWorkingStatus>  
    <isSupportGettingChannelInfoByCondition>  
      <!--optional, xs:boolean-->  
    </isSupportGettingChannelInfoByCondition>  
    <isSupportDiagnosedDataParameter>  
      <!--optional, xs:boolean-->  
    </isSupportDiagnosedDataParameter>  
    <isSupportSimpleDevStatus>  
      <!--optional, xs: boolean, whether it supports getting device working  
status-->  
    </isSupportSimpleDevStatus>  
    <isSupportFlexible>  
      <!--optional, xs: boolean, whether it supports getting channel status by  
condition-->
```



```
</isSupportFlexible>
<isSupportPTZChannels>
  <!--optional, xs:boolean, whether it supports returning PTZ channel
(which is different from the video channel)-->
</isSupportPTZChannels>
<isSupportSubscribeEvent>
  <!--optional, xs:boolean, whether it supports alarm or event
subscription: "true,false"-->
</isSupportSubscribeEvent>
<isSupportDiagnosedData>
  <!--optional, xs:boolean, "true,false", whether it supports diagnosis
data-->
</isSupportDiagnosedData>
<isSupportTimeCap>
  <!--optional, xs:boolean, whether it supports time capability-->
</isSupportTimeCap>
<isSupportThermalStreamData>
  <!--optional, xs:boolean, whether it supports uploading thermal stream
data in real-time. If it is supported, the returned value is "true"; otherwise,
this node will not be returned-->
</isSupportThermalStreamData>
<isSupportPostUpdateFirmware>
  <!--optional,xs:boolean,"true,false", whether it supports upgrading the
firmware-->
</isSupportPostUpdateFirmware>
<isSupportPostConfigData>
  <!--optional, xs:boolean,"true,false", whether it supports importing or
exporting the configuration file-->
</isSupportPostConfigData>
<isSupportUserLock>
  <!--optional, xs:boolean,"true,false", whether it supports locking user-->
</isSupportUserLock>
<isSupportModuleLock><!--optional, xs:boolean, whether it supports locking
the module: "true,false"--></isSupportModuleLock>
<isSupportSoundCfg><!--optional, xs:boolean--></isSupportSoundCfg>
<isSupportMetadata>
  <!--optional, xs:boolean, if it is supported, return "true", otherwise,
this node will not be returned-->
</isSupportMetadata>
<isSupportShutdown><!--optional, xs:boolean, whether it supports shutdown
configuration--></isSupportShutdown>
<supportSmartOverlapChannles opt="1"/><!--optional, xs:boolean, whether it
supports stream configuration of smart events. If this function is supported,
this node and the corresponding channel ID will be returned; otherwise, this
node will not be returned-->
<isSupportConsumptionMode><!--optional, xs:boolean, whether it supports
switching power consumption mode:true (yes), this node is not returned (no).
Related URI: /ISAPI/System/consumptionMode/capabilities?format=json--></
isSupportConsumptionMode>
<isSupportManualPowerConsumption><!--optional, xs:boolean, whether it
supports control the power consumption mode manually: true (yes), this node is
not returned (no)--></isSupportManualPowerConsumption>
```

```
</SysCap>
<voicetalkNums><!--optional, xs:integer, the number of two-way audio
channels--></voicetalkNums>
<isSupportSnapshot><!--optional, xs:boolean, whether it supports capture:
"true, false"--></isSupportSnapshot>
<SecurityCap/><!--optional, security capability-->
<EventCap/><!--optional, event capability-->
<ITCCap><!--optional--></ITCCap>
<ImageCap/><!--optional, image capability-->
<RacmCap/><!--optional, storage capability-->
<PTZCtrlCap>
  <isSupportPatrols><!--optional, xs:boolean--></isSupportPatrols>
  <isSupportCombinedPath><!--optional, xs:boolean, whether the device
supports the PTZ combined path-->true</isSupportCombinedPath>
</PTZCtrlCap>
<SmartCap/><!--optional, intelligent capability-->
<isSupportEhome><!--optional, xs:boolean--></isSupportEhome>
<isSupportStreamingEncrypt><!--optional, xs:boolean--></
isSupportStreamingEncrypt>
<TestCap>
  <isSupportEmailTest><!--optional, xs:boolean--></isSupportEmailTest>
</TestCap>
<ThermalCap/><!--optional, temperature measurement capability-->
<WLAAlarmCap/><!--optional, wireless alarm capability-->
<SecurityCPCapabilities/><!--optional, security control panel capability-->
<isSupportGIS>
  <!--optional, xs:boolean, whether it supports GIS capability-->
</isSupportGIS>
<isSupportCompass>
  <!--optional, xs:boolean-->
</isSupportCompass>
<isSupportRoadInfoOverlays>
  <!--optional, xs:boolean-->
</isSupportRoadInfoOverlays>
<isSupportFaceCaptureStatistics>
  <!--optional, xs:boolean-->
</isSupportFaceCaptureStatistics>
<isSupportExternalDevice>
  <!--optional, xs:boolean-->
</isSupportExternalDevice>
<isSupportElectronicsEnlarge>
  <!--optional, xs:boolean, whether it supports digital zoom-->
</isSupportElectronicsEnlarge>
<isSupportRemoveStorage>
  <!--optional, xs:boolean-->
</isSupportRemoveStorage>
<isSupportCloud>
  <!--optional, xs:boolean-->
</isSupportCloud>
<isSupportRecordHost>
  <!--optional, xs:boolean-->
</isSupportRecordHost>
```

```
<isSupportEagleEye>
  <!--optional, xs:boolean, whether it supports PanoVu series camera-->
</isSupportEagleEye>
<isSupportPanorama>
  <!--optional, xs:boolean, whether it supports panorama-->
</isSupportPanorama>
<isSupportFirmwareVersionInfo>
  <!--optional, xs:boolean, whether it supports displaying firmware version
information-->
</isSupportFirmwareVersionInfo>
<isSupportExternalWirelessServer>
  <!--optional, xs: boolean-->
</isSupportExternalWirelessServer>
<isSupportSetupCalibration>
  <!--optional, xs:boolean, whether it supports setting calibration-->
</isSupportSetupCalibration>
<isSupportGetmutexFuncErrMsg>
  <!--optional, xs:boolean, whether it supports getting mutex information-->
</isSupportGetmutexFuncErrMsg>
<isSupportTokenAuthenticate><!--optional, xs:boolean--></
isSupportTokenAuthenticate>
<isSupportStreamDualVCA><!--optional, xs:boolean--></isSupportStreamDualVCA>
<isSupportlaserSpotManual>
  <!--optional, boolean, whether it supports laser spot configuration-->
</isSupportlaserSpotManual>
<isSupportRTMP><!--optional, xs:boolean--></isSupportRTMP>
<isSupportTraffic><!--optional, xs:boolean--></isSupportTraffic>
<isSupportLaserSpotAdjustment>
  <!--optional, boolean, whether it supports adjusting laser spot size-->
</isSupportLaserSpotAdjustment>
<VideoIntercomCap/><!--optional, video intercom capability-->
<isSupportSafetyCabin>
  <!--optional, xs:boolean-->
</isSupportSafetyCabin>
<isSupportPEA>
  <!--optional, xs:boolean, whether it supports one-touch security control
panel capability-->
</isSupportPEA>
<isSupportCurrentLock>
  <!--optional, xs:boolean, whether it supports locking current
configuration-->
</isSupportCurrentLock>
<isSupportGuardAgainstTheft>
  <!--optional, xs:boolean, whether it supports device anti-theft
configuration-->
</isSupportGuardAgainstTheft>
<isSupportPicInfoOverlap>
  <!--optional, xs:boolean, whether it supports picture information overlay-->
</isSupportPicInfoOverlap>
<isSupportPlay>
  <!--optional, xs: boolean, whether it supports live view: "true,false"-->
</isSupportPlay>
```

```
<isSupportPlayback>
  <!--optional, xs: boolean, whether it supports playback: "true,false"-->
</isSupportPlayback>
<UHFRFIDReader>
  <!--optional, supported capability of UHF RFID card reader-->
  <isSupportBasicInformation>
    <!--optional, xs:boolean, whether it supports basic parameters of UHF
RFID card reader-->
  </isSupportBasicInformation>
  <isSupportHardDiskStorageTest>
    <!--optional, xs:boolean, whether it supports hard disk storage test of
UHF RFID card reader-->
  </isSupportHardDiskStorageTest>
</UHFRFIDReader>
<isSupportIntelligentStructureAnalysis>
  <!--optional, xs:boolean, whether it supports structured VCA-->
</isSupportIntelligentStructureAnalysis>
<isSupportIntelligentAnalysisEngines>
  <!--optional, xs:boolean, whether it supports VCA engine configuration-->
</isSupportIntelligentAnalysisEngines>
<PreviewDisplayNum>
  <!--optional, xs:integer, the number of live view windows, which is the
number of simultaneous live view windows controlled by the device. Limited by
the performance of DeepinMind series network video recorder, currently only
live view of a network camera is supported, and playback is not supported-->
</PreviewDisplayNum>
<isSupportBoard opt="true,false">
  <!--optional, xs:boolean, whether it supports protocol related to sub-
board-->
</isSupportBoard>
<ResourceSwitch>
  <workMode opt="4KPreview,educationRecord">
    <!--req, xs:string, device working mode: "4KPreview"-4K live view mode,
"educationRecord"-education recording mode-->
  </workMode>
</ResourceSwitch>
<isSupportCustomStream><!--optional, xs:boolean--></isSupportCustomStream>
<isSupportTriggerCapCheck>
  <!--optional, xs:boolean, whether it supports verifying capability of alarm
linkage actions-->
</isSupportTriggerCapCheck>
<isSupportActiveMulticast>
  <!--optional, xs: boolean, whether it supports active multicast-->
</isSupportActiveMulticast>
<isSupportChannelEventCap>
  <!--optional, xs:boolean, whether it supports getting event capability by
channel-->
</isSupportChannelEventCap>
<isSupportPictureServer>
  <!-- opt, xs:boolean, whether it supports picture storage server-->
</isSupportPictureServer>
<isSupportVideoCompositeAlarm>
```

```
<!--optional, xs:boolean, whether it supports video double check alarm-->
</isSupportVideoCompositeAlarm>
<isSupportSensorCalibrating>
  <!--optional, xs:boolean, whether it supports double sensor calibration-->
</isSupportSensorCalibrating>
<isSupportChannelEventListCap>
  <!--optional, xs:boolean, whether it supports getting event capability of
all channels-->
</isSupportChannelEventListCap>
<VCAResourceChannelsCap>
  <!--optional, whether it supports independently switching to another VCA
resource by channel-->
  <ChannelsList>
    <channelsID>
      <!--req, xs:integer, channel No. supported by the device-->
    </channelsID>
  </ChannelsList>
</VCAResourceChannelsCap>
<SensorCap/><!--optional, intelligent cabinet capability-->
<isSupportSecurityCP/>
  <!--optional, xs:boolean, whether it supports the applications of security
control panel: "true, false"-->
</isSupportSecurityCP>
<isSupportClientProxyWEB>
  <!--optional, xs:boolean, whether it supports the function that the client
proxy passes through the remote web configuration: "true"-->
</isSupportClientProxyWEB>
<WEBLocation>
  <!--optional, string type, web page location: "local"-local device,
"remote"-remote location. If this node is not returned, the web page will be in
the local device by default-->
</WEBLocation>
<isSupportTime/>
  <!--optional, xs:boolean, "true, false", whether it supports time
configuration-->
</isSupportTime>
<isSupportTimeZone/>
  <!--optional, xs:boolean, "true, false", whether it supports daylight
saving time (DST) configuration-->
</isSupportTimeZone>
<isSupportCityManagement>
  <!--optional, boolean, ro, whether it supports intelligent city management--
>true
</isSupportCityManagement>
<isSupportMixedTargetDetection>
  <!--optional, xs:boolean, "true, false", whether it supports multi-target-
type detection-->
</isSupportMixedTargetDetection>
<isSupportFaceContrastMode>
  <!--optional, xs:boolean, whether it supports face picture comparison mode--
>
</isSupportFaceContrastMode>
```

```
<isSupportPictureCaptureComparision>
  <!--optional, xs:boolean, whether it supports face picture N:1 comparison
between face pictures captured by the camera and imported face pictures-->
</isSupportPictureCaptureComparision>
<isSupportGPSCalibratation>
  <!--optional, xs:boolean, whether it supports GPS calibration capability-->
</isSupportGPSCalibratation>
<isSupportChannelFullEventListCap>
  <!--optional, xs:boolean, whether it supports getting event list capability
of all channels-->
</isSupportChannelFullEventListCap>
<isSupportAUXInfoCap>
  <!--optional, xs:boolean, whether it supports getting property capability
of all channels-->
</isSupportAUXInfoCap>
<isSupportCalibrationFile>
  <!--optional, xs:boolean, whether it supports importing calibration file-->
</isSupportCalibrationFile>
<isSupportDisplayTrajectory>
  <!--optional, xs:boolean, whether it supports displaying trajectory-->
</isSupportDisplayTrajectory>
<maximumSuperPositionTime opt="5,10,20,30">
  <!--dep,xs:integer, the maximum time of trajectory displaying, unit:
second, it is valid only when displaying trajectory is supported-->
</maximumSuperPositionTime>
<isSupportUnitConfig>
  <!--optional, xs:boolean, whether it supports unit configuration-->
</isSupportUnitConfig>
<isSupportAutoMaintenance>
  <!--optional, xs:boolean, whether it supports automatic maintenance. When
this node exists and values "true", it indicates support-->
</isSupportAutoMaintenance>
<isSupportGetLinkSocketIP>
  <!--optional, xs: boolean, "true,false", whether it supports getting the
SocketIP of current connection-->
</isSupportGetLinkSocketIP>
<isSupportIntelligentSearch>
  <!--optional, xs:boolean, whether it supports intelligent search-->
</isSupportIntelligentSearch>
<IOTCap><!--optional, xs:boolean, IoT device access capability-->
  <supportChannelNum>
    <!--req, xs:integer, number of supported channels of IoT device-->
  </supportChannelNum>
  <startChannelNo>
    <!--optional, xs:integer, initial channel ID, if this node is not
inputted, it indicates that the initial channel ID is 1-->
  </startChannelNo>
  <isSupportlinkageChannelsSearch>
    <!--optional, boolean, returns "true" if support, returns "false" if not
support-->
  </isSupportlinkageChannelsSearch>
</IOTCap>
```

```
<isSupportEncryption>
  <!--optional, xs: boolean, stream encryption capability-->
</isSupportEncryption>
<AIDEventSupport opt="abandonedObject, pedestrian, congestion, roadBlock,
construction, trafficAccident, fogDetection, wrongDirection, illegalParking,
SSharpDriving, lowSpeed, dragRacing">
  <!--optional, xs:string, supported traffic incident type: "abandonedObject"-
objects dropped down, "pedestrian"-pedestrian, "congestion"-congestion,
"roadBlock"-roadblock, "construction"-construction, "trafficAccident"-traffic
accident, "fogDetection"-fog, "wrongDirection"-wrong-way driving,
"illegalParking"-illegal parking, "SSharpDriving"-slalom driving, "lowSpeed"-
driving in low speed, "dragRacing"-street racing-->
  </AIDEventSupport>
  <TFSEventSupport
opt="illegalParking ,wrongDirection,crossLane, laneChange, vehicleExist, turnRound,
parallelParking, notKeepDistance, notSlowZebraCrossing, overtakeRightSide, lowSpeed,
dragRacing, changeLaneContinuously, SSharpDriving, largeVehicleOccupyLine, jamCrossL
ine">
  <!--optional, xs:string, supported enforcement event type: "illegalParking"-
illegal parking, "wrongDirection"-wrong-way driving, "crossLane"-driving on the
lane line, "laneChange"-illegal lane change, "vehicleExist"-motor vehicle on
non-motor vehicle lane, "turnRound"-illegal U-turn, "parallelParking"-parallel
parking, "notKeepDistance"-not keeping vehicle distance, "notSlowZebraCrossing"-
not slowing down at zebra corssing, "overtakeRightSide"-overtaking on the
right, "lowSpeed"-driving in low speed, "dragRacing"-street racing,
"changeLaneContinuously"-continuous lane change, "SSharpDriving"-slalom
driving, "largeVehicleOccupyLine"-lane occupation by large-sized vehicle,
"jamCrossLine"-queue jumping-->
  </TFSEventSupport>
  <isVehicleStatisticsSupport>
    <!--optional, xs: boolean, whether it supports setting parameters for
traffic data collection-->
  </isVehicleStatisticsSupport>
  <isSupportIntersectionAnalysis>
    <!--optional, xs: boolean, whether it supports intersection analysis-->
  </isSupportIntersectionAnalysis>
  <supportRemoteCtrl
opt="up, down, left, right, enter, menu, num, power, esc, edit, F1, .prev, rec, play, stop, not
Support"/><!--whether it supports remote control-->
  <isSptDiagnosis>
    <!--optional, xs:boolean, whether it supports device diagnosis: "true",
"false"-->
  </isSptDiagnosis>
  <isSptSerialLogCfg>
    <!--optional, xs:boolean, whether it supports configuring serial port log
redirection: "true", "false"-->
  </isSptSerialLogCfg>
  <isSptFileExport>
    <!--optional, xs:boolean, whether it supports exporting files from the
device: "true", "false"-->
  </isSptFileExport>
  <isSptCertificationStandard>
```

```
<!--optional, xs:boolean, whether it supports configuring authentication
standard for security control panel: "true", "false"-->
</isSptCertificationStandard>
<isSptKeypadLock>
  <!--optional, xs:boolean, whether it supports locking keypad: "true",
"false"-->
  </isSptKeypadLock>
  <MixedTargetDetection><!--optional, whether the device supports recognizing
specific target among mixed targets-->
    <isSupportFaceRecognition><!--optional, xs:boolean, whether it supports
face recognition--></isSupportFaceRecognition>
    <isSupportHumanRecognition><!--optional, xs:boolean, whether it supports
human body recognition--></isSupportHumanRecognition>
    <isSupportVehicleRecognition><!--optional, xs:boolean, whether it supports
vehicle recognition--></isSupportVehicleRecognition>
  </MixedTargetDetection>
  <isSupportDiscoveryMode><!--optional, xs:boolean--></isSupportDiscoveryMode>
  <streamEncryptionType>
    <!--dep, xs:string, stream encryption type: "RTP/TLS", "SRTP/UDP", "SRTP/
MULTICAST". This node is valid when <b>isSupportEncryption</b> is "true", and the
device can support one or more stream encryption types-->
    </streamEncryptionType>
    <isSupportLms><!--optional, xs:boolean, whether it supports laser--></
isSupportLms>
    <isSupportLCDScreen><!--optional, xs:boolean, whether it supports LCD screen--
></isSupportLCDScreen>
    <isSupportBluetooth><!--optional, xs:boolean, whether it supports bluetooth--
></isSupportBluetooth>
    <isSupportAcsUpdate>
      <!--optional, whether it supports upgrading sub access control devices or
peripheral modules: "true"-yes, this node is not returned-no-->
      </isSupportAcsUpdate>
      <isSupportAccessControlCap>
        <!--optional, whether it supports access control capability: "true"-yes,
this node is not returned-no-->
        </isSupportAccessControlCap>
        <isSupportIDCardInfoEvent><!--optional, whether it supports ID card swiping
event: "true"-yes. This node will not be returned if this function is not
supported--></isSupportIDCardInfoEvent>
        <OpenPlatformCap><!--optional, embedded open platform capability, refer to
the message XML_OpenPlatformCap for details-->
        <isSupportInstallationAngleCalibration>
          <!--optional, xs:boolean, whether it supports installation angle
calibration-->
          </isSupportInstallationAngleCalibration>
          <isSupportZeroBiasCalibration>
            <!--optional, xs:boolean, whether it supports zero bias calibration-->
            </isSupportZeroBiasCalibration>
            <isSupportDevStatus><!--optional, xs:boolean, whether device supports getting
device status--></isSupportDevStatus>
            <isSupportRadar><!--optional, xs:boolean, whether it supports the security
radar--></isSupportRadar>
```



```
<isSupportRadarChannels><!--optional, xs:boolean, whether it supports getting
radar channels--></isSupportRadarChannels>
<radarIPDForm><!--optional, xs:string, radar form: "single"-single radar,
"double_diagonal"-two radars forming an 180° diagonal, "double_vertical"-two
radars forming a 90° vertical angle--></radarIPDForm>
<isSupportRadarFieldDetection><!--optional, xs:boolean, whether it supports
intrusion detection (radar)--></isSupportRadarFieldDetection>
<isSupportRadarLineDetection><!--optional, xs:boolean, whether it supports
line crossing detection (radar)--></isSupportRadarLineDetection>
<mixedTargetDetectionWebNoDisplay><!--optional, xs:boolean, whether to enable
not displaying multi-target-type recognition--></
mixedTargetDetectionWebNoDisplay>
<SHMCap><!--opt-->
  <isSupportHighHDDTemperature><!--optional, xs:boolean, whether it supports
HDD high temperature detection--></isSupportHighHDDTemperature>
  <isSupportLowHDDTemperature><!--optional, xs:boolean, whether it supports
HDD low temperature detection--></isSupportLowHDDTemperature>
  <isSupportHDDImpact><!--optional, xs:boolean, whether it supports HDD impact
detection--></isSupportHDDImpact>
  <isSupportHDBadBlock><!--optional, xs:boolean, whether it supports HDD bad
sector detection--></isSupportHDBadBlock>
  <isSupportSevereHDDFailure><!--optional, xs:boolean, whether it supports HDD
severe fault detection--></isSupportSevereHDDFailure>
</SHMCap>
<isSupportBVCorrect><!--optional, xs:boolean, whether it supports configuring
camera correction parameters--></isSupportBVCorrect>
<guideEventSupport opt="linkageCapture">
  <!--optional,xs:string, events which support quick setup by instruction,
"linkageCapture"-capture by linkage-->
</guideEventSupport>
<isSupportAutoSwitch><!--optional, xs:boolean, whether it supports auto
switch--> true</isSupportAutoSwitch>
<isSupportDataPrealarm><!--optional,xs:boolean, whether it supports traffic
pre-alarm event--></isSupportDataPrealarm>
<supportGISEvent opt="AID,TPS,ANPR,mixedTargetDetection">
  <!--optional, xs:string, event types that support GIS information access:
AID (corresponding SDK event: COMM_ALARM_AID_V41), TPS (corresponding SDK
event: COMM_ALARM_TPS_REAL_TIME), ANPR (corresponding SDK event:
COMM_ITS_PLATE_RESULT), mixedTargetDetection-mixed targets detection-->
</supportGISEvent>
<isSupportIntelligentMode><!--optional, xs:boolean, whether it supports
intelligent scene switch (related URI:/ISAPI/System/IntelligentSceneSwitch?
format=json)--></isSupportIntelligentMode>
<isSupportCertificateCaptureEvent><!--optional, xs:boolean, whether it
supports certificate capture and comparison events: true=yes. If this function
is not supported, this node will not be returned--></
isSupportCertificateCaptureEvent>
<isSupportAlgorithmsInfo><!--optional, xs:boolean, whether it supports
getting the algorithm library version information: true=yes. If this function
is not supported, this node will not be returned--></isSupportAlgorithmsInfo>
<isSupportVibrationDetection><!--optional, xs:boolean, whether it supports
vibration detection--></isSupportVibrationDetection>
```

```
<isSupportFaceTemperatureMeasurementEvent><!--optional, xs:boolean, whether
it supports uploading face thermography events (eventType:
"FaceTemperatureMeasurementEvent")--></isSupportFaceTemperatureMeasurementEvent>
<isSupportQRCodeEvent><!--optional, xs:boolean, whether it supports uploading
QR code events (eventType: "QRCodeEvent")--></isSupportQRCodeEvent>
<isSupportPersonArmingTrack><!--optional, xs:boolean, whether device supports
person arming (related URI: /ISAPI/Intelligent/channels/<ID>/personArmingTrack/
capabilities?format=json)--></isSupportPersonArmingTrack>
<isSupportManualPersonArmingTrack><!--optional, xs:boolean, whether device
supports manual person arming (related URI: /ISAPI/Intelligent/channels/<ID>/
manualPersonArmingTrack?format=json)--></isSupportManualPersonArmingTrack>
<isSupportGPSCalibrationMode><!--optional, xs:boolean, whether device
supports GPS calibration (related URI: /ISAPI/System/GPSCalibration/channels/
<ID>/mode?format=json)--></isSupportGPSCalibrationMode>
<isSupportGPSVerification><!--optional, xs:boolean, whether device supports
GPS verification (related URI: /ISAPI/System/GPSVerification/channels/<ID>/
points?format=json)--></isSupportGPSVerification>
<isSupportHBDLib><!--optional, xs:boolean, whether device supports human body
picture library (related URI: /ISAPI/Intelligent/HBDLib/capabilities?
format=json)--></isSupportHBDLib>
<isSupportFireEscapeDetection><!--optional, xs:boolean, whether the device
supports fire engine access detection (related URI: /ISAPI/Intelligent/channels/
<ID>/fireEscapeDetection/capabilities?format=json)--></
isSupportFireEscapeDetection>
<isSupportTakingElevatorDetection><!--optional, xs:boolean, whether the
device supports elevator detection (related URI: /ISAPI/Intelligent/channels/
<ID>/takingElevatorDetection/capabilities?format=json)--></
isSupportTakingElevatorDetection>
<isSupportSSDFileSystemUpgrade><!--optional, xs:boolean, whether the device
supports SSD file system upgrade (related URI: /ISAPI/System/SSDFileSystem/
upgrade?format=json)--></isSupportSSDFileSystemUpgrade>
<isSupportSSDFileSystemFormat><!--optional, xs:boolean, whether the device
supports SSD file system formatting (related URI: /ISAPI/System/SSDFileSystem/
format?format=json)--></isSupportSSDFileSystemFormat>
<isSupportSSDFileSystemCapacity><!--optional, xs:boolean, whether the device
supports getting space distribution information of SSD file system (related
URI: /ISAPI/System/SSDFileSystem/capacity?format=json)--></
isSupportSSDFileSystemCapacity>
<isSupportAIOpenPlatform><!--optional, xs:boolean, whether the device
supports AI open platform capabilities; if supports, this node will be returned
and its value is true; if not, this node will not be returned--></
isSupportAIOpenPlatform>
<isSupportPictureDownloadError><!--optional, xs:boolean, whether the device
supports reporting picture download failure--></isSupportPictureDownloadError>
<characteristicCode min="1" max="128"><!--optional, xs:string, device
attribute code (related URI: /ISAPI/System/deviceInfo/characteristicCode?
format=json)--></characteristicCode>
<isSupportContainerDetection><!--optional, xs:boolean, whether the device
supports container detection (if this node is not returned, refer to the value
returned by /ISAPI/Traffic/ContentMgmt/InputProxy/channels/<ID>/ocrScene/
capabilities to find whether the device supports container detection)--></
isSupportContainerDetection>
```

```
<isSupportLensParamFile><!--optional, xs:boolean, whether the device supports
exporting and importing the lens parameters file--></isSupportLensParamFile>
<isSupportCounting><!--optional, xs:boolean, ro, whether it supports people
counting--></isSupportCounting>
<isSupportFramesPeopleCounting><!--optional, xs:boolean, ro, whether it
supports regional people counting--></isSupportFramesPeopleCounting>
<zoomFocusWebDisplay
opt="ROI,roadTrafficDetection,SMD,mixedTargetDetection,faceCapture"><!--
optional, string, zoom and focus page supported by the Web Client--></
zoomFocusWebDisplay>
<isSupportDebugLogModuleType
opt="playService,communicationService,attendanceService,faceService"><!--
optional, xs:boolean, whether to export the debugging logs by module type; the
value of <moduleType> in the URI (/ISAPI/System/debugLog?
format=json&moduleType=<moduleType>) can be: "playService",
"communicationService", "attendanceService", "faceService"--></
isSupportDebugLogModuleType>
</isSupportPlateQuaAlarm>
<isSupportWiegand><!--optional, xs:boolean, ro, whether it supports the
Wiegand protocol (related URI: /ISAPI/System/Wiegand/<wiegandID>/capabilities?
format=json)-->true</isSupportWiegand>
<isSupportChannelOccupy><!--optional, xs:boolean, whether it supports
detection of outdoor fire escape occupied by vehicle--></isSupportChannelOccupy>
<isSupportOffDuty><!--optional, xs:boolean, whether it supports detection of
person absent in fire control room--></isSupportOffDuty>
<isSupportNoCertificate><!--optional, xs:boolean, whether it supports
detection of authenticated staff not enough in fire control room--></
isSupportNoCertificate>
<isSupportSmokeAlarm><!--optional, xs:boolean, whether it supports smoke
alarm--></isSupportSmokeAlarm>
<isSupportBatteryCarDisobey><!--optional, xs:boolean, whether it supports
electric scooter parking violation detection--></isSupportBatteryCarDisobey>
<isSupportNoFireExtinguisherRecog><!--optional, xs:boolean, whether it
supports fire extinguisher missing detection--></
isSupportNoFireExtinguisherRecog>
<isSupportIndoorPasswayBlock><!--optional, xs:boolean, whether it supports
indoor channel blockage detection--></isSupportIndoorPasswayBlock>
<isSupportFireSmartFireDetect><!--optional, xs:boolean, whether it supports
fire source detection--></isSupportFireSmartFireDetect>
<isSupportDetectorRunningStatus><!--optional, xs:boolean, whether it supports
detector running status--></isSupportDetectorRunningStatus>
<isSupportDetectorOperationStatus><!--optional, xs:boolean, whether it
supports detector operation status--></isSupportDetectorOperationStatus>
<isSupportDetectorTemperatureAlarm
opt="highTemperature,riseTemperature,flame"><!--optional, xs:boolean, whether
it supports temperature alarm: "highTemperature" (high temperature alarm),
"riseTemperature" (temperature rising alarm), "flame" (flame alarm)--></
isSupportDetectorTemperatureAlarm>
<isSupportDetectorShelterAlarm><!--optional, xs:boolean, whether it supports
detector video tampering alarm--></isSupportDetectorShelterAlarm>
<isSupportDetectorMotionAlarm><!--optional, xs:boolean, whether it supports
detector movement alarm--></isSupportDetectorMotionAlarm>
```

```
<isSupportDetectorTamperAlarm><!--optional, xs:boolean, whether it supports
detector tampering alarm--></isSupportDetectorTamperAlarm>
<isSupportDetectorEmergencyAlarm><!--optional, xs:boolean, whether it
supports detector emergency alarm--></isSupportDetectorEmergencyAlarm>
<isSupportSmokingDetectAlarm><!--optional, xs:boolean, whether it supports
smoking alarm--></isSupportSmokingDetectAlarm>
<isSupportDetectorSmokeAlarm><!--optional, xs:boolean, whether it supports
smoke alarm--></isSupportDetectorSmokeAlarm>
<isSupportDetectorCombustibleGasAlarm><!--optional, xs:boolean, whether it
supports gas alarm--></isSupportDetectorCombustibleGasAlarm>
<isSupportFireControlData><!--optional, xs:boolean, whether it supports
uploading real-time fire protection data--></isSupportFireControlData>
<isSupportFireNoRegulation><!--optional, xs:boolean, whether it supports fire
no regulation alarm--></isSupportFireNoRegulation>
<isSupportSmokeFireRecognize><!--optional, xs:boolean, whether it supports
uploading the smoke and fire detection event--></isSupportSmokeFireRecognize>
</DeviceCap>
```

E.32 XML_EventNotificationAlert_AlarmEventInfo

EventNotificationAlert message with alarm/event information in XML format.

```
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--dep, xs:string, device IPv4 address--></ipAddress>
  <ipv6Address><!--dep, xs:string, device IPv6 address--></ipv6Address>
  <portNo><!--opt, xs:integer, device port number--></portNo>
  <protocol><!--opt, xs:string, protocol type for uploading alarm/event
information, "HTTP,HTTPS"--></protocol>
  <macAddress><!--opt, xs:string, MAC address--></macAddress>
  <channelID><!--dep, xs:string, device channel No., starts from 1--></
channelID>
  <dateTime><!--req, alarm/event triggered or occurred time, format:
2017-07-19T10:06:41+08:00--></dateTime>
  <activePostCount><!--req, xs:integer, alarm/event frequency, starts from 1--
></activePostCount>
  <eventType><!--req, xs:string, alarm/event type, "peopleCounting, ANPR,..."--
></eventType>
  <eventState>
    <!--req, xs:string, durative alarm/event status: "active"-valid, "inactive"-
invalid, e.g., when a moving target is detected,
    the alarm/event information will be uploaded continuously until the status
is set to "inactive"-->
  </eventState>
  <eventDescription><!--req, xs:string, alarm/event description--></
eventDescription>
  <...><!--opt, for different alarm/event types, the nodes are different, see
the message examples in different applications--></...>
</EventNotificationAlert>
```

E.33 XML_EventNotificationAlert_HeartbeatInfo

EventNotificationAlert message with heartbeat information (when there is no alarm is triggered) in XML format

```
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <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, heartbeat uploaded time, format:
2017-07-19T10:06:41+08:00--></dateTime>
  <activePostCount><!--req, xs:integer, heartbeat frequency, starts from 1--></
activePostCount>
  <eventType><!--req, xs:string, for heartbeat, it is "videoloss"--></eventType>
  <eventState>
    <!--req, xs:string, for heartbeat, it is "inactive"-->
  </eventState>
  <eventDescription><!--req, xs: string, description--></eventDescription>
</EventNotificationAlert>
```

Remarks

- For network camera or network speed dome with the version 5.5.0 and lower, the heartbeat frequency is 300 ms per heartbeat.
- For network camera or network speed dome with the version 5.5.0 and higher, the heartbeat frequency is 10 s per heartbeat. If no heartbeat received for continuous 30 s, it indicates that the heartbeat is timed out.

Example

Message Example of Heartbeat

```
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.com/ver20/
XMLSchema">
  <ipAddress>10.17.133.46</ipAddress>
  <portNo>80</portNo>
  <protocol>HTTP</protocol>
  <macAddress>44:19:b6:6d:24:85</macAddress>
  <channelID>1</channelID>
  <dateTime>2017-05-04T11:20:02+08:00</dateTime>
  <activePostCount>0</activePostCount>
  <eventType>videoloss</eventType>
  <eventState>inactive</eventState>
  <eventDescription>videoloss alarm</eventDescription>
</EventNotificationAlert>
```

E.34 XML_EventTrigger

Linkage parameter message in XML format

```
<EventTrigger version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:string, ID--></id>
  <eventType>
    <!--required, xs:string, see details in the "Remarks" below-->
  </eventType>
  <eventDescription><!--optional, xs:string--></eventDescription>
  <inputIOPortID><!--dependent, xs:string, alarm input ID--></inputIOPortID>
  <dynInputIOPortID><!--dependent, xs:string, dynamic alarm input ID--></
dynInputPortID>
  <videoInputChannelID>
    <!--dependent, xs:string, video input channel ID, it is valid when
<event><b>eventType</b></event> is "VMD, videoloss, tamperdetection, regionEntrance, regionExiting,
loitering, group, rapidMove, parking, unattendedBaggage, attendedBaggage"-->
  </videoInputChannelID>
  <dynVideoInputChannelID><!--dependent, xs:string, dynamic video input channel
ID--></dynVideoInputChannelID>
  <intervalBetweenEvents><!--optional, xs:integer, event time interval, unit:
second--></intervalBetweenEvents>
  <WLSensorID><!--dependent, xs:string, ID--></WLSensorID>
  <EventTriggerNotificationList/><!--optional, alarm/event linkage actions, see
details in the message of XML_EventTriggerNotificationList-->
</EventTrigger>
```

Remarks

The node **<eventType>** can be the following values: IO, VMD, videoloss, raidfailure, recordingfailure, badvideo, POS, analytics, fanfailure, overheat, tamperdetection, diskfull, diskerror, nicbroken, ipconflict, illaccess, videomismatch, resolutionmismatch, radifailure, PIR, WLSensor, spareException, poePowerException, heatmap, counting, linedetection, fieldddetection, regionEntrance, regionExiting, loitering, group,rapidMove, parking, unattendedBaggage, attendedBaggage, HUMANATTRIBUTE, blacklist, whitelist, peopleDetection, allVehicleList, otherVehicleList, vehicledetection, storageDetection, shipsDetection, humanAttribute, faceContrast, blacklistFaceContrast, whiteListFaceContrast, faceSnap, faceLib, personDensityDetection, personQueueDetecton, mixedTargetDetection, HVTVehicleDetection, illegalParking, pedestrian, trafficAccident, construction, roadblock, abandonedObject, parallelParking, parkingState, congestion, intersectionAnalysis, heatMap, thermometry, shipsFlowDetection, dredgerDetection, reverseEntrance, luma, highHDTemperature, lowHDTemperature, hdImpact, hdBadBlock, SevereHDFailure, safetyHelmetDetection, vibrationDetection, HBDLib,TMPA,faceThermometry,noMaskDetection, detectorTemp, detectorSmoke, detectorTamper, smokeFireRecognize, indoorPasswayBlock, detectorShelter, detectorMotion, fireNoRegulation.

See Also

[XML_EventTriggerNotificationList](#)

E.35 XML_EventTriggerCapType

XML message about capability of alarm linkage action types

```
<EventTriggerCapType version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <isSupportCenter><!--optional, xs:boolean--></isSupportCenter>
  <isSupportRecord><!--optional, xs:boolean--></isSupportRecord>
  <isSupportMonitorAlarm><!--optional, xs:boolean--></isSupportMonitorAlarm>
  <isSupportBeep><!--optional, xs: boolean, whether it supports audible
warning--></isSupportBeep>
  <isSupportIO><!--optional, xs:boolean--></isSupportIO>
  <isSupportFTP><!--optional, xs:boolean--></isSupportFTP>
  <isSupportEmail><!--optional, xs:boolean--></isSupEmail>
  <isSupportLightAudioAlarm><!--optional, xs:boolean--></
isSupportLightAudioAlarm>
  <isSupportFocus><!--optional, xs:boolean--></isSupportFocus>
  <isSupportPTZ><!--optional, xs:boolean--></isSupportPTZ>
  <maxPresetActionNum>
    <!--dependent, xs:integer, it is valid only when <isSupportPTZ> is "true"-->
  </maxPresetActionNum>
  <maxPatrolActionNum>
    <!--dependent, xs:integer, it is valid only when <isSupportPTZ> is "true"-->
  </maxPatrolActionNum>
  <maxPatternActionNum>
    <!--dependent, xs:integer, it is valid only when <isSupportPTZ> is "true"-->
  </maxPatternActionNum>
  <isSupportTrack><!--optional, xs:boolean, whether it supports PTZ linked
tracking--></isSupportTrack>
  <isSupportWhiteLight>
    <!--optional, xs: boolean, whether it supports supplement light alarm
linkage-->
  </isSupportWhiteLight>
  <isSupportCloud><!--optional, xs:boolean, whether it supports upload to the
cloud--></isSupportCloud>
  <targetNotificationInterval max="1000" min="0" default="30"><!--xs:integer,
range: [0, 1000], the default value is 30, unit: seconds, this node is valid
for <MotionDetectionTriggerCap> and <TamperDetectionTriggerCap> and this node
is valid when <isSupportPTZ> is "true"--></targetNotificationInterval>
  <direction opt="both,forward,reverse"><!--xs:string, triggering direction,
this node is valid for the node <BlackListTriggerCap>, <WhiteListTriggerCap>,
and <VehicleDetectionTriggerCap>--></direction>
  <presetDurationTime min="" max=""><!--dependent, xs:integer--></
presetDurationTime>
  <isSupportSMS><!--optional, xs:boolean, whether to support SMS (Short Message
Service)--></isSupportSMS>
  <maxCellphoneNum><!--dependent, xs:integer, the maximum number of cellphones,
```

```

which is node is valid only when <isSupportSMS> is "true"--></maxCellphoneNum>
  <isSupportOSD><!--optional, xs:boolean--></isSupportOSD>
  <isSupportAudio><!--optional, xs:boolean, whether it supports setting audio
alarm independently. If this node is set to "true", audio alarm and buzzer
alarm can be linked separately, and the lineage method is audio--></
isSupportAudio>
  <AudioAction><!--dependent, this node is valid when <isSupportBeep> is "true"
or <isSupportAudio> is "true"-->
    <audioTypeList>
      <audioType><!--list-->
        <audioID><!--required, xs:integer, alarm sound type--></audioID>
        <audioDescription><!--required, xs:string, alarm sound description, it
should correspond to the alarm sound type--></audioDescription>
      </audioType>
    </audioTypeList>
    <alarmTimes opt="0,1,2,3,4,5,6,7,8,9,255"><!--required, xs:integer, alarm
times, it is between 0 and 9, 255-continuous alarm, unit: time--></alarmTimes>
  </AudioAction>
  <isSupportSMS><!--optional, xs:boolean --></isSupportSMS>
  <maxCellphoneNum><!--dependent, if <isSupportSMS> is true, xs:integer--></
maxCellphoneNum>
  <isNotSupportCenterModify><!--optional, xs:boolean, whether editing
configuration parameters of the monitoring center is not supported: "true"-yes
(configuration parameters of the monitoring center cannot be edited), "false"
or this node is not returned-no (configuration parameters of the monitoring
center can be edited)--></isNotSupportCenterModify>
  <isSupportMessageConfig>
    <!--optional, xs:boolean, whether it supports SMS configuration, if
supports, set cellphoneNumber to null-->
  </isSupportMessageConfig>
  <isSupportAnalogOutput><!--optional, xs:boolean, whether it supports IO
output of linkage analog channel--></isSupportAnalogOutput>
  <isSupportIOOutputUnify><!--optional, xs:boolean, whether it supports
configuration of IO output--></isSupportIOOutputUnify>
  <isSupportFaceContrast><!--optional, xs:boolean, whether it supports face
picture comparison linkage--></isSupportFaceContrast>
  <isSupportSiren><!--optional, xs:boolean, whether it supports siren linkage--
></isSupportSiren>
  <isSupportOutput><!--optional, xs:boolean, whether it supports relay linkage--
></isSupportOutput>
</EventTriggerCapType>

```

E.36 XML_EventTriggerNotification

Event linkage notification message in XML format

```

<EventTriggerNotification><!--opt-->
  <id><!--required, xs:string, device ID--></id>
  <notificationMethod>
    <!--required, xs:string, linkage actions,

```



```
opt="email,IM,IO,syslog,HTTP,FTP,beep,ptz,record, monitorAlarm, center,
LightAudioAlarm,focus,trace,cloud,SMS,whiteLight,audio,whiteLight,faceContrast,s
iren,output"-->
</notificationMethod>
<notificationRecurrence>
  <!--optional, xs:string, "beginning,beginningandend,recurring"-->
</notificationRecurrence>
<notificationInterval><!--dependent, xs:integer, unit: millisecond--></
notificationInterval>
  <outputIOPortID><!--dependent, xs:string, video output No., it is required
only when notificationMethod is "IO"--></outputIOPortID>
  <dynOutputIOPortID><!--dependent, xs:string, dynamic video output No., it is
required only when notificationMethod is "IO"--></dynOutputIOPortID>
  <videoInputID><!--dependent, xs:string, video input No., it is required only
when notificationMethod is "record"--></videoInputID>
  <dynVideoInputID><!--dependent, xs:string, dynamic video input No., it is
required only when notificationMethod is "record"--></dynVideoInputID>
  <ptzAction><!--dependent, it is required only when notificationMethod is
"ptz"-->
    <ptzChannelID><!--required, xs:string, PTZ channel ID--></ptzChannelID>
    <actionName><!--required, xs:string, PTZ control type: "preset", "pattern",
"patrol"--></actionName>
    <actionNum><!--dependent, xs:integer></actionNum>
  </ptzAction>
  <WhiteLightAction><!--dependent, white light linkage parameters, this node is
valid when notificationMethod is "whiteLight"-->
    <whiteLightDurationTime><!--required, xs:integer, white light flashing
duration, it is between 1 and 60, unit: second--></whiteLightDurationTime>
  </WhiteLightAction>
  <cellphoneNumber><!--dependent, xs:string, min="0" max="11",cellphone number--
></cellphoneNumber-->
</EventTriggerNotification>
```

E.37 XML_EventTriggerNotificationList

EventTriggerNotificationList message in XML format

```
<EventTriggerNotificationList version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <EventTriggerNotification/><!--opt, see details in the message of
XML_EventTriggerNotification-->
</EventTriggerNotificationList>
```

See Also

[XML_EventTriggerNotification](#)

E.38 XML_EventTriggersCap

XML message about linkage capabilities of different alarm categories

```
<EventTriggersCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <DiskfullTriggerCap><!--optional, xs: EventTriggerCapType--></
DiskfullTriggerCap>
  <DiskerrorTriggerCap><!--optional, xs: EventTriggerCapType--></
DiskerrorTriggerCap>
  <NicbrokenTriggerCap><!--optional, xs: EventTriggerCapType--></
NicbrokenTriggerCap>
  <IpconflictTriggerCap><!--optional, xs: EventTriggerCapType--></
IpconflictTriggerCap>
  <IllaccesTriggerCap><!--optional, xs: EventTriggerCapType--></
IllaccesTriggerCap>
  <BadvideoTriggerCap><!--optional, xs: EventTriggerCapType--></
BadvideoTriggerCap>
  <VideomismatchTriggerCap><!--optional, xs: EventTriggerCapType--></
VideomismatchTriggerCap>
  <IOTriggerCap><!--optional, xs: EventTriggerCapType--></IOTriggerCap>
  <LineDetectTriggerCap><!--optional, xs: EventTriggerCapType--></
LineDetectTriggerCap>
  <RegionEntranceTriggerCap><!--optional, xs: EventTriggerCapType--></
RegionEntranceTriggerCap>
  <RegionExitingTriggerCap><!--optional, xs: EventTriggerCapType--></
RegionExitingTriggerCap>
  <LoiteringTriggerCap><!--optional, xs: EventTriggerCapType--></
LoiteringTriggerCap>
  <GroupDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
GroupDetectionTriggerCap>
  <RapidMoveTriggerCap><!--optional, xs: EventTriggerCapType--></
RapidMoveTriggerCap>
  <ParkingTriggerCap><!--optional, xs: EventTriggerCapType--></
ParkingTriggerCap>
  <UnattendedBaggageTriggerCap><!--optional, xs: EventTriggerCapType--></
UnattendedBaggageTriggerCap>
  <AttendedBaggageTriggerCap><!--optional, xs: EventTriggerCapType--></
AttendedBaggageTriggerCap>
  <FireDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
FireDetectionTriggerCap>
  <FireDetectionCap><!--optional, xs: EventTriggerCapType--></FireDetectionCap>
  <StorageDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
StorageDetectionTriggerCap>
  <ShipsDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
ShipsDetectionTriggerCap>
  <ThermometryCap><!--optional, xs: EventTriggerCapType--></ThermometryCap>
  <VandalProofTriggerCap><!--optional, xs: EventTriggerCapType--></
VandalProofTriggerCap>
  <BlackListTriggerCap><!--opt, xs: EventTriggerCapType, configuration
capability of blocklist arming linkage--></BlackListTriggerCap>
  <WhiteListTriggerCap><!--opt, xs: EventTriggerCapType, configuration
```

```
capability of allowlist arming linkage--></WhiteListTriggerCap>
  <AllVehicleListTriggerCap><!--optional, xs:EventTriggerCapType, configuration
capability of other list arming linkage--></AllVehicleListTriggerCap>
  <OtherVehicleListTriggerCap><!--optional, xs:EventTriggerCapType--></
OtherVehicleListTriggerCap>
  <PeopleDetectionTriggerCap><!--optional, xs:EventTriggerCapType--></
PeopleDetectionTriggerCap>
  <PIRAAlarmCap><!--optional, xs: EventTriggerCapType--></PIRAAlarmCap>
  <TamperDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
TamperDetectionTriggerCap>
  <DefocusDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
DefocusDetectionTriggerCap>
  <FaceDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
FaceDetectionTriggerCap>
  <SceneChangeDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
SceneChangeDetectionTriggerCap>
  <VandalProofAlarmCap><!--optional, xs: EventTriggerCapType--></
VandalProofAlarmCap>
  <JudgmentTriggerCap><!--optional, xs: EventTriggerCapType--></
JudgmentTriggerCap>
  <FightingTriggerCap><!--optional, xs: EventTriggerCapType--></
FightingTriggerCap>
  <RisingTriggerCap><!--optional, xs: EventTriggerCapType--></RisingTriggerCap>
  <DozingTriggerCap><!--optional, xs: EventTriggerCapType--></DozingTriggerCap>
  <CountingTriggerCap><!--optional, xs: EventTriggerCapType--></
CountingTriggerCap>
  <VideoLossTriggerCap><!--optional, xs: EventTriggerCapType--></
VideoLossTriggerCap>
  <HideTriggerCap><!--optional, xs:EventTriggerCapType--></HideTriggerCap>
  <AlarmInTriggerCap><!--optional, xs: EventTriggerCapType--></
AlarmInTriggerCap>
  <VehicleDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
VehicleDetectionTriggerCap>
  <AudioExceptionCap><!--optional, xs: EventTriggerCapType--></
AudioExceptionCap>
  <FiledDetectTriggerCap><!--optional, xs: EventTriggerCapType--></
FiledDetectTriggerCap>
  <MotionDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
MotionDetectionTriggerCap>
  <TemperatureCap><!--optional, xs: EventTriggerCapType--></TemperatureCap>
  <IntelligentTriggerCap><!--optional, xs: EventTriggerCapType--></
IntelligentTriggerCap>
  <FaceContrastTriggerCap><!--optional, xs: EventTriggerCapType, face picture
comparison alarm linkage--></FaceContrastTriggerCap>
  <PersonDensityDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
PersonDensityDetectionTriggerCap>
  <PersonQueueDetectionTriggerCap><!--optional, xs: EventTriggerCapType, queue
management alarm linkage--></PersonQueueDetectionTriggerCap>
  <HumanRecognitionTriggerCap><!--optional, xs: EventTriggerCapType--></
HumanRecognitionTriggerCap>
  <FaceSnapTriggerCap><!--optional, xs: EventTriggerCapType--></
FaceSnapTriggerCap>
```

```
<isSupportWhiteLightAction>
  <!--dependent, xs: boolean, see details in EventTriggerCapType, it is valid
when isSupportWhiteLight is "true"-->
</isSupportWhiteLightAction>
<isSupportAudioAction>
  <!--dependent, xs: boolean, see details in EventTriggerCapType, it is valid
when isSupportBeep is "true"-->
</isSupportAudioAction>
<HFPDTriggerCap><!--optional, xs: EventTriggerCapType--></HFPDTriggerCap>
<MixedTargetDetectionCap><!--optional, xs: EventTriggerCapType--></
MixedTargetDetectionCap>
<HVTVehicleDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
HVTVehicleDetectionTriggerCap>
<VCATriggerCap><!--optional, xs: EventTriggerCapType--></VCATriggerCap>
<PIRCap><!--optional, xs: EventTriggerCapType--></PIRCap>
<IllegalParkingTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports illegal parking detection--></IllegalParkingTriggerCap>
<PedestrianTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports pedestrian detection--></PedestrianTriggerCap>
<TrafficAccidentTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports traffic accident detection--></TrafficAccidentTriggerCap>
<ConstructionTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports construction detection--></ConstructionTriggerCap>
<RoadBlockTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports roadblock detection--></RoadBlockTriggerCap>
<AbandonedObjectTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports objects dropped down detection--></AbandonedObjectTriggerCap>
<ParallelParkingTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports parallel parking detection--></ParallelParkingTriggerCap>
<ParkingStateTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports parking space status detection, currently this node is not supported--
></ParkingStateTriggerCap>
<CongestionTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports congestion detection--></CongestionTriggerCap>
<IntersectionAnalysisCap><!--optional, xs: EventTriggerCapType, whether it
supports intersection analysis--></IntersectionAnalysisCap>
<ShipsFlowDetectionTriggerCap><!--optional,xs:EventTriggerCapType, ship flow
detection--></ShipsFlowDetectionTriggerCap>
<dredgerDetectionTriggerCap><!--optional,xs:EventTriggerCapType, dredger
detection--></dredgerDetectionTriggerCap>
<voltageInstableTriggerCap><!--optional,xs:EventTriggerCapType, supply
voltage exception--></voltageInstableTriggerCap>
<HighHDDTemperatureTriggerCap><!--optional, xs:EventTriggerCapType, HDD high
temperature detection--></HighHDDTemperatureTriggerCap>
<LowHDDTemperatureTriggerCap><!--optional, xs:EventTriggerCapType, HDD low
temperature detection--></LowHDDTemperatureTriggerCap>
<HDImpactTriggerCap><!--optional, xs:EventTriggerCapType, HDD impact
detection--></HDImpactTriggerCap>
<HDBadBlockTriggerCap><!--optional, xs:EventTriggerCapType, HDD bad sector
detection--></HDBadBlockTriggerCap>
<SevereHDDFailureTriggerCap><!--optional, xs:EventTriggerCapType, HDD severe
fault detection--></SevereHDDFailureTriggerCap>
```

```
<HUMANATTRIBUTECap><!--optional, xs:EventTriggerCapType--></HUMANATTRIBUTECap>
<HumanAttributeTriggerCap><!--optional, xs:EventTriggerCapType, human body
attribute--></HumanAttributeTriggerCap>
<BlackListFaceContrastTriggerCap><!--opt, xs:EventTriggerCapType, alarm
linkage capability of blacklist face comparison--></
BlackListFaceContrastTriggerCap>
<FaceLibTriggerCap><!--optional, xs:EventTriggerCapType--></FaceLibTriggerCap>
<SafetyHelmetDetectionTriggerCap><!--optional, xs:EventTriggerCapType, alarm
linkage capability of hard hat detection--></SafetyHelmetDetectionTriggerCap>
<VibrationDetectionTriggerCap><!--optional, xs:EventTriggerCapType, alarm
linkage capability of vibration detection--></VibrationDetectionTriggerCap>
<RadarLineDetectionTriggerCap><!--optional, xs:EventTriggerCapType, alarm
linkage capability of radar line crossing detection--></
RadarLineDetectionTriggerCap>
<RadarFieldDetectionTriggerCap><!--optional, xs:EventTriggerCapType, alarm
linkage capability of radar intrusion detection--></
RadarFieldDetectionTriggerCap>
<HBDLibTriggerCap><!--optional, xs:EventTriggerCapType, alarm linkage
capability of human body picture library--></HBDLibTriggerCap>
<FaceThermometryCap><!--optional, xs:EventTriggerCapType--></
FaceThermometryCap>
<NoMaskDetectionTriggerCap><!--optional, xs:EventTriggerCapType, alarm
linkage capability of no wearing mask detection--></NoMaskDetectionTriggerCap>
<TMPATriggerCap><!--optional, xs:EventTriggerCapType, alarm linkage
capability of temperature measurement pre-alarm--></TMPATriggerCap>
<FireEscapeDetectionTriggerCap><!--optional, xs:EventTriggerCapType, alarm
linkage capability of fire engine access detection--></
FireEscapeDetectionTriggerCap>
<TakingElevatorDetectionTriggerCap><!--optional, xs:EventTriggerCapType,
alarm linkage capability of elevator detection--></
TakingElevatorDetectionTriggerCap>
<RuleTriggerCap><!--optional, linkage capability of rule triggered alarm -->
  <isSupportCityManagement>
    <!--optional, xs:boolean, whether the city management supports setting
linkage actions by area; if supports, the value is true, otherwise, this node
will not be returned-->
  </isSupportCityManagement>
</RuleTriggerCap>
<ThermalCalibrationFileExceptionCap><!--optional, xs:EventTriggerCapType,
alarm linkage capability of thermography calibration file exception--></
ThermalCalibrationFileExceptionCap>
</EventTriggersCap>
```

See Also

[XML_EventTriggerCapType](#)

E.39 XML_HttpHostNotification

XML message about parameters of a HTTP listening server

```

<HttpHostNotification version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <id><!--required, xs:string, ID--></id>
  <url><!--required, xs:string, the absolute path, e.g., http://
<ipAddress>:<portNo>/<uri>--></url>
  <protocolType><!--required, xs:string, "HTTP,HTTPS,EHome". When the value of
this node is EHome, this message can be used to configure the information (such
as IP address and port No.) for the AMS (Alarm Management Server) of ISUP 5.0
to listen on. For communication via ISUP 5.0 (MQTT), the listening parameters
can be configured and obtained; For other communication methods (such as
communication via HTTP, HTTPS, EZVIZ Open Platform, etc.), the listening
parameters can only be obtained and cannot be configured--></protocolType>
  <parameterFormatType><!--required, xs:string, alarm/event information format,
"XML,JSON"--></parameterFormatType>
  <addressingFormatType><!--required, xs:string, "ipaddress,hostname"--></
addressingFormatType>
  <hostName><!--dependent, xs:string--></hostName>
  <ipAddress><!--dependent, xs:string--></ipAddress>
  <ipv6Address><!--dependent, xs:string--></ipv6Address>
  <portNo><!--optional, xs:integer--></portNo>
  <userName><!--dependent, xs:string--></userName>
  <password><!--dependent, xs:string--></password>
  <httpAuthenticationMethod><!--required, xs:string, "MD5digest,none"--></
httpAuthenticationMethod>
  <ANPR><!--optional-->
    <detectionUploadPicturesType>
      <!--optional, xs:string, types of alarm picture to be uploaded: "all",
"licensePlatePicture", "detectionPicture". When configuring the types of
captured pictures to be uploaded to the HTTP listening server, the node
detectionUploadPicturesType supported to be configured with the following
values: "licensePlatePicture", "detectionPicture", and "all"-->
      <!--The picture type configured for this node must be included in that
configured for the node capturePicType in XML_PicParam. For example, if the
values of the node capturePicType are "licensePlatePicture", "vehiclePicture",
and "detectionPicture", the value of this node can be any; if the value of the
node capturePicType is "licensePlatePicture", the value of this node can only
be "licensePlatePicture"; if the values of the node capturePicType are
"licensePlatePicture" and "vehiclePicture", the value of this node can be
"licensePlatePicture" or "all"; if the values of the node capturePicType are
"detectionPicture" and "vehiclePicture", the value of this node can be
"detectionPicture" or "all"-->
    </detectionUploadPicturesType>
  </ANPR>
  <eventType optional="AID,TFS,TPS"><!--required, xs:string--></eventType>
  <uploadImagesDataType>
    <!--optional, xs:string, "URL", "binary" (default), for cloud storage, only
"URL" is supported-->
  </uploadImagesDataType>
  <httpBroken>
    <!--optional, xs:boolean, automatic network replenishment (ANR). The ANR
function will be applied to all events once enabled-->true
  </httpBroken>

```

```
<eventMode><!--optional, xs:string, "all,list"--></eventMode>
<EventList><!--dependent, it is valid only when eventMode is "list"-->
  <Event><!--required-->
    <type><!--required, xs:string--></type>
  </Event>
</EventList>
<channels><!--optional, xs:string, "1,2,3,4..."--></channels>
<SubscribeEvent/><!--optional, event subscription parameters, see details in
the message of XML_SubscribeEvent-->
</HttpHostNotification>
```

E.40 XML_HttpHostNotificationCap

XML message about capability of HTTP listening server

```
<HttpHostNotificationCap version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <hostNumber>2</hostNumber>
  <urlLen max=""/>
  <protocolType opt="HTTP,HTTPS,EHome"/>
  <parameterFormatType opt="XML,querystring,JSON"/>
  <addressingFormatType opt="ipaddress,hostname"/>
  <ipAddress opt="ipv4,ipv6"/>
  <portNo min="" max=""/>
  <userNameLen min="" max=""/>
  <passwordLen min="" max=""/>
  <httpAuthenticationMethod opt="MD5digest,none"/>
  <Extensions>
    <intervalBetweenEvents min="" max=""/>
  </Extensions>
  <uploadImagesDataType opt="URL,binary"/>
  <ANPR><!--optional-->
    <detectionUpLoadPicturesType
opt="all,licensePlatePicture,detectionPicture..."/><!--optional, xs:string, types
of alarm pictures to be uploaded-->
    <alarmHttpPushProtocol opt="baseline,custom"/>
  </ANPR>
  <httpBroken opt="true,false" def="true"><!--optional, xs:boolean, whether to
enable global ANR: true, false--></httpBroken>
  <SubscribeEventCap>
    <heartbeat min="" max=""/><!--optional, heartbeat time interval, unit:
second-->
    <channelMode opt="all,list"/><!--required, all-subscribe events of all
channels, list-subscribe event by channel-->
    <eventMode opt="all,list"/><!--required, event subscription mode: all-
subscribe all events of all channels, list-subscribe events by type, channel,
and target-->
    <!--if the values of the two nodes channelMode and eventMode are both
"all", it indicates that the device does not support subscribe events by type
and channel-->
```

```
<EventList><!--dependent, alarm uploading mode, this node is valid only
when eventMode is "list"-->
  <Event><!--required-->
    <type><!--required, xs:string, event types--></type>
    <pictureURLType opt="binary,localURL,cloudStorageURL" def=""/>
    <!--optional, xs:string, transmission format of alarm picture: "binary"-
picture binary data, "localURL"-picture URL from local device,
"cloudStorageURL"-picture URL from cloud storage-->
  </Event>
</EventList>
<pictureURLType opt="binary,localURL,cloudStorageURL" def=""/>
<!--optional, xs:string, transmission format of all alarm pictures:
"binary"-picture binary data (default for camera), "localURL"-picture URL from
local device (default for NVR/DVR), "cloudStorageURL"-picture URL from cloud
storage; this node is in highest priority-->
  <ChangedUploadSub>
    <interval/><!--optional, xs:integer, the life cycle of arming GUID, unit:
second, the default life cycle is 5 minutes; if the reconnection is not started
during the life cycle, a new GUID will be generated-->
    <StatusSub>
      <all/><!--optional, xs:boolean, whether to subscribe all-->
      <channel/><!--optional, xs:boolean, subscribe channel status, this node is
not required when the node all is "true"-->
      <hd/><!--optional, xs:boolean, subscribe the HDD status, this node is not
required when the node all is "true"-->
      <capability/><!--optional, xs:boolean, subscribe the capability changed
status, this node is not required when the node all is "true"-->
    </StatusSub>
  </ChangedUploadSub>
</SubscribeEventCap>
</HttpHostNotificationCap>
```

E.41 XML_HttpHostNotificationList

HttpHostNotificationList message in XML format

```
<HttpHostNotificationList version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <HttpHostNotification>
    <id><!--req, xs:string, ID--></id>
    <url><!--req, xs:string--></url>
    <protocolType><!--req, xs:string, "HTTP,HTTPS,EHome". When the value of
this node is EHome, this message can be used to configure the information (such
as IP address and port No.) for the AMS (Alarm Management Server) of ISUP 5.0
to listen on. For communication via ISUP 5.0 (MQTT), the listening parameters
can be configured and obtained; For other communication methods (such as
communication via HTTP, HTTPS, EZVIZ Open Platform, etc.), the listening
parameters can only be obtained and cannot be configured--></protocolType>
    <parameterFormatType><!--req, xs:string, alarm/event information format,
"XML,JSON"--></parameterFormatType>
```



```
<addressingFormatType><!--req, xs:string, "ipaddress,hostname"--></addressingFormatType>
<hostName><!--dep, xs:string--></hostName>
<ipAddress><!--dep, xs:string--></ipAddress>
<ipv6Address><!--dep, xs:string--></ipv6Address>
<portNo><!--opt, xs:integer--></portNo>
<userName><!--dep, xs:string--></userName>
<password><!--dep, xs:string--></password>
<httpAuthenticationMethod><!--req, xs:string, "MD5digest,none"--></httpAuthenticationMethod>
<uploadImagesDataType>
  <!--opt, xs:string, "URL", "binary" (default), for cloud storage, only "URL" is supported-->
</uploadImagesDataType>
<eventMode><!--opt, xs:string, "all,list"--></eventMode>
<EventList><!--dep, it is valid only when eventMode is "list"-->
  <Event><!--req-->
    <type><!--req, xs:string--></type>
  </Event>
</EventList>
<channels><!--opt, xs:string, "1,2,3,4..."--></channels>
</HttpHostNotification>
</HttpHostNotificationList>
```

Example

HttpHostNotificationList Message Example

```
<HttpHostNotificationList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <HttpHostNotification>
    <id>1</id>
    <url></url>
    <protocolType>HTTP</protocolType>
    <parameterFormatType>XML</parameterFormatType>
    <addressingFormatType>ipaddress</addressingFormatType>
    <ipAddress>0.0.0.0</ipAddress>
    <portNo>80</portNo>
    <userName></userName>
    <httpAuthenticationMethod>none</httpAuthenticationMethod>
  </HttpHostNotification>
</HttpHostNotificationList>
```

E.42 XML_HttpHostTestResult

HttpHostTestResult message in XML format.

```
<HttpHostTestResult version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <errorDescription>
    <!--req, xs:string-->
```

```
</errorDescription>
</HttpHostTestResult>
```

E.43 XML_IntelliCap

XML message about intelligent capability

```
<IntelliCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <isFaceSupport><!--optional, xs:boolean, whether it supports face detection-->
</isFaceSupport>
  <isBehaviorSupport><!--optional, xs:boolean, whether it supports behavior
analysis--></isBehaviorSupport>
  <isLineDetectionSupport><!--optional, xs:boolean, whether it supports line
crossing detection--></isLineDetectionSupport>
  <isFieldDetectionSupport><!--optional, xs:boolean, whether it supports
intrusion detection--></isFieldDetectionSupport>
  <isRegionEntranceSupport><!--optional, xs:boolean, whether it supports
region entrance detection--></isRegionEntranceSupport>
  <isRegionExitingSupport><!--optional, xs:boolean, whether it supports region
exiting detection--></isRegionExitingSupport>
  <isLoiteringSupport><!--optional, xs:boolean, whether it supports loitering
detection--></isLoiteringSupport>
  <isGroupSupport><!--optional, xs:boolean, whether it supports people
gathering detection--></isGroupSupport>
  <isRapidMoveSupport><!--optional, xs:boolean, whether it supports fast
moving detection--></isRapidMoveSupport>
  <isParkingSupport><!--optional, xs:boolean, whether it supports parking
detection--></isParkingSupport>
  <isUnattendedBaggageSupport><!--optional, xs:boolean, whether it supports
unattended baggage detection--></isUnattendedBaggageSupport>
  <isAttendedBaggageSupport><!--optional, xs:boolean, whether it supports
object removal detection--></isAttendedBaggageSupport>
  <isTeacherSupport><!--optional, xs:boolean, whether it supports teacher's
behavior detection--></isTeacherSupport>
  <isStudentSupport><!--optional, xs:boolean, whether it supports student's
behavior detection--></isStudentSupport>
  <isCombinedSupport><!--optional, xs:boolean, whether it supports combined
detection--></isCombinedSupport>
  <isTrafficSupport><!--optional, xs:boolean, whether it supports traffic
detection--></isTrafficSupport>
  <RestoreLib>
    <libName opt="studentsStoodUp,peopleCounting,shipDetection"><!--optional,
xs:string,--></libName>
  </RestoreLib>
  <RestoreDefParamForbid>
    <!--optional, xs:boolean "true, false", forbid restoring algorithm library
to default. When the node does not exist, it indicates restoring to default is
not supported. When the node exists and its value is "ture", it indicates
support, when the value is "false", it indicates not support-->
  </RestoreDefParamForbid>
```

```
<RestoreAlgLibParam>
  <!--optional, the capabilities of restoring algorithm library parameters
  according to algorithm library name-->
  <libName opt="faceSnap,HMS,behaviorAnalysis"><!--optional, xs:string--></
  libName>
</RestoreAlgLibParam>
<isFaceCaptureStatisticsSupport><!--whether it supports face picture
statistics--></isFaceCaptureStatisticsSupport>
<isSupportPersonQueueDetection><!--whether it supports queue management--></
isSupportPersonQueueDetection>
<isSupportIntersectionAnalysis>
  <!--optional, xs:boolean, whether it supports intersection analysis-->
</isSupportIntersectionAnalysis>
<isSupportCityManagement><!--optional, xs:boolean, whether it supports
intelligent city management. Refer to URI: /ISAPI/Intelligent/channels/<ID>/
cityManagement/capabilities?format=json--></isSupportCityManagement>
<mixedTargetDetectionWithoutAttribute><!--optional, xs:boolean--></
mixedTargetDetectionWithoutAttribute>
<isSupportUploadFacePictureByForm><!--optional, xs:boolean, whether it
supports uploading face pictures by form--></isSupportUploadFacePictureByForm>
<isSupportUploadFacePictureByUrl><!--optional, xs:boolean, whether it
supports uploading face pictures by URL--></isSupportUploadFacePictureByUrl>
<isSupportUploadHumanPictureByForm><!--optional, xs:boolean, whether it
supports uploading human pictures in form--></isSupportUploadHumanPictureByForm>
<isSupportMisinfoFilterStatisticalMode><!--optional, xs:boolean, whether it
supports statistics of filtering face comparison--></
isSupportMisinfoFilterStatisticalMode>
<isSupportFaceScore><!--optional, xs:boolean, whether it supports face
grading configuration (camera)--></isSupportFaceScore>
<HumanRecognitionModeSearchCap>
  <searchTargetsNumMax><!--optional, xs:integer, maximum number of sample
  pictures that can be imported for searching by picture--></searchTargetsNumMax>
  <HumanMode>
    <searchCond opt="age_group,gender,jacet_color,glass,bag,ride,unlimit"/
  ><!--setting multiple search conditions is supported-->
    <similarity min="0.0" max="100.0"/><!--optional, xs:float, similarity,
  range: [0.0,100.0]-->
  </HumanMode>
  <HumanInfo><!--optional, person attribute information, it is required when
  searching for picture by attributes-->
    <age_group
  opt="child,infant,kid,middle,middleAged,old,prime,teenager,unknown,young">
    <!--optional, xs:string, age group:
  "child,infant,kid,middle,middleAged,old,prime,teenager,unknown,young"-->
    </age_group>
    <gender opt="female,male,unknown"><!--optional, xs:string, gender:
  "female,male,unknown"--></gender>
    <jacet_color
  opt="black,black,blue,brown,cyan,deepBlue,deepGray,gray,green,mixture,orange,pin
  k,purple,red,silver,unknown,white,yellow">
    <!--optional, xs:string, tops color-->
    </jacet_color>
```

```
<glass opt="yes,no,unknown,sunglasses"><!--optional, xs:string, whether
the person is wearing glasses--></glass>
<bag opt="yes,no,unknown"><!--optional, xs:string, whether the person is
carrying a bag--></bag>
<ride opt="yes,no,unknown"><!--optional, xs:string, whether the person is
riding--></ride>
</HumanInfo>
<positive opt="true,false"/><!--optional, xs:boolean, whether it is false
human body recognition alarm: "true"-yes, "false"-no-->
<eventType opt="unlimit,humanRecognition">
  <!--optional, xs:string, event type: "unlimit"-no limit,
"humanRecognition"-human body detection alarm-->
</eventType>
<isSupportMultiChannelSearch>
  <!--optional, xs:boolean, whether it supports multi-channel search-->
</isSupportMultiChannelSearch>
<isSupportTotalSearchResult>
  <!--optional, xs:boolean, whether it supports limiting number of results
that can be obtained after a single search-->
</isSupportTotalSearchResult>
</HumanRecognitionModeSearchCap>
<VehicleRecognitionModeSearchCap>
  <searchTargetsNumMax><!--optional, xs:integer, maximum number of sample
pictures that can be imported for searching by picture--></searchTargetsNumMax>
  <eventType opt="unlimit,vehicleBlackList,vehicleWhiteList"/><!--xs:string,
event type: "unlimit"-no limit, "vehicleBlackList"-vehicle blacklist,
"vehicleWhiteList"-vehicle allowlist-->
  <VehicleMode>
    <searchCond
opt="licensePlate,vehicleLogo,vehicleSubLogoRecog,vehicleType,vehicleColor,unlim
it"/>
    <similarity min="0.0" max="100.0"/><!--optional, xs:float, similarity,
range: [0.0,100.0]-->
  </VehicleMode>
  <isSupportMultiChannelSearch>
    <!--optional, xs:boolean, whether it supports multi-channel search-->
  </isSupportMultiChannelSearch>
  <isSupportTotalSearchResult>
    <!--optional, xs:boolean, whether it supports limiting number of results
that can be obtained after a single search-->
  </isSupportTotalSearchResult>
</VehicleRecognitionModeSearchCap>
<FaceContrastAnalyzeCap><!--capability of manual comparison and analysis of
face pictures-->
  <isSupportTrigger opt="false,true"><!--optional, xs:boolean--></
isSupportTrigger>
  <SearchCondition><!--optional-->
    <startTime><!--required, xs:time, ISO8601 time--></startTime>
    <endTime><!--required, xs:time, ISO8601 time--></endTime>
  </SearchCondition>
</FaceContrastAnalyzeCap>
<FramesPeopleCountingCap><!--capability of people counting statistics in a
```

```
single frame-->
  <Statistics>
    <SearchCondition>
      <startTime><!--required, xs:time,ISO8601 time--></startTime>
      <endTime><!--required, xs:time,ISO8601 time--></endTime>
    </SearchCondition>
  </Statistics>
  <MaskRegion><!--optional, shielded region-->
    <maxRegionNum><!--optional, xs:integer, number of regions--></
maxRegionNum>
    <Region>
      <vertexNum min="3" max="10"><!--optional, xs:integer, number of region
vertexes--></vertexNum>
    </Region>
  </MaskRegion>
</FramesPeopleCountingCap>
<FaceContrastPersonInfoExtend><!--configuration capability of face comparison
tag-->
  <personInfoCap>
    <maxPersonInfo min="0" max="4">
      <!--optional, xs:integer, maximum number of person tags-->
    </maxPersonInfo>
  </personInfoCap>
  <personInfoFDlibCap>
    <maxPersonInfo min="0" max="4">
      <!--optional, xs:integer, maximum number of person tags-->
    </maxPersonInfo>
  </personInfoFDlibCap>
</FaceContrastPersonInfoExtend>
<isSupportSafetyHelmetDetection>
  <!-- opt, xs:boolean, whether it supports hard hat detection-->
</isSupportSafetyHelmetDetection>
<isSupportDistanceRecognition><!--optional, xs:boolean, whether supports
distance measurement--></isSupportDistanceRecognition>
<isSupportATM><!--optional, xs:boolean, ATM intelligent configuraiton,
corresponds to URI: /ISAPI/Intelligent/ATM/capabilities?format=json--></
isSupportATM>
<isSupportFaceTemperature><!--optional, xs:boolean, whether the device
supports face thermography, corresponds to URI: /ISAPI/Intelligent/
faceTemperature/capabilities?format=json--></isSupportFaceTemperature>
<isSupportFireEscapeDetectionSearch>
  <!--optional, xs:boolean, whether the device supports searching for fire
engine access detection events of multiple channels (related URI: URI/ISAPI/
Intelligent/fireEscapeDetection/search/capabilities?format=json)-->
</isSupportFireEscapeDetectionSearch>
<isSupportTakingElevatorDetectionSearch>
  <!--optional, xs:boolean, whether the device supports searching for
elevator detection events of multiple channels (related URI: /ISAPI/Intelligent/
takingElevatorDetection/search/capabilities?format=json)-->
</isSupportTakingElevatorDetectionSearch>
<isSupportCityManagementSearch>
  <!--optional, xs:boolean, whether the device supports searching for city
```

```
management events of multiple channels (related URI: /ISAPI/Intelligent/
cityManagement/search/capabilities?format=json)-->
</isSupportCityManagementSearch>
<isSupportSafetyHelmetDetectionSearch>
  <!--optional, xs:boolean, whether the device supports searching for hard
hat detection events of multiple channels (related URI: /ISAPI/Intelligent/
safetyHelmetDetection/search/capabilities?format=json)-->
</isSupportSafetyHelmetDetectionSearch>
<isSupportFramesPeopleCountingSearch>
  <!--optional, boolean, ro, whether it supports searching for the regional
people counting event (related URI: /ISAPI/Intelligent/framesPeopleCounting/
search/capabilities?format=json)-->true
</isSupportFramesPeopleCountingSearch>
<isSupportChannelPollingSchedules>
  <!--optional, xs:boolean, whether the device supports channel polling
configuration: "true,false"-->
</isSupportChannelPollingSchedules>
<AnalysisImageFaceCap>
  <!--optional, face picture analysis type (related URI:/ISAPI/Intelligent/
analysisImage/face)-->
  <modeling opt="false"><!--optional,xs:boolean, whether the device
supports modeling: "true,false"--></modeling>
</AnalysisImageFaceCap>
</IntelliCap>
```

E.44 XML_Schedule

Schedule message in XML format

```
<Schedule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:string, ID--></id>
  <eventType>
    <!--optional, xs:string, alarm/event types, see details in the "Remarks"
below-->
  </eventType>
  <inputIOPortID><!--read-only, dependent, xs:string, alarm input No.--></
inputIOPortID>
  <outputIOPortID><!--read-only, dependent, xs:string, alarm output No.--></
inputIOPortID>
  <videoInputChannelID><!--read-only, dependent, xs:string, video input channel
ID--></videoInputChannelID>
  <TimeBlockList size="8"><!--required-->
    <TimeBlock><!--list-->
      <dayOfWeek>
        <!--optional, xs:integer, day of the week based on ISO8601,
"1"=Monday, ...-->
      </dayOfWeek>
      <TimeRange><!--required-->
        <beginTime><!--required, xs:time, ISO 8601 time--></beginTime>
        <endTime><!--required, xs:time, ISO 8601 time--></endTime>
```

```
</TimeRange>
<CustomExtension>
  <vehicleDetectSceneID>
    <!--required, xs:interger-->
  </vehicleDetectSceneID>
</CustomExtension>
</TimeBlock>
</TimeBlockList>
<HolidayBlockList><!--optional-->
  <TimeBlock><!--list-->
    <TimeRange><!--required-->
      <beginTime><!--required, xs:time, ISO 8601 time--></beginTime>
      <endTime><!--required, xs:time, ISO 8601 time--></endTime>
    </TimeRange>
  </TimeBlock>
</HolidayBlockList>
</Schedule>
```

Remarks

The node **<eventType>** can be set to the following values: IO, VMD, videoloss, PIR, linedetection, fielddetection, audioexception, facedetection, regionEntrance, regionExiting, loitering, group, rapidMove, parking, unattendedBaggage, attendedBaggage, storageDetection, shipsDetection, HUMANATTRIBUTE, humanAttribute, faceContrast, faceSnap, faceLib, whiteListFaceContrast, personDensityDetection, personQueueDetection, mixedTargetDetection, fireDetection, illegalParking, pedestrian, trafficAccident, construction, roadblock, abandonedObject, parallelParking, parkingState, congestion, intersectionAnalysis, heatMap, reverseEntrance, vehicledetect, safetyHelmetDetection, vibrationDetection, TMPA, faceThermometry, HBDLib, detectorTemp, detectorSmoke, detectorTamper, smokeFireRecognizesmokeFireRecognize, indoorPasswayBlock, detectorShelter, detectorMotion, fireNoRegulation, peopleDetections.

E.45 XML_ResponseStatus

XML message about response status

```
<?xml version="1.0" encoding="utf-8"?>
<ResponseStatus version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <requestURL>
    <!--required, read-only, xs:string, request URL-->
  </requestURL>
  <statusCode>
    <!--required, read-only, xs:integer, status code: 0,1-OK, 2-Device Busy, 3-Device Error, 4-Invalid Operation, 5-Invalid XML Format, 6-Invalid XML Content, 7-Reboot Required, 9-Additional Error-->
  </statusCode>
  <statusString>
    <!--required, read-only, xs:string, status description: OK, Device Busy, Device Error, Invalid Operation, Invalid XML Format, Invalid XML Content, Reboot, Additional Error-->
```

```
</statusString>
<subStatusCode>
  <!--required, read-only, xs:string, describe the error reason in detail-->
</subStatusCode>
<MErrCode>
  <!--optional, xs:string, error code categorized by functional modules,
e.g., 0x12345678-->
</MErrCode>
<MErrDevSelfEx>
  <!--optional, xs:string, extension field of MErrCode. It is used to define
the custom error code, which is categorized by functional modules-->
</MErrDevSelfEx>
</ResponseStatus>
```


Appendix F. Response Codes of Text Protocol

The response codes returned during the text protocol integration is based on the status codes of HTTP. 7 kinds of status codes are predefined, including 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid Message Format), 6 (Invalid Message Content), and 7 (Reboot Required). Each kind of status code contains multiple sub status codes, and the response codes are in a one-to-one correspondence with the sub status codes.

StatusCode=1

SubStatusCode	Error Code	Description
ok	0x1	Operation completed.
riskPassword	0x10000002	Risky password.
armProcess	0x10000005	Arming process.

StatusCode=2

Sub Status Code	Error Code	Description
noMemory	0x20000001	Insufficient memory.
serviceUnavailable	0x20000002	The service is not available.
upgrading	0x20000003	Upgrading.
deviceBusy	0x20000004	The device is busy or no response.
reConnectIpc	0x20000005	The video server is reconnected.
transferUpgradePackageFailed	0x20000006	Transmitting device upgrade data failed.
startUpgradeFailed	0x20000007	Starting upgrading device failed.
getUpgradeProcessfailed.	0x20000008	Getting upgrade status failed.
certificateExist	0x2000000B	The Authentication certificate already exists.

StatusCode=3

Sub Status Code	Error Code	Description
deviceError	0x30000001	Hardware error.
badFlash	0x30000002	Flash operation error.
28181Uninitialized	0x30000003	The 28181 configuration is not initialized.
socketConnectError	0x30000005	Connecting to socket failed.
receiveError	0x30000007	Receive response message failed.
deletePictureError	0x3000000A	Deleting picture failed.
pictureSizeExceedLimit	0x3000000C	Too large picture size.
clearCacheError	0x3000000D	Clearing cache failed.
updateDatabasError	0x3000000F	Updating database failed.
searchDatabaseError	0x30000010	Searching in the database failed.
writeDatabaseError	0x30000011	Writing to database failed.
deleteDatabaseError	0x30000012	Deleting database element failed.
searchDatabaseElementError	0x30000013	Getting number of database elements failed.
cloudAutoUpgradeException	0x30000016	Downloading upgrade packet from cloud and upgrading failed.
HBPException	0x30001000	HBP exception.
UDEPException	0x30001001	UDEP exception
elasticSearchException	0x30001002	Elastic exception.
kafkaException	0x30001003	Kafka exception.
HBaseException	0x30001004	Hbase exception.
sparkException	0x30001005	Spark exception.
yarnException	0x30001006	Yarn exception.
cacheException	0x30001007	Cache exception.

Sub Status Code	Error Code	Description
trafficException	0x30001008	Monitoring point big data server exception.
faceException	0x30001009	Human face big data server exception.
SSDFileSystemsIsError	0x30001013	SSD file system error (Error occurs when it is non-Ext4 file system)
insufficientSSDCapacityForFPD	0x30001014	Insufficient SSD space for person frequency detection.
wifiException	0x3000100A	Wi-Fi big data server exception
structException	0x3000100D	Video parameters structure server exception.
noLinkageResource	0x30001015	Insufficient linkage resources.
engineAbnormal	0x30002015	Engine exception.
engineInitialization	0x30002016	Initializing the engine.
algorithmLoadingFailed	0x30002017	Loading the model failed.
algorithmDownloadFailed	0x30002018	Downloading the model failed.
algorithmDecryptionFailed	0x30002019	Decrypting the model failed.
unboundChannel	0x30002020	Delete the linked channel to load the new model.
unsupportedResolution	0x30002021	Invalid resolution.
unsupportedStreamType	0x30002022	Invalid stream type.
insufficientDecRes	0x30002023	Insufficient decoding resources.
insufficientEnginePerformance	0x30002024	Insufficient engine performance (The number of channels to be analyzed exceeds the engine's capability).
improperResolution	0x30002025	Improper resolution (The maximum resolution allowed is 4096×4096).

Sub Status Code	Error Code	Description
improperPicSize	0x30002026	Improper picture size (The maximum size allowed is 5MB).
URLDownloadFailed	0x30002027	Downloading the picture via the URI failed.
unsupportedImageFormat	0x30002028	Invalid picture format (Only JPG is supported currently).
unsupportedPollingIntervalTime	0x30002029	Invalid polling interval (The interval should be more than 10s).
exceedImagesNumber	0x30002030	The number of pictures exceeds the limit (The platform can apply 1 to 100 picture URIs per time, the maximum number allowed is 100).
unsupportedMPID	0x30002031	The applied MPID does not exist in the device, so updating this MPID is not supported.
modelPackageNotMatchLabel	0x30002032	The model and the description file mismatch.
modelPackageNotMatchTask	0x30002033	The task and the model type mismatch.
insufficientSpace	0x30002034	Insufficient space (When the number of model packages does not reach the maximum number allowed but their size together exceeds the free space, the model packages cannot be added).
engineUnLoadingModelPackage	0x30002035	Applying the task failed. This engine is not linked to a model package (Canceling the linkage failed, this engine is not linked to a model package).
engineWithModelPackage	0x30002036	Linking the engine to this model package failed. The engine has been linked to

Sub Status Code	Error Code	Description
		another model package. Please cancel their linkage first.
modelPackageDelete	0x30002037	Linking the model package failed. The model package has been deleted.
deleteTaskFailed	0x30002038	Deleting the task failed (It is returned when the user fails to end a task).
modelPackageNumberslimited	0x30002039	Adding the model package failed. The number of model package has reached the maximum number allowed.
modelPackageDeleteFailed	0x30002040	Deleting the model package failed.
noArmingResource	0x30001016	Insufficient arming resources.
calibrationTimeout	0x30002051	Calibration timed out.
captureTimeout	0x30006000	Data collection timed out.
lowScore	0x30006001	Low quality of collected data.
uploadingFailed	0x30007004	Uploading failed.

StatusCode=4

Sub Status Code	Error Code	Description
notSupport	0x40000001	Not supported.
lowPrivilege	0x40000002	No permission.
badAuthorization	0x40000003	Authentication failed.
methodNotAllowed	0x40000004	Invalid HTTP method.
notSetHdiskRedund	0x40000005	Setting spare HDD failed.
invalidOperation	0x40000006	Invalid operation.
notActivated	0x40000007	Inactivated.
hasActivated	0x40000008	Activated.
certificateAlreadyExist	0x40000009	The certificate already exists.

Sub Status Code	Error Code	Description
operateFailed	0x4000000F	Operation failed.
USBNotExist	0x40000010	USB device is not connected.
upgradePackageMorethan2GB	0x40001000	Up to 2GB upgrade package is allowed to be uploaded.
IDNotExist	0x40001001	The ID does not exist.
interfaceOperationError	0x40001002	API operation failed.
synchronizationError	0x40001003	Synchronization failed.
synchronizing	0x40001004	Synchronizing.
importError	0x40001005	Importing failed.
importing	0x40001006	Importing.
fileAlreadyExists	0x40001007	The file already exists.
invalidID	0x40001008	Invalid ID.
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.
exportItemsExceedLimit	0x40001010	No more items can be exported.
noFiles	0x40001011	The file does not exist.
beingExportedByAnotherUser	0x40001012	Being exported by others.
needReAuthentication	0x40001013	Authentication is needed after upgrade.
unitAddNotOnline	0x40001015	The added data analysis server is offline.
unitControl	0x40001016	The data analysis server is already added.

Sub Status Code	Error Code	Description
analysis unitFull	0x40001017	No more data analysis server can be added.
unitIDError	0x40001018	The data analysis server ID does not exist.
unitExit	0x40001019	The data analysis server already exists in the list.
unitSearch	0x4000101A	Searching data analysis server in the list failed.
unitNotOnline	0x4000101B	The data analysis server is offline.
unitInfoError	0x4000101C	Getting data analysis server information failed.
unitGetNodeInfoError	0x4000101D	Getting node information failed.
unitGetNetworkInfoError	0x4000101E	Getting the network information of data analysis server failed
unitSetNetworkInfoError	0x4000101F	Setting the network information of data analysis server failed
setSmartNodeInfoError	0x40001020	Setting node information failed.
setUnitNetworkInfoError	0x40001021	Setting data analysis server network information failed.
unitRestartCloseError	0x40001022	Rebooting or shutting down data analysis server failed.
virtualIPnotAllowed	0x40001023	Adding virtual IP address is not allowed.
unitInstalled	0x40001024	The data analysis server is already installed.
badSubnetMask	0x40001025	Invalid subnet mask.
uintVersionMismatched	0x40001026	Data analysis server version mismatches.
deviceModelMismatched	0x40001027	Adding failed. Device model mismatches.
unitAddNotSelf	0x40001028	Adding peripherals is not allowed.
noValidUnit	0x40001029	No valid data analysis server.
unitNameDuplicate	0x4000102A	Duplicated data analysis server name.
deleteUnitFirst	0x4000102B	Delete the added data analysis server of the node first.
getLocalInfoFailed	0x4000102C	Getting the server information failed.

Sub Status Code	Error Code	Description
getClientAddedNodeFailed	0x4000102D	Getting the added node information of data analysis server failed.
taskExit	0x4000102E	The task already exists.
taskInitError	0x4000102F	Initializing task failed.
taskSubmitError	0x40001030	Submitting task failed.
taskDelError	0x40001031	Deleting task failed.
taskPauseError	0x40001032	Pausing task failed.
taskContinueError	0x40001033	Starting task failed.
taskSeverNoCfg	0x40001035	Full-text search server is not configured.
taskPicSeverNoCfg	0x40001036	The picture server is not configured.
taskStreamError	0x40001037	Streaming information exception.
taskRecSDK	0x40001038	History recording is not supported.
taskCasaError	0x4000103A	Cascading is not supported.
taskVCARuleError	0x4000103B	Invalid VCA rule.
taskNoRun	0x4000103C	The task is not executed.
unitLinksNoStorageNode	0x4000103D	No node is linked with the data analysis server. Configure the node first.
searchFailed	0x4000103E	Searching video files failed.
searchNull	0x4000103F	No video clip.
userScheOffline	0x40001040	The task scheduler service is offline.
updateTypeUnmatched	0x40001041	The upgrade package type mismatches.
userExist	0x40001043	The user already exists.
userCannotDelAdmin	0x40001044	The administrator cannot be deleted.
userInexistence	0x40001045	The user name does not exist.
userCannotCreatAdmin	0x40001046	The administrator cannot be created.
monitorCamExceed	0x40001048	Up to 3000 cameras can be added.

Sub Status Code	Error Code	Description
monitorCunitOverLimit	0x40001049	Adding failed. Up to 5 lower-levels are supported by the control center.
monitorReginOverLimit	0x4000104A	Adding failed. Up to 5 lower-levels are supported by the area.
monitorArming	0x4000104B	The camera is already armed. Disarm the camera and try again.
monitorSyncCfgNotSet	0x4000104C	The system parameters are not configured.
monitorFdSyncing	0x4000104E	Synchronizing. Try again after completing the synchronization.
monitorParseFailed	0x4000104F	Parsing camera information failed.
monitorCreatRootFailed	0x40001050	Creating resource node failed.
deleteArmingInfo	0x40001051	The camera is already . Disarm the camera and try again.
cannotModify	0x40001052	Editing is not allowed. Select again.
cannotDel	0x40001053	Deletion is not allowed. Select again.
deviceExist	0x40001054	The device already exists.
IPErrorConnectFailed	0x40001056	Connection failed. Check the network port.
cannotAdd	0x40001057	Only the capture cameras can be added.
serverExist	0x40001058	The server already exists.
fullTextParamError	0x40001059	Incorrect full-text search parameters.
storParamError	0x4000105A	Incorrect storage server parameters.
picServerFull	0x4000105B	The storage space of picture storage server is full.
NTPUnconnect	0x4000105C	Connecting to NTP server failed. Check the parameters.
storSerConnectFailed	0x4000105D	Connecting to storage server failed. Check the network port.
storSerLoginFailed	0x4000105E	Logging in to storage server failed. Check the user name and password.

Sub Status Code	Error Code	Description
searchSerConnectFailed	0x4000105F	Connecting to full-text search server failed. Check the network port.
searchSerLoginFailed	0x40001060	Logging in to full-text search server failed. Check the user name and password.
kafkaConnectFailed	0x40001061	Connecting to Kafka failed. Check the network port.
mgmtConnectFailed	0x40001062	Connecting to system failed. Check the network port.
mgmtLoginFailed	0x40001063	Logging in to system failed. Check the user name and password.
TDAConnectFailed	0x40001064	Connecting to traffic data access server failed. Checking the server status.
86sdkConnectFailed	0x40001065	Connecting to listening port of iVMS-8600 System failed. Check the parameters.
nameExist	0x40001066	Duplicated server name.
batchProcessFailed	0x40001067	Processing in batch failed.
IDNotExist	0x40001068	The server ID does not exist.
serviceNumberReachesLimit	0x40001069	No more service can be added.
invalidServiceType.	0x4000106A	Invalid service type.
clusterGetInfo	0x4000106B	Getting cluster group information failed.
clusterDelNode	0x4000106C	Deletion node failed.
clusterAddNode	0x4000106D	Adding node failed.
clusterInstalling	0x4000106E	Creating cluster...Do not operate.
clusterUninstall	0x4000106F	Reseting cluster...Do not operate.
clusterInstall	0x40001070	Creating cluster failed.
clusterIpError	0x40001071	Invalid IP address of task scheduler server.
clusterNotSameSeg	0x40001072	The main node and sub node must be in the same network segment.
clusterVirIpError	0x40001073	Automatically getting virtual IP address failed. Enter manually.

Sub Status Code	Error Code	Description
clusterNodeUnadd	0x40001074	The specified main (sub) node is not added.
clusterNodeOffline	0x40001075	The task scheduler server is offline.
nodeNotCurrentIP	0x40001076	The analysis node of the current IP address is required when adding main and sub nodes.
addNodeNetFailed	0x40001077	Adding node failed. The network disconnected.
needTwoMgmtNode	0x40001078	Two management nodes are required when adding main and sub nodes.
ipConflict	0x40001079	The virtual IP address and data analysis server's IP address conflicted.
ipUsed	0x4000107A	The virtual IP address has been occupied.
cloudAlalyseOnline	0x4000107B	The cloud analytic server is online.
virIP&mainIPnotSame NetSegment	0x4000107C	The virtual IP address is not in the same network segment with the IP address of main/sub node.
getNodeDispatchInfoFailed	0x4000107D	Getting node scheduler information failed.
unableModifyManagementNetworkIP	0x4000107E	Editing management network interface failed. The analysis board is in the cluster.
notSpecifyVirtualIP	0x4000107F	Virtual IP address should be specified for main and sub cluster.
armingFull	0x40001080	No more device can be armed.
armingNoFind	0x40001081	The arming information does not exist.
disArming	0x40001082	Disarming failed.
getArmingError	0x40001084	Getting arming information failed.
refreshArmingError	0x40001085	Refreshing arming information failed.
ArmingPlateSame	0x40001086	The license plate number is repeatedly armed.
ArmingParseXLSError	0x40001087	Parsing arming information file failed.
ArmingTimeError	0x40001088	Invalid arming time period.
ArmingSearchTimeError	0x40001089	Invalid search time period.

Sub Status Code	Error Code	Description
armingRelationshipReachesLimit	0x4000108A	No more relation can be created.
duplicateAarmingName	0x4000108B	The relation name already exists.
noMoreArmingListAdded	0x4000108C	No more blocklist library can be armed.
noMoreCamerasAdded	0x4000108D	No more camera can be armed.
noMoreArmingListAddedWithCamera	0x4000108E	No more library can be linked to the camera.
noMoreArmingPeriodAdded	0x4000108F	No more time period can be added to the arming schedule.
armingPeriodsOverlapped	0x40001090	The time periods in the arming schedule are overlapped.
noArmingAlarmInfo	0x40001091	The alarm information does not exist.
armingAlarmUnRead	0x40001092	Getting number of unread alarms failed.
getArmingAlarmError	0x40001093	Getting alarm information failed.
searchByPictureTimeout	0x40001094	Searching picture by picture timeout. Search again.
comparisonTimeRangeError	0x40001095	Comparison time period error.
selectMonitorNumberUpperLimit	0x40001096	No more monitoring point ID can be filtered.
noMoreComparisonTasksAdded	0x40001097	No more comparison task can be executed at the same time.
GetComparisonResultFailed	0x40001098	Getting comparison result failed.
comparisonTypeError	0x40001099	Comparison type error.
comparisonUnfinished	0x4000109A	The comparison is not completed.
facePictureModelInvalid	0x4000109B	Invalid face model.
duplicateLibraryName.	0x4000109C	The library name already exists.
noRecord	0x4000109D	No record found.

Sub Status Code	Error Code	Description
countingRecordsFailed.	0x4000109E	Calculate the number of records failed.
getHumanFaceFrameFailed	0x4000109F	Getting face thumbnail from the picture failed.
modelingFailed.	0x400010A0	Modeling face according to picture URL failed.
1V1FacePictureComparisonFailed	0x400010A1	Comparison 1 VS 1 face picture failed.
libraryArmed	0x400010A2	The blocklist library is armed.
licenseExceedLimit	0x400010A3	Dongle limited.
licenseExpired	0x400010A4	Dongle expired.
licenseDisabled	0x400010A5	Unavailable dongle.
licenseNotExist	0x400010A6	The dongle does not exist.
SessionExpired	0x400010A7	Session expired .
beyondConcurrentLimit	0x400010A8	Out of concurrent limit.
stopSync	0x400010A9	Synchronization stopped.
getProgressFailed	0x400010AA	Getting progress failed.
uploadExtraCaps	0x400010AB	No more files can be uploaded.
timeRangeError	0x400010AC	Time period error.
dataPortNotConnected	0x400010AD	The data port is not connected.
addClusterNodeFailed	0x400010AE	Adding to the cluster failed. The device is already added to other cluster.
taskNotExist	0x400010AF	The task does not exist.
taskQueryFailed	0x400010B0	Searching task failed.
modifyTimeRuleFailed	0x400010B2	The task already exists. Editing time rule is not allowed.
modifySmartRuleFailed	0x400010B3	The task already exists. Editing VAC rule is not allowed.
queryHistoryVideoFailed	0x400010B4	Searching history video failed.
addDeviceFailed	0x400010B5	Adding device failed.

Sub Status Code	Error Code	Description
addVideoFailed	0x400010B6	Adding video files failed.
deleteAllVideoFailed	0x400010B7	Deleting all video files failed.
createVideoIndexFailed	0x400010B8	Indexing video files failed.
videoCheckTypeFailed	0x400010B9	Verifying video files types failed.
configStructuredAddressFailed	0x400010BA	Configuring IP address of structured server failed.
configPictureServerAddressFailed	0x400010BB	Configuring IP address of picture stored server failed.
storageServiceIPNotExist	0x400010BD	The storage server IP address does not exist.
syncBackupDatabaseFailed	0x400010BE	Synchronizing sub database failed. Try again.
syncBackupNTPTimeFailed	0x400010BF	Synchronizing NTP time of sub server failed.
clusterNotSelectLoopbackAddress	0x400010C0	Loopback address is not supported by the main or sub cluster.
addFaceRecordFailed	0x400010C1	Adding face record failed.
deleteFaceRecordFailed	0x400010C2	Deleting face record failed.
modifyFaceRecordFailed	0x400010C3	Editing face record failed.
queryFaceRecordFailed	0x400010C4	Searching face record failed.
faceDetectFailed	0x400010C5	Detecting face failed.
libraryNotExist	0x400010C6	The library does not exist.
blackListQueryExporting	0x400010C7	Exporting matched blocklists.
blackListQueryExported	0x400010C8	The matched blocklists are exported.
blackListQueryStopExporting	0x400010C9	Exporting matched blocklists is stopped.
blackListAlarmQueryExporting	0x400010CA	Exporting matched blocklist alarms.

Sub Status Code	Error Code	Description
blackListAlarmQueryExported	0x400010CB	The matched blocklists alarms are exported.
blackListAlarmQueryStopExporting	0x400010CC	Exporting matched blocklist alarms is stopped.
getBigDataCloudAnalysisFailed	0x400010CD	Getting big data cloud analytic information failed.
setBigDataCloudAnalysisFailed	0x400010CE	Configuring big data cloud analytic failed.
submitMapSearchFailed	0x400010CF	Submitting search by picture task failed.
controlRelationshipNotExist	0x400010D0	The relation does not exist.
getHistoryAlarmInfoFailed	0x400010D1	Getting history alarm information failed.
getFlowReportFailed	0x400010D2	Getting people counting report failed.
addGuardFailed	0x400010D3	Adding arming configuration failed.
deleteGuardFailed	0x400010D4	Deleting arming configuration failed.
modifyGuardFailed	0x400010D5	Editing arming configuration failed.
queryGuardFailed	0x400010D6	Searching arming configurations failed.
uploadUserSuperCaps	0x400010D7	No more user information can be uploaded.
bigDataServerConnectFailed	0x400010D8	Connecting to big data server failed.
microVideoCloudRequestInfoBuildFailed	0x400010D9	Adding response information of micro video cloud failed.
microVideoCloudResponseInfoBuildFailed	0x400010DA	Parsing response information of micro video cloud failed.
transcodingServerRequestInfoBuildFailed	0x400010DB	Adding response information of transcoding server failed.
transcodingServerResponseInfoParseFailed	0x400010DC	Parsing response information of transcoding server failed.
transcodingServerOffline	0x400010DD	Transcoding server is offline.

Sub Status Code	Error Code	Description
microVideoCloudOffline	0x400010DE	Micro video cloud is offline.
UPSServerOffline	0x400010DF	UPS monitor server is offline.
statisticReportRequestInfoBuildFailed	0x400010E0	Adding response information of statistics report failed.
statisticReportResponseInfoParseFailed	0x400010E1	Parsing response information of statistics report failed.
DisplayConfigInfoBuildFailed	0x400010E2	Adding display configuration information failed.
DisplayConfigInfoParseFailed	0x400010E3	Parsing display configuration information failed.
DisplayConfigInfoSaveFailed	0x400010E4	Saving display configuration information failed.
notSupportDisplayConfigType	0x400010E5	The display configuration type is not supported.
passError	0x400010E7	Incorrect password.
upgradePackageLarge	0x400010EB	Too large upgrade package.
sessionUserReachesLimit	0x400010EC	No more user can log in via session.
ISO8601TimeFormatError	0x400010ED	Invalid ISO8601 time format.
clusterDissolutionFailed	0x400010EE	Deleting cluster failed.
getServiceNodeInfoFailed	0x400010EF	Getting service node information failed.
getUPSInfoFailed	0x400010F0	Getting UPS configuration information failed.
getDataStatisticsReportFailed	0x400010F1	Getting data statistic report failed.
getDisplayConfigInfoFailed	0x400010F2	Getting display configuration failed.
namingAnalysisBoardNotAllowed	0x400010F3	Renaming analysis board is not allowed.

Sub Status Code	Error Code	Description
onlyDrawRegionsOfConvexPolygon	0x400010F4	Only drawing convex polygon area is supported.
bigDataServerResponseInfoParseFailed	0x400010F5	Parsing response message of big data service failed.
bigDataServerReturnFailed	0x400010F6	No response is returned by big data service.
microVideoReturnFailed	0x400010F7	No response is returned by micro video cloud service.
transcodingServerReturnFailed	0x400010F8	No response is returned by transcoding service.
UPSServerReturnFailed	0x400010F9	No response is returned by UPS monitoring service.
forwardingServerReturnFailed	0x400010FA	No response is returned by forwarding service.
storageServerReturnFailed	0x400010FB	No response is returned by storage service.
cloudAnalysisServerReturnFailed	0x400010FC	No response is returned by cloud analytic service.
modelEmpty	0x400010FD	No model is obtained.
mainAndBackupNodeCannotModifyManagementNetworkInterfaceIP	0x400010FE	Editing the management interface IP address of main node and backup node is not allowed.
IDTooLong	0x400010FF	The ID is too long.
pictureCheckFailed	0x40001100	Detecting picture failed.
pictureModelingFailed	0x40001101	Modeling picture failed.
setCloudAnalysisDefaultProvinceFailed	0x40001102	Setting default province of cloud analytic service failed.
InspectionAreasNumberExceedLimit	0x40001103	No more detection regions can be added.
picturePixelsTooLarge	0x40001105	The picture resolution is too high.
picturePixelsTooSmall	0x40001106	The picture resolution is too low.
storageServiceIPEmpty	0x40001107	The storage server IP address is required.

Sub Status Code	Error Code	Description
bigDataServerRequestInfoBuildFail	0x40001108	Creating request message of big data service failed.
analysisTimedOut	0x40001109	Analysis time out.
high-performanceModeDisabled.	0x4000110A	Please enable high-performance mode.
configuringUPSMonitoringServerTimedOut	0x4000110B	Configuring the UPS monitoring server time out. Check IP address.
cloudAnalysisRequestInformationBuildFailed	0x4000110C	Creating request message of cloud analytic service failed.
cloudAnalysisResponseInformationParseFailed	0x4000110D	Parsing response message of cloud analytic service failed.
allCloudAnalysisInterfaceFailed	0x4000110E	Calling API for cloud analytic service failed.
cloudAnalysisModelCompareFailed	0x4000110F	Model comparison of cloud analytic service failed.
cloudAnalysisFacePictureQualityRatingFailed	0x40001110	Getting face quality grading of cloud analytic service failed.
cloudAnalysisExtractFeaturePointsFailed	0x40001111	Extracting feature of cloud analytic service failed.
cloudAnalysisExtractPropertyFailed	0x40001112	Extracting property of cloud analytic service failed.
getAddedNodeInformationFailed	0x40001113	Getting the added nodes information of data analysis server failed.
noMoreAnalysisUnitsAdded	0x40001114	No more data analysis servers can be added.
detectionAreaInvalid	0x40001115	Invalid detection region.
shieldAreaInvalid	0x40001116	Invalid shield region.
noMoreShieldAreasAdded	0x40001117	No more shield region can be drawn.
onlyAreaOfRectangleShapeAllowed	0x40001118	Only drawing rectangle is allowed in detection area.
numberReachedLimit	0x40001119	Number reached the limit.

Sub Status Code	Error Code	Description
wait1~3MinutesGetIPAfterSetupDHCP	0x4000111A	Wait 1 to 3 minutes to get IP address after configuring DHCP.
plannedTimeMustbeHalfAnHour	0x4000111B	Schedule must be half an hour.
oneDeviceCannotBuildCluster	0x4000111C	Creating main and backup cluster requires at least two devices.
updatePackageFileNotUploaded	0x4000111E	Upgrade package is not uploaded.
highPerformanceTasksNotSupportDrawingDetectionRegions	0x4000111F	Drawing detection area is not allowed under high-performance mode.
controlCenterIDDoesNotExist	0x40001120	The control center ID does not exist.
regionIDDoesNotExist	0x40001121	The area ID does not exist.
licensePlateFormatError	0x40001122	Invalid license plate format.
managementNodesNotSupportThisOperation	0x40001123	The operation is not supported.
searchByPictureResourceNotConfiged	0x40001124	The conditions for searching picture by picture are not configured.
videoFileEncapsulationFormatNotSupported	0x40001125	The video container format is not supported.
videoPackageFailure	0x40001126	Converting video container format failed.
videoCodingFormatNotSupported	0x40001127	Video coding format is not supported.
monitorOfDeviceArmingdeleteArmingInfo	0x40001129	The camera is armed. Disarm it and try again.
getVideoSourceTypeFailed	0x4000112A	Getting video source type failed.
smartRulesBuildFailed	0x4000112B	Creating VAC rule failed.
smartRulesParseFailed	0x4000112C	Parsing VAC rule failed.
timeRulesBuildFailed	0x4000112D	Creating time rule failed.

Sub Status Code	Error Code	Description
timeRulesParseFailed	0x4000112E	Parsing time rule failed.
monitoInfoInvalid	0x4000112F	Invalid camera information.
addingFailedVersionMismatches	0x40001130	Adding failed. The device version mismatches.
theInformationReturnedAfterCloudAnalysisIsEmpty	0x40001131	No response is returned by the cloud analytic service.
selectingIpAddressOfHostAndSpareNodeFailedCheckTheStatus	0x40001132	Setting IP address for main node and backup node failed. Check the node status.
theSearchIdDoesNotExist	0x40001133	The search ID does not exist.
theSynchronizationIdDoesNotExist	0x40001134	The synchronization ID does not exist.
theUserIdDoesNotExist	0x40001136	The user ID does not exist.
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.
editingStreamingTypesNotSupported	0x40001140	Editing streaming type is not supported.
switchingScheduledTaskToTemporaryTaskIsNotSupported	0x40001141	Switching scheduled task to temporary task is not supported.

Sub Status Code	Error Code	Description
switchingTemporaryTaskToScheduledTasksNotSupported	0x40001142	Switching temporary task to scheduled task is not supported.
theTaskIsNotDispatchedOrItIsUpdating	0x40001143	The task is not dispatched or is updating.
thisTaskDoesNotExist	0x40001144	This task does not exist in the cloud analytic service.
duplicatedSchedule	0x40001145	Schedule period cannot be overlapped.
continuousScheduleWithSameAlgorithmTypeShouldBeMerged	0x40001146	The continuous schedule periods with same algorithm type should be merged.
invalidStreamingTimeRange	0x40001147	Invalid streaming time period.
invalidListLibraryType	0x40001148	Invalid list library type.
theNumberOfMatchedResultsShouldBeLargerThan0	0x40001149	The number of search results should be larger than 0.
invalidValueRangeOfSimilarity	0x4000114A	Invalid similarity range.
invalidSortingType	0x4000114B	Invalid sorting type.
noMoreListLibraryCanBeLinkedToTheDevice	0x4000114C	No more lists can be added to one device.
InvalidRecipientAddressFormat	0x4000114D	Invalid address format of result receiver.
creatingClusterFailedTheDongleIsNotPluggedIn	0x4000114E	Insert the dongle before creating cluster.
theURLIsTooLong	0x4000114F	No schedule configured for the task.
noScheduleIsConfiguredForTheTask	0x40001150	No schedule configured for the task.
theDongleIsExpired	0x40001151	Dongle has expired.
dongleException	0x40001152	Dongle exception.
invalidKey	0x40001153	Invalid authorization service key.

Sub Status Code	Error Code	Description
decryptionFailed	0x40001154	Decrypting authorization service failed.
encryptionFailed	0x40001155	Encrypting authorization service failed.
AuthorizeServiceResponseError	0x40001156	Authorization service response exception.
incorrectParameter	0x40001157	Authorization service parameters error.
operationFailed	0x40001158	Operating authorization service error.
noAnalysisResourceOrNoDataInTheListLibrary	0x40001159	No cloud analytic resources or no data in the list library.
calculationException	0x4000115A	Calculation exception.
allocatingList	0x4000115B	Allocating list.
thisOperationIsNotSupportedByTheCloudAnalytics	0x4000115C	This operation is not supported by the cloud analytic service.
theCloudAnalyticsIsInterrupted	0x4000115D	The operation of cloud analytic service is interrupted.
theServiceIsNotReady	0x4000115E	The service is not ready.
searchingForExternalApiFailed	0x4000115F	Searching external interfaces failed.
noOnlineNode	0x40001160	No node is online.
noNodeAllocated	0x40001161	No allocated node.
noMatchedList	0x40001162	No matched list.
allocatingFailedTooManyFacePictureLists	0x40001163	Allocation failed. Too many lists of big data service.
searchIsNotCompletedSearchAgain	0x40001164	Current searching is not completed. Search again.
allocatingListIsNotCompleted	0x40001165	Allocating list is not completed.
searchingForCloudAnalyticsResultsFailed	0x40001166	Searching cloud analytic service overtime.
noDataOfTheCurrentLibraryFound	0x40001167	No data in the current library. Make sure there is data in the Hbase.

Sub Status Code	Error Code	Description
noFacePictureLibraryIsArmed	0x40001168	No face picture library is armed for big data service.
noAvailableDataSlicingVersionInformationArmedFirstAndSliceTheData	0x40001169	Invalid standard version information.
duplicatedOperationDataSlicingIsExecuting	0x4000116A	Slicing failed. Duplicated operation.
slicingDataFailedNoArmedFacePictureLibrary	0x4000116B	Slicing failed. No arming information in the face big data.
GenerateBenchmarkFileFailedSlicingAgain	0x4000116C	Generating sliced file failed. Slice again.
NonprimaryNodesProhibitedFromSlicingData	0x4000116D	Slicing is not allowed by the backup node.
NoReadyNodeToClusterServers	0x4000116E	Creating the cluster failed. No ready node.
NodeManagementServicesOffline	0x4000116F	The node management server is offline.
theCamera(s)OfTheControlCenterAreAlreadyArmed.DisarmThemFirst	0x40001170	Some cameras in control center are already armed. Disarm them and try again.
theCamera(s)OfTheAreaAreAlreadyArmed.DisarmThemFirst	0x40001171	Some cameras in this area are already armed. Disarm them and try again.
configuringHigh-frequencyPeopleDetectionFailed	0x40001172	Configuring high frequency people detection failed.
searchingForHigh-frequencyPeopleDetectionLogsFailed.	0x40001173	Searching detection event logs of high-frequency people detection failed.
gettingDetailsOfSearchesHigh-frequencyPeopleDetectionLogsFailed.	0x40001174	Getting the search result details of frequently appeared person alarms failed.

Sub Status Code	Error Code	Description
theArmedCamerasAlreadyExistInTheControlCenter	0x40001175	Some cameras in control center are already armed.
disarmingFailedTheCamerasIsNotArmed	0x40001177	Disarming failed. The camera is not armed.
noDataReturned	0x40001178	No response is returned by the big data service.
preallocFailure	0x40001179	Pre-allocating algorithm resource failed.
overDogLimit	0x4000117A	Configuration failed. No more resources can be pre-allocated.
analysisServicesDoNotSupport	0x4000117B	Not supported.
commandAndDispatchServiceError	0x4000117C	Scheduling service of cloud analytic service error.
engineModuleError	0x4000117D	Engine module of cloud analytic service error.
streamingServiceError	0x4000117E	Streaming component of cloud analytic service error.
faceAnalysisModuleError	0x4000117F	Face analysis module of cloud analytic service error.
vehicleAnalysisModuleError	0x40001180	Vehicle pictures analytic module of cloud analytic service error.
videoStructuralAnalysisModuleError	0x40001181	Video structuring module of cloud analytic service error.
postprocessingModuleError	0x40001182	Post-processing module of cloud analytic service error.
frequentlyAppearedPersonAlarmsAlreadyConfiguredForListLibrary	0x40001183	Frequently appeared person alarm is already armed for blocklist library.
creatingListLibraryFailed	0x40001184	Creating list library failed.
invalidIdentityKeyOfListLibrary	0x40001185	Invalid identity key of list library.
noMoreDevicesCanBeArmed	0x40001186	No more camera can be added.

Sub Status Code	Error Code	Description
settingAlgorithmTypeForDeviceFailed	0x40001187	Allocating task resource failed.
gettingHighFrequencyPersonDetectionAlarmInformationFailed	0x40001188	Setting frequently appeared person alarm failed.
invalidSearchConfiton	0x40001189	Invalid result.
theTasksIsNotCompleted	0x4000118B	The task is not completed.
resourceOverRemainLimit	0x4000118C	No more resource can be pre-allocated.
frequentlyAppearedPersonAlarmsAlreadyConfiguredForTheCameraDisarmFirstAndTryAgain	0x4000118D	The frequently appeared person alarm of this camera is configured. Delete the arming information and try again.
switchtimedifflesslimit	0x4000123b	Time difference between power on and off should be less than 10 minutes.
associatedFaceLibNumOverLimit	0x40001279	Maximum number of linked face picture libraries reached.
noMorePeopleNumChangeRulesAdded	0x4000128A	Maximum number of people number changing rules reached.
noMoreViolentMotionRulesAdded	0x4000128D	Maximum number of violent motion rules reached.
noMoreLeavePositionRulesAdded	0x4000128E	Maximum number of leaving position rules reached.
SMRDiskNotSupportRaid	0x40001291	SMR disk does not support RAID.
OnlySupportHikAndCustomProtocol	0x400012A3	IPv6 camera can only be added via Device Network SDK or custom protocols.
vehicleEnginesNoResource	0x400012A6	Insufficient vehicle engine resources.
noMoreRunningRulesAdded	0x400012A9	Maximum number of running rules reached.

Sub Status Code	Error Code	Description
noMoreGroupRulesAdded	0x400012AA	Maximum number of people gathering rules reached.
noMoreFailDownRulesAdded	0x400012AB	Maximum number of people falling down rules reached.
noMorePlayCellphoneRulesAdded	0x400012AC	Maximum number of playing cellphone rules reached.
ruleEventTypeDuplicate	0x400012C8	Event type duplicated.
noMoreRetentionRulesAdded	0x400015AD	Maximum number of people retention rules reached.
noMoreSleepOnDutyRulesAdded	0x400015AE	Maximum number of sleeping on duty rules reached.
polygonNotAllowedCrossing	0x400015C2	Polygons are not allowed to cross.
configureRuleBeforeAdvanceParam	0x400015F8	Advanced parameters fail to be configured as no rule is configured, please configure rule information first.
behaviorCanNotPackToPic	0x40001603	The behavior model cannot be packaged as a picture algorithm.
noCluster	0x40001608	No cluster created.
NotAssociatedWithOwnChannel	0x400019C1	Current channel is not linked.
AITargetBPCaptureFail	0x400019C5	Capturing reference picture for AI target comparison failed.
AITargetBPToDSPFail	0x400019C6	Sending reference picture to DSP for AI target comparison failed.
AITargetBPDuplicateName	0x400019C7	Duplicated name of reference picture for AI target comparison.
audioFileNameWrong	0x400019D0	Incorrect audio file name.
audioFileImportFail	0x400019D1	Importing audio file failed.
NonOperationalStandbyMachine	0x400019F0	Non-operational hot spare.

Sub Status Code	Error Code	Description
MaximumNumberOfDevices	0x400019F1	The maximum number of devices reached.
StandbyMmachineCannotBeDeleted	0x400019F2	The hot spare cannot be deleted.
alreadyRunning	0x40002026	The application program is running.
notRunning	0x40002027	The application program is stopped.
packNotFound	0x40002028	The software packet does not exist.
alreadyExist	0x40002029	The application program already exists.
noMemory	0x4000202A	Insufficient memory.
invalidLicense	0x4000202B	Invalid License.
noClientCertificate	0x40002036	The client certificate is not installed.
noCACertificate	0x40002037	The CA certificate is not installed.
authenticationFailed	0x40002038	Authenticating certificate failed. Check the certificate.
clientCertificateExpired	0x40002039	The client certificate is expired.
clientCertificateRevocation	0x4000203A	The client certificate is revoked.
CACertificateExpired	0x4000203B	The CA certificate is expired.
CACertificateRevocation	0x4000203C	The CA certificate is revoked.
connectFail	0x4000203D	Connection failed.
loginNumExceedLimit	0x4000203F	No more user can log in.
HDMIResolutionIllegal	0x40002040	The HDMI video resolution cannot be larger than that of main and sub stream.
hdFormatFail	0x40002049	Formatting HDD failed.
formattingFailed	0x40002056	Formatting HDD failed.
encryptedFormattingFailed	0x40002057	Formatting encrypted HDD failed.
wrongPassword	0x40002058	Verifying password of SD card failed. Incorrect password.

Sub Status Code	Error Code	Description
audiosPlayingPleaseWait	0x40002067	Audio is playing. Please wait.
twoWayAudioInProgressPleaseWait	0x40002068	Two-way audio in progress. Please wait.
calibrationPointNumFull	0x40002069	The maximum number of calibration points reached.
completeTheLevelCalibrationFirst	0x4000206A	The level calibration is not set.
completeTheRadarCameraCalibrationFirst	0x4000206B	The radar-camera calibration is not set.
pointsOnStraightLine	0x4000209C	Calibrating failed. The calibration points cannot be one the same line.
TValueLessThanOrEqualZero	0x4000209D	Calibration failed. The T value of the calibration points should be larger than 0.
HBDLibNumOverLimit	0x40002092	The number of human body picture libraries reaches the upper limit
theShieldRegionError	0x40002093	Saving failed. The shielded area should be the ground area where the shielded object is located.
theDetectionAreaError	0x40002094	Saving failed. The detection area should only cover the ground area.
invalidLaneLine	0x40002096	Saving failed. Invalid lane line.
enableITSFunctionOfThisChannelFirst	0x400020A2	Enable ITS function of this channel first.
noCloudStorageServer	0x400020C5	No cloud storage server
NotSupportWithVideoTask	0x400020F3	This function is not supported.
noDetectionArea	0x400050df	No detection area
armingFailed	0x40008000	Arming failed.
disarmingFailed	0x40008001	Disarming failed.
clearAlarmFailed	0x40008002	Clearing alarm failed.
bypassFailed	0x40008003	Bypass failed.

Sub Status Code	Error Code	Description
bypassRecoverFailed	0x40008004	Bypass recovery failed.
outputsOpenFailed	0x40008005	Opening relay failed.
outputsCloseFailed	0x40008006	Closing relay failed.
registerTimeOut	0x40008007	Registering timed out.
registerFailed	0x40008008	Registering failed.
addedByOtherHost	0x40008009	The peripheral is already added by other security control panel.
alreadyAdded	0x4000800A	The peripheral is already added.
armedStatus	0x4000800B	The partition is armed.
bypassStatus	0x4000800C	Bypassed.
zoneNotSupport	0x4000800D	This operation is not supported by the zone.
zoneFault	0x4000800E	The zone is in fault status.
pwdConflict	0x4000800F	Password conflicted.
audioTestEntryFailed	0x40008010	Enabling audio test mode failed.
audioTestRecoveryFailed	0x40008011	Disabling audio test mode failed.
addCardMode	0x40008012	Adding card mode.
searchMode	0x40008013	Search mode.
addRemoterMode	0x40008014	Adding keyfob mode.
registerMode	0x40008015	Registration mode.
exDevNotExist	0x40008016	The peripheral does not exist.
theNumberOfExDevLimited	0x40008017	No peripheral can be added.
sirenConfigFailed	0x40008018	Setting siren failed.
chanCannotRepeatedBinded	0x40008019	This channel is already linked by the zone.
inProgramMode	0x4000801B	The keypad is in programming mode.
inPaceTest	0x4000801C	In pacing mode.
arming	0x4000801D	Arming.

Sub Status Code	Error Code	Description
masterSlaveIsEnable	0x4000802c	The main-sub relationship has taken effect, the sub radar does not support this operation.
forceTrackNotEnabled	0x4000802d	Mandatory tracking is disabled.
isNotSupportZoneConfigByLocalArea	0x4000802e	This area does not support the zone type.
alarmLineCross	0x4000802f	Trigger lines are overlapped.
zoneDrawingOutOfRange	0x40008030	The drawn zone is out of detection range.
alarmLineDrawingOutOfRange	0x40008031	The drawn alarm trigger line is out of detection range.
hasTargetInWarningArea	0x40008032	The warning zone already contains targets. Whether to enable mandatory arming?
radarModuleConnectFail	0x40008033	Radar module communication failed.
importCfgFilePasswordErr	0x40008034	Incorrect password for importing configuration files.
overAudioFileNumLimit	0x40008038	The number of audio files exceeds the limit.
audioFileNameIsLong	0x40008039	The audio file name is too long.
audioFormatIsWrong	0x4000803a	The audio file format is invalid.
audioFileIsLarge	0x4000803b	The size of the audio file exceeds the limit.
pircamCapTimeOut	0x4000803c	Capturing of pircam timed out.
pircamCapFail	0x4000803d	Capturing of pircam failed.
pircamIsCaping	0x4000803e	The pircam is capturing.
audioFileHasExisted	0x4000803f	The audio file already exists.
subscribeTypeErr	0x4000a016	This metadata type is not supported to be subscribed.
EISError	0x4000A01C	Electronic image stabilization failed. The smart event function is enabled.
jpegPicWithAppendDataError	0x4000A01D	Capturing the thermal graphic failed. Check if the temperature measurement parameters

Sub Status Code	Error Code	Description
		(emissivity, distance, reflective temperature) are configured correctly.
startAppFail	/	Starting running application program failed.
yuvconflict	/	The raw video stream conflicted.
overMaxAppNum	/	No more application program can be uploaded.
noFlash	/	Insufficient flash.
platMismatch	/	The platform mismatches.
emptyEventName	0x400015E0	Event name is empty.
sameEventName	0x400015E1	A same event name already exists.
emptyEventType	0x400015E2	Event type is required.
sameEventType	0x400015E3	A same event type already exists.
maxEventNameReached	0x400015E4	Maximum of events reached.
hotSpareNotAllowedExternalStorage	0x400015FC	External storage is not allowed when hot spare is enabled.
sameCustomProtocolName	0x400015FD	A same protocol name already exists.
maxPTZTriggerChannelReached	0x400015FE	Maximum of channels linked with PTZ reached.
POSCannotAddHolidayPlan	0x400015FF	No POS events during holidays.
eventTypesTooLong	0x40001600	Event type is too long.
eventNamesTooLong	0x40001601	Event name is too long.
PerimeterEnginesNoResource	0x40001602	No more perimeter engines.
invalidProvinceCode	0x40001607	Invalid province code.

StatusCode=5

Sub Status Code	Error Code	Description
badXmlFormat	0x50000001	Invalid XML format.

StatusCode=6

Sub Status Code	Error Code	Description
badParameters	0x60000001	Invalid parameter.
badHostAddress	0x60000002	Invalid host IP address.
badXmlContent	0x60000003	Invalid XML content.
badIPv4Address	0x60000004	Invalid IPv4 address.
badIPv6Address	0x60000005	Invalid IPv6 address.
conflictIPv4Address	0x60000006	IPv4 address conflicted.
conflictIPv6Address	0x60000007	IPv6 address conflicted.
badDomainName	0x60000008	Invalid domain name.
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.
incorrentUserNameOrPasswor d	0x600000012	Incorrect user name or password.
invalidStoragePoolOfCloudServ er	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.

Sub Status Code	Error Code	Description
userPwdLenUnder8	0x60000023	At least two kinds of characters, including digits, letters, and symbols, should be contained in the password.
userPwdNameSame	0x60000025	Duplicated password.
userPwdNameMirror	0x60000026	The password cannot be the reverse order of user name.
beyondARGSRangeLimit	0x60000027	The parameter value is out of limit.
DetectionLineOutofDetectionRegion	0x60000085	The rule line is out of region.
DetectionRegionError	0x60000086	Rule region error. Make sure the rule region is convex polygon.
DetectionRegionOutOfCountingRegion	0x60000087	The rule region must be marked as red frame.
PedalAreaError	0x60000088	The pedal area must be in the rule region.
DetectionAreaABError	0x60000089	The detection region A and B must be in the a rule frame.
ABRegionCannotIntersect	0x6000008a	Region A and B cannot be overlapped.
customHBPIDError	0x6000008b	Incorrect ID of custom human body picture library
customHBPIDRepeat	0x6000008c	Duplicated ID of custom human body picture library
dataVersionsInHBDLibMismatches	0x6000008d	Database versions mismatches of human body picture library
invalidHBPID	0x6000008e	Invalid human body picture PID
invalidHBDID	0x6000008f	Invalid ID of human body picture library
humanLibraryError	0x60000090	Error of human body picture library

Sub Status Code	Error Code	Description
humanLibraryNumError	0x60000091	No more human body picture library can be added
humanImagesNumError	0x60000092	No more human body picture can be added
noHumanInThePicture	0x60000093	Modeling failed, no human body in the picture
analysisEnginesNoResourceError	0x60001000	No analysis engine.
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.
analysisEnginesFacelibImporting	0x60001005	Importing pictures to face picture library. Failed to edit analysis engine parameters.
analysisEnginesAssociatedChannel	0x60001006	The analysis engine is linked to channel.
smdEncodingNoResource	0x60001007	Insufficient motion detection encoding resources.
smdDecodingNoResource	0x60001008	Insufficient motion detection decoding resources.
diskError	0x60001009	HDD error.
diskFull	0x6000100a	HDD full.
facelibDataProcessing	0x6000100b	Handling face picture library data.
capturePackageFailed	0x6000100c	Capturing packet failed.
capturePackageProcessing	0x6000100d	Capturing packet.
noSupportWithPlaybackAbstract	0x6000100e	This function is not supported. Playback by video synopsis is enabled.

Sub Status Code	Error Code	Description
insufficientNetworkBandwidth	0x6000100f	Insufficient network bandwidth.
tapeLibNeedStopArchive	0x60001010	Stop the filing operation of tape library first.
identityKeyError	0x60001011	Incorrect interaction command.
identityKeyMissing	0x60001012	The interaction command is lost.
noSupportWithPersonDensityDetect	0x60001013	This function is not supported. The people density detection is enabled.
ipcResolutionOverflow	0x60001014	The configured resolution of network camera is invalid.
ipcBitrateOverflow	0x60001015	The configured bit rate of network camera is invalid.
tooGreatTimeDifference	0x60001016	Too large time difference between device and server.
noSupportWithPlayback	0x60001017	This function is not supported. Playback is enabled.
channelNoSupportWithSMD	0x60001018	This function is not supported. Motion detection is enabled.
channelNoSupportWithFD	0x60001019	This function is not supported. Face capture is enabled.
illegalPhoneNumber	0x6000101a	Invalid phone number.
illegalCertificateNumber	0x6000101b	Invalid certificate No.
linkedCameraOutLimit	0x6000101c	Connecting camera timed out.
achieveMaxChannelLimit	0x6000101e	No more channels are allowed.
humanMisInfoFilterEnabledChannelNumError	0x6000101f	No more channels are allowed to enable preventing false alarm.
humanEnginesNoResource	0x60001020	Insufficient human body analysis engine resources.
taskNumberOverflow	0x60001021	No more tasks can be added.

Sub Status Code	Error Code	Description
collisionTimeOverflow	0x60001022	No more comparison duration can be configured.
invalidTaskID	0x60001023	Invalid task ID.
eventNotSupport	0x60001024	Event subscription is not supported.
invalidEZVIZSecretKey	0x60001034	Invalid verification code for Hik-Connect.
needDoubleVerification	0x60001042	Double verification required
noDoubleVerificationUser	0x60001043	No double verification user
timeSpanNumOverLimit	0x60001044	Max. number of time buckets reached
channelNumOverLimit	0x60001045	Max. number of channels reached
noSearchIDResource	0x60001046	Insufficient searchID resources
noSupportDeleteStrangerLib	0x60001051	Deleting stranger library is not supported
noSupportCreateStrangerLib	0x60001052	Creating stranger library is not supported
behaviorAnalysisRuleInfoError	0x60001053	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.

Sub Status Code	Error Code	Description
AdaptiveStreamNotEnabled	0x60002001	The stream self-adaptive function is not enabled.
AdaptiveStreamAndVariableBit rateEnabled	0x60002002	Stream self-adaptive and variable bitrate function cannot be enabled at the same time.
noSafetyHelmetRegion	0x60002023	The hard hat detection area is not configured (if users save their settings without configuring the arming area, they should be prompted to configure one).
unclosedSafetyHelmet	0x60002024	The hard hat detection is enabled (If users save their settings after deleting the arming area, they should be prompted to disable hard hat detection first and then delete the arming area).
width/heightRatioOfPictureError	0x6000202C	The width/height ratio of the uploaded picture should be in the range from 1:2 to 2:1.
PTZNotInitialized	0x6000202E	PTZ is not initialized.
PTZSelfChecking	0x6000202F	PTZ is self-checking.
PTZLocked	0x60002030	PTZ is locked.
advancedParametersError	0x60002031	Auto-switch interval in advanced parameters cannot be shorter than parking tolerance for illegal parking detection in speed dome rule settings.
resolutionError	0x60005003	Invalid resolution
deployExceedMax	0x60006018	The arming connections exceed the maximum number.
detectorTypeMismatch	0x60008000	The detector type mismatched.
nameExist	0x60008001	The name already exists.

Sub Status Code	Error Code	Description
uploadImageSizeError	0x60008016	The size of the uploaded picture is larger than 5 MB.
laneAndRegionOverlap	/	The lanes are overlapped.
unitConfigurationNotInEffect	/	Invalid unit parameter.
ruleAndShieldingMaskConflict	/	The line-rule region overlaps with the shielded area.
wholeRuleInShieldingMask	/	There are complete temperature measurement rules in the shielded area.
LogDiskNotSetReadOnlyInGroupMode	0x60001100	The log HDD in the HDD group cannot be set to read-only.
LogDiskNotSetRedundancyInGroupMode	0x60001101	The log HDD in the HDD group cannot be set to redundancy.
holidayNameContainChineseOrSpecialChar	0x60001080	No Chinese and special characters allowed in holiday name.
genderValueError	0x60001081	Invalid gender.
certificateTypeValueError	0x60001082	Invalid identification type.
personInfoExtendValuesTooLong	0x60001083	The length of customized tags exceeds limit.
personInfoExtendValueContainsInvalidChar	0x60001084	Invalid characters are not allowed in customized tags of the face picture library.
excelHeaderError	0x60001085	Excel header error.
intelligentTrafficMutexWithHighFrames	0x60008014	Please disable all functions of traffic incident detection, violation enforcement, and traffic data collection, or adjust the video frame rate to that lower than 50 fps.
intelligentTrafficMutexWithHighFramesEx	0x60008018	Please disable all functions of traffic incident detection, violation enforcement, traffic data collection, and vehicle

Sub Status Code	Error Code	Description
		detection, or adjust the video frame rate to that lower than 50 fps.

StatusCode=7

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

