



**Intelligent Security API**

**Developer Guide**

## **About this Document**

- This Document includes instructions for using and managing the Product. Pictures, charts, images and all other information hereinafter are for description and explanation only. Unless otherwise agreed, our company makes no warranties, express or implied.
- Please use this Document with the guidance and assistance of professionals trained in supporting the Product.

## **Trademarks Acknowledgment**

All trademarks and logos mentioned are the properties of their respective owners.

## **LEGAL DISCLAIMER**

- TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THIS DOCUMENT AND THE PRODUCT DESCRIBED, WITH ITS HARDWARE, SOFTWARE AND FIRMWARE, ARE PROVIDED "AS IS" AND "WITH ALL FAULTS AND ERRORS". OUR COMPANY MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, MERCHANTABILITY, SATISFACTORY QUALITY, OR FITNESS FOR A PARTICULAR PURPOSE. THE USE OF THE PRODUCT BY YOU IS AT YOUR OWN RISK. IN NO EVENT WILL OUR COMPANY BE LIABLE TO YOU 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), PRODUCT LIABILITY, OR OTHERWISE, IN CONNECTION WITH THE USE OF THE PRODUCT, EVEN IF OUR COMPANY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR LOSS.
- YOU ACKNOWLEDGE THAT THE NATURE OF THE INTERNET PROVIDES FOR INHERENT SECURITY RISKS, AND OUR COMPANY SHALL NOT TAKE ANY RESPONSIBILITIES FOR ABNORMAL OPERATION, PRIVACY LEAKAGE OR OTHER DAMAGES RESULTING FROM CYBER-ATTACK, HACKER ATTACK, VIRUS INFECTION, OR OTHER INTERNET SECURITY RISKS; HOWEVER, OUR COMPANY WILL PROVIDE TIMELY TECHNICAL SUPPORT IF REQUIRED.
- YOU AGREE TO USE THIS PRODUCT IN COMPLIANCE WITH ALL APPLICABLE LAWS, AND YOU ARE SOLELY RESPONSIBLE FOR ENSURING THAT YOUR USE CONFORMS TO THE APPLICABLE LAW. ESPECIALLY, YOU ARE RESPONSIBLE, FOR USING THIS PRODUCT IN A MANNER THAT DOES NOT INFRINGE ON THE RIGHTS OF THIRD PARTIES, INCLUDING WITHOUT LIMITATION, RIGHTS OF PUBLICITY, INTELLECTUAL PROPERTY RIGHTS, OR DATA PROTECTION AND OTHER PRIVACY RIGHTS, YOU SHALL NOT USE THIS PRODUCT FOR ANY PROHIBITED END-USAGES, INCLUDING THE DEVELOPMENT OR PRODUCTION OF WEAPONS OF MASS DESTRUCTION, THE DEVELOPMENT OR PRODUCTION OF CHEMICAL OR BIOLOGICAL WEAPONS, ANY ACTIVITIES IN THE CONTEXT RELATED TO ANY NUCLEAR EXPLOSIVE OR UNSAFE NUCLEAR FUEL-CYCLE, OR IN SUPPORT OF HUMAN RIGHTS ABUSES.
- IN THE EVENT OF ANY CONFLICTS BETWEEN THIS DOCUMENT AND THE APPLICABLE LAW, THE LATTER PREVAILS.

# 1 Reading Guide

Chapter	Description
Overview	Includes the ISAPI overview, applicable products, terms and definitions, abbreviations, and update history.
ISAPI Framework	Read the chapter to take a quick look at the ISAPI framework and basic functions.
Quick Start Guide	Read the chapter to quickly understand the programming process of basic functions such as authentication, message parsing, real-time live view, playback, and event uploading.
API Reference	Start programming according to API definitions.
How-To Video Guidance	How-to videos demonstrate detailed steps of different integration tasks.

## 2 Overview

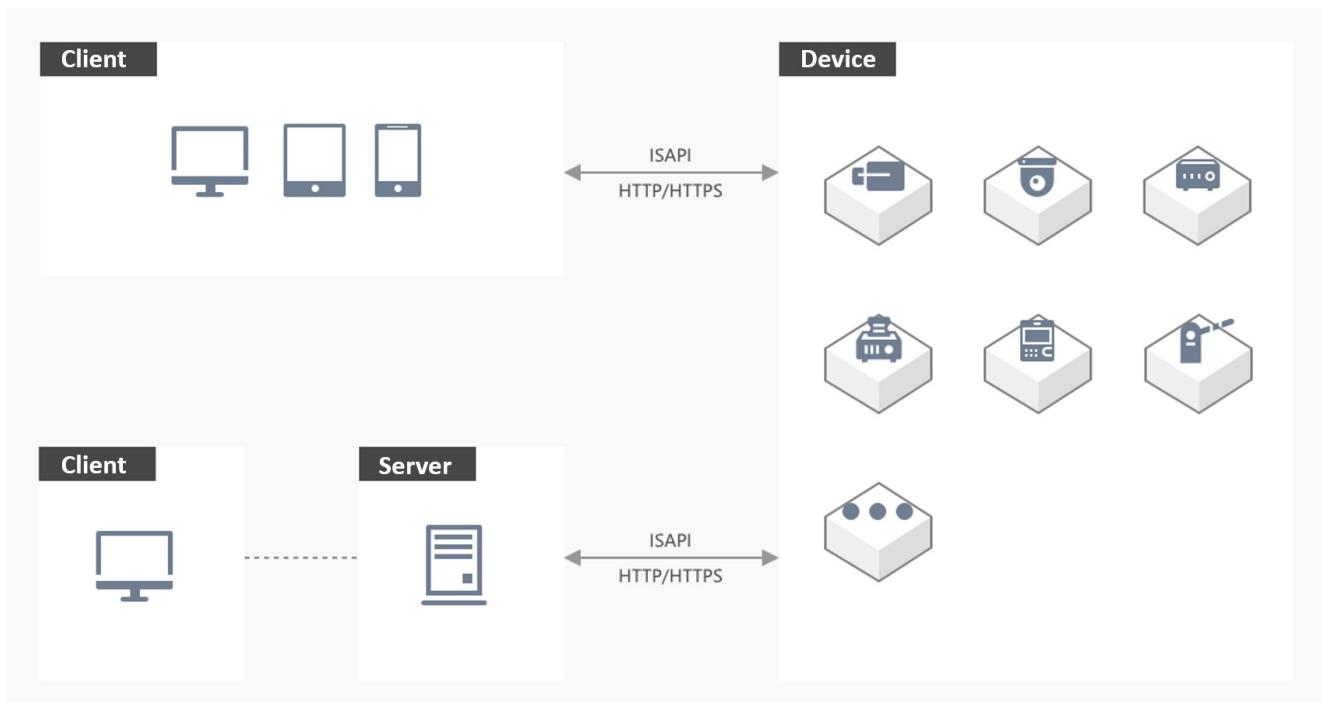
### 2.1 Introduction

Intelligent Security API (hereinafter referred to as ISAPI) is an application layer protocol based on [HTTP \(Hypertext Transfer Protocol\)](#) and adopts the REST (Representational State Transfer) architecture for communication between security devices (cameras, DVRs, NVRs, etc.) and the platform or client software.

Since established in 2013, ISAPI has included more than 11,000 APIs for different functions, including device management, vehicle recognition, parking lot management, intelligent facial application, access control management, interrogation management, and recording management. It is applicable to industries such as traffic, fire protection, education, and security inspection.

#### 2.1.1 Application Scenario

When you integrate devices via ISAPI, the device acts as the server to listen on the fixed port and the user's application acts as the client to actively log in to the device for communication. To achieve the above goals, the device should be configured with a fixed IP address and the requests from the client can reach the server.



## 2.1.2 Layers in the Network Model

ISAPI is an application layer protocol based on HTTP, thereby it inherits all specifications and properties from HTTP.

Protocols frequently used along with ISAPI include SADP (Search Active Device Protocol) based on multicast for discovering and activating devices, [RTSP \(Real-Time Streaming Protocol\)](#) based on TCP/UDP for live view and video playback of devices, etc.

Application Layer	Device Discovery SADP	Signaling Interaction ISAPI (HTTP)	Media Stream RTSP/RTP
Transport Layer	MCAST	TCP	TCP/UDP
Network Layer	IP		
Network Interface	Hardware Drive Interface		

## 2.2 Product Scope

- Network Cameras

- Pro Series(EasyIP)

DS-2CD2021G1-I, DS-2CD2023G2-I(U), DS-2CD2023G2-LI(2U), DS-2CD2025FHW-I, DS-2CD2026G2-I(U), DS-2CD2026G2-IU/SL, DS-2CD2027G2-L(U), DS-2CD2043G0-I, DS-2CD2043G2-I(U), DS-2CD2043G2-LI(2U), DS-2CD2045FWD-I, DS-2CD2046G2-I(U), DS-2CD2046G2-IU/SL, DS-2CD2046G2H-I(U), DS-2CD2047G2-L(U), DS-2CD2047G2-LU/SL, DS-2CD2047G2H-LI(U), DS-2CD2047G2H-LIU/SL, DS-2CD2048G2-I(U)K(US), DS-2CD2063G2-I(U), DS-2CD2063G2-LI(2U), DS-2CD2065G1-I, DS-2CD2066G2-I(U), DS-2CD2066G2-IU/SL, DS-2CD2066G2H-I(U), DS-2CD2067G2-L(U), DS-2CD2067G2H-LI(U), DS-2CD2067G2H-LIU/SL, DS-2CD2083G2-I(U), DS-2CD2083G2-LI(2U), DS-2CD2085G1-I, DS-2CD2086G2-I(U), DS-2CD2086G2-IU/SL, DS-2CD2086G2H-I(U), DS-2CD2087G2-L(U), DS-2CD2087G2-L(U)K, DS-2CD2087G2H-LI(U), DS-2CD2087G2H-LIU/SL, DS-2CD2121G0-I(W)(S), DS-2CD2123G2-I(S), DS-2CD2123G2-IU, DS-2CD2123G2-LI(S)(2U), DS-2CD2125FHW-I(S), DS-2CD2125G0-IMS, DS-2CD2126G2-I(SU), DS-2CD2126G2-IMS, DS-2CD2127G2-(SU), DS-2CD2143G0-I(S), DS-2CD2143G0-IU, DS-2CD2143G2-I(S), DS-2CD2143G2-IU, DS-2CD2143G2-LI(S)(2U), DS-2CD2145FWD-I(S), DS-2CD2146G2-I(SU), DS-2CD2146G2H-I(SU), DS-2CD2147G2(-SU), DS-2CD2147G2-L(SU), DS-2CD2147G2H-LI(SU), DS-2CD2148G2-LI(SU)K(US), DS-2CD2163G2-I(S), DS-2CD2163G2-IU, DS-2CD2163G2-LI(S)(2U), DS-2CD2165G0-I(S), DS-2CD2166G2-I(SU), DS-2CD2166G2H-I(SU), DS-2CD2167G2H-LI(SU), DS-2CD2183G2-I(S), DS-2CD2183G2-IU, DS-2CD2183G2-LI(S)(2U), DS-2CD2185G0-IMS, DS-2CD2186G2-I(SU), DS-2CD2186G2-IMS, DS-2CD2186G2H-I(SU), DS-2CD2187G2-L(SU), DS-2CD2187G2H-LI(SU), DS-2CD2321G0-I/NF, DS-2CD2323G2-I(U), DS-2CD2323G2-LI(2U), DS-2CD2326G2-I(U), DS-2CD2326G2-ISU/SL, DS-2CD2327G2-L(U), DS-2CD2343G0-I(U), DS-2CD2343G2-I(U), DS-2CD2343G2-LI(2U), DS-2CD2345FWD-I, DS-2CD2345G0P-I, DS-2CD2346G2-I(U), DS-2CD2346G2-ISU/SL, DS-2CD2346G2H-I(U), DS-2CD2346G2H-IS2U/S(L)(RB), DS-2CD2346G2P-ISU/SL, DS-2CD2347G1-L(U), DS-2CD2347G2-L(U), DS-2CD2347G2-LSU/SL, DS-2CD2348G2-LI(U)K(US), DS-2CD2363G2-I(U), DS-2CD2363G2-LI(2U), DS-2CD2365G1-I, DS-2CD2366G2-I(U), DS-2CD2366G2-ISU/SL, DS-2CD2366G2H-I(U), DS-2CD2366G2H-IS2U/S(L)(RB), DS-2CD2366G2P-ISU/SL, DS-2CD2367G2-L(U), DS-2CD2367G2H-LI(U), DS-2CD2367G2H-ISU/SL, DS-2CD2367G2P-LSU/SL, DS-2CD2383G2-I(U), DS-2CD2383G2-LI(2U), DS-2CD2385G1-I, DS-2CD2386G2-I(U), DS-2CD2386G2-ISU/SL, DS-2CD2386G2H-I(U), DS-2CD2386G2H-IS2U/S(L)(RB), DS-2CD2387G2-L(U), DS-2CD2387G2-L(U)K, DS-2CD2387G2-LSU/SL, DS-2CD2387G2H-LI(U), DS-2CD2387G2H-ISU/SL, DS-2CD2387G2P-LSU/SL, DS-2CD2421G0-I(D)(W), DS-2CD2423G2-I, DS-2CD2423G2-IW, DS-2CD2426G2-I, DS-2CD2426G2-IW, DS-2CD2441G0-I(W), DS-2CD2443G0-I(W), DS-2CD2443G2-I, DS-2CD2443G2-IW, DS-2CD2446G2-I, DS-2CD2446G2-IW, DS-2CD2463G2-I, DS-

2CD2466G2-I, DS-2CD2483G2-I, DS-2CD2523G2-I(S), DS-2CD2523G2-LI(W)(S)(2U), DS-2CD2526G2-I(S), DS-2CD2527G2-LS, DS-2CD2543G0-I(W)(S), DS-2CD2543G2-I(WS), DS-2CD2543G2-LI(W)(S)(2U), DS-2CD2545FWD-I(W)(S), DS-2CD2546G2-I(WS), DS-2CD2547G2-LS, DS-2CD2563G2-I(S), DS-2CD2563G2-LI(W)(S)(2U), DS-2CD2566G2-I(S), DS-2CD2583G2-I(S), DS-2CD2583G2-LI(W)(S)(2U), DS-2CD2586G2-I(S), DS-2CD2621G0-I(Z)(S), DS-2CD2623G2-IZS, DS-2CD2623G2-LIZS2U, DS-2CD2623G2-LIZS2U/SL, DS-2CD2625FWD-IZS, DS-2CD2626G2-IZS, DS-2CD2626G2-IZSU/SL, DS-2CD2626G2T-IZS, DS-2CD2643G0-IZS, DS-2CD2643G1-IZ(S), DS-2CD2643G2-IZS, DS-2CD2643G2-LIZS2U, DS-2CD2643G2-LIZS2U/SL, DS-2CD2645FWD-IZS, DS-2CD2645G1-IZS, DS-2CD2646G2-IZS, DS-2CD2646G2-IZSU/SL, DS-2CD2646G2HT-IZS, DS-2CD2646G2HT-IZS2U/S(L)(RB), DS-2CD2646G2T-IZS, DS-2CD2646G2T-IZSY, DS-2CD2647G2-LZS, DS-2CD2647G2HT-LIZS, DS-2CD2647G2T-LZS, DS-2CD2648G2T-LIZS(US), DS-2CD2663G2-IZS, DS-2CD2663G2-LIZS2U, DS-2CD2663G2-LIZS2U/SL, DS-2CD2665G0-IZS, DS-2CD2665G1-IZS, DS-2CD2666G2-IZS, DS-2CD2666G2-IZSU/SL, DS-2CD2666G2HT-IZS, DS-2CD2666G2HT-IZS2U/S(L)(RB), DS-2CD2666G2T-IZS, DS-2CD2667G2HT-LIZS, DS-2CD2667G2T-LZS, DS-2CD2668G2T-LIZS(US), DS-2CD2683G2-IZS, DS-2CD2683G2-LIZS2U, DS-2CD2683G2-LIZS2U/SL, DS-2CD2685G0-IZS, DS-2CD2685G1-IZS, DS-2CD2686G2-IZS, DS-2CD2686G2-IZSU/SL, DS-2CD2686G2HT-IZS, DS-2CD2686G2HT-IZS2U/S(L)(RB), DS-2CD2686G2T-IZS, DS-2CD2686G2T-IZSY, DS-2CD2687G2HT-LIZS, DS-2CD2687G2T-LZS, DS-2CD2688G2T-LIZS(US), DS-2CD2721G0-I(Z)(S), DS-2CD2723G2-IZS, DS-2CD2723G2-LIZS2U, DS-2CD2725FWD-IZS, DS-2CD2726G2-IZS, DS-2CD2726G2T-IZS, DS-2CD2743G0-IZS, DS-2CD2743G1-IZ(S), DS-2CD2743G2-IZS, DS-2CD2743G2-LIZS2U, DS-2CD2745FWD-IZS, DS-2CD2745G1-IZS, DS-2CD2746G2-IZS, DS-2CD2746G2HT-IZS, DS-2CD2746G2T-IZS, DS-2CD2747G2-LZS, DS-2CD2747G2HT-LIZS, DS-2CD2747G2T-LZS, DS-2CD2748G2T-LZS(US), DS-2CD2763G2-IZS, DS-2CD2763G2-LIZS2U, DS-2CD2765G0-IZS, DS-2CD2765G1-IZS, DS-2CD2766G2-IZS, DS-2CD2766G2HT-IZS, DS-2CD2766G2T-IZS, DS-2CD2767G2HT-LIZS, DS-2CD2767G2T-LZS, DS-2CD2768G2T-LIZS(US), DS-2CD2783G2-IZS, DS-2CD2783G2-LIZS2U, DS-2CD2785G0-IZS, DS-2CD2786G2-IZS, DS-2CD2786G2HT-IZS, DS-2CD2786G2T-IZS, DS-2CD2787G2HT-LIZS, DS-2CD2787G2T-LZS, DS-2CD2788G2T-LIZS(US), DS-2CD2821G0, DS-2CD2935FWD-I, DS-2CD2955FWD-I, DS-2CD2955G0-IS(U), DS-2CD2E23G2-U, DS-2CD2E43G2-U, DS-2CD2H23G2-IZS, DS-2CD2H23G2-LIZS2U, DS-2CD2H26G2-IZS, DS-2CD2H26G2T-IZS, DS-2CD2H43G2-IZS, DS-2CD2H43G2-LIZS2U, DS-2CD2H45FWD-IZS, DS-2CD2H45G1-IZS, DS-2CD2H46G2-IZS, DS-2CD2H46G2T-IZS, DS-2CD2H63G2-IZS, DS-2CD2H63G2-LIZS2U, DS-2CD2H66G2-IZS, DS-2CD2H66G2T-IZS, DS-2CD2H83G2-IZS, DS-2CD2H83G2-LIZS2U, DS-2CD2H86G2-IZS, DS-2CD2H86G2T-IZS, DS-2CD2T23G2-2I/4I, DS-2CD2T23G2-2LI(2U)/4LI(2U), DS-2CD2T25FHWI-15/I8, DS-2CD2T24G2-2I/4I, DS-2CD2T26G2-2I/4I, DS-2CD2T26G2-15/I8, DS-2CD2T27G2-L, DS-2CD2T43G0-15/I8, DS-2CD2T43G2-2I/4I, DS-2CD2T43G2-2LI(2U)/4LI(2U), DS-2CD2T45FWD-15/I8, DS-2CD2T45G0P-I, DS-2CD2T46G2-2I/4I, DS-2CD2T46G2-4IY, DS-2CD2T46G2-15/I8, DS-2CD2T46G2H-2I/4I, DS-2CD2T46G2H-IS2U/S(L)(RB), DS-2CD2T46G2P-15/I8, DS-2CD2T47G1-L, DS-2CD2T47G2-L, DS-2CD2T47G2-LSU/SL, DS-2CD2T47G2H-LI, DS-2CD2T47G2H-LISU/SL, DS-2CD2T47G2P-15/I8, DS-2CD2T48G2-LIK(US), DS-2CD2T63G2-2I/4I, DS-2CD2T63G2-2LI(2U)/4LI(2U), DS-2CD2T65G1-15/I8, DS-2CD2T66G2-2I/4I, DS-2CD2T66G2-15/I8, DS-2CD2T66G2H-2I/4I, DS-2CD2T66G2H-IS2U/S(L)(RB), DS-2CD2T67G2-L, DS-2CD2T67G2H-LI, DS-2CD2T67G2H-LISU/SL, DS-2CD2T67G2P-15/I8, DS-2CD2T83G2-2I/4I, DS-2CD2T83G2-2LI(2U)/4LI(2U), DS-2CD2T85G1-15/I8, DS-2CD2T86G2-2I/4I, DS-2CD2T86G2-4IY, DS-2CD2T86G2-15/I8, DS-2CD2T86G2H-2I/4I, DS-2CD2T86G2H-IS2U/S(L)(RB), DS-2CD2T87G2-L, DS-2CD2T87G2-LK, DS-2CD2T87G2-LSU/SL, DS-2CD2T87G2H-LI, DS-2CD2T87G2H-LISU/SL, DS-2CD2T87G2P-LSU/SL

## 2.3 Terms And Definitions

### 2.3.1 Full Power Consumption Mode

Device operates in full consumption mode.

### 2.3.2 Low Power Consumption Mode

For multi-chip devices, only the main chip is powered on; for single-chip devices, the video image quality, frame rate, bit rate, etc. is not affected; 4G/5G chips work normally.

### 2.3.3 Real-Time Low Power Consumption Mode

For multi-chip devices, only the main chip is powered on; for single-chip devices, the video image quality, frame rate, bit rate, etc. might be affected; 4G/5G chips work in low power consumption keepalive mode (the device sends a heartbeat packet every 5 minutes).

### **2.3.4 Standby Mode**

If no operation is performed on the device within 5 minutes or the device battery is lower than 20%, the device will go to sleep.

### **2.3.5 Event**

Event refers to the information uploaded by the device actively. Event needs to be uploaded by the device in real time for the immediate response from the platform. If the device is offline, the event can be stored in the cache first and then be uploaded again when the connection is restored.

### **2.3.6 Arming**

Arming means that the client establishes connection with the device so that events can be uploaded to the client via the connection. The client can subscribe to some event types, and the device will upload the specified events only, otherwise the device will upload all types of events to the client.

### **2.3.7 Metadata**

In this document, metadata refers to the private intelligent structured information.

### **2.3.8 Motion Detection**

It can realize image changes recognition and moving objects detection of a specified area by camera. For example, if a person passes by or a camera is moved, the motion will be recognized and then an alarm will be uploaded for notifications, and the event can trigger recording and alarm output.

### **2.3.9 Traffic Enforcement Event Detection**

The traffic enforcement event detection supports the function of violation enforcement, which can automatically detect and capture the evidence of events, including illegal parking, wrong-way driving, illegal U-turning, driving on the lane line, illegal lane change, and motor vehicle on non-motor vehicle lane, on the roads with no more than four lanes. The enforcement results will be uploaded as alarms in real time. The difference between traffic incident detection and traffic enforcement event detection is that the former only detects events which are not always illegal (e.g., congestion, roadblock, etc.) and the latter detects events of violating the traffic rules. For the traffic enforcement event detection (e.g., illegal parking, wrong-way driving, illegal U-turning, driving on the lane line, illegal lane change, etc.), the detailed violation information needs to be recorded, including the background picture, close-up picture, vehicle picture, and license plate number.

### **2.3.10 Traffic Incident Detection**

AID (Automatic (Traffic) Incident Detection) is an intelligent traffic function for detecting traffic incidents that occurred on the roads with no more than four lanes and triggering alarms in real time. Supported incidents include thrown object detection, pedestrian detection, congestion detection, roadblock detection, traffic accident detection, fog detection, etc. AID is also known as traffic incidents of vehicles. The difference between traffic incident detection and traffic enforcement event detection is that the former only detects events which are not always illegal (e.g., congestion, roadblock, etc.) and the latter detects events of traffic rule violation. For the traffic enforcement event detection (e.g., illegal parking, wrong-way driving, illegal U-turning, driving on the lane line, illegal lane change, etc.), the detailed violation information needs to be recorded, including the background picture, close-up picture, vehicle picture, and license plate number.

## **2.4 Symbols And Acronyms**

HEOP: Embedded Open Platform

HTTP: Hypertext Transfer Protocol.

SDP (Session Description Protocol)

RTSP (Real Time Streaming Protocol)

OSD: On-screen display

VMD: Motion Detection

ANPR: Automatic Number Plate Recognition

TFS: Traffic Enforcement Event Detection (TFS is the abbreviation for Traffic Forensics System)

AID: Automatic (Traffic) Incident Detection

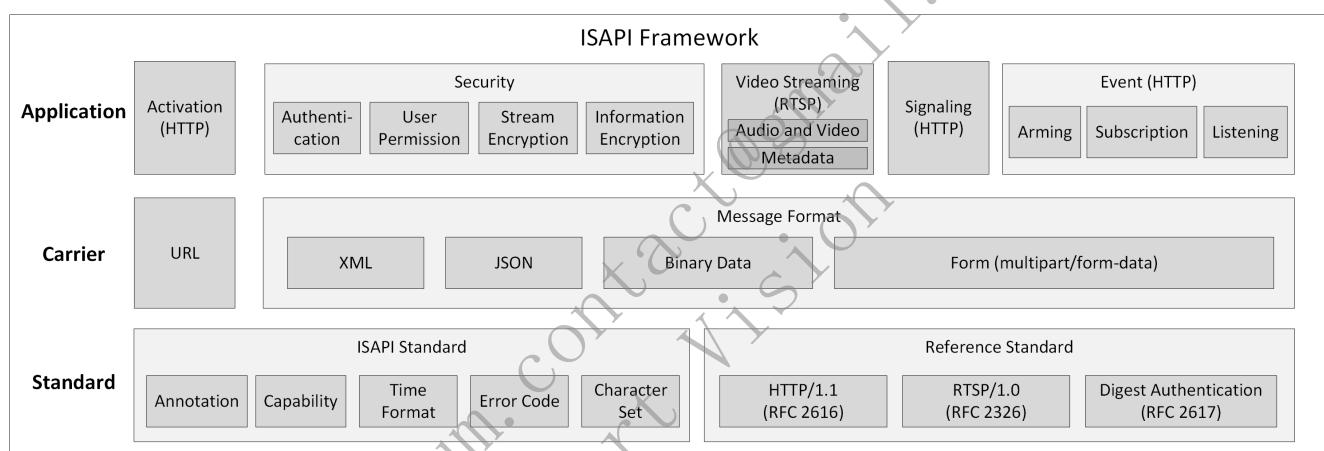
PIR: Passive Infrared Detector.

## 2.5 Update History

No update record

# 3 ISAPI Framework

## 3.1 Overview



\*\*Notes\*\*:

In general, ISAPI refers to the communication protocol based on the HTTP standard. As ISAPI is usually used along with RTSP (Real-Time Streaming Protocol), the RTSP standard is brought into the ISAPI system.

The metadata scheme for transmitting additional information of the stream is extended based on the RTSP standard to transmit the video stream and the structured intelligent information of the stream simultaneously. It is compatible with the RTSP standard.

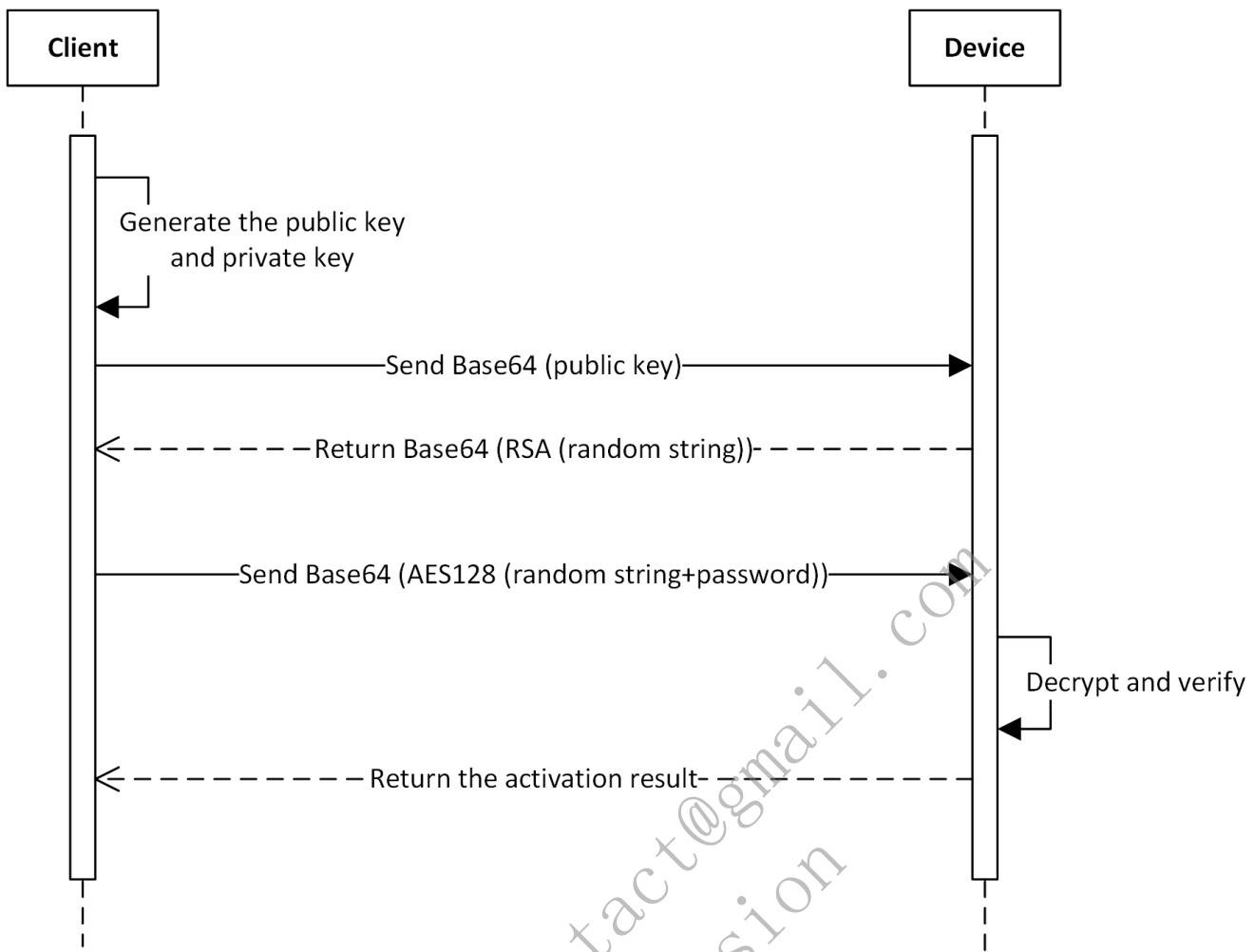
## 3.2 Activation

The purpose of activation is to ensure that the user can set the password for the device and the password meets the security requirement. After the device is activated, you can use the related functions.

ISAPI is a communication protocol running on the application layer. When activating the device via ISAPI, you should know the device's IP address and make sure that the device is connected to the client.

The web application built in the device supports activating the device via ISAPI. When you enter the device's IP address in the address bar of the web browser on the PC, you can activate the device according to the activation guide.

If you want to activate the device on your own application, you need to integrate the activation function via ISAPI. The API calling flow and related APIs are shown below.



Firstly, two operations are defined:

- bytesToHexString: it is used to convert a byte array (the length is N) to a hexadecimal string (the length is 2N). For example, `127,10,23` -> `7f0a17`
  - hexStringToBytes: it is used to convert a hexadecimal string (the length is 2N) to a byte array (the length is N). For example, `7f0a17` -> `127,10,23`
1. The client generates a public and private key pair (1024 bits), and gets the 128-byte modulus in the public key (hereinafter referred to as public key modulus). If the length is longer than 128, the leading 0 needs to be removed.
  2. The client converts the public key modulus to a 256-byte public key string via bytesToHexString and sends the public key string to the device in XML message (related URI: `POST /ISAPI/Security/challenge`) after being encoded by Base64.
  3. The device parses the request to obtain a 256-byte public key string decoded by Base64 and converts it to a 128-byte public key modulus via hexStringToBytes. The complete public key is the combination of obtained public key modulus and public exponent (the default value is '`010001`').
  4. The device generates a 32-byte hexadecimal random string, calls the RSA API to encrypt the random string with the private key, converts the encrypted data to a string via bytesToHexString, encodes the string by Base64, and then sends it to the client.
  5. The client decodes the string from the device by Base64, converts it via hexStringToBytes to get the encrypted data, decrypts the encrypted data with the private key via RSA to obtain a 32-byte hexadecimal random string, converts the obtained string via hexStringToBytes to get a 16-byte AES key. Then the client uses the AES key to encrypt the "string consisting of the first 16 characters of the random string and the real password" by AES128 ECB mode (with zero-padding method) to get a ciphertext, and converts the ciphertext via bytesToHexString, encodes it by Base64, and sends it to the device in XML message (related URI: `PUT /ISAPI/System/activate`).
- Note: If the first 16 characters of the random string are `aaaabbcccccddd` and the real password is `Abc12345`, the data before encryption is `aaaabbcccccdddAbc12345`. This can ensure that the client uses the random string as the key for encryption.
6. The device decodes the string by Base64, converts it via hexStringToBytes to get the ciphertext, uses the AES key to decrypt the ciphertext by AES128 ECB mode, and gets the real password via removing the first 16 characters.

7. The device verifies the password and returns the activation result.

#### Notes:

- You can get the device's activation status by calling the URI `GET /SDK/activateStatus` which requires no authentication.
- Devices also support to be activated via SADP (Search Active Device Protocol) which is based on the communication protocol of the data link layer. With SADP, you do not have to know the IP address of the device but need to ensure that the device and the application running SADP are connected to the same router. SADP also supports discovering devices in the LAN, changing the password of the devices, and so on. The HCSadpSDK is provided for SADP integration, including the developer guide, plug-in, and sample demo which can be used as a simple SADP tool.

## 3.3 Security Mechanism

### 3.3.1 Authentication

When the client applications send requests to devices, they need to use digest authentication (see details in [RFC 7616](#)) for identity authentication.

Currently, all mainstream request class libraries of HTTP have encapsulated digest authentication. See details in [Authentication](#) of Quick Start Guide.

### 3.3.2 User Permission

There are three kinds of users with different permissions for access control and management.

**Administrator:** Has the permission to access all supported resources and should keep activated all the time. It is also known as "admin".

**Operator:** Has the permission to access general resources and a part of advanced resources.

**Normal User:** Only has the permission to access general resources.

### 3.3.3 Information Encryption

During ISAPI integration, the HTTPS service of devices is enabled by default. When the client applications communicate with devices via HTTPS, the information can be transmitted securely.

## 3.4 Video Streaming

### 3.4.1 Audio and Video Stream

ISAPI supports getting and setting stream media parameters of the device, such as video resolution, encoding format, and stream.

Cameras support standard RTSP (Real-Time Streaming Protocol, see details in [RFC 7826](#)). Client applications can get the stream from devices via RTSP.

For details about real-time streaming and video playback, refer to [Real-Time Live View](#) and [Playback](#) in Quick Start Guide.

### 3.4.2 Metadata

The metadata is the structured intelligent information generated by intelligent devices. When the client applications get the audio and/or video stream from devices via RTSP, the metadata will be returned by the device at the same time. For example, to display the face target frame, face information, vehicle target frame, license plate number, vehicle information, and other information on the video stream, the client applications can overlay the above information on the video image.

Before using the metadata, you need to enable the metadata function of the device and then get the stream from the device via RTSP. Some devices support subscribing to the metadata by type. For details about the process of integrating the metadata function, refer to [Metadata Management](#).

# 4 Quick Start Guide

## 4.1 Authentication

When the client applications send requests to the devices, they need to use digest authentication (see details in [RFC 7616](#)) for identity authentication.

Client applications only need to call APIs of the class library to implement the digest authentication. The sample code is shown below.

### 4.1.1 C/C++ (libcurl)

```
// #include <curl/curl.h>
// Callback Function
static size_t OnWriteData(void* buffer, size_t size, size_t nmemb, void* lpVoid)
{
    std::string* str = dynamic_cast<std::string*>((std::string *)lpVoid);
    if( NULL == str || NULL == buffer )
    {
        return -1;
    }

    char* pData = (char*)buffer;
    str->append(pData, size * nmemb);
    return nmemb;
}

std::string strUrl = "http://192.168.18.84:80/ISAPI/System/deviceInfo";
std::string strresponseData;
CURL *pCurlHandle = curl_easy_init();
curl_easy_setopt(pCurlHandle, CURLOPT_CUSTOMREQUEST, "GET");
curl_easy_setopt(pCurlHandle, CURLOPT_URL, strUrl.c_str());
// Set the user name and password
curl_easy_setopt(pCurlHandle, CURLOPT_USERPWD, "admin:admin12345");
// Set the authentication method to the digest authentication
curl_easy_setopt(pCurlHandle, CURLOPT_HTTPAUTH, CURLAUTH_DIGEST);
// Set the callback function
curl_easy_setopt(pCurlHandle, CURLOPT_WRITEFUNCTION, OnWriteData);
// Set the parameters of the callback function to get the returned information
curl_easy_setopt(pCurlHandle, CURLOPT_WRITEDATA, &strresponseData);
// Timeout settings for receiving the data. If receiving data is not completed within 5 seconds, the application will exit directly
curl_easy_setopt(pCurlHandle, CURLOPT_TIMEOUT, 5);
// Set the redirection times to avoid too many redirections
curl_easy_setopt(pCurlHandle, CURLOPT_MAXREDIRS, 1);
// Connection timeout duration. If the duration is too short, the client application will be disconnected before the data request sent by the application reaches the device
curl_easy_setopt(pCurlHandle, CURLOPT_CONNECTTIMEOUT, 5);
CURLcode nRet = curl_easy_perform(pCurlHandle);
if (0 == nRet)
{
    // Output the received message
    std::cout << strresponseData << std::endl;
}
curl_easy_cleanup(pCurlHandle);
```

### 4.1.2 C# (WebClient)

```
// using System.Net;
// using System.Net.Security;
try
{
    string strUrl = "http://192.168.18.84:80/ISAPI/System/deviceInfo";
    WebClient client = new WebClient();
    // Set the user name and password
    client.Credentials = new NetworkCredential("admin", "admin12345");
    byte[] responseData = client.DownloadData(strUrl);
    string strresponseData = Encoding.UTF8.GetString(responseData);
    // Output received information
    Console.WriteLine(strresponseData);
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
```

### 4.1.3 Java (HttpClient)

```

// import org.apache.commons.httpclient.HttpClient;
String url = "http://192.168.18.84:80/ISAPI/System/deviceInfo";
HttpClient client = new HttpClient();
// Set the user name and password
UsernamePasswordCredentials creds = new UsernamePasswordCredentials("admin", "admin12345");
client.getState().setCredentials(AuthScope.ANY, creds);
GetMethod method = new GetMethod(url);
method.setDoAuthentication(true);
int statusCode = client.executeMethod(method);
byte[] responseData = method.getResponseBodyAsString().getBytes(method.getResponseCharSet());
String strresponseData = new String(responseData, "utf-8");
method.releaseConnection();
// Output received information
System.out.println(strresponseData);

```

## 4.1.4 Python (requests)

```

# -*- coding: utf-8 -*-

import requests
request_url = 'http://192.168.18.84:80/ISAPI/System/deviceInfo'
# Set the authentication information
auth = requests.auth.HTTPDigestAuth('admin', 'admin12345')
# Send the request and receive response
response = requests.get(request_url, auth=auth)
# Output response content
print(response.text)

```

## 4.2 Message Parsing

### 4.2.1 Message Format

During the process of communication and interaction via ISAPI, the request and response messages are often text data in XML or JSON format. Besides that, the data of firmware packages and configuration files is in binary format. A request can also be in form format with multiple formats of data (multipart/form-data).

#### 4.2.1.1 XML

Generally, the Content-Type in the headers of the HTTP request is application/xml; charset="UTF-8".

Request and response messages in XML format are all encoded with UTF-8 standards in ISAPI.

The namespace http://www.isapi.org/ver20/XMLSchema and ISAPI version number 2.0 of XML messages are configured by default, see the example below.

```

<?xml version="1.0" encoding="UTF-8"?>
<NodeList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <Node>
    <id>1</id>
    <enabled>true</enabled>
    <nodeName>nodeName</nodeName>
    <level>level1</level>
  </Node>
</NodeList>

```

#### 4.2.1.2 JSON

The Content-Type in the headers of the HTTP request is often application/json.

To distinguish between APIs with XML messages and those with JSON messages, ISAPI adds the query parameter format=json to all request URLs with JSON messages, e.g.,

<http://192.168.1.1:80/ISAPI/System/Sensor/thermometrySensor?format=json>. Messages of request URLs without the query parameter format=json are usually in XML format. However, there may be some exceptions, and the message format is subject to the API definition.

Request and response messages in JSON format are all encoded by UTF-8 in ISAPI.

#### 4.2.1.3 Binary Data

For the firmware and configuration files, the Content-Type in the header of an HTTP request is often application/octet-stream.

#### 4.2.1.4 Form (multipart/form-data)

When multiple pieces of data are submitted at the same time in an ISAPI request (e.g., the person information and face picture need to be submitted at the same time when a face record is added to the face picture library), the Content-Type in the header of the corresponding HTTP request is usually multipart/form-data, boundary=AaB03x, where the boundary is a variable used to separate the entire HTTP body into multiple units and each unit is a piece of data with its own headers and body. In Content-Disposition of form unit headers, the name property refers to the form unit name, which is required for all form units; the filename property refers to the file name of form unit body, which is required only when the form unit body is a file. In headers of form units, Content-Length refers to the body length, which starts after CRLF(\r\n) and ends before two hyphens (--) of next form. There should be a CRLF used as the delimiter of two form units before two hyphens (--), and the Content-Length of previous form unit does not include the CRLF length. For the detailed format description, refer to [RFC 1867 \(Form-Based File Upload in HTML\)](#). Pay attention to two hyphens (--) before and after the boundary.

#### Notes

- In RFC specifications, it is strongly recommended to contain the field Content-Length in the entity header, and there is no requirement that the field Content-Length should be contained in the header of each form element. The absence of field Content-Length in the header should be considered when the client and device programs parse the form data.
- To avoid the conflict between message content and boundary value, it is recommended to use a longer and more complex string as the boundary value.

The example of ISAPI form data submitted by a client to a device is as follows.

```
POST /ISAPI/Intelligent/FDLib/pictureUpload
Content-Type: multipart/form-data; boundary=e5c2f8c5461142aea117791dade6414d
Content-Length: 56789

--e5c2f8c5461142aea117791dade6414d
Content-Disposition: form-data; name="PictureUploadData";
Content-Type: application/xml
Content-Length: 1234

<PictureUploadData/>
--e5c2f8c5461142aea117791dade6414d
Content-Disposition: form-data; name="face_picture"; filename="face_picture.jpg";
Content-Type: image/jpeg
Content-Length: 34567

Picture Data
--e5c2f8c5461142aea117791dade6414d--
```

The example of ISAPI form data responded from a device to a client is as follows.

In ISAPI messages, when there are multiple form units, three nodes (pid, contentid, and filename) are used for linking form units. The corresponding relations are as follows:

Node Name	Form Field	Description
pid	name	pid in XML/JSON messages corresponds to the name property of Content-Disposition in form headers.
contentid	Content-ID	contentid in XML/JSON messages corresponds to Content-ID in form headers.
filename	filename	filename in XML/JSON messages corresponds to filename property of Content-Disposition in form headers.

```

HTTP/1.1 200 OK
Content-Type: multipart/form-data; boundary=136a73438ecc4618834b999409d05bb9
Content-Length: 56789

--136a73438ecc4618834b999409d05bb9
Content-Disposition: form-data; name="mixedTargetDetection";
Content-Type: application/json
Content-Length: 811

{
    "ipAddress": "172.6.64.7",
    "macAddress": "01:17:24:45:D9:F4",
    "channelID": 1,
    "dateTime": "2009-11-14T15:27+08:00",
    "eventType": "mixedTargetDetection",
    "eventDescription": "Mixed target detection",
    "deviceID": "123456789",
    "CaptureResult": [
        {
            "targetID": 1,
            "Human": {
                "Rect": {
                    "height": 1.0,
                    "width": 1.0,
                    "x": 0.0,
                    "y": 0.0
                },
                "contentID1": "humanImage", /*human body thumbnail*/
                "contentID2": "humanBackgroundImage", /*human body background picture*/
                "pId1": "9d48a26f7b8b4f2390c16808f93f3534", /*human body thumbnail ID */
                "pId2": "5EE7078E07BB47CF860DE8E4E9A85F28" /*ID of human body background picture*/
            }
        }
    ]
}
--136a73438ecc4618834b999409d05bb9
Content-Disposition: form-data; name="9d48a26f7b8b4f2390c16808f93f3534"; filename="humanImage.jpg";
Content-Type: image/jpeg
Content-Length: 34567
Content-ID: humanImage

Picture Data
--136a73438ecc4618834b999409d05bb9-
Content-Disposition: form-data; name="5EE7078E07BB47CF860DE8E4E9A85F28"; filename="humanBackgroundImage.jpg";
Content-Type: image/jpeg
Content-Length: 345678
Content-ID: humanBackgroundImage

Picture Data
--136a73438ecc4618834b999409d05bb9--

```

## 4.2.2 Annotation

The field descriptions of ISAPI request and response messages are marked as annotations in the example messages as shown below.

```

<?xml version="1.0" encoding="UTF-8"?>

<NodeList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, node List, attr:version{req, string, version No., range:[,]}-->
    <Node>
        <!--ro, opt, object, node information-->
        <id>
            <!--ro, req, int, node No., range:[,], step:, unit:, unitType:-->1
        </id>
        <enabled>
            <!--ro, opt, bool, whether to enable-->true
        </enabled>
        <nodeName>
            <!--ro, req, string, node name, range:[1,32]-->test
        </nodeName>
        <level>
            <!--ro, opt, enum, Level, subType:string,
            [Level1#Level 1,Level2#Level 2,Level3#Level 3]-->level1
        </level>
    </Node>
</NodeList>

```

```

{
    "name": "test",
    /*ro, req, string, name, range:[1,32]*/
    "type": "type1",
    /*ro, req, enum, type, subType:string, [type1#type 1,type2#type 2]*/
    "enabled": true,
    /*ro, opt, bool, enable or not, desc:xxxxxxxx*/
    "NodeList": {
        /*opt, object, node list, dep:and,{$.enabled,eq,true}*/
        "scene": 1,
        /*req, enum, scene, subType:int, [1#scene 1; 2#scene 2; 3#scene 3]*/
        "ID": 1
        /*req, int, No., range:[1,8], step:, unit:, unitType:*/
    }
}

```

Key annotations are shown in the table below.

Annotation	Description	Remark
ro	Attribute: Read-Only	This field can only be obtained and cannot be edited.
wo	Attribute: Write-Only	This field can only be edited and cannot be obtained.
req	Attribute: Required	This field is required for request messages sent to the device and response messages returned from the device.
opt	Attribute: Optional	This field is optional for request messages sent to the device and response messages returned from the device.
dep	Attribute: Dependent	This field is valid and required when specific conditions are satisfied.
object	Field Type: Object	The field of type object contains multiple sub-fields.
list	Field Type: List	The subType following it refers to the data type of sub-items in the list.
subType	Field Type: String	The range following it refers to the maximum and the minimum string size of the field.
int	Field Type: Int	The range following it refers to the maximum and the minimum value of the field.
float	Field Type: Float	The range following it refers to the maximum and the minimum value of the field.
bool	Field Type: Boolean	The value can be true or false.
enum	Field Type: Enumeration	The subType following it indicates that the enumerators are of type string or int. The [] following the subType contains all enumerators.
subType	Sub-Type of Field	When the type of field is list or enum, the value of subType is the data type of each sub-object.
desc	Field Description	The detailed description of the field.

### 4.2.3 Capability Set

ISAPI has designed capability sets for almost all functions, APIs, and fields. URLs for getting the capability set end with `/capabilities`. Some URLs may contain query parameters in the format: `/capabilities?format=json&type=xxx`.

There are two types of fields in the capability message of ISAPI: whether the device supports a function and the value range of a field in an API.

**Whether the device supports a function:** it is often in the format `isSupportXXXXXXXX`, which indicates that whether the device supports a function and a set of APIs for implementing this function.

The capability message example in JSON format is shown below:

```
{
    "isSupportMap": true,
    /*ro, opt, bool, whether it supports the e-map function, desc:/ISAPI/SDT/Management/map/capabilities?format=json*/
    "isSupportAlgTrainResourceInfo": true,
    /*ro, opt, bool, whether it supports only getting the resource information of the algorithm training platform,
desc:/ISAPI/SDT/algorithmTraining/ResourceInfo?format=json*/
    "isSupportAlgTrainAuthInfo": true,
    /*ro, opt, bool, whether it supports only getting the authorization information of the algorithm training platform,
desc:/ISAPI/SDT/algorithmTraining/SoftLock/AuthInfo?format=json*/
    "isSupportAlgTrainNodeList": true,
    /*ro, opt, bool, whether it supports only getting the node information of the algorithm training platform, desc:/ISAPI/SDT/algorithmTraining/NodeList?
format=json*/
    "isSupportNAS": true
    /*ro, opt, bool, whether it supports mounting and unmounting NAS, desc:/ISAPI/SDT/Management/NAS/capabilities?format=json*/
}
```

The capability message example in XML format is shown below:

```
<isSupportNetworkStatus>
    <!--ro, opt, bool, whether it supports searching the network status, desc: related API (/ISAPI/System/Network/status?format=json)-->true
</isSupportNetworkStatus>
```

**The value range of the field:** the maximum value, minimum value, the maximum size, the minimum size, options, and so on of each field of the API.

The example of JSON format is shown below:

```
{
    "boolType": {
        /*req, object, example of the capability of type bool*/
        "@opt": [true, false]
        /*req, array, options, subType: bool*/
    },
    "integerType": {
        /*req, object, example of the capability of type integer*/
        "@min": 0,
        /*ro, req, int, the minimum value*/
        "@max": 100
        /*ro, req, int, the maximum value*/
    },
    "stringType": {
        /*req, object, example of the capability of type string*/
        "@min": 0,
        /*ro, req, int, the minimum string size*/
        "@max": 32
        /*ro, req, int, the maximum string size*/
    },
    "enumType": {
        /*req, object, capability example of type enum*/
        "@opt": ["enum1", "enum2", "enum3"]
        /*req, array, options, subType: string*/
    }
}
```

The example of XML format is shown below:

```
<boolType opt="true,false" def="true">
    <!--ro, opt, bool, example of the capability of type bool-->true
</boolType>
<integerType min="0" max="100">
    <!--ro, opt, int, example of the capability of type int-->50
</integerType>
<stringType min="0" max="64">
    <!--ro, opt, string, example of the capability of type string-->test
</stringType>
<enumType opt="red,white,black" def="red">
    <!--ro, opt, string, example of the capability of type enum-->white
</enumType>
```

**Note:** For the same capability set, devices of different models and versions may return different results. The values shown in this document are only examples for reference. The capability set actually returned by the device takes precedence.

#### 4.2.4 Time Format

ISAPI adopts [ISO 8601 Standard Time Format](#), which is the same as [W3C Standard Date and Time Formats](#).

Format: YYYY-MM-DDThh:mm:ss.sTZD

YYYY = the year consisting of four decimal digits  
MM = the month consisting of two decimal digits (01-January, 02-February, and so forth)  
DD = the day consisting of two decimal digits (01 to 31)  
hh = the hour consisting of two decimal digits (00 to 23, a.m. and p.m. are not allowed)  
mm = the minute consisting of two decimal digits (00 to 59)  
ss = the second consisting of two decimal digits (00 to 59)  
s = one or more digits representing the fractional part of a second  
TZD = time zone identifier (Z or +hh:mm or -hh:mm)

**Example:** 2017-08-16T20:17:06.123+08:00 refers to 20:17:06.123 on August 16, 2017 (local time which is 8 hours ahead of UTC). The plus sign (+) indicates that the local time is ahead of UTC, and the minus sign (-) means that the local time is behind UTC.

After the DST is enabled, the local time and time difference will change compared with UTC, and the values of related fields also need to be changed. Disabling the DST will bring into the opposite effect.

**Example:** In 1986, the DST was in effect from May 4 at 2:00 a.m. (GMT+8). During the DST period, the clocks were moved one hour ahead, which means that there was one less hour on that day. When the DST ends at 2:00 a.m. on September 14, 1986, the clocks were moved one hour back and there was an extra hour on that day. The changes of the time are as follows:

- DST Starts: 1986-05-04T02:00:00+08:00 --> 1986-05-04T03:00:00+09:00
- DST Ends: 1986-09-14T02:00:00+09:00 --> 1986-09-14T01:00:00+08:00

#### Notes:

- The time difference cannot be simply used to determine the time zone. Because when the DST starts, the time difference will change and it cannot represent the actual time zone.
- Both TZ (UTC time, e.g., 1986-05-03T18:00:00Z) and TD (local time and time difference, e.g., 1986-05-04T02:00:00+08:00) meet the time format standards of ISO 8601. In ISAPI, the TD format is recommended to be used in messages sent from the user applications and the devices.
- A few old-version devices will return the time in TZ format. For representing the time difference information and forward compatibility, an extra field timeDiff is added as shown in the example below. User applications need to support both TD format and TZ format when parsing the time in the messages returned by devices.

```
{  
    "dateTime": "1986-05-03T18:00:00Z", /*device time. The value in TZ format is the UTC time and the value in TD format is the time difference between the device's local time and UTC*/  
    "timeDiff": "+08:00" /*optional, time difference between the local time and UTC time. If this field does not exist, the user application will convert the dateTime into the local time for use*/  
}
```

## 4.2.5 Character Set

To prevent characters not commonly used from resulting in exceptions in device programs and user applications, ISAPI limits the valid field values of type string to a specific range of characters. Character sets allowed to be used in the fields of type string in ISAPI are listed below.

- Single-byte character set: lowercase letters (a-z), uppercase letters (A-Z), digits (0-9), and special characters (see details in the table below).
- Multi-byte character set: language characters based on Unicode and encoded by UTF-8 (UTF-8 encoding is a transformation format of Unicode character set. For details, refer to [RFC 2044](#)).

No.	Name	Special Character	No.	Name	Special Character
1	Open Parenthesis	(	18	Dollar Sign	\$
2	Close Parenthesis	)	19	Percent Sign	%
3	Plus Sign	+	20	Ampersand	&
4	Comma	,	21	Close Single Quotation Mark	'
5	Minus Sign	-	22	Asterisk	*
6	Period	.	23	Slash	/
7	Semicolon	;	24	Smaller Than	<
8	Equal Sign	=	25	Greater Than	>
9	At Sign	@	26	Question Mark	?
10	Open Square Bracket	[	27	Caret	^
11	Close Square Bracket	]	28	Open Single Quotation Mark	'
12	Underscore	_	29	Vertical Bar	
13	Open Brace	{	30	Tilde	~
14	Close Brace	}	31	Double Quotation Marks	"
15	Space		32	Colon	:
16	Exclamation Mark	!	33	Backslash	\
17	Octothorpe	#			

The valid characters that can be used in some special fields are listed below.

- User name: lowercase letters (a-z), uppercase letters (A-Z), digits (0-9), and characters from No. 1 to No. 30 in the special character table.
- Password: User Name: lowercase letters (a-z), uppercase letters (A-Z), digits (0-9), and characters from No. 1 to No. 33 in the special character table.
- Names displayed on the UI (device name, person name, face picture library name, etc.): lowercase letters (a-z), uppercase letters (A-Z), digits (0-9), characters from No. 1 to No. 15 in the special character table, and multi-byte characters.
- Normal fields of type string support lowercase letters (a-z), uppercase letters (A-Z), digits (0-9), characters from No. 1 to No. 15 in the special character table, and multi-byte characters by default.

#### 4.2.6 Error Processing

When requesting via ISAPI failed (the HTTP status code is not 200), the device will return the HTTP status code and ISAPI error code. For HTTP status codes, refer to 10 Status Code Definitions in [RFC 2616](#). For ISAPI error codes, refer to Error Code Dictionary.

Message Example:

```
HTTP/1.1 403 Forbidden
Content-Type: application/json; charset=UTF-8
Date: Thu, 15 Jul 2021 20:43:30 GMT
Content-Length: 229
Connection: Keep-Alive
```

```
{
  "requestURL": "/ISAPI/Event/triggers/notifications/channels/whiteLightAlarm",
  "statusCode": 4,
  "statusString": "Invalid Operation",
  "subStatusCode": "notSupport",
  "errorCode": 1073741825,
  "errorMsg": "notSupport"
}
```

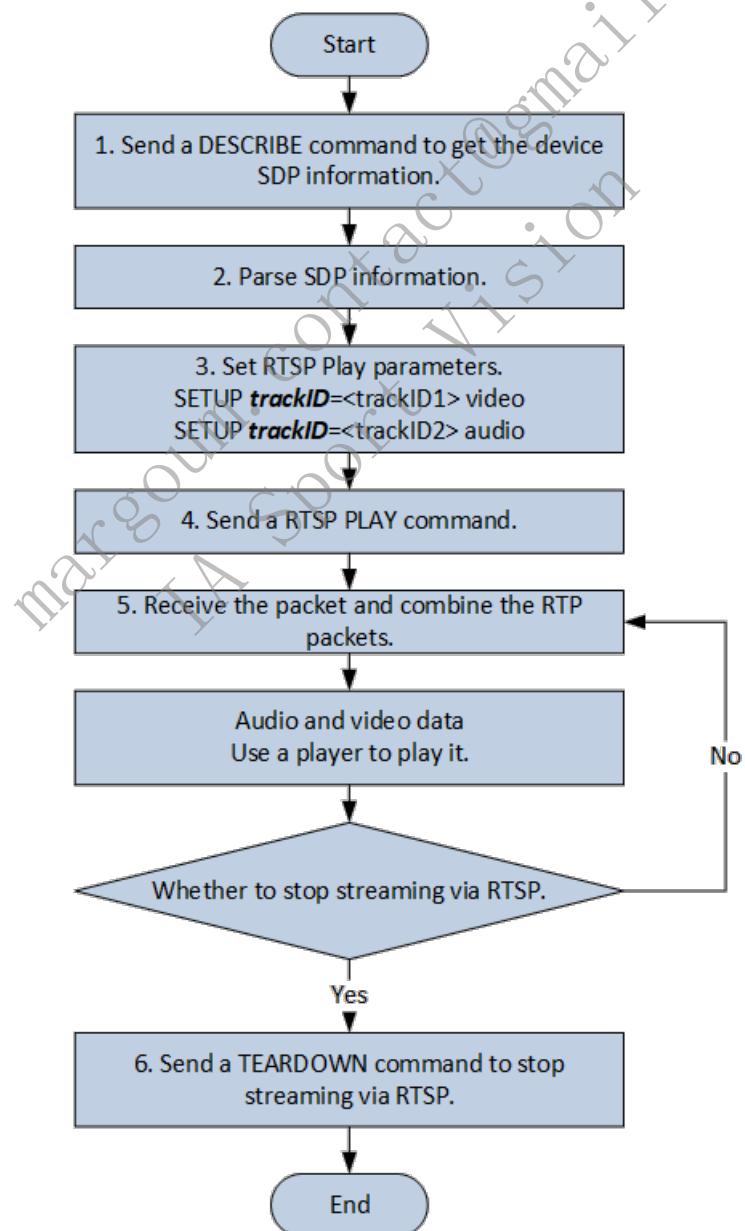
## 4.3 Real-Time Live View

### 4.3.1 Introduction to the Function

Supports getting and setting stream media parameters of devices such as resolution, coding format, and stream type.

Supports streaming from products via RTSP (Real Time Streaming Protocol, see details in [RFC 7826](#)).

### 4.3.2 API Calling Flow



1. A client sends RTSP DESCRIBE commands such as `DESCRIBE /ISAPI/Streaming/channels/101 RTSP/1.0`. Digest authentication with devices is required before this step. 2. The client parses the media SDP information returned by the device. 3. Set RTSP play parameters, that is to set the track ID parsed from SDP information via SETUP commands. For

example, trackID=1 indicates videos while trackID=2 indicates audios. 4. The client sends an RTSP PLAY command, and the device will send audio stream, video stream, and metadata in the format of `PLAY /ISAPI/Streaming/channels/101 RTSP/1.0` . 5. The client receives the RTP packet sent by the device. Divided RTP packets should be assembled on the client before being parsed. 6. The client sends the command RTSP TEARDOWN to stop streaming.

**Notes:**

- Digest authentication is required in RTSP playback. The method is the same as that of ISAPI digest authentication.
- The address format for streaming from devices is `rtsp:// <host>[:port]/ISAPI/Streaming/channels/<ID>`, of which `<host>` is the device IP address; `[:port]` is optional, and 554 by default; `<ID>` is the device channel ID \* 100 + stream type (1-main stream, 2-sub-stream, 3-third stream). For example, the IP address of the target device is `172.7.203.11`, and the streaming address of main stream for channel 17 will be `rtsp://172.7.203.11:554/ISAPI/Streaming/channels/1701`.
- RTSP also supports containing user names and passwords in URL. The format is `rtsp://username:password@[address]:[port]/Streaming/Channels/[id](?parm1=value1&parm2-=value2...)`, such as `/Streaming/Channels/101?transportmode=unicast`.

### 4.3.3 Example

1. A client sends an RTSP DESCRIBE command.

```
DESCRIBE rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101 RTSP/1.0
CSeq:0
Accept:application/sdp
User-Agent:NKPlayer-1.00.00.081112
```

2. Server responds that authentication is required.

```
RTSP/1.0 401 Unauthorized
CSeq: 0
WWW-Authenticate: Digest realm="3521781c29acb312330dd668", nonce="026019333", algorithm="MD5"
```

3. The client sends an RTSP DESCRIBE request with authentication information again.

```
DESCRIBE rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101 RTSP/1.0
CSeq:1
Accept:application/sdp
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101",
response="76a29c5b8edb49838013cf1cf27941"
User-Agent:NKPlayer-1.00.00.081112
```

4. The device responds to SDP information.

5. The client sends RTSP SETUP requests, and the server responds to them.

```
SETUP rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101/trackID=1 RTSP/1.0
CSeq:2
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="02601933", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101",
response="ff343f5ff82de028dd9b4932cc44201"
Transport:RTP/AVP/TCP;unicast;interleaved=0-1:ssrc=0
User-Agent:NikPlayer-1.00.00.081112
```

RTSP/1.0 200 OK  
Session: 1127293610;timeout=60  
Transport: RTP/AVP/TCP;unicast;interleaved=0-1:ssrc=433122aa  
CSeq: 2  
Accept-Ranges: NPT  
Media-Properties: No-Seeking, Time-Progressing, Time-Duration=0  
Date: Tue, 17 Nov 2020 02:09:45 GMT

```
SETUP rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101/trackID=2 RTSP/1.0
CSeq:3
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101",
response="ff343f5ff82deb028dd9b4932cc44201"
Session:1127293610;timeout=60
Transport:RTP/AVP/TCP;unicast;interleaved=2-3:ssrc=0
User-Agent:NikPlayer-1.00.00.081112
```

```
RTSP/1.0 200 OK
Session: 1127293610;timeout=60
Transport: RTP/AVP/TCP;unicast;interleaved=2-3:ssrc=433122ab
CSeq: 3
Accept-Ranges: NPT
Media-Properties: No-Seeking, Time-Progressing, Time-Duration=0
Date: Tue, 17 Nov 2020 02:09:45 GMT
```

6. The client sends an RTSP PLAY request.

PLAY rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101 RTSP/1.0  
CSeq:4  
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101", response="24edf8a6ff3ef767f7c49d1c847200bd"  
Session:1127293610;timeout=60  
Range:npt=0.000000-0.000000  
User-Agent:NKPlayer-1.00.00.081112

7. The server sends audio and video stream data.

```

RTSP/1.0 200 OK
Session: 1127293610
CSeq: 4
Date: Tue, 17 Nov 2020 02:09:45 GMT
$. ....d1.w....c....".T....g....).i.....a....7.S..~J.....X....X.

```

8. The client sends an RTSP TEARDOWN request, and the server responds to it.

```

TEARDOWN rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101 RTSP/1.0
CSeq:5
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101",
response="24edf8a6ff3ef767f7c49d1c847200bd"
Session:1127293610;timeout=60
Range:npt=0.00000-0.00000
User-Agent:NKPlayer-1.00.00.081112

```

```

RTSP/1.0 200 OK
Session: 1127293610
CSeq: 5
Date: Tue, 17 Nov 2020 02:09:50 GMT

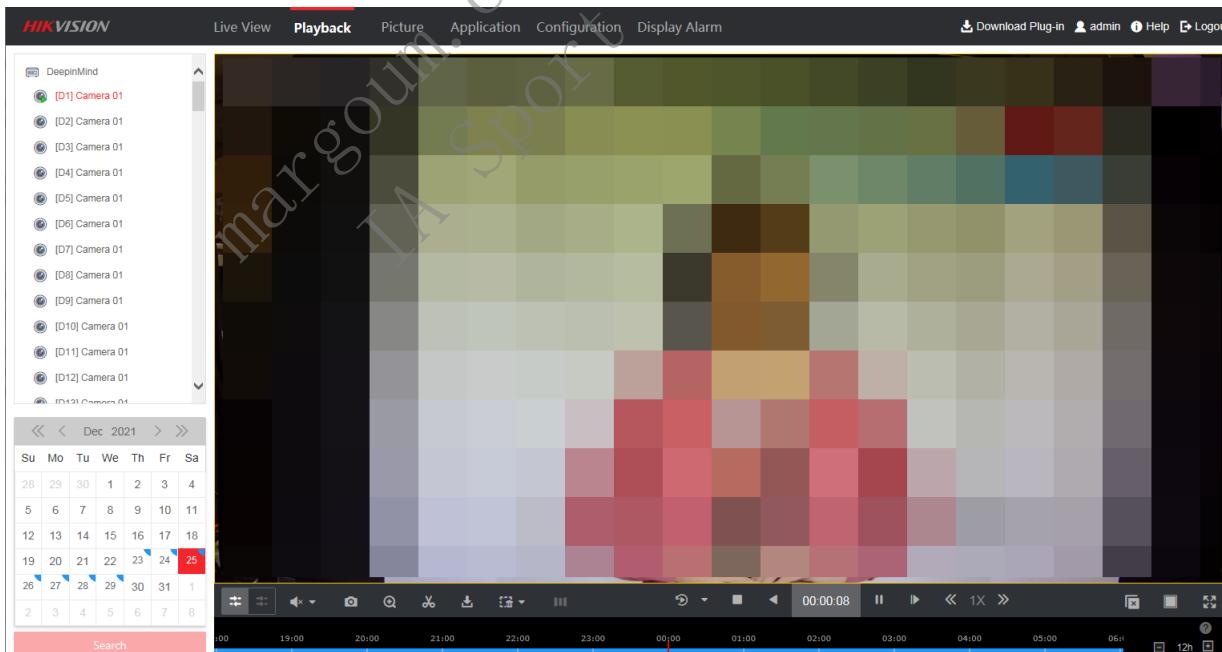
```

## 4.4 Playback

### 4.4.1 Introduction to the Function

Devices such as cameras and NVRs can store videos. Storage mediums such as SD card, TF card and HDDs are needed before recording videos. The process of playback starts with searching for footages and then getting video stream via RTSP (Real Time Streaming Protocol, see details in [RFC 7826](#)).

Below is an example of playback on a web client. The calendar in the bottom-left corner shows the results of searching for videos. Dates with videos are shown on the calendar and you can click a date to play back searched videos of the corresponding day.



### #### 4.4.2 API Calling Flow

1. (Optional) Check the calendar to find dates with stored videos: `POST /ISAPI/ContentMgmt/record/tracks/<trackStreamID>/dailyDistribution`. `<trackStreamID>` = channel number \* 100 + stream type (1-main stream, 2-sub-stream, 3-third stream). For example, The rackStreamID of the main stream for channel 17 is 1701.
2. Searching for videos by parsing the `playbackURI` of the device to get the RTSP address `POST /ISAPI/ContentMgmt/search`.
3. Play videos back via RTSP. Steps of playback via RTSP is the same as that of real-time live view. Refer to streaming via RTSP in real-time streaming. See details in [RFC 7826](#) (Real Time Streaming Protocol).

**Notes:**

- Playback via RTSP needs digest authentication. The method is the same as the one of ISAPI digest authentication.
- Send PAUSE command to pause playback. Send PLAY command to continue. If you want to perform fast forward and slow forward, you can send a PLAY command and set Scale in headers. See details in 10.6 PAUSE and 12.34 Scale in [RFC 7826](#) (Real Time Streaming Protocol).

### 4.4.3 Example

#### 4.4.3.1 Search for Videos According to Calendar

##### Request from Client

```
POST /ISAPI/ContentMgmt/record/tracks/101/dailyDistribution HTTP/1.1
Host: 10.14.97.40
Connection: keep-alive
Content-Length: 119
Cache-Control: max-age=0
Accept: */*
X-Requested-With: XMLHttpRequest
If-Modified-Since: 0
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://10.14.97.40
Referer: http://10.14.97.40/doc/index.html
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: WebSession_78aedfcc66=3b451c9d37cb637827da0815086e7ecfd9984b0095b3cb7198e8197a424a3279

<?xml version="1.0" encoding="utf-8"?>
<trackDailyParam>
    <year>2021</year>
    <monthOfYear>08</monthOfYear>
</trackDailyParam>
```

##### Response from Device

HTTP/1.1 200 OK

Vary: Accept-Encoding

X-Frame-Options: SAMEORIGIN

Content-Type: application/xml; charset="UTF-8"

X-Content-Type-Options: nosniff

Date: Wed, 18 Aug 2021 15:47:43 GMT

Content-Length: 2915

X-XSS-Protection: 1; mode=block

Connection: Keep-Alive

Accept-Ranges: bytes

```
<?xml version="1.0" encoding="UTF-8" ?>
<trackDailyDistribution version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <dayList>
        <day><id>1</id><dayOfMonth>1</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>2</id><dayOfMonth>2</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>3</id><dayOfMonth>3</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>4</id><dayOfMonth>4</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>5</id><dayOfMonth>5</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>6</id><dayOfMonth>6</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>7</id><dayOfMonth>7</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>8</id><dayOfMonth>8</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>9</id><dayOfMonth>9</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>10</id><dayOfMonth>10</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>11</id><dayOfMonth>11</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>12</id><dayOfMonth>12</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>13</id><dayOfMonth>13</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>14</id><dayOfMonth>14</dayOfMonth><record>false</record></day>
        <day><id>15</id><dayOfMonth>15</dayOfMonth><record>false</record></day>
        <day><id>16</id><dayOfMonth>16</dayOfMonth><record>false</record></day>
        <day><id>17</id><dayOfMonth>17</dayOfMonth><record>false</record></day>
        <day><id>18</id><dayOfMonth>18</dayOfMonth><record>true</record><recordType>time</recordType></day>
        <day><id>19</id><dayOfMonth>19</dayOfMonth><record>false</record></day>
        <day><id>20</id><dayOfMonth>20</dayOfMonth><record>false</record></day>
        <day><id>21</id><dayOfMonth>21</dayOfMonth><record>false</record></day>
        <day><id>22</id><dayOfMonth>22</dayOfMonth><record>false</record></day>
        <day><id>23</id><dayOfMonth>23</dayOfMonth><record>false</record></day>
        <day><id>24</id><dayOfMonth>24</dayOfMonth><record>false</record></day>
        <day><id>25</id><dayOfMonth>25</dayOfMonth><record>false</record></day>
        <day><id>26</id><dayOfMonth>26</dayOfMonth><record>false</record></day>
        <day><id>27</id><dayOfMonth>27</dayOfMonth><record>false</record></day>
        <day><id>28</id><dayOfMonth>28</dayOfMonth><record>false</record></day>
        <day><id>29</id><dayOfMonth>29</dayOfMonth><record>false</record></day>
        <day><id>30</id><dayOfMonth>30</dayOfMonth><record>false</record></day>
        <day><id>31</id><dayOfMonth>31</dayOfMonth><record>false</record></day>
    </dayList>
</trackDailyDistribution>
```

#### 4.4.3.2 Search for Videos

##### Request from Client

```
POST /ISAPI/ContentMgmt/search HTTP/1.1
Host: 10.14.97.40
Connection: keep-alive
Content-Length: 486
Cache-Control: max-age=0
Accept: /*
X-Requested-With: XMLHttpRequest
If-Modified-Since: 0
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://10.14.97.40
Referer: http://10.14.97.40/doc/index.html
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: WebSession_78aedfcc66=3b451c9d37cb637827da0815086e7ecfd9984b0095b3cb7198e8197a424a3279

<?xml version="1.0" encoding="utf-8"?>
<CMSearchDescription>
    <searchID>88C2D4D-D3FA-4AD4-BD80-555C182050CC</searchID>
    <trackList>
        <trackID>101</trackID>
    </trackList>
    <timeSpanList>
        <timeSpan>
            <startTime>2021-08-16T00:00:00Z</startTime>
            <endTime>2021-08-18T23:59:59Z</endTime>
        </timeSpan>
    </timeSpanList>
    <maxResults>100</maxResults>
    <searchResultPosition>0</searchResultPosition>
    <metadataList>
        <metadataDescriptor>//recordType.meta.std-cgi.com</metadataDescriptor>
    </metadataList>
</CMSearchDescription>
```

## Response from Device

```
HTTP/1.1 200 OK
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
Content-Type: application/xml; charset="UTF-8"
X-Content-Type-Options: nosniff
Date: Wed, 18 Aug 2021 15:19:13 GMT
Content-Length: 1021
X-XSS-Protection: 1; mode=block
Connection: Keep-Alive
Accept-Ranges: bytes

<?xml version="1.0" encoding="UTF-8" ?>
<CMSearchResult version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <searchID>{88c2cd4d-d3fa-4ad4-bd80-555c18205dcc}</searchID>
    <responseStatus>true</responseStatus>
    <responseStatusStrg>OK</responseStatusStrg>
    <numOfMatches>1</numOfMatches>
    <matchList>
        <searchMatchItem>
            <sourceID>{00000000-0000-0000-0000-000000000000}</sourceID>
            <trackID>101</trackID>
            <timeSpan>
                <startTime>2021-08-18T15:18:15Z</startTime>
                <endTime>2021-08-18T15:19:08Z</endTime>
            </timeSpan>
            <mediaSegmentDescriptor>
                <contentType>video</contentType>
                <codecType>H.264-BP</codecType>
                <playbackURI>rtsp://10.14.97.40/Streaming/tracks/101/?starttime=20210818T151815Z&endtime=20210818T151908Z&name=00000004667000100&size=1400788</playbackURI>
                <lockStatus>unlock</lockStatus>
                <name>00000004667000100</name>
            </mediaSegmentDescriptor>
            <metadataMatches>
                <metadataDescriptor>recordType.meta..com/timing</metadataDescriptor>
            </metadataMatches>
        </searchMatchItem>
    </matchList>
</CMSearchResult>
```

### 4.4.3.3 Playback via RTSP

1. A client sends an RTSP DESCRIBE command.

```
DESCRIBE rtsp://10.14.97.40:554/Streaming/tracks/101/?starttime=20210818T151815Z&endtime=20210818T151908Z&name=00000004667000100&size=1400788 RTSP/1.0
CSeq: 4
Authorization: Digest username="", realm="323852ae0234c718f2d4198b", nonce="34e30476d", uri="rtsp://10.14.97.40:554/Streaming/tracks/101/?starttime=20210818T151815Z&endtime=20210818T151908Z&name=00000004667000100&size=1400788", response="883546f4c19dd156fb3a490266c99715"
User-Agent: LibVLC/3.0.3 (LIVE555 Streaming Media v2016.11.28)
```

2. The server responds that authentication is required.

```
RTSP/1.0 401 Unauthorized
CSeq: 5
WWW-Authenticate: Digest realm="323852ae0234c718f2d4198b", nonce="55e5895b9", algorithm="MD5"
```

3. The client sends an RTSP DESCRIBE request with authentication information again.

```
DESCRIBE rtsp://10.14.97.40:554/Streaming/tracks/101/?starttime=20210818T151815Z&endtime=20210818T151908Z&name=00000004667000100&size=1400788 RTSP/1.0
CSeq: 7
Authorization: Digest username="admin", realm="323852ae0234c718f2d4198b", nonce="55e5895b9", uri="rtsp://10.14.97.40:554/Streaming/tracks/101/?starttime=20210818T151815Z&endtime=20210818T151908Z&name=00000004667000100&size=1400788", response="cf33e4dc6b86a2fdd2e5b26d25e7b99d"
User-Agent: LibVLC/3.0.3 (LIVE555 Streaming Media v2016.11.28)
Accept: application/sdp
```

4. The server responds to SDP information.

5. The client sends RTSP SETUP commands two times, and the server responds to them.

```
SETUP rtsp://10.14.97.40:554/Streaming/tracks/101/?starttime=20210818T151815Z&endtime=20210818T151908Z&name=00000004667000100&size=1400788&trackID=video
RTSP/1.0
CSeq: 8
Authorization: Digest username="admin", realm="323852ae0234c718f2d4198b", nonce="5e5895b9", uri="rtsp://10.14.97.40:554/Streaming/tracks/101/?starttime=20210818T151815Z&endtime=20210818T151908Z&name=00000004667000100&size=1400788", response="44d9a3758b7f19f1a31fe042b1464b5"
User-Agent: LibVLC/3.0.3 (LIVE555 Streaming Media v2016.11.28)
Transport: RTP/AVP;unicast;client port=61048-61049
```

```
RTSP/1.0 200 OK
Session: 225263317;timeout=60
Transport: RTP/AVP;unicast;client_port=61048-61049;server_port=62000-62001:ssrc=d6d3ed5
CSeq: 8
Accept-Ranges: UTC
Media-Properties: Random-Access=1.0s, Unlimited, Immutable,Scales="-1, 0.5, 0.25, 0.125,:1, 2, 4"
Date: Wed, 18 Aug 2021 07:29:25 GMT
```

SETUP rtsp://10.14.97.40:554/Streaming/tracks/101/?starttime=20210818T151815Z&endtime=20210818T151908Z&name=00000004667000100&size=1400788/trackID=audio  
RTSP/1.0  
CSeq: 9  
Authorization: Digest username="admin", realm="323852ae0234c718f2d4198b", nonce="55e5895b9", uri="rtsp://10.14.97.40:554/Streaming/tracks/101/?starttime=20210818T151815Z&endtime=20210818T151908Z&name=00000004667000100&size=1400788", response="44d97a3758b7f19f1a31fe042b1464b5"  
User-Agent: LibVLC/3.0.3 (LIVE555 Streaming Media v2016.11.28)  
Transport: RTP/AVP;unicast;client\_port=61050-61051  
Session: 225263317

```
RTSP/1.0 200 OK
Session: 225263317;timeout=60
Transport: RTP/AVP;unicast;client_port=61050-61051;server_port=62002-62003;ssrc=d6d3ed6
CSeq: 9
Accept-Ranges: UTC
Media-Properties: Random-Access=1.0s, Unlimited, Immutable,Scales="-1, 0.5, 0.25, 0.125,:1, 2, 4"
Date: Wed, 18 Aug 2021 07:29:25 GMT
```

6. The client sends an RTSP PLAY request, and the server starts to push streams for playback.

PLAY rtsp://10.14.97.40:554/Streaming/tracks/101?starttime=20210818T151815Z&endtime=20210818T151908Z&name=00000004667000100&size=1400788 RTSP/1.0  
CSeq: 10  
Authorization: Digest username="admin", realm="323852ae0234c718f2d4198b", nonce="55e5895b9", uri="rtsp://10.14.97.40:554/Streaming/tracks/101?starttime=20210818T151815Z&endtime=20210818T151908Z&name=00000004667000100&size=1400788", response="4d8b7ceb301c06b40a9ae411189a90f"  
User-Agent: LibVLC/3.0.3 (LIVE555 Streaming Media V2016.11.28)  
Session: 225263317  
Range: npt=0.000-

```
RTSP/1.0 200 OK
Range: clock=20210818T151815Z-20210818T151908Z
Session: 225263317
CSeq: 10
Date: Wed, 18 Aug 2021 07:29:25 GMT
```

7. The client sends an RTSP TEARDOWN request to stop playback.

```
TEARDOWN rtsp://10.14.97.40:554/Streaming/tracks/101/?starttime=20210818T151815Z&endtime=20210818T151908Z&name=0000004667000100&size=1400788 RTSP/1.0
CSeq: 11
Authorization: Digest username="admin", realm="323852ae0234c718f2d4198b", nonce="55e5895b9", uri="rtsp://10.14.97.40:554/Streaming/tracks/101/?starttime=20210818T151815Z&endtime=20210818T151908Z&name=0000004667000100&size=1400788", response="1d3f6f8d07d7087d341560b125445456"
User-Agent: LibVLC/3.0.3 (LIVE555 Streaming Media v2016.11.28)
Session: 225263317
```

```
RTSP/1.0 200 OK
CSeq: 11
Date: Wed, 18 Aug 2021 07:29:38 GMT
Session: 225263317
Connection: close
```

## 5 Device (General)

### 5.1 App Custom & Auto-Parsing Event(s)

#### 5.1.1 Introduction to the Function

The HEOP's third-party app has the requirement of third-party platform communication. In order to improve the communication efficiency between third-party app and platforms, HEOP provides an event upload mode of reusing the existing communication link of HCNetSDK, ISAPI, ISUP, and OpenNetStream.

As the third-party apps are imported to the embedded devices, when apps reuse the existing communication link to upload events, the event content cannot be predicted by embedded devices. Therefore, it is necessary to constrain the events format uploaded by the third-party app to ensure the unified processing of embedded devices, and implement the auto-parsing and displaying of business platform.

This chapter mainly introduces the technical requirements and constraints on the content of events uploaded by third-party apps.

Two types of events are provided for uploading custom events of the third-party app.

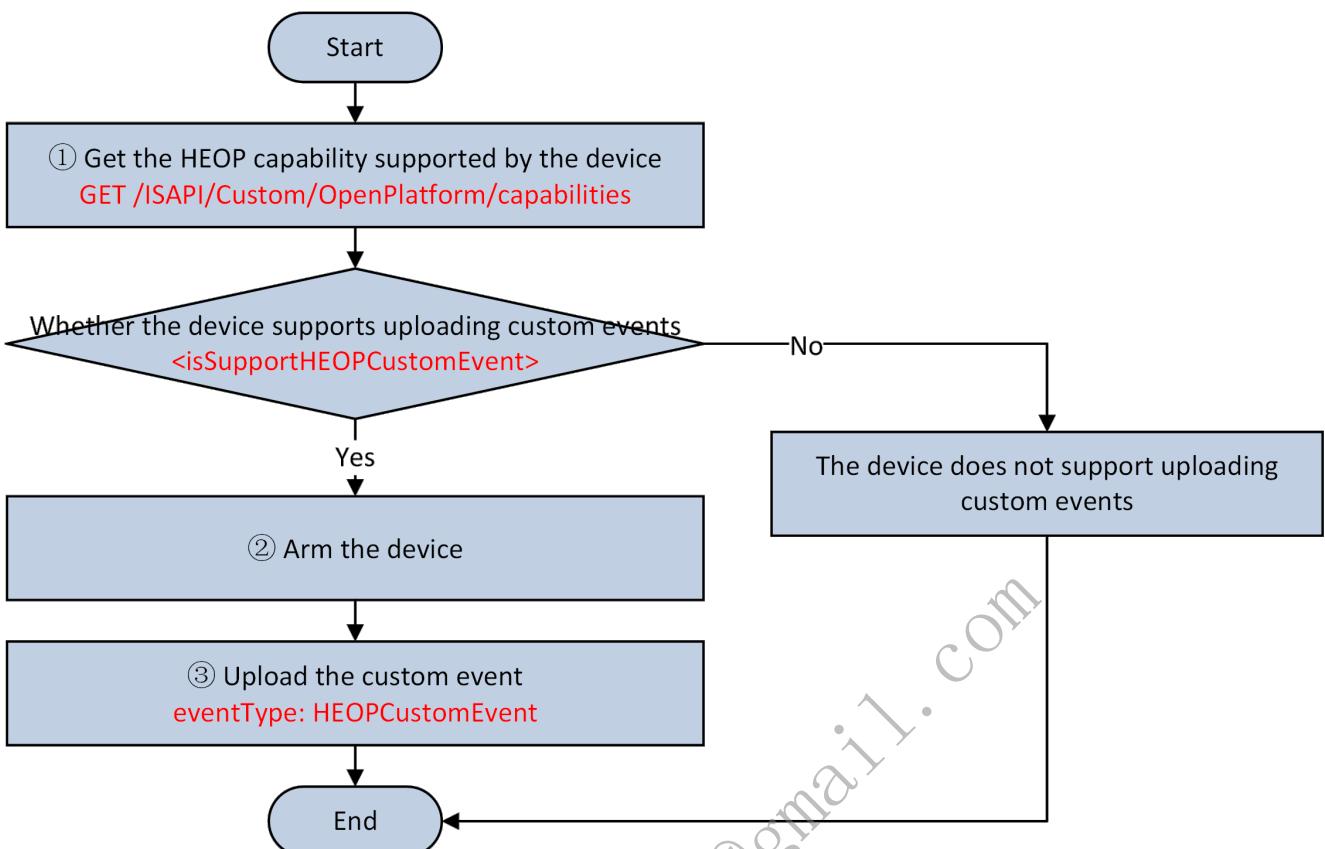
1. Uploading the custom event: the data is in JSON format, and the data content is agreed between the device and the platform. Embedded devices have no constraints on the uploaded content.
2. Uploading the auto-parsing event: the data is in JSON format, third-party apps need to comply with constraints on auto-parsing, and embedded devices verify the format of data uploaded by third-party apps. It realize the automatic parsing of platforms.

If the third-party app needs to use the existing communication link, you need to know the type of supported event uploading before programming.

#### 5.1.2 API Calling Flow

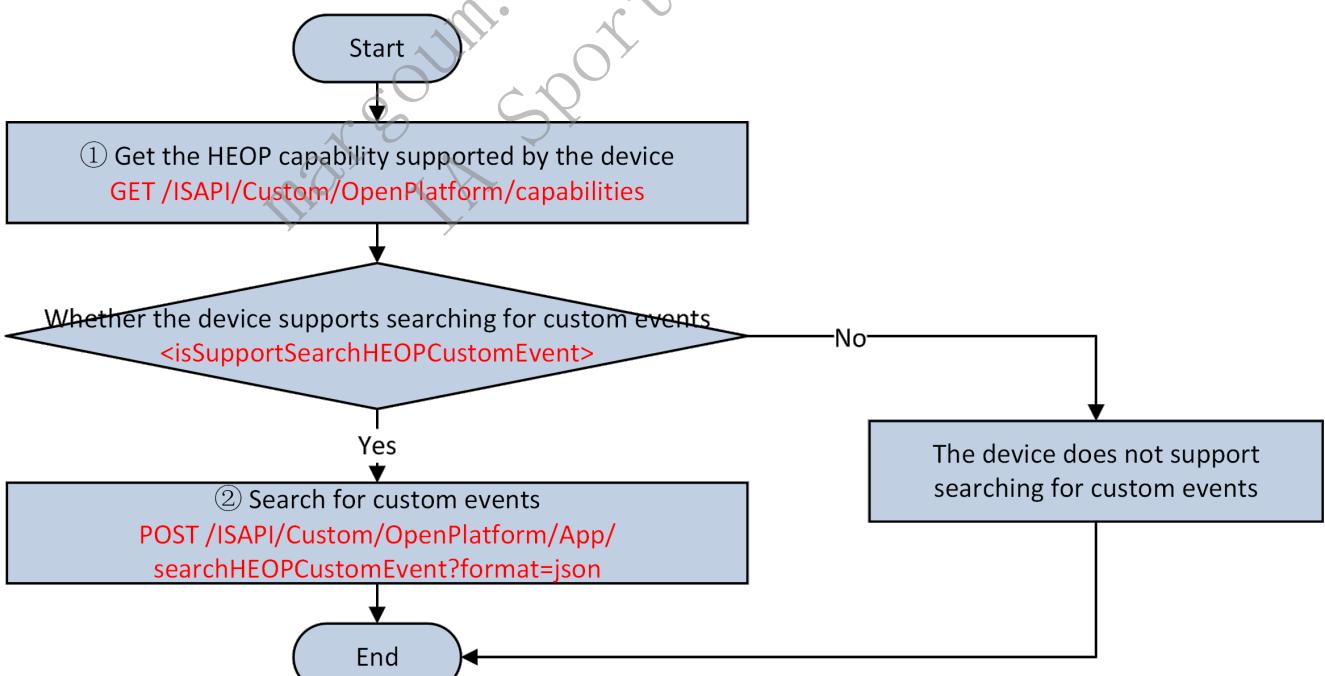
##### 5.1.2.1 Custom Event(s)

###### API Calling Flow of Uploading Custom Events



1. Get the HEOP capability supported by the device: `GET /ISAPI/Custom/OpenPlatform/capabilities`. If the node `<isSupportHEOPCustomEvent>` is returned in the capability and its value is true, it indicates that uploading custom events is supported by the device.
2. Arm the device, see "Event Arming and Subscription Management" for details.
3. Upload the custom event (eventType: `HEOPCustomEvent`).

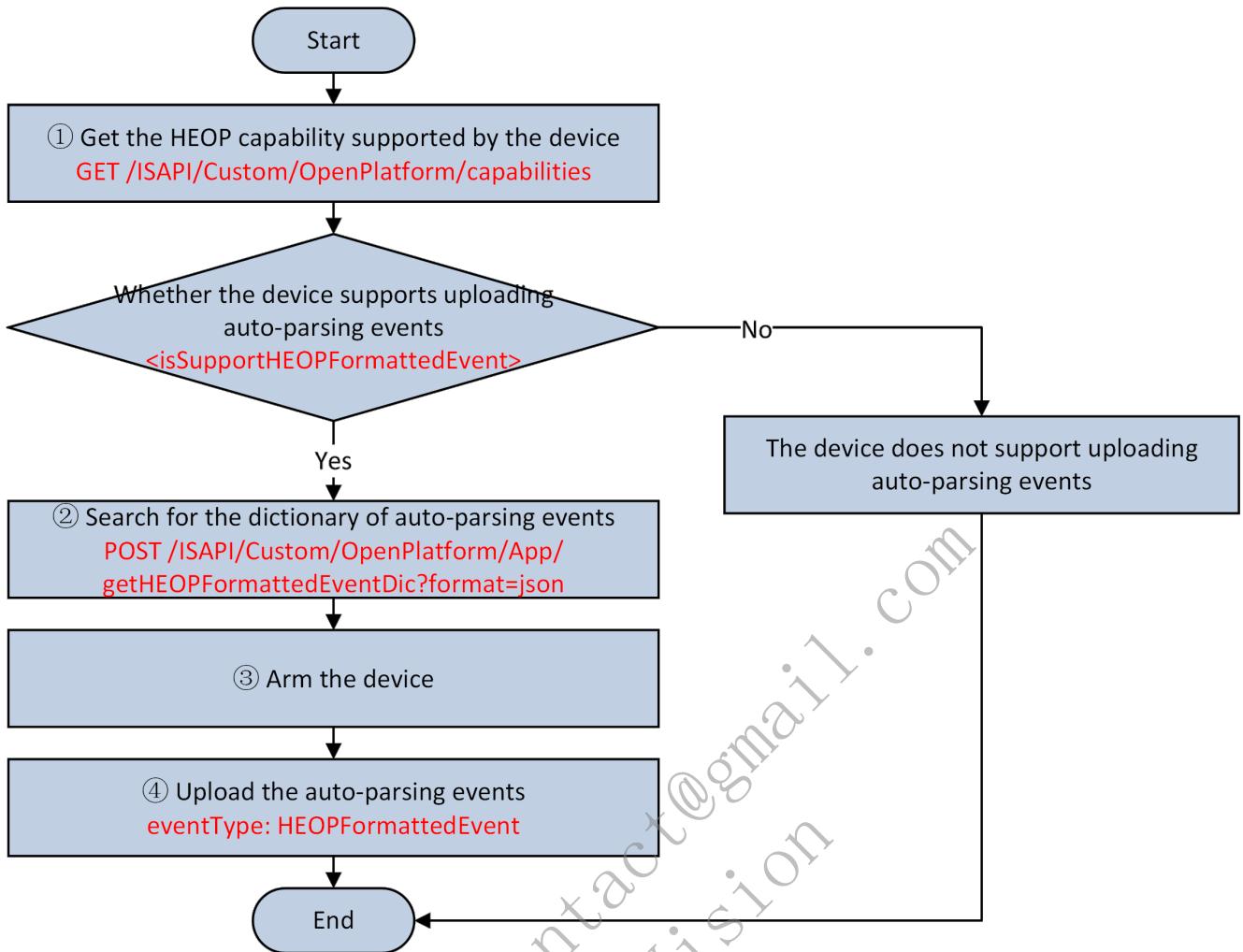
#### API Calling Flow of Searching for Custom Events



1. Get the HEOP capability supported by the device: `GET /ISAPI/Custom/OpenPlatform/capabilities`. If the node `<isSupportSearchHEOPCustomEvent>` is returned in the capability and its value is true, it indicates that searching for custom events is supported by the device.
2. Search for custom events: `POST /ISAPI/Custom/OpenPlatform/App/searchHEOPCustomEvent?format=json`.

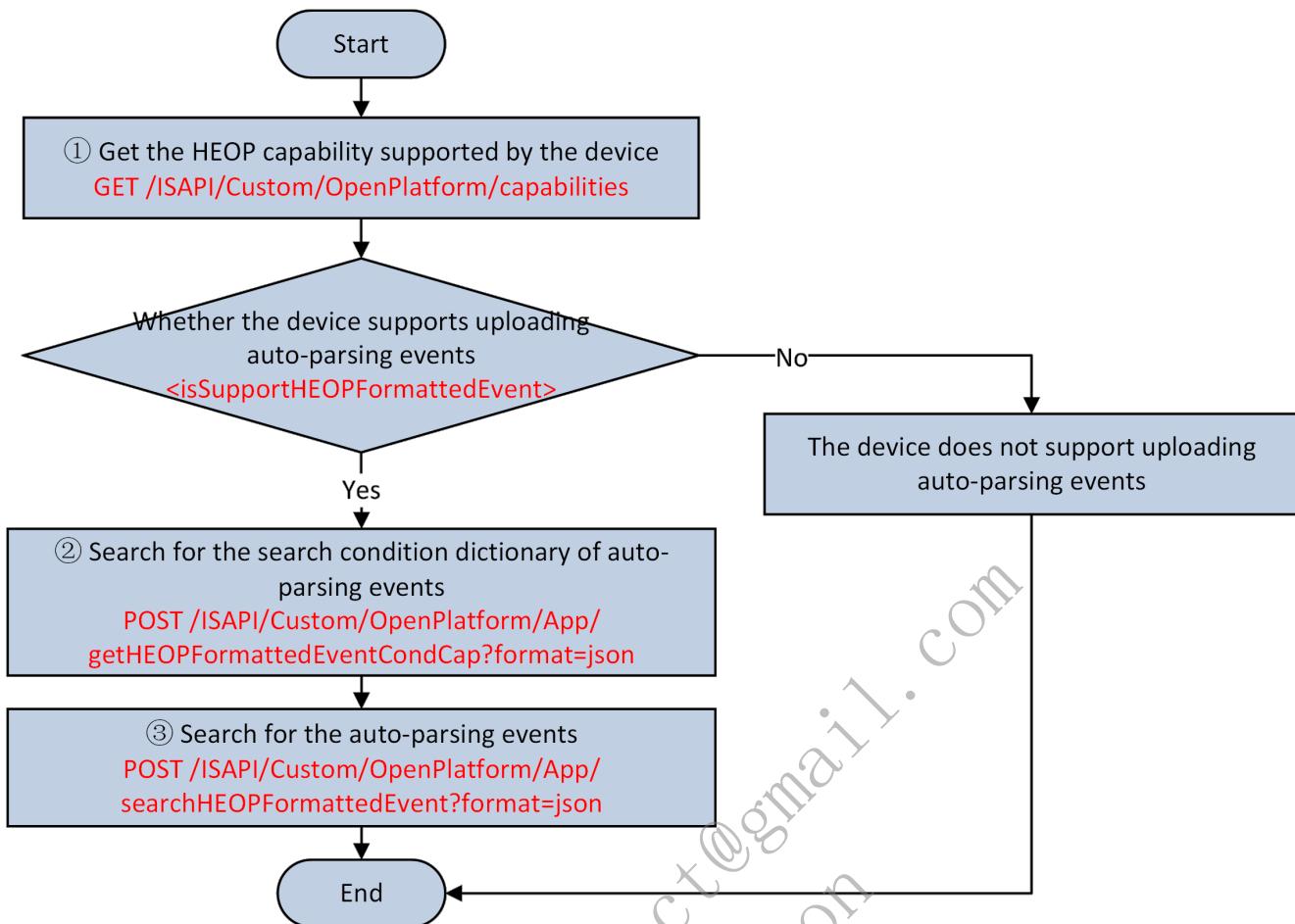
#### 5.1.2.2 Auto-Parsing Event(s)

##### API Calling Flow of Uploading Auto-Parsing Events



1. Get the HEOP capability supported by the device: GET /ISAPI/Custom/OpenPlatform/capabilities. If the node <isSupportHEOPFormattedEvent> is returned in the capability and its value is true, it indicates that uploading auto-parsing events is supported by the device.
2. Search for the dictionary of auto-parsing events: POST /ISAPI/Custom/OpenPlatform/App/getHEOPFormattedEventDic?format=json.
3. Arm the device, see "Event Arming and Subscription Management" for details.
4. Upload the auto-parsing event (event type: HEOPFormattedEvent).

#### API Calling Flow of Searching for Auto-Parsing Events



1. Get the HEOP capability supported by the device: GET /ISAPI/Custom/OpenPlatform/capabilities. If the node <isSupportSearchHEOPFormattedEvent> is returned in the capability and its value is true, it indicates that searching for auto-parsing events is supported by the device.
  2. Search for the search condition dictionary of auto-parsing events: POST /ISAPI/Custom/OpenPlatform/App/getHEOPFormattedEventCondCap?format=json.
  3. Search for auto-parsing event: POST /ISAPI/Custom/OpenPlatform/App/searchHEOPFormattedEvent?format=json.

### 5.1.2.3 Message Example and UI Example

## 1. Example of Searching for Auto-Parsing Events Dictionary

## Request Message Example

```
{  
    "AppID": "12345"  
}
```

## Response Message Example

```
{
    "HEOPFormattedEventDic": [
        {
            "subEventType": "faceInfo",
            "subEventTypeName": "Face Picture Information",
            "ParamsInfo": [
                {
                    "paramName": "gender",
                    "description": "Gender",
                    "type": "enumValue",
                    "enumValue": ["female", "male", "unknown"],
                    "enumValueDescription": ["Female", "Male", "Unknown"]
                },
                {
                    "paramName": "name",
                    "description": "Name",
                    "type": "int",
                },
                {
                    "paramName": "age",
                    "description": "Age",
                    "type": "int",
                },
                {
                    "paramName": "labels",
                    "description": "Labels",
                    "type": "stringArray",
                },
                {
                    "paramName": "faceRect",
                    "description": "Face Frame",
                    "type": "Region",
                }
            ]
        },
        {
            "subEventType": "vehicleInfo",
            "subEventTypeName": "Vehicle Information",
            "ParamsInfo": [
                {
                    "paramName": "plateNo",
                    "description": "License Plate No.",
                    "type": "string",
                    "enumValue": [],
                    "enumValueDescription": []
                }
            ]
        }
    ]
}
```

## 2. Example of Uploading Auto-Parsing Event

### Complete Example

```

POST /HEOPFormattedEvent HTTP/1.1
Content-Type: multipart/form-data; boundary=7e13971310878
Content-Length: 1343

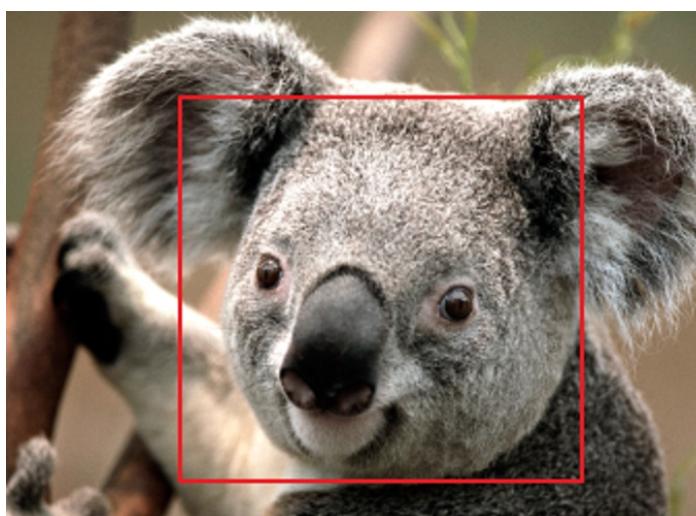
--7e13971310878
Content-Disposition: form-data; name="HEOPFormattedEvent.json";filename="HEOPFormattedEvent.json";
Content-Type: application/json
Content-Length: 997

{
    "ipAddress": "172.6.64.7",
    "portNo": 80,
    "protocol": "HTTP",
    "macAddress": "01:17:24:45:D9:F4",
    "channelID": 1,
    "dateTime": "2004-05-03T17:30:08+08:00",
    "activePostCount": 1,
    "eventType": "HEOPFormattedEvent",
    "eventState": "active",
    "eventDescription": "HEOP Custom Event",
    "channelName": "test",
    "deviceID": "test0123",
    "serialNumber": "test",
    "AppID": "12345",
    "subEventType": "faceInfo",
    "HEOPFormattedEvent": {
        "gender": "male",
        "name": "Tom",
        "age": 25,
        "labels": ["Young Artist", "Bank Staff"],
        "faceRect": [
            {
                "x": 0.234,
                "y": 0.234,
            },
            {
                "x": 0.234,
                "y": 0.456,
            },
            {
                "x": 0.456,
                "y": 0.456,
            },
            {
                "x": 0.456,
                "y": 0.234,
            },
        ]
    }
}
--7e13971310878
Content-Disposition: form-data; name="picture1";filename="picture1.jpg";
Content-Type: image/jpeg
Content-Length: 516876

.....JFIF.....
--7e13971310878--

```

### 3. Example of Result for Parsing Uploaded Auto-Parsing Event



Face Frame

**Name:** Koala  
**Gender:** Unknown  
**Age:** 3  
**Label:** Artist, Bank Employee

### 4. Example of Searching for the Search Condition Dictionary of Auto-Parsing Events

#### Request Message Example

```
{  
    "AppID": "12345"  
}
```

### Response Message Example

```
{  
    "HEOPFormattedEventCondCap": [  
        {  
            "subEventType": "faceInfo",  
            "subEventTypeName": "Face Picture Information",  
            "ParamsInfo": [  
                {  
                    "paramName": "gender",  
                    "description": "Gender",  
                    "type": "enumValueArray",  
                    "enumValueArrayCap": {  
                        "size": 3,  
                        "enumValue": ["female", "male", "unknown"],  
                        "enumValueDescription": ["Female", "Male", "Unknown"]  
                    }  
                },  
                {  
                    "paramName": "name",  
                    "description": "Name",  
                    "type": "string",  
                    "stringCap": {  
                        "size": 64  
                    }  
                },  
                {  
                    "paramName": "age",  
                    "description": "Age",  
                    "type": "intRange",  
                    "IntRangeCap": {  
                        "minRange": [0, 100],  
                        "maxRange": [0, 100],  
                    }  
                }  
            ]  
        }  
    ]  
}
```

### 5. UI Example of Search Condition of Auto-Parsing Events

Gender



Name

Age

 — 

### 6. Example of Searching for Auto-Parsing Event

#### Request Message Example

```
{
  "searchID": "123",
  "searchResultPosition": 1,
  "maxResults": 100,
  "startTime": "2020-01-01T08:00+08:00",
  "endTime": "2020-01-02T08:00+08:00",
  "AppID": "12345",
  "subEventType": "faceInfo",
  "HEOPFormattedEventCond": {
    "gender": ["female", "male"],
    "name": "Tom",
    "age": [25, 35]
  }
}
```

## Response Message Example

```
{
  "searchID": "123",
  "responseStatusStrg": "OK",
  "numOfMatches": 1,
  "totalMatches": 1,
  "Results": [
    {
      "ipAddress": "172.6.64.7",
      "portNo": 80,
      "protocol": "HTTP",
      "macAddress": "01:17:24:45:D9:F4",
      "channelID": 1,
      "dateTme": "2004-05-03T17:30:08+08:00",
      "activePostCount": 1,
      "eventType": "HEOPFormattedEvent",
      "eventState": "active",
      "eventDescription": "HEOP Custom Event",
      "channelName": "test",
      "deviceID": "test0123",
      "serialNumber": "test",
      "AppID": "12345",
      "subEventType": "faceInfo",
      "HEOPFormattedEvent": {
        "gender": "male",
        "name": "Tom",
        "age": 25,
        "labels": ["Artist", "Bank Employee"],
        "faceRect": [
          {
            "x": 0.234,
            "y": 0.234
          },
          {
            "x": 0.234,
            "y": 0.456
          },
          {
            "x": 0.456,
            "y": 0.456
          },
          {
            "x": 0.456,
            "y": 0.234
          }
        ]
      }
    }
  ],
  "picURL": "http://xxx.xxx.xxx/xxxx",
  "URLCertificationType": "digest"
}
```

## 5.2 Arming and Subscription

### 5.2.1 Introduction to the Function

With arming and subscription, the client can establish HTTP persistent connection with the device, and continuously receive the event messages from the device.

For ISAPI event arming, the client can receive all types of events by GET method, or receive the subscribed events only by POST method.

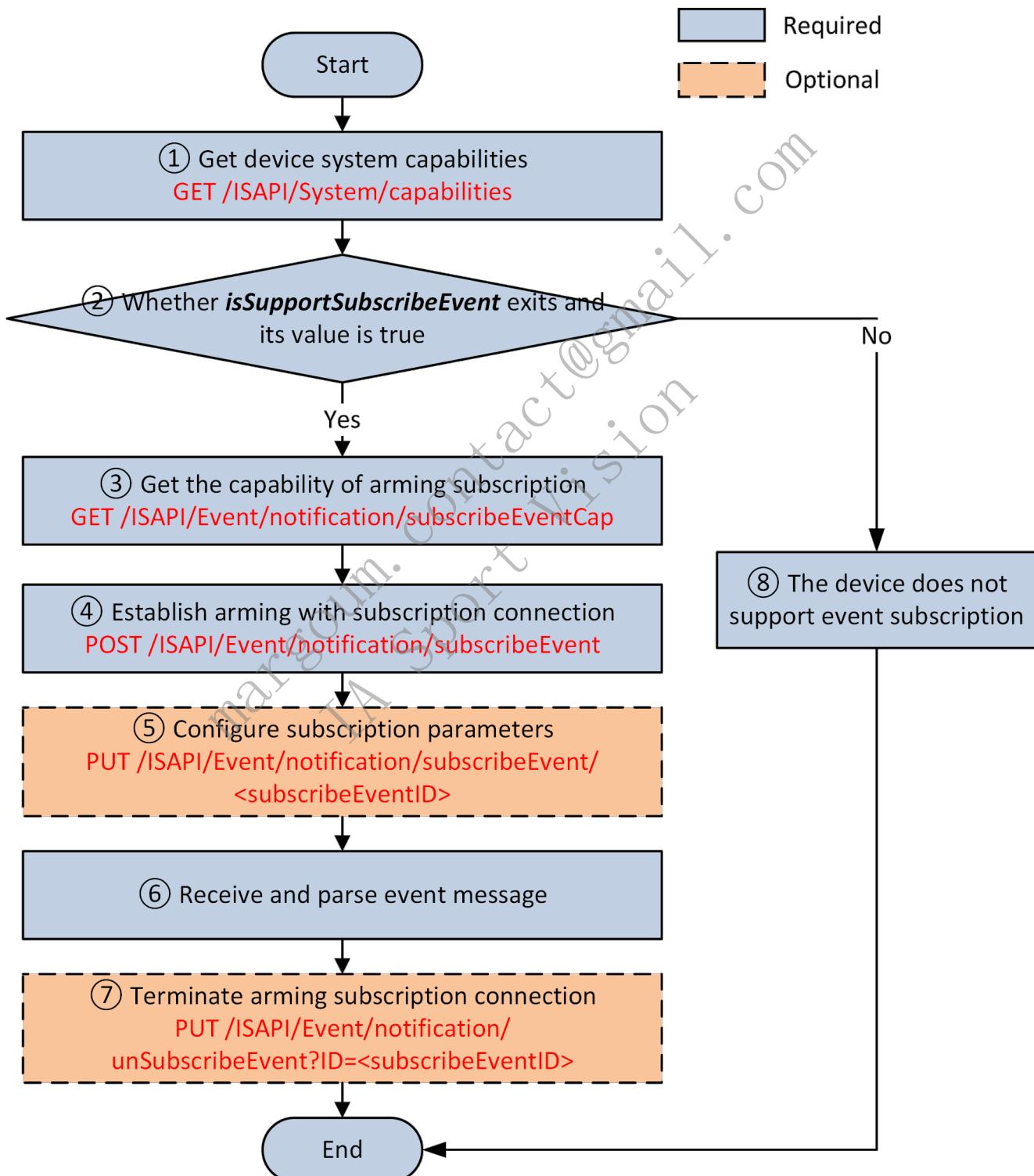
### 5.2.2 API Calling Flow

### 5.2.2.1 Without Subscription

1. Establish a connection for arming: `GET /ISAPI/Event/notification/alertStream`. You need to set `Connection: keep-alive` in HTTP Headers.
2. When receiving events sent by the device, the event messages can be separated and parsed by boundary. See "Parsing Event Messages" below for details.
3. Disable the arming connection when you do not need to receive event messages.

### 5.2.2.2 Subscription

#### 5.2.2.2.1 API Calling Flow



1. Get device system capabilities: `GET /ISAPI/System/capabilities`.
2. Check if event subscription is supported: `isSupportSubscribeEvent` exists and its value is `true`. When `isSupportSubscribeEvent` does not exist or its value is `false`, the device does not support event subscription.

3. Get the capability of arming with subscription: `GET /ISAPI/Event/notification/subscribeEventCap`.
4. Establish a connection of arming with subscription: `POST /ISAPI/Event/notification/subscribeEvent`. You need to set `Connection: keep-alive` in HTTP Headers.
5. (Optional) Edit parameters of the existing subscription. You need to get the subscription parameters first: `GET /ISAPI/Event/notification/subscribeEvent/<subscribeEventID>`. Then, edit the parameters based on the existing subscription configurations: `PUT /ISAPI/Event/notification/subscribeEvent/<subscribeEventID>`.
6. Receive events sent by the device. The event messages will be separated and parsed by boundary. For parsing description, see **Event Messages Parsing** below.
7. (Optional) Terminate the connection of arming with subscription: `PUT /ISAPI/Event/notification/unSubscribeEvent?ID=<subscribeEventID>`. When communicating with the device via HTTP directly, there is no need to call this API. You can just terminate the connection.

**Note:**

Three types of data will be transmitted on the arming link: `<SubscribeEventResponse/>`, `<EventNotificationAlert/>`, and picture data. `<SubscribeEventResponse/>` is the data of first form sent by the device after arming established, see the response parameters of URL (`POST /ISAPI/Event/notification/subscribeEvent`) for details; and `<EventNotificationAlert/>` is the event content or heartbeat, you can identify the event type via field `eventType`, e.g., for heartbeat, the value of `eventType` is `heartBeat`.

#### 5.2.2.2 Example

```

POST /ISAPI/Event/notification/subscribeEvent HTTP/1.1
Host: device_ip
Accept-Language: zh-cn
Date: YourDate
Content-Type: application/xml;
Content-Length: text_length
Connection: Keep-Alive

<SubscribeEvent/>
HTTP/1.1 401 Unauthorized
Date: Sun, 01 Apr 2018 18:58:53 GMT
Server:
Content-Length: 178
Content-Type: text/html
Connection: keep-alive
Keep-Alive: timeout=10, max=99
WWW-Authenticate: Digest qop="auth", realm="IP Camera(C2183)", nonce="4e5468694e7a42694e7a4d364f4449354d7a6b354d54513d", stale="FALSE"

POST /ISAPI/Event/notification/subscribeEvent HTTP/1.1
Authorization: Digest username="admin",realm="IP
Camera(C2183)",nonce="4e5468694e7a42694e7a4d364f4449354d7a6b354d54513d",uri="/ISAPI/Event/notification/alertStream",cnonce="3d183a245b8729121ae4ca3d41b90f18
",nc=00000001,qop="auth",response="f2e0728991bb031f83df557a8f185178"
Host: device_ip

<SubscribeEvent/>
HTTP/1.1 200 OK
MIME-Version: 1.0
Connection: close
Content-Type: multipart/mixed; boundary=<frontier>

--<frontier>
Content-Type: application/xml; charset="UTF-8" <!--some event messages are uploaded in JSON format, and the upper layer needs to distinguish the message format accroding to Content-Type when parsing event messages-->
Content-Length: text_length

<SubscribeEventResponse/>
--<frontier>
Content-Type: application/xml; charset="UTF-8" <!--some event messages are uploaded in JSON format, and the upper layer needs to distinguish the message format accroding to Content-Type when parsing event messages-->
Content-Length: text_length

<EventNotificationAlert/>
--<frontier>
Content-Disposition: form-data; name="Picture_Name"
Content-Type: image/jpeg
Content-Length: image_length

[Picture Data]
--<frontier>

```

#### 5.2.2.3 Event Messages Parsing

After the arming connection with the device is established, the data sent by the device is in HTTP form format (multipart/form-data). In an HTTP request, Content-Type in Headers is usually multipart/form-data, boundary=AaB03x, and boundary is a variable used to divide HTTP Body into multiple units, each being a set of data and has its own Headers and Body. For detailed format description, see [RFC 1867 \(Form-based File Upload in HTML\)](#). An example is shown below. Note the dash -- before and after boundary.

```
HTTP/1.1 200 OK
Content-Type: multipart/form-data; boundary=AaB03x
Connection: keep-alive
--AaB03x
Content-Disposition: form-data; name="ANPR.xml"; filename="ANPR.xml";
Content-Type: application/xml
Content-Length: 9

<ANPR/>
--AaB03x
Content-Disposition: form-data; name="licensePlatePicture.jpg"; filename="licensePlatePicture.jpg";
Content-Type: image/jpeg
Content-Length: 14

Image Data
--AaB03x--
```

The description of some keywords are as follows:

Keyword	Example	Description
Content-Type	multipart/form-data; boundary=AaB03x	Content type. multipart/form-data means the message is in form format.
boundary	AaB03x	Delimiter of the form message. --boundary is the start of a form. --boundary-- is the end of the whole HTTP form message.
Content-Disposition	form-data; name="ANPR.xml"; filename="ANPR.xml";	Content description.
name	"ANPR.xml"	Form name.
filename	"ANPR.xml"	File name of the form.
Content-Length	9	Content length, starting from the next \r\n to the next --boundary.

### 5.2.3 Restriction Description

Note that ISAPI arming (with or without subscription) uses HTTP/HTTPS persistent connection. Due to the simplex channel communication mode of HTTP, after establishing the arming connection, the device will send out event messages continuously, while you cannot send any message to the device via the connection.

After the heartbeat time, if you do not receive any message from the device, you should disable the arming connection and try establishing a new one.

### 5.2.4 Sample Messages

#### 5.2.4.1 Establish Arming Subscription

```
POST /ISAPI/Event/notification/subscribeEvent HTTP/1.1
Authorization: Digest username="admin",realm="IP
Camera(C2183)"
,nonce="4e5468694e7a42694e7a4d364f449354d7a6b354d54513d",uri="/ISAPI/Event/notification/alertStream",cnonce="3d183a245b8729121ae4ca3d41b90f18
",nc=00000001,qop="auth",response="f2e0728991bb031f83df557a8f185178"
Host: device_ip

<SubscribeEvent/>
```

#### 5.2.4.2 The Device Responses and Uploads an Event Message

```

HTTP/1.1 200 OK
MIME-Version: 1.0
Connection: close
Content-Type: multipart/mixed; boundary=<frontier>

--<frontier>
Content-Type: application/xml; charset="UTF-8" <!--Some alarm messages are in JSON format, so when parsing messages, the upper-Layer should distinguish them according to the Content-Type field.-->
Content-Length: text_length

<SubscribeEventResponse/>
--<frontier>
Content-Type: application/xml; charset="UTF-8" <!--Some alarm messages are in JSON format, so when parsing messages, the upper-Layer should distinguish them according to the Content-Type field.-->
Content-Length: text_length

<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/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>heartBeat</eventType>
  <eventState>active</eventState>
  <eventDescription>heartBeat</eventDescription>
</EventNotificationAlert>
--<frontier>
Content-Disposition: form-data; name="Picture_Name"
Content-Type: image/jpeg
Content-Length: image_length
Content-ID: image_ID

[Picture Data]
--<frontier>

```

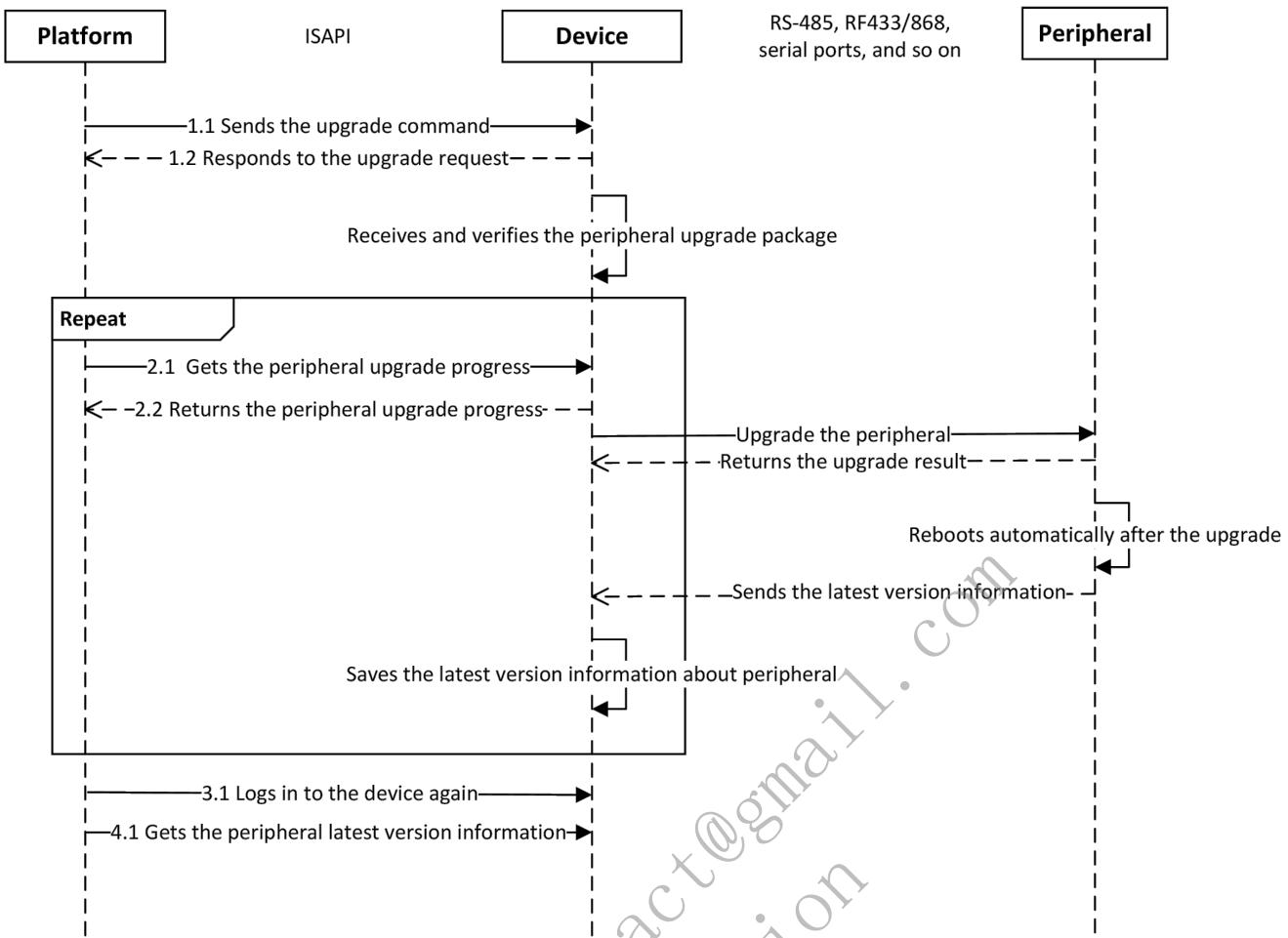
## 5.3 Device Peripherals Upgrade

### 5.3.1 Introduction to the Function

The platform or client software or web client under the LAN upgrades device peripherals via ISAPI.

### 5.3.2 API Calling Flow

The sequence diagram of upgrading device peripherals by the platform is shown below.



1. Get the device system capability `GET /ISAPI/System/capabilities` and check whether the device supports upgrading peripherals. If the field `isSupportAcsUpdate` is returned and its value is true, it indicates that the device supports this function, otherwise, the device does not support this function.
2. Get the capability of upgrading the peripherals module `GET /ISAPI/System/AcsUpdate/capabilities`, and get the types and IDs of peripherals that support upgrading.
3. The platform sends the upgrade command `POST /ISAPI/System/updateFirmware?type=<type>&moduleAddress=<moduleAddress>&id=<indexID>`. In the URL `type` refers to the peripheral type, `moduleAddress` refers to the peripheral module address, and `indexID` refers to the ID of peripheral to be upgraded. The platform will apply the upgrade peripheral package to the device.
4. Get the peripheral upgrade progress `GET /ISAPI/System/upgradeStatus?type=<Type>`.
5. Log in to the device again.
6. Get the peripheral latest version information.

## 5.4 Device Time Sync

### 5.4.1 Introduction to the Function

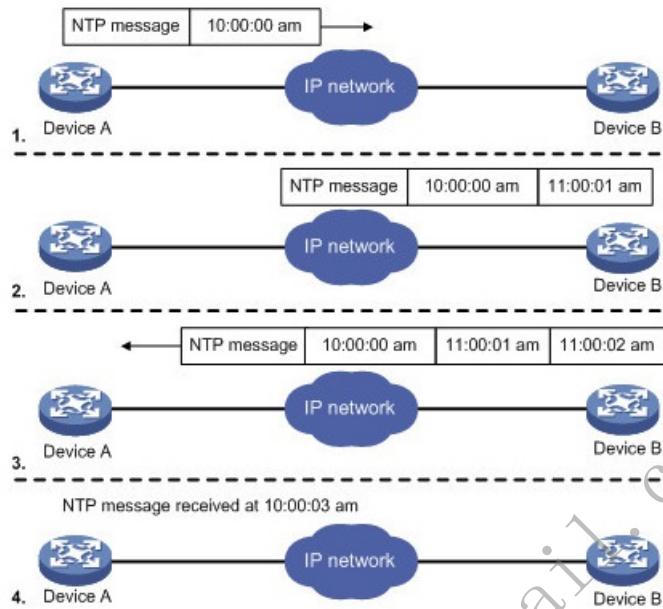
Time sync is a method to synchronize the `time` of all devices connecting to the NTP server, so that all devices can share the same clock `time` for providing related functions based on `time`. Supported `time` sync types: NTP `time` sync, manual sync, satellite `time` sync, platform `time` synchronization, GB28181 `time` sync, etc. The following describes the method of NTP `time` sync.

#### 5.4.1.1 NTP Time Sync

The local system of running NTP can receive sync from other clock sources (self as client), other clocks can sync from the local system (self as server), and sync with other devices.

The basic working principle of NTP is shown in the picture. Device A and Device B are connected via the network, and their systems follow their own independent system time. To auto sync their time, you can set device time auto sync via NTP. For example:

Before time sync between Device A and Device B, the time of Device A is 10:00:00 am, and that of Device B is 11:00:00 am. Device B is set as the server of NTP server, so that the time of Device A should be synchronized with that of Device B. The time of NTP message transmitted between Device A and Device B is 1 second.



The working process of system clock synchronization is as follows:

Device A sends an NTP message to Device B with a timestamp of 10:00:00 am (T1) that is when it leaves Device A.

When the NTP message reaches Device B, Device B will add its own timestamp, which is 11:00:01 am (T2).

Then the NTP message leaves Device B with Device B's timestamp, which is 11:00:02 am (T3).

Device A receives the response message, and the local time of Device A is 10:00:03 am (T4).

Above all, Device A can calculate two important parameters:

Round-trip delay of NTP message: Delay = (T4-T1) - (T3-T2) = 2 seconds.

Time difference between Device A and Device B: offset = ((T2-T1)+(T3-T4))/2=1 h.

Device A can sync its own time with that of Device B according to calculation results.

## 5.4.2 API Calling Flow

### 5.4.2.1 Time Sync Configuration

#### 1. Get the Capability of Device Time synchronization Management

You can call this API to get the time sync types currently supported by the device, such as NTP time sync, manual time sync, satellite time sync, Connect platform time sync, and GB28181 time sync.

Get the capability: `GET /ISAPI/System/time/capabilities`.

#### 2. Set device time synchronization management parameters

You can configure the time synchronization mode as follows.

NTP time synchronization: See 4.2.2 NTP Time Sync (Client).

Manual time synchronization: Set the value of `timeMode` to `manual`, and set the device local time in nodes `localTime`, `timeZone`.

Satellite time synchronization: Set the value of `timeMode` to `satellite`, and set the device local time in nodes `satelliteInterval`.

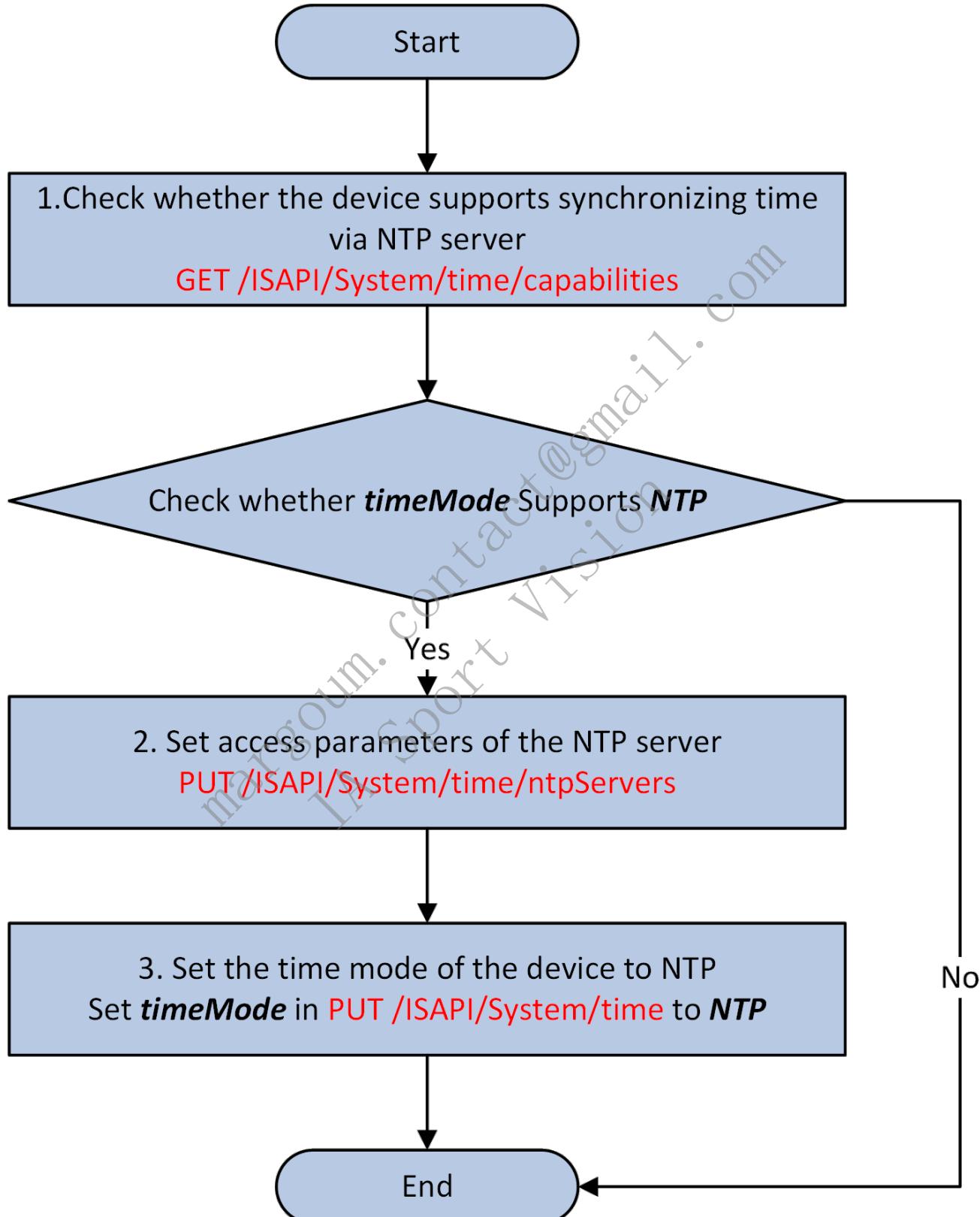
Platform time synchronization: Set the value of `timeMode` to `platform`.

GB28181 time synchronization: Set the value of `timeMode` to `GB28181`. If there are two GB28181 platforms, you can select platform No. via `platformNo`. It is the only ID, which is configured via `platformNo` in `GB28181List`, related URL: `/ISAPI/System/Network/SIP/<SIPServerID>`

Get device time synchronization management parameters: `GET /ISAPI/System/time`; Set device time synchronization management parameters: `PUT /ISAPI/System/time`.

#### 5.4.2.2 NTP Time Sync (Client)

The local system running the NTP server can receive sync information from other clock sources (self as client), sync other clocks (self as server) as clock sources, and sync with other devices. Calling flow (self as client):



1. Check whether the device supports synchronizing time via NTP server Get the capability of the device: `GET /ISAPI/System/time/capabilities`; and check whether `timeMode` supports `NTP`.

##### 2. Set access parameters of the NTP server

Supports accessing the NTP server by IP address to synchronize the device time.

Get the access parameter capability of the NTP server: GET /ISAPI/System/time/ntpServers/capabilities

Set access parameters of the NTP server: PUT /ISAPI/System/time/ntpServers

Get access parameters of the NTP server: GET /ISAPI/System/time/ntpServers

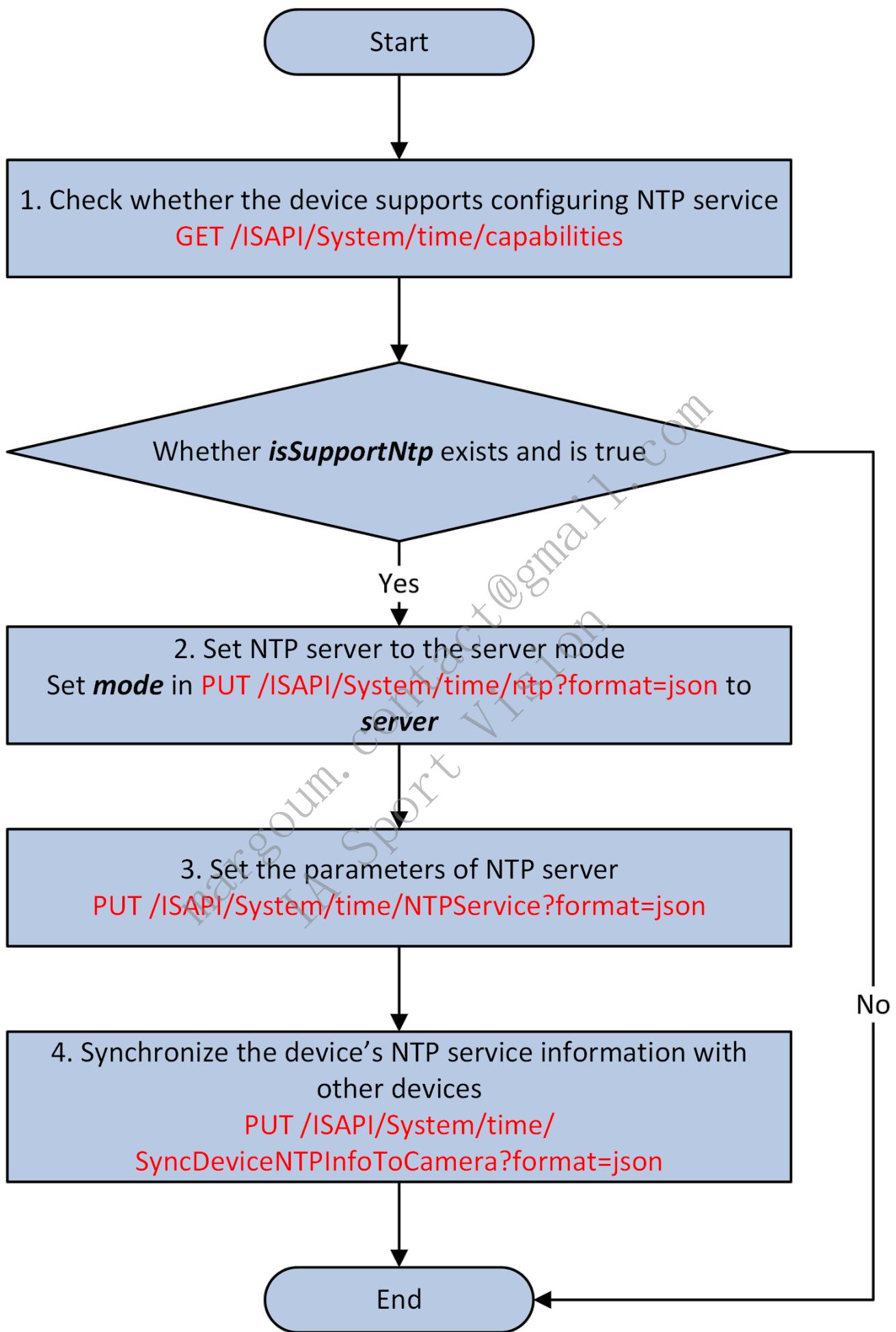
### **3. Set the time mode of the device to NTP**

Supports setting the value of `timeMode` to `NTP`.

Get device time synchronization management parameters: GET /ISAPI/System/time Set device time synchronization management parameters: PUT /ISAPI/System/time

#### **5.4.2.2 NTP Time Sync (Server Mode)**

The local system running the NTP server can receive sync information from other clock sources (self as client), sync other clocks (self as server) as clock sources, and sync with other devices. Calling flow (self as server):



1. Check whether the device supports configuring NTP service Get the capability of device time synchronization management: GET /ISAPI/System/time/capabilities; If *isSupportNtp* is returned, it indicates that the device supports time synchronization management.

## **2. Set NTP server to the server mode**

Supports setting the value of `mode` to `server`.

Get the capability of server mode: `GET /ISAPI/System/time/ntp/capabilities?format=json`

Set NTP to server mode: `PUT /ISAPI/System/time/ntp?format=json`

Get parameters of NTP server mode: `GET /ISAPI/System/time/ntp?format=json`

## **3. Set the parameters of NTP server**

Supports setting the IP address of the NTP server.

Get the capability of NTP server: `GET /ISAPI/System/time/NTPService/capabilities?format=json`

Set the NTP server parameters: `PUT /ISAPI/System/time/NTPService?format=json`

Get the parameters of the NTP server: `GET /ISAPI/System/time/NTPService?format=json`

## **4. Synchronize the device's NTP service information with other devices**

Supports synchronizing the time information to the camera.

Get the capability set of synchronizing device's NTP service information with the camera: `GET /ISAPI/System/time/SyncDeviceInfoToCamera/capabilities?format=json`

Synchronize device's NTP service information with the camera: `PUT /ISAPI/System/time/SyncDeviceInfoToCamera?format=json`

Get the progress of synchronizing device's NTP service information with the camera: `GET /ISAPI/System/time/SyncDeviceInfoToCamera/Progress?format=json`

Search for the results of synchronizing device's NTP service information with the camera: `POST /ISAPI/System/time/SyncDeviceInfoToCamera/SearchResult?format=json`

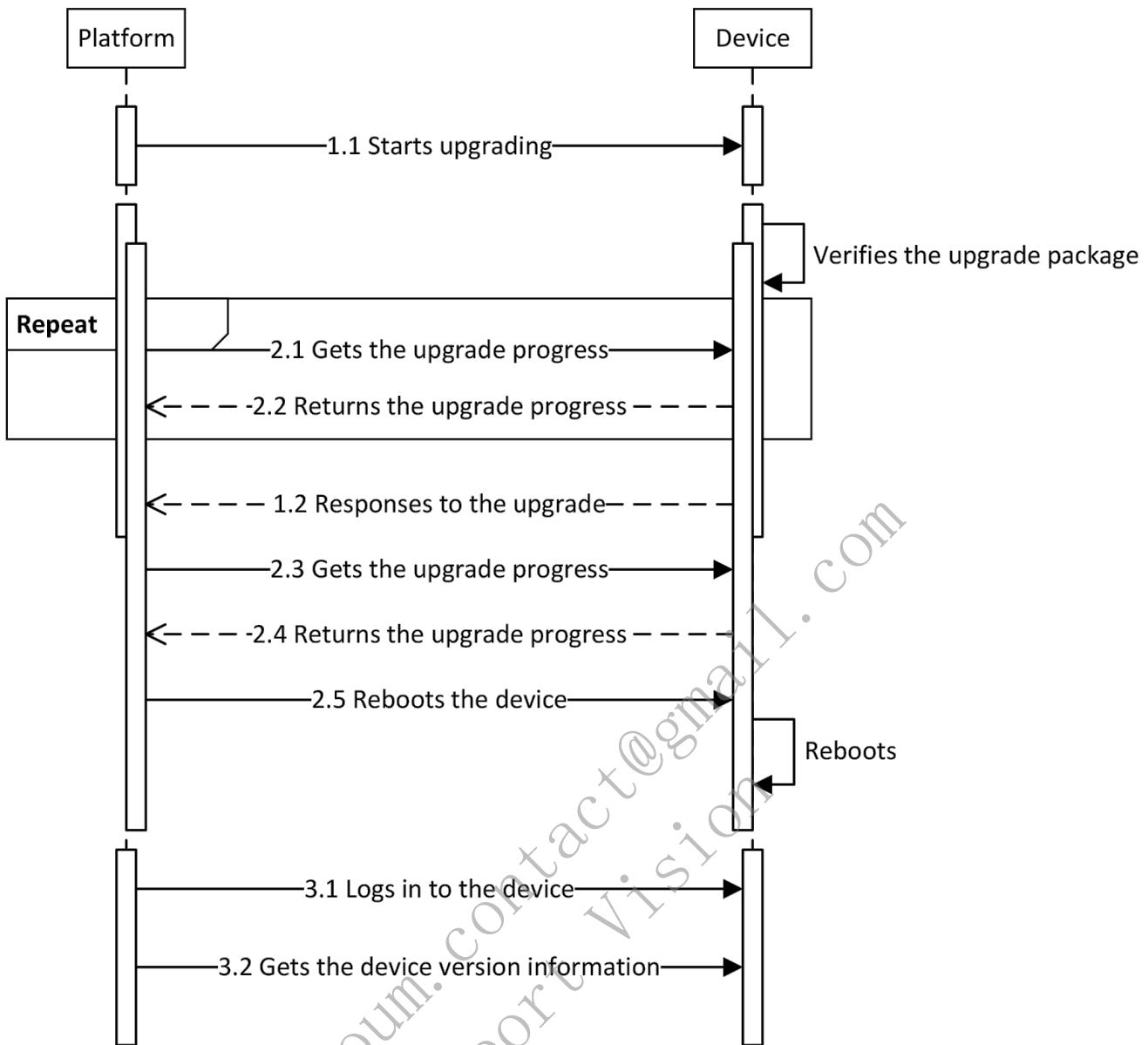
## **5.5 Device Upgrade**

### **5.5.1 Introduction to the Function**

The platform or client software or web client under the LAN upgrades devices via ISAPI.

### **5.5.2 API Calling Flow**

The sequence diagram of upgrading devices by the platform is shown below.



### 1. Upgrade devices.

Upgrade the device firmware: `POST /ISAPI/System/updateFirmware`.

### 2. Get the device upgrade progress.

Get the device upgrade progress: `GET /ISAPI/System/upgradeStatus`.

### 3. Reboot devices.

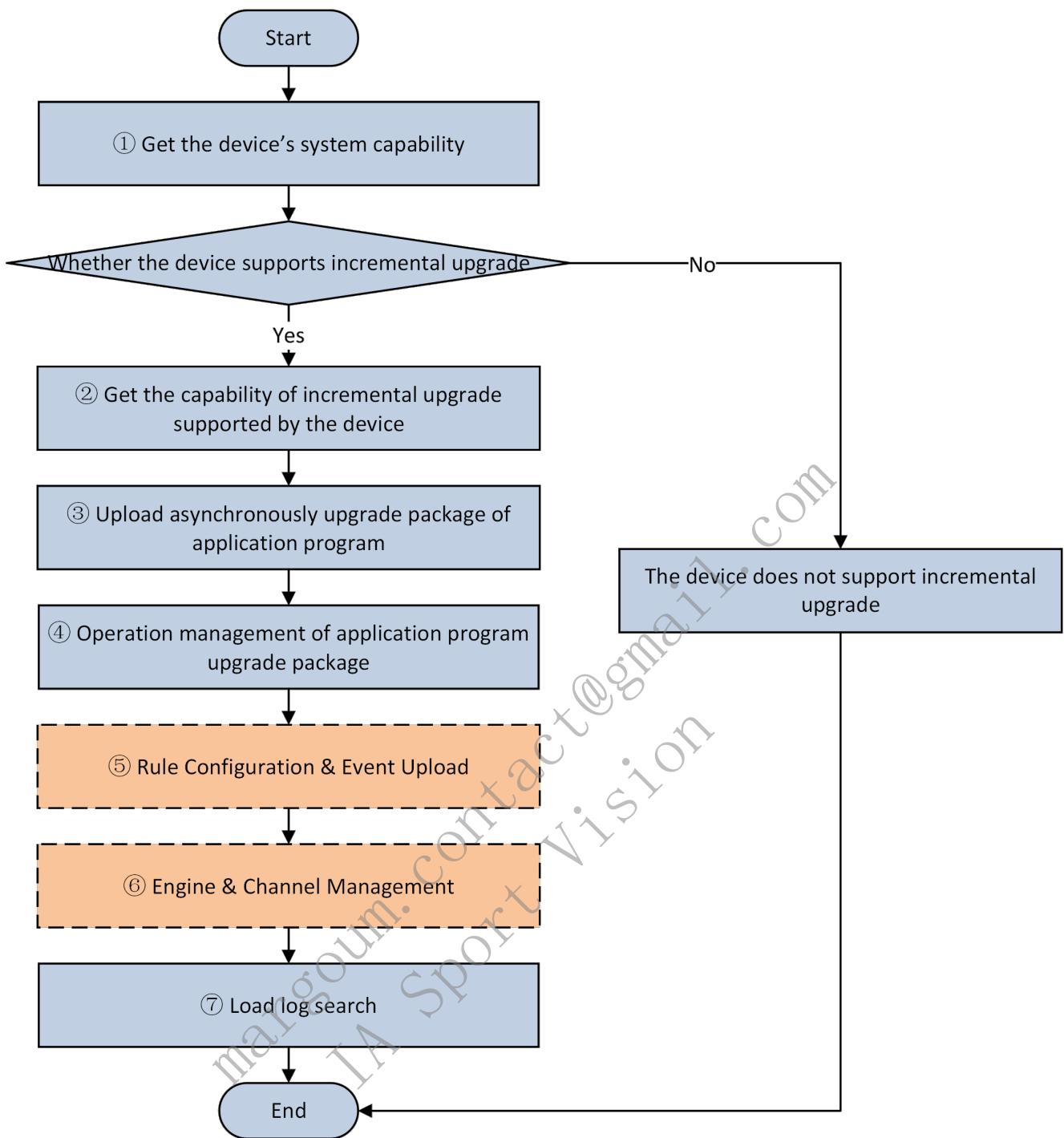
Reboot devices: `PUT /ISAPI/System/reboot`.

## 5.6 HEOP Basics

### 5.6.1 Introduction to the Function

The HEOP integrated by Intelligent Security API (ISAPI) provides runtime environments and management APIs for third-party application programs to connect to and run on network cameras.

### 5.6.2 API Calling Flow



1. Get the device system capability: `GET /ISAPI/System/capabilitie`. If the node `<OpenPlatformCap>` is returned in the capability message and its value is true, it indicates that the HEOP function is supported by the device.
2. Get the HEOP capability supported by the device: `GET /ISAPI/OpenPlatform/capabilities`.
3. Search the HEOP remaining resources: `GET /ISAPI/Custom/OpenPlatform/resource`.
4. Upgrade synchronously the application program package:
  - i. Configure synchronous upgrade of the application program full package: `PUT /ISAPI/Custom/OpenPlatform/uploadApp`.
  - ii. Get the synchronous upgrading progress of the application program full package: `GET /ISAPI/Custom/OpenPlatform/uploadAppStatus`.
5. Application program information management:
  - i. Get page information of application program: `GET /ISAPI/Custom/OpenPlatform/pageItem`.
  - ii. Get the application program information list: `GET /ISAPI/Custom/OpenPlatform/App`.
  - iii. Get the information of a single application program: `GET /ISAPI/Custom/OpenPlatform/App/<AppID>`.
  - iv. Get the logs of the application program: `GET /ISAPI/Custom/OpenPlatform/App/<AppID>/syslog`.
6. Application program operation management:
  - i. Run the application program: `PUT /ISAPI/Custom/OpenPlatform/App/<AppID>/start?engineID=<engineID>`.

ii. Stop running the application program: `PUT /ISAPI/Custom/OpenPlatform/App/<AppID>/stop?engineID=<engineID>`.

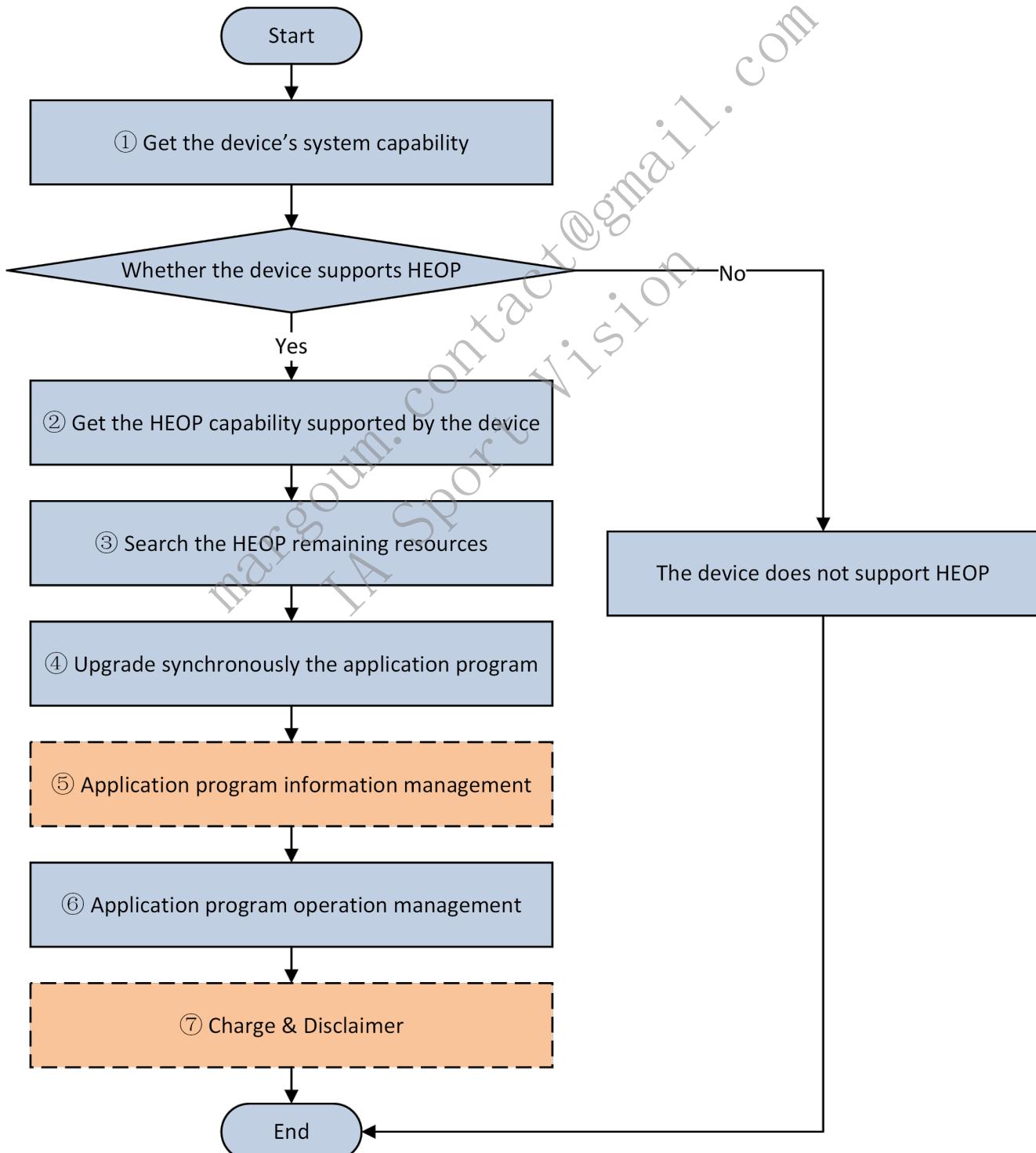
iii. Delete the application program: `DELETE /ISAPI/Custom/OpenPlatform/App/<AppID>/delete`.

## 5.7 Incremental Upgrade

### 5.7.1 Introduction to the Function

If intelligent detection models of the front-end and back-end devices mismatch, the model data collected by the front-end device cannot be used for the comparison on back-end devices. At present, the device package is becoming larger, and the size some device upgrade packages reach level GB, which makes it difficult to upgrade the device (such as the upgrade package delivery, long upgrade time). The unified algorithm switch of front-end and back-end devices is designed to support HEOP framework, which supports patch upgrade of algorithm package.

### 5.7.2 API Calling Flow



1. Get the device system capability: `GET /ISAPI/System/capabilitie`. If the node `<isSupportPatchUpgrade>` is

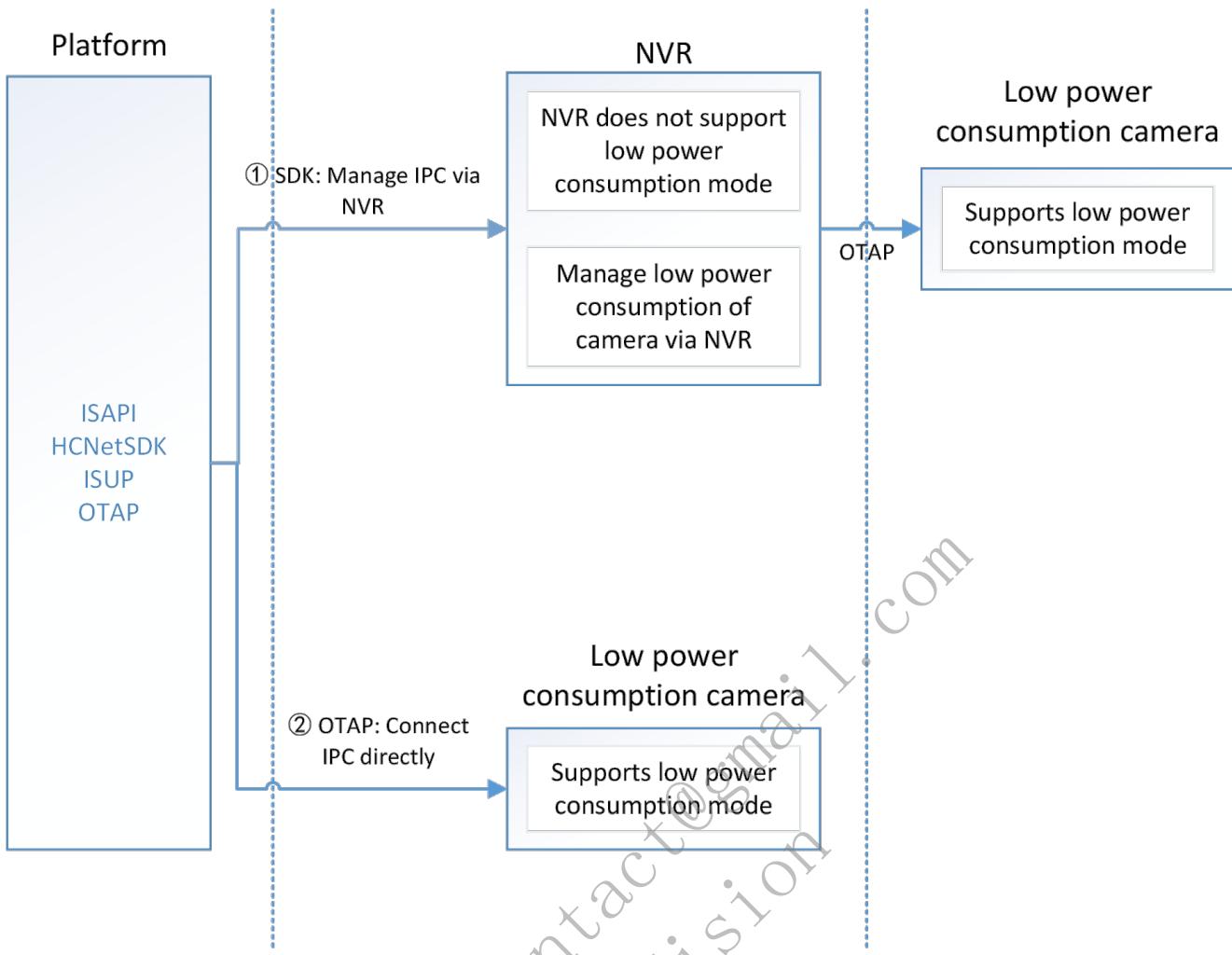
returned in the capability message and its value is "true", it indicates that HEOP Synchronization is supported by the device.

2. Get the incremental upgrade capability supported by device: GET  
`/ISAPI/Custom/OpenPlatform/patchUpgrade/capabilities?format=json`.
3. Upload asynchronously the application program package:
  - i. Upload asynchronously the application program package: POST  
`/ISAPI/Custom/OpenPlatform/uploadApp/async?format=json`.
  - ii. Get the asynchronous uploading progress of the application program package: GET  
`/ISAPI/Custom/OpenPlatform/uploadAppStatus/async/<AppID>?format=json`.
4. Operation management of application program upgrade package:
  - i. Run the specified application program upgrade package: POST  
`/ISAPI/Custom/OpenPlatform/appPackage/<AppUUID>/start`.
  - ii. Stop running the application program upgrade package: PUT  
`/ISAPI/Custom/OpenPlatform/appPackage/<AppUUID>/stop`.
  - iii. Delete the specified application program upgrade package: DELETE  
`/ISAPI/Custom/OpenPlatform/appPackage/<AppUUID>`.
5. Rule Configuration & Event Upload: Reuse the existing protocol system (HCNetSDK and ISAPI), and the device selects whether to provide this function according to different intelligent types based on the existing intelligent rules.
6. Engine & Channel Management:
  - i. Get the information about engine status: GET `/ISAPI/Custom/OpenPlatform/APP/engines?format=json`.
  - ii. Get the information about the application program package that is bound to a specified engine: GET  
`/ISAPI/Custom/OpenPlatform/APP/loading/<engineID>?format=json`.
  - iii. Set the information about the application program package that is bound to a specified engine: PUT  
`/ISAPI/Custom/OpenPlatform/APP/loading/<engineID>?format=json`.
  - iv. Delete the binding between the application program package and the engine: DELETE  
`/ISAPI/Custom/OpenPlatform/APP/unloading/<engineID>?format=json`.
  - v. Unbind the specified application program(s) from the engine: PUT  
`/ISAPI/Custom/OpenPlatform/APP/unloading?format=json`.

## 5.8 Low Power Consumption Management of Network Camera via NVR

### 5.8.1 Introduction to the Function

For NVRs without low power consumption functions, they can be accessed by low power consumption cameras and manage the low power consumption functions including low power consumption management configuration, low power consumption mode, sleep/wakeup operation, etc. via the gateway sub-device solution.



## 5.8.2 API Calling Flow

Sub-device access management: See details of sub-device transparent transmission in Sub-device Transparent Transmission.

1. Check whether the NVR device supports sub-device: `GET /ISAPI/ContentMgmt/InputProxy/channels/capabilities`. If the `isSupportDevIndex` is returned and its value is true, it indicates that the NVR supports the transparent transmission function of sub-device.
2. Get the sub-device Information accessed by NVR: `GET /ISAPI/ContentMgmt/InputProxy/channels?security=<security>&iv=<iv>`. Each digital channel includes the parameter `devIndex` and digital channels from the same physical device have the same `devIndex`.
3. Get the low power consumption function supported by network camera: `GET /ISAPI/System/consumptionMode/capabilities?format=json&devIndex=<devIndex>`. Check whether the functions are supported according to the returned nodes in the response message.

### Note:

- For operations on sub-device via NVR by the platform, the transparent transmission command carries the sub-device information `devIndex` in the request information, i.e., the URL parameter `devIndex` is suffixed to all ISAPI URLs.
- During ISAPI packaging, the URL parameter `devIndex=<devIndex>` suffixed to the URL represents a direct command to the sub-device; without `devIndex=<devIndex>`, it is a command to the NVR.

## 5.9 Low Power Consumption Mode Management

### 5.9.1 Introduction to the Function

In low power consumption mode, the device power consumption is lowered to improve the chip reliability and extend the battery life. For devices deployed in the wild or overhead that cannot be maintained frequently, the low power consumption management helps to extend the device service life and reduce the maintenance cost.

As the low power consumption devices are usually deployed in the wide without stable or available wired network but mostly cellular data without fixed IP address, the ISUP service should be enabled on deployment for device monitoring and management. Besides, when the device is sleeping and manual wakeup is required, it can only be woken up via the ISUP protocol. Therefore, it is recommended to integrate the ISUP protocol first for integration of low power consumption management function. So far the low power consumption management includes multiple management modes and can be selected according to the specific device and scenario. See details of consumption modes in Terms and Definitions.

#### Note:

- In sleeping mode, the device only maintain the heartbeat keepalive via the ISUP protocol (V5.0). Live view, playback, two-way audio, configuration, and other operations are not supported.
- The device will remain asleep for at most 5 minutes.
- If there are online users accessing functions like live view and playback, the device will not go to sleep.

## 5.9.2 API Calling Flow

1. Power consumption management: configure power consumption mode.
2. Sleep management: configure low power consumption sleep mode; configure sleeping schedule; configure auto-sleep countdown.
3. When the device is sleeping, operations such as live view and playback will fail.
4. Wakeup management: send a wakeup command to device via the ISUP protocol (V5.0); configure the wakeup schedule.

## 5.10 Management of Devices Access Serial Port

### 5.10.1 Introduction to the Function

Information management of device accessed the serial are as follows: 1. Configure manufacturer, type, and model information of the specific serial port access device. 2. Search for the device type or model supported by the specific serial port.

### 5.10.2 API Calling Flow

1. **Check whether the device supports information management of devices access the serial port:** GET /ISAPI/System/Serial/capabilities; If <isSupportDeviceInfo> is returned, it indicates that the device supports information configuration of devices access the serial port.

3. **Set the information of devices access the serial port:**

Get the capability of device information parameters of a single serial port: GET /ISAPI/System/Serial/ports/<portID>/deviceInfo?format=json;

Get device information parameters access single serial port: GET /ISAPI/System/Serial/ports/<portID>/deviceInfo?format=json;

Set device information parameter of single serial port: PUT /ISAPI/System/Serial/ports/<portID>/deviceInfo?format=json;

4. **Check whether the device supports linking information of devices access the serial port:** GET /ISAPI/System/Serial/capabilities; If <isSupportSearchDeviceInfoRelations> is returned, it indicates that the device supports searching for linked information od devices access the serial port.

5. **Search for linked information of devices access the serial port:**

Get the capability of searching for linked parameters of information of devices access a single serial port: GET /ISAPI/System/Serial/ports/<portID>/searchDeviceInfoRelations/capabilities?format=json;

Search for linked parameters of information of devices access a single serial port: POST /ISAPI/System/Serial/ports/<portID>/searchDeviceInfoRelations?format=json;

## 5.11 Module Sleep/Wakeup

### 5.11.1 Introduction to the Function

Besides the system-level chip, devices also consist of other modules (e.g., fall detection radar and vital sign detection radar consist of the system-level chip and radar module). When the integrator sends a sleep command to device, the device puts the module to sleep but the system-level chip is still operating normally. The integrator can wake up the module via the ISAPI protocol.

## 5.11.2 API Calling Flow

1. (Client) Get the system capability: `GET /ISAPI/System/capabilities`. If `isSupportConsumptionMode` is returned and its value is `true`, it indicates that the device supports switching power consumption modes.
2. (Client) Get the power consumption mode capability: `GET /ISAPI/System/consumptionMode/capabilities?format=json`. If `isSupportSleep` is returned and its value is `true`, it indicates that the device supports sleep; if `isSupportWakeUp` is returned and its value is `true`, it indicates that the device supports wakeup; if `isSupportSleeping` is returned and its value is `true`, it indicates that the device supports sleeping mode.
3. Put the module to sleep immediately: `PUT /ISAPI/System/consumptionMode/Sleep?format=json`.
4. Wake up the module immediately: `PUT /ISAPI/System/consumptionMode/WakeUp?format=json`.
5. Get the current sleep status: `GET /ISAPI/System/consumptionMode/SleepStatus?format=json`.

## 5.12 Mutually Exclusive Functions

### 5.12.1 Introduction to the Function

Some functions are mutually exclusive due to the device performance (for example, function A and function B cannot run at the same time, and only one of them is allowed at one time).

### 5.12.2 API Calling Flow

The following three APIs are available for getting the mutually exclusive function information:

1. Get the information of mutually exclusive functions: `GET /ISAPI/System/mutexFunction/capabilities?format=json`. You can call this URL to get the list of existing mutually exclusive functions supported by the device. Note: NVR devices only support setting exclusive function "perimeter" (perimeter protection), and do not support "linedetection" (line crossing detection), "fielddetection" (intrusion detection), "regionEntrance" (region entrance), and "regionExiting" (region exiting).
2. Search for the functions that are mutually exclusive with a specified function: `POST /ISAPI/System/mutexFunction?format=json`. Based on the list of mutually exclusive functions returned by `GET /ISAPI/System/mutexFunction/capabilities?format=json`, you can search for the mutual exclusion status of a specified function.
3. Get the mutual exclusion information when device function exception occurs: `GET /ISAPI/System/mutexFunctionErrorMsg`. After getting the error code, you can call this API to get the current mutually exclusive functions.

## 5.13 Sensor-Triggered Wakeup

### 5.13.1 Introduction to the Function

Connect sensors to the low power consumption device via RS-485 interface. When the device is sleeping, if data collected by the sensors exceeds the configured threshold condition, the device will wake up in full power consumption mode and upload alarm information and relevant pictures/videos according to the triggering rules. This function is commonly applied to the wild environment.

### 5.13.2 API Calling Flow

1. (Client) Get the system capability: `GET /ISAPI/System/capabilities`. If `isSupportConsumptionMode` is returned, it indicates that the device supports switching power consumption modes.

2. (Client) Get the power consumption mode capability: GET `/ISAPI/System/consumptionMode/capabilities?format=json`. If `isSupportSensorTriggeringWakeParams` is returned, it indicates that the device supports configuring sensor-triggered wakeup rules; if `isSupportSensorTriggeringWakeRange` is returned, it indicates that the device supports getting the sensor-triggered wakeup range.
3. (Client) Get the sensor-triggered wakeup range: GET `/ISAPI/System/consumptionMode/SensorTriggeringWakeRange?format=json`; `sensorDataType` is the sensor-triggered wakeup sensor data supported by device. See details of sensor data in the dictionary file obtained from GET `/ISAPI/IoTGateway/SensorDataFlowMgr/SearchSensorDictFile?format=json`.
4. (Client) Configure sensor-triggered wakeup rules for device:

Get capability: GET `/ISAPI/System/consumptionMode/SensorTriggeringWakeParams/capabilities?format=json`. Pay attention to the range of `@size` and `ruleID` in the response message; other nodes represents the variable types commonly used for sensor-triggered wakeup rules. See details of sensor data in the dictionary file obtained from Step 3.

Get sensor-triggered wakeup rule parameters: GET `/ISAPI/System/consumptionMode/SensorTriggeringWakeParams?format=json`.

Configure sensor-triggered wakeup rules: PUT `/ISAPI/System/consumptionMode/SensorTriggeringWakeParams?format=json`.

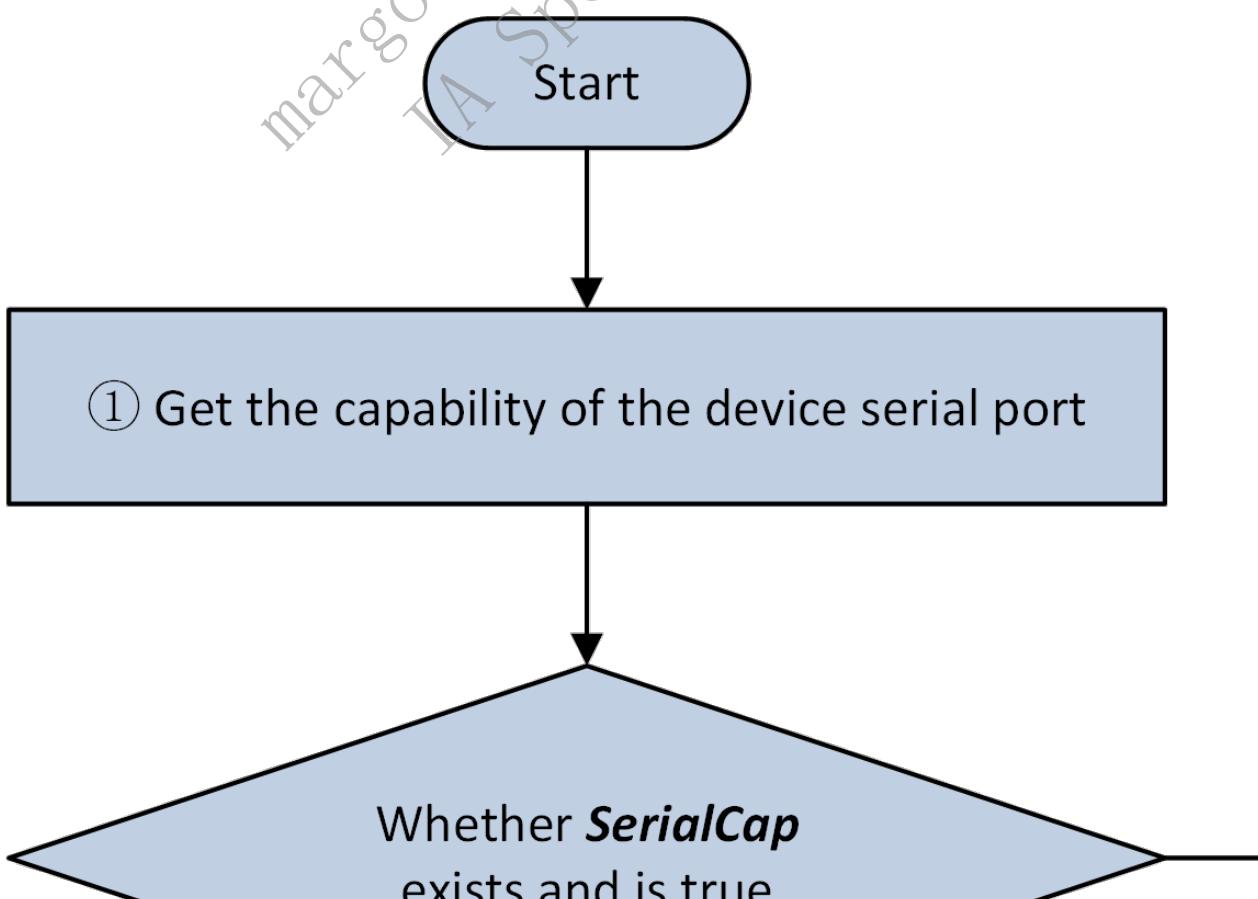
**Note:** If sensors with data exception are not disconnected, the device will wake up frequently. Therefore, after 3 times of rules being triggered for the same sensor within a day, the device will not wake up anymore.

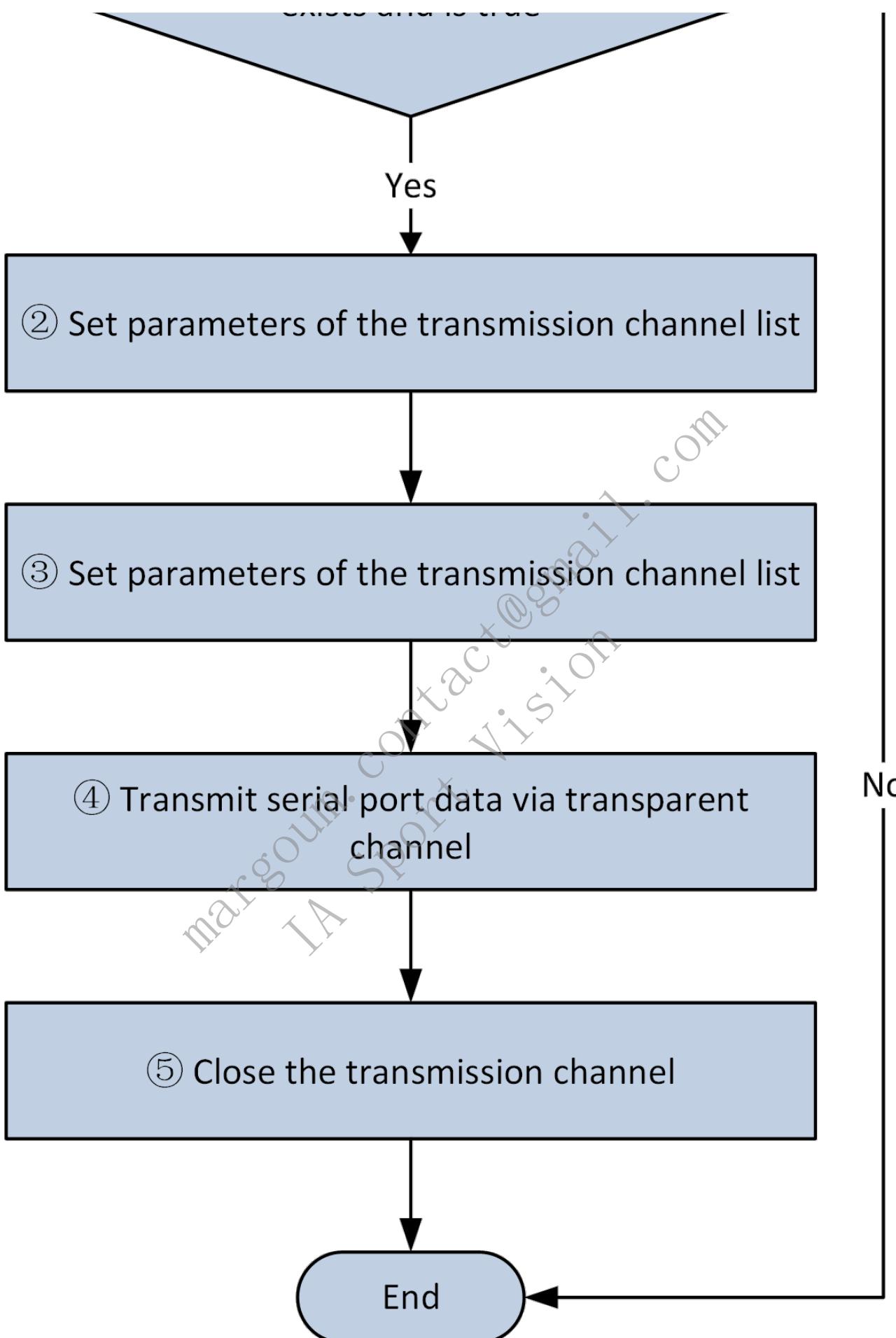
## 5.14 Serial Port Data Transparent Transmission

### 5.14.1 Introduction to the Function

RS485, RS422 and RS232 serial ports external to the device are used as transparent channels to transmit serial port data.

### 5.14.2 API Calling Flow





**1. Check whether the device supports serial port data transmission:**

Get the capability of the device serial port: `GET /ISAPI/System/capabilities`. If `SerialCap` is returned and the value is true, it indicates that the device supports the functions of the serial port.

**2. Set parameters of the transmission channel list:**

Get parameters of the specific transmission channel: GET

/ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>;

Configure parameters of the specific transmission channel: GET

/ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>.

### 3. Open the transmission channel: PUT

/ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>/open;

### 4. Transmit serial port data via transparent channel:

Receive data uploaded by device serial port through transmission channel: GET

/ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>/transData;

Send data to device serial port through transmission channel: PUT

/ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>/transData;

### 5. Close the transmission channel: PUT

/ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>/close;

## 5.15 Serial Port Parameter Configuration

### 5.15.1 Introduction to the Function

Serial port parameter configuration.

### 5.15.2 API Calling Flow

#### 1. Check whether the device supports configuring parameters of the serial port:

Get the capability of device serial port: GET /ISAPI/System/capabilities; If is returned and its value is true, it indicates that the device supports functions of serial port.

#### 3. Get parameters of all serial ports:

Get the capability of all serial ports: GET /ISAPI/System/Serial/capabilities;

Get control parameters of all serial ports: GET /ISAPI/System/Serial/ports?permissionController=<indexID>;

#### 4. Set control parameters of single serial port:

Get control parameters of single serial port: GET /ISAPI/System/Serial/ports/<portID>?permissionController=<indexID>;

Configure control parameters of single serial port: PUT /ISAPI/System/Serial/ports/<portID>?permissionController=<indexID>;

#### 5. Get the status of single serial port: GET /ISAPI/System/Serial/ports/<portID>/status

## 5.16 Sleep

In full power consumption mode or real-time low power consumption mode, you can configure the sleep conditions. Once the conditions are met, the device will go to sleep.

### 5.16.1 Low Battery Sleep

#### 5.16.1.1 Introduction to the Function

In full power consumption mode or real-time low power consumption mode, the device goes to sleep when the battery is lower than the sleep threshold.

#### 5.16.1.2 API Calling Flow

1. (Client) Get the system capability: GET /ISAPI/System/capabilities. If isSupportConsumptionMode is returned, it indicates that the device supports switching power consumption modes.
2. (Client) Get the capability of switching power consumption modes: GET /ISAPI/System/consumptionMode/capabilities?format=json. If Sleep is returned, it indicates that the device supports low battery sleep.

3. (Client) Configure the low battery sleep threshold powerThreshold: PUT /ISAPI/System/consumptionMode?format=json.

### 5.16.1.3 Exception Handling

Error Code:

statusCode	statusString	subStatusCode	errorCode	errorMsg	Description	Remarks
4	Invalid Operation	deviceSleep	0x40002097	Please wake up device.	The device is sleeping.	

## 5.16.2 Sleep Schedule

### 5.16.2.1 Introduction to the Function

In full power consumption mode or real-time low power consumption mode, the device goes to sleep according to the configured sleeping time periods, wakes up and captures pictures at intervals, and goes back to sleep afterwards.

### 5.16.2.2 API Calling Flow

1. (Client) Get the system capability: GET /ISAPI/System/capabilities. If isSupportConsumptionMode is returned, it indicates that the device supports switching power consumption modes.
2. (Client) Get the capability of switching power consumption modes: GET /ISAPI/System/consumptionMode/capabilities?format=json. If TimingSleep is returned, it indicates that the device supports sleep schedule.,
3. (Client) Configure the sleep schedule TimingSleep according to supported response parameters: PUT /ISAPI/System/consumptionMode?format=json.

#### Note:

- Besides the general device sleep, a certain module in device can be specified to sleep, e.g., the thermal imaging camera module (thermalMovement) and visible light camera module (visibleMovement) of dual-channel thermal imaging devices. Wake up the thermal imaging camera module manually: PUT /ISAPI/System/consumptionMode/manualArouseThermalMovement?format=json.

### 5.16.2.3 Exception Handling

Error Code:

statusCode	statusString	subStatusCode	errorCode	errorMsg	Description	Remarks
4	Invalid Operation	deviceSleep	0x40002097	Please wake up device.	The device is sleeping.	
4	Invalid Operation	restoreThermalimagingVideoFailed	0x40002150	The device is under over-temperature protection and cannot restore the thermal imaging video.	The device is under over-temperature protection and cannot restore the thermal imaging video.	

## 5.16.3 Auto Sleep Countdown

### 5.16.3.1 Introduction to the Function

After the set time, the device will go to sleep automatically.

### 5.16.3.2 API Calling Flow

In full power consumption mode, the device can perform auto sleep via the following 2 methods. Integrate the function according to the capability returned by device.

Method 1:

1. (Client) Get the system capability: `GET /ISAPI/System/capabilities`. If `isSupportConsumptionMode` is returned, it indicates that the device supports switching power consumption modes.
2. (Client) Get the capability of power consumption modes: `GET /ISAPI/System/consumptionMode/capabilities?format=json`. If `AutoSleep` is returned, it indicates that the device supports auto sleep.
3. (Client) Configure auto sleep `AutoSleep` according to supported response parameters: `PUT /ISAPI/System/consumptionMode?format=json`.

Method 2:

1. (Client) Get the system capability: `GET /ISAPI/System/capabilities`. If `isSupportSleep` is returned, it indicates that the device supports sleep.
2. (Client) Get the capability of sleep: `GET /ISAPI/System/sleep/capabilities?format=json`.
3. (Client) Configure sleep according to supported response parameters: `PUT /ISAPI/System/sleep?format=json`.

### 5.16.3.3 Exception Handling

Error Code:

statusCode	statusString	subStatusCode	errorCode	errorMsg	Description	Remarks
4	Invalid Operation	deviceSleep	0x40002097	Please wake up device.	The device is sleeping.	

## 5.17 Wakeup Schedule

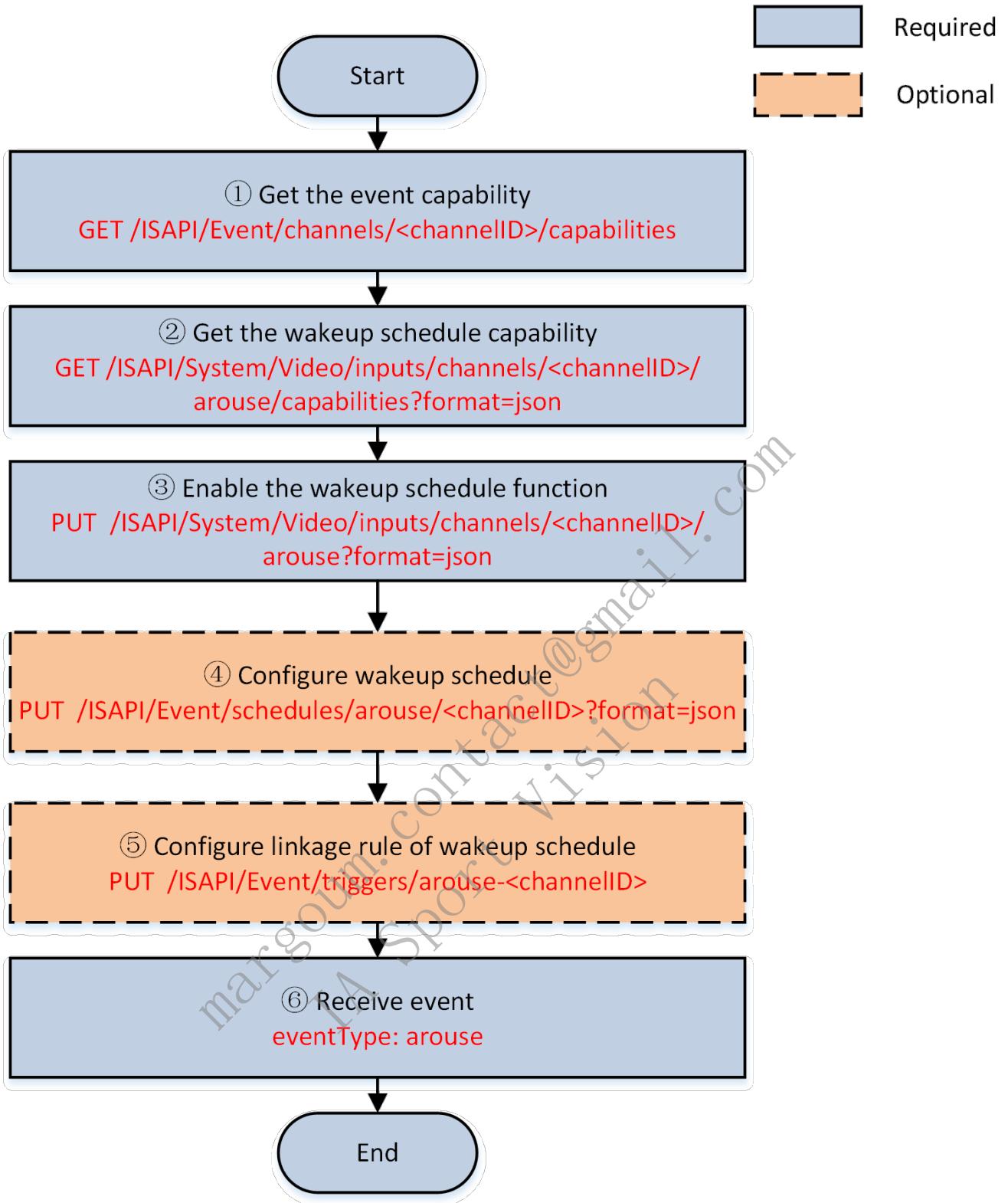
### 5.17.1 Introduction to the Function

So far there are 4 methods to wake up devices in sleeping mode:

1. After the sleep schedule is configured, the device auto wakes up in wakeup mode, or auto wakes up according to the configured capture interval in sleeping mode.
2. For devices registered with ISUP service (V5.0), they wake up by applying ISUP command to themselves in sleeping mode.
3. (Client) Configure the device wakeup schedule and the device will wake up at set time.
4. Connect the low power consumption device with sensors. (Client) Configures the threshold condition for sensor-triggered device wakeup. Once the data collected by sensor exceeds the threshold, the device will wake up.

If the integration scenario requires configuring regular sleep schedule and capturing pictures as scheduled, Method 1 is recommended and see details in the API calling flow of sleep schedule; for client manually waking up the device, Method 2 is recommended and see details in the user manual of HCISUPSDK; if the integration scenario requires waking up device regularly to capture pictures when it is consistently sleeping, Method 3 is recommended and shown below.

### 5.17.2 API Calling Flow



1. (Client) Get the event capability: `GET /ISAPI/Event/channels/<channelID>/capabilities`. If `eventType` is returned and its value includes "arouse", it indicates the channel supports wakeup schedule.
2. (Client) Get the wakeup schedule capability: `GET /ISAPI/System/Video/inputs/channels/<channelID>/arouse/capabilities?format=json`.
3. (Client) Enable the wakeup schedule function: `PUT /ISAPI/System/Video/inputs/channels/<channelID>/arouse?format=json`.
4. (Optional) (Client) Configure wakeup schedule: `PUT /ISAPI/Event/schedules/arouse/<channelID>?format=json`.
5. (Optional) (Client) Configure linkage rule of wakeup schedule: `PUT /ISAPI/Event/triggers/arouse-<channelID>`.
6. The device wakes up as scheduled and uploads the event `eventType: arouse`. If the capture function is configured in Step 3, the uploaded event will include pictures captured when the device is awake.

**Note:**

- Check if single-channel devices support the wakeup schedule function: If Step 1 and 2 are both not supported, the function is not supported.
- When the wakeup schedule is enabled on the client and the sleep schedule is configured (scheduled capture enabled), the scheduled wakeup event will be triggered on scheduled capture.
- Cancel the device low battery sleep via Connect: `PUT /ISAPI/System/consumptionMode/cancelSleep?format=json`.

## 6 Video (General)

---

### 6.1 . OSD

#### 6.1.1 Introduction to the Function

On-screen display is applied in the display interface of cameras, and is used to display some characters or graphics on the video to provide some information to the audience. After OSD configurations on the client software, the configured information text will be overlaid on the video stream of cameras and displayed on the screen.

The typical application scenarios of dynamic OSD: elevator floor number overlay for elevator monitoring, characters overlay for toll stations on highway/road, notification information overlay during live video programs. The typical application scenarios of static OSD: fixed information (such as camera position, date, time) overlay on the video signal.

#### 6.1.2 API Calling Flow

##### 6.1.2.1 OSD

1. The client software gets the OSD capability of a specified channel: `GET /ISAPI/System/Video/inputs/channels/<channelID>/overlayscapabilities`.
2. The client software configures OSD parameters, including whether to enable OSD, custom text overlay, time overlay, and overlay format, for a specified channel.

Get: `GET /ISAPI/System/Video/inputs/channels/<channelID>/overlays`.

Set: `PUT /ISAPI/System/Video/inputs/channels/<channelID>/overlays`.

##### Remarks:

- The configuration URL supports various parameters, including custom text overlay, channel name overlay, time overlay, overlay format, and so on. Some of these parameters can also be configured by other URLs, as shown below: `TextOverlayList` can also be configured via URL of custom text overlay.

`channelNameOverlay` can also be configured via URL of channel name overlay.

`DateTimeOverlay` can also be configured via URL of channel time overlay.

- The overlaid information alignment mode (defined by `alignment`) refers to the alignment mode of custom text:

`customize`: custom mode. The client software sends the overlay position coordinates of custom text to the device, and the device will overlay the characters on the stream for display according to the position coordinates.

`alignRight`: right align. The device will display custom text on the right boundary of screen.

`alignLeft`: left align. The device will display custom text on the left boundary of screen.

`allRight`: all right align. The device will display all overlay contents, including custom text, the channel name, the time on the right boundary of screen.

`allLeft`: all left align. The device will display all overlay contents, including custom text, the channel name,

the time on the left boundary of screen.

#### 6.1.2.2 Custom Text Overlay

1. The client software gets the OSD capability of a specified channel: GET  
`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/capabilities`. If `TextOverlayList` is returned, it indicates that custom text overlay is supported and `size` defines the maximum number of custom texts.

2. The client software configures custom text OSD. `textID` refers to the custom text No., and the value range is: [1, `size`]:

Create a custom text for a specified channel: POST

`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/text.`

Get overlay parameters of a custom text for a specified channel: GET

`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/text/<textID>.`

Set overlay parameters of a custom text for a specified channel: PUT

`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/text/<textID>.`

Delete a custom text: DELETE `/ISAPI/System/Video/inputs/channels/<channelID>/overlays/text/<textID>.`

Get overlay parameters of all custom texts for a specified channel: GET

`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/text.`

Set overlay parameters of all custom texts for a specified channel: PUT

`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/text.`

Delete all custom texts for a specified channel: DELETE

`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/text.`

#### 6.1.2.3 Channel Name Overlay

1. The client software gets the OSD capability of a specified channel: GET  
`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/capabilities`. If `channelNameOverlay` is returned, it indicates that channel name overlay is supported.

2. The client software configures the channel name overlay, including whether to enable channel name overlay, overlay position, and so on.

Get channel name overlay parameters: GET

`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/channelNameOverlay.`

Set channel name overlay parameters: PUT

`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/channelNameOverlay.`

#### 6.1.2.4 Channel Time Overlay

1. The client software gets the OSD capability of a specified channel: GET  
`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/capabilities`. If `DateTimeOverlay` is returned, it indicates that channel time overlay is supported.

2. The client software configures the channel time overlay, including whether to enable channel time overlay, overlay position, date format, and so on.

Get channel time overlay parameters: GET

`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/dateTimeOverlay.`

Set channel time overlay parameters: PUT

`/ISAPI/System/Video/inputs/channels/<channelID>/overlays/dateTimeOverlay.`

## 6.2 AIOP Metadata

### 6.2.1 Introduction to the Function

The client obtains the AIOP metadata together with the audio and video streams via RTSP, and then the client can selectively overlay the intelligent structured information of metadata on the video, or use the metadata for secondary development.

The AIOP detection rule is based on the training model and model description files imported to the device. Once the training model is changed, the attributes and category of targets carried by AIOP metadata changes respectively. The client will analyze the target attributes in AIOP metadata based on the latest training model information.

#### 1. Metadata Management

- The client enables uploading the AIOP detection event. The device pushes the model and target detection information.

#### 2. RTSP Interaction

- The client analyzes the description file information of AIOP model, target coordinates, target attributes, and other information.

### 6.2.2 API Calling Flow

1. The steps from 1st to 6th are the same as the calling flow in Chapter 4.2. To obtain the AIOP metadata, the client needs to subscribe the metadata event of the video channel: AIOP detection event (event type ID: AIOP).
2. For operations of streaming via RTSP, refer to the Chapter 4.3 Integration Based on RTSP. Send a PLAY command to parse the metadata packet.
  - After sending a PLAY command, the client will first receive the metadata packet (triggered when the algorithm is updated) sent by the device. See 5.3.1 for detailed message.
  - And then the client will receive the metadata packet (triggered when the target is detected) sent by the device. See 5.3.2 for detailed message.

Notes:

- During the streaming process, if the AIOP model is changed, the device will send the metadata packet containing the model information again on the established RTSP session, and the client should support parsing the newly sent packet. You must use the latest attributes to parse the metadata packet sent afterwards.

### 6.2.3 Message

#### 6.2.3.1 Message Uploaded when the Algorithm is Updated

```

{
    "Metadata": {
        "type": "algoPackageUpdate",
        "subType": "AIOP",
        "time": "2017-01-01T00:00:00+08:00",
        "DevInfo": {
            "ipAddress": "192.168.0.64",
            "portNo": "554",
            "macAddress": "28:57:be:ee:d1:5a",
            "channel": "1"
        },
        "ModelPackageInfo": [
            {
                "modelId": "34d8670d8e6f4d6899c4c3c3c277714f",
                "attributes": [
                    {
                        "attributeName": "1",
                        "attributeTag": "",
                        "subAttributes": []
                    },
                    {
                        "attributeName": "2",
                        "attributeTag": "",
                        "subAttributes": []
                    }
                ]
            },
            {
                "modelId": "758cf1d0446840419808a4be4d637db5",
                "attributes": [
                    {
                        "attributeName": "0",
                        "attributeTag": "",
                        "subAttributes": [
                            {
                                "attributeName": 0,
                                "attributeTag": ""
                            },
                            {
                                "attributeName": 1,
                                "attributeTag": ""
                            }
                        ]
                    }
                ]
            }
        ]
    }
}

```

#### Field Description:

Field	Value	Required	Description
type	algoPackageUpdate	Yes	It is uploaded when the algorithm is updated.
subType	AIOP	Yes	AIOP detection
time	-	Yes	Format: ISO 8601. It is accurate to millisecond, e.g., 17-01-11T17:43:30.256+08:00
DevInfo	-	Yes	Device Information
ModelPackageInfo	-	Yes	Model description

#### DevInfo

Field	Value	Required	Description
ipAddress	string	Yes	IPv4 address
portNo	string	Yes	Port number, value range: [1,65535]
macAddress	string	Yes	Format: ISO 8601. It should be accurate to millisecond, e.g., 2017-01-11T17:43:30.256+08:00
channel	string	Yes	Channel number

#### ModelPackageInfo:

Field	Value	Required	Description
modelId	string	Yes	Model ID, which is unique.
attributeName	string	Yes	Attribute name
attributeTag	string	Yes	Attribute tag

### 6.2.3.2 Message Uploaded when the Target is Detected

```
{
  "Metadata": {
    "type": "activityTarget",
    "subType": "AIOP",
    "time": "2017-01-01T00:00:00+08:00",
    "DevInfo": {
      "ipAddress": "192.168.0.64",
      "portNo": "554",
      "macAddress": "28:57:be:ee:d1:5a",
      "channel": "1"
    },
    "TargetDetection": {
      "TargetList": [
        {
          "targetID": "1",
          "RegionList": [
            {
              "x": 0.001,
              "y": 0.001
            }
          ],
          "PropertyList": [
            {
              "description": "confidence",
              "value": "100"
            }
          ],
          "AttributeInfo": [
            {
              "modelID": "34d8670d8e6f4d6899c4c3c277714f",
              "attributeName": "1"
            },
            {
              "modelID": "758cf1d0446840419808a4be4d637db5",
              "attributeName": "0",
              "subAttributeName": "0"
            }
          ]
        }
      ]
    }
  }
}
```

#### Field Description:

Descriptions of type, subType, time, DevInfo are same as those in 5.3.1

Field	Value	Required	Description
TargetDetection	object	Yes	Target detection information
TargetList	array	Yes	Target list. Up to 64 targets are supported.
targetID	string	No	Target ID
RegionList	array	Yes	Target coordinates, which consists of multiple points (X-coordinates and Y-coordinates).
PropertyList	array	No	Target property list
AttributeList	array	No	Target attribute list

#### RegionList

Field	Value	Required	Description
x	float	Yes	The origin is the upper-left corner of the screen. Range: [0.000,1.000].
y	float	Yes	The origin is the upper-left corner of the screen. Range: [0.000,1.000].

## PropertyList

description	Value	Required	Description
confidence	string	No	Target confidence. Value: [0,100].

## AttributeList

Field	Value	Required	Description
modelID	string	Yes	Corresponds to the "ModelPackageInfo.modelID" in 5.3.1
attributeName	string	Yes	Corresponds to the "ModelPackageInfo.attributes.attributeName" in 5.3.1
subAttributeName	string	No	Corresponds to the "ModelPackageInfo.attributes.subAttributes.attributeName" in 5.3.1

## 6.3 Async Capturing Manually

### 6.3.1 Introduction to the Function

The client sends a capture command to the device, and the device immediately responds by returning the URL of the storage server where the captured picture is stored. The client can request image data from the storage server based on the received URL. It is suitable for the limited network bandwidth between the client and the device or the client centrally manages a large number of devices. The client can choose to use the image URL address carried in the alarm content to request pictures from the storage server when the network conditions improve or there is no concurrency of a large number of requests, so as to reduce the probability of image download failure and the risk of network congestion.

### 6.3.2 API Calling Flow

1. The client gets the system capability: GET /ISAPI/System/capabilities, and if `isSupportSnapshotAsync` is returned and its value is `true`, async manual capturing shall be supported.
2. (Optional) The client gets the async manual capturing capability of a specified channel: GET /ISAPI/Streaming/channels/<channelID>/picture/async/capabilities?format=json.
3. The client controls the specified channel to manually and asynchronously capture pictures: GET /ISAPI/Streaming/channels/<channelID>/picture/async?format=json&imageType=<imageType>&URLType=<URLType>, and the device returns the URL of the storage server for capturing pictures.

#### Remarks:

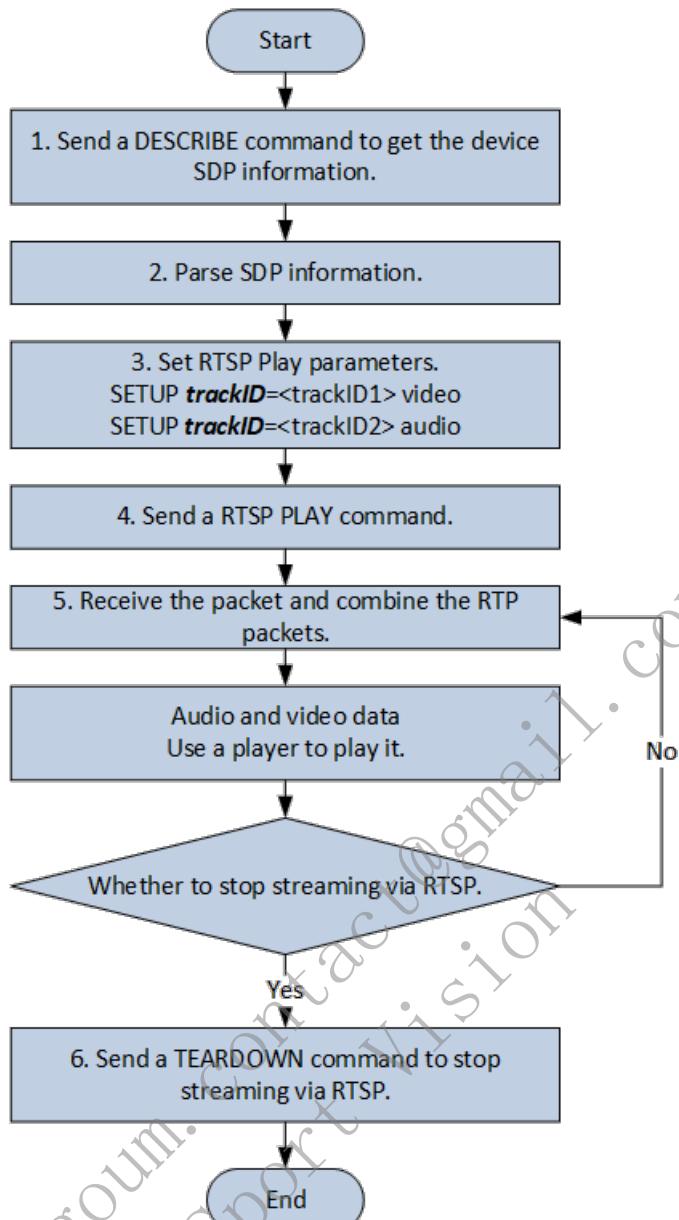
- If the step 2 returns a failure during the integration, it is recommended that you check whether the channel supports async manual capturing according to steps 1 and 3.

## 6.4 Live View

### 6.4.1 Introduction to the Function

Supports getting and setting stream media parameters of devices such as resolution, coding format, and stream type. Supports streaming from products via RTSP (Real Time Streaming Protocol, see details in [RFC 7826](#)).

### 6.4.2 API Calling Flow



1. A client sends RTSP DESCRIBE commands such as `DESCRIBE /ISAPI/Streaming/channels/101 RTSP/1.0`; Digest authentication with devices is required.
2. The client parses the media SDP information returned by the device.
3. Set RTSP play parameters, set the track ID parsed from SDP information via SETUP commands. For example, `trackID=1` indicates videos while `trackID=2` indicates audios.
4. The client sends an RTSP PLAY command, and device will send audio stream, video stream, and metadata in the format of `PLAY /ISAPI/Streaming/channels/101 RTSP/1.0`.
5. The client receives the RTP package sent by device. Divided RTP packets should be assembled on the client before being parsed.
6. The client sends the command RTSP TEARDOWN to stop streaming.

Notes:

- Digest authentication is required in RTSP playback. The method is the same as that of ISAPI digest authentication.
- The address format for streaming from devices is `rtsp:// <host>[:port]/ISAPI/Streaming/channels/<ID>`, of which `<host>` is the device IP address, `[:port]` is optional, and 554 by default, `<ID>` is the device channel ID \* 100 + stream type (1-main stream, 2-sub-stream, 3-third stream); for example, the IP address of the target device is `172.7.203.11`, and if the channel number is 172, the streaming address will be `rtsp://172.7.203.11:554/ISAPI/Streaming/channels/1701`.
- RTSP also supports containing user names and passwords in URL. The format is `rtsp://username:password@[address]:[port]/Streaming/Channels/[id](?parm1=value1&parm2-=value2...)`, such as `/Streaming/Channels/101?transportmode=unicast`.

### 6.4.3. Example

1. A client sends an RTSP DESCRIBE request.

```
DESCRIBE rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101 RTSP/1.0  
CSeq:0  
Accept:application/sdp  
User-Agent:NPKPlayer-1.00.00.081112
```

2. Server responds that authentication is required.

```
RTSP/1.0 401 Unauthorized
CSeq: 0
WWW-Authenticate: Digest realm="3521781c29acb312330dd668", nonce="026019333", algorithm="MD5"
```

3. The client sends an RTSP DESCRIBE request with authentication information again.

DESCRIBE rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101 RTSP/1.0  
CSeq:1  
Accept:application/sdp  
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101", response="76a2c9c5b8edbd49838013cf1cf27941"  
User-Agent:NKPlayer-1.00.00.081112

4. The device responds to SDP information.

5. The client sends RTSP SETUP requests, and the server responds to them.

SETUP rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101/trackID=1 RTSP/1.0  
CSeq:2  
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101", response="ff343f5ff82deb028dd9b4932cc44201"  
Transport:RTP/AVP/TCP;unicast;interleaved=0-1:ssrc=0  
User-Agent:NikPlayer-1.00.00.081112

```
RTSP/1.0 200 OK
Session: 1127293610;timeout=60
Transport: RTP/AVP/TCP;unicast;interleaved=0-1:ssrc=433122aa
CSeq: 2
Accept-Ranges: NPT
Media-Properties: No-Seeking, Time-Progressing, Time-Duration=0
Date: Tue, 17 Nov 2020 02:09:45 GMT
```

```
SETUP rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101/trackID=2 RTSP/1.0
CSeq:3
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101",
response="ff343f5ff82deb028dd9b4932cc44201"
Session:1127293610;timeout=60
Transport:RTP/AVP/TCP;unicast;interleaved=2-3:ssrc=0
User-Agent:NKPlayer-1.0.00.081112
```

```
RTSP/1.0 200 OK
Session: 1127293610;timeout=60
Transport: RTP/AVP/TCP;unicast;interleaved=2-3:ssrc=433122ab
CSeq: 3
Accept-Ranges: NPT
Media-Properties: No-Seeking, Time-Progressing, Time-Duration=0
Date: Tue, 17 Nov 2020 02:09:45 GMT
```

6. The client sends an RTSP PLAY request.

```
PLAY rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101 RTSP/1.0
CSeq:4
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101",
response="24edf8a6ff3ef767f7c49d1c847200bd"
Session:1127293610;timeout=60
Range:npt=0.00000-0.00000
User-Agent:NKPlayer-1.0.00.081112
```

7. The server sends audio and video stream data.

```
RTSP/1.0 200 OK
Session: 1127293610
CSeq: 4
Date: Tue, 17 Nov 2020 02:09:45 GMT
$. ....d1.w....c....".T....g....).i.....a....7.S..~J.....X....X.
```

8. The client sends an RTSP TEARDOWN request, and the server responds to it.

```
TEARDOWN rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101 RTSP/1.0
CSeq:5
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101",
response="24edf8a6ff3ef767f7c49d1c847200bd"
Session:1127293610;timeout=60
Range:npt=0.00000-0.00000
User-Agent:NKPlayer-1.0.00.081112
```

```
RTSP/1.0 200 OK
Session: 1127293610
CSeq: 5
Date: Tue, 17 Nov 2020 02:09:50 GMT
```

## 6.5 Manual Capturing Events

### 6.5.1 Introduction to the Function

According to the preset information and capture interval configured by the client, the device manually captures images at the specified preset and reports the image information in the form of events. It is used for the client to identify emergency situations and perform an emergency patrol on some presets for analysis. It has a higher priority than the daily scheduled patrol capture. Common usage scenarios: according to the scheduled capturing event (eventType: imageCapture) rules, the store camera regularly audits in a number of key areas. If in the event of an emergency, and a worker wants to immediately re-inspect each area, you can set a manual capture rule to complete an independent audit. After the audit is completed, the camera will continue to audit according to the scheduled capturing rules.

### 6.5.2 API Calling Flow

1. The client gets the system capability: `GET /ISAPI/System/capabilities`, and if `isSupportManualImageCapture` returns `true`, manual capturing shall be supported.
2. The client gets manual capturing capability for a specified channel: `GET /ISAPI/System/Video/inputs/channels/<channelID>/manualImageCapture/capabilities?format=json`.
3. The client sets the parameters of manual capture events for a specified channel: Get parameters: `GET /ISAPI/System/Video/inputs/channels/<channelID>/manualImageCapture?format=json`. Set parameters: `PUT /ISAPI/System/Video/inputs/channels/<channelID>/manualImageCapture?format=json`.
4. Receive the event uploaded by the device in arming or listening mode: `eventType:manualImageCapture`.

#### Remarks:

- Each time the client sets manual capture event parameters, the channel immediately takes a patrol capture of the relevant presets. When the last preset capture is finished, the manual capture event ends.

## 6.6 Metadata

### 6.6.1 Introduction to the Function

To deal with the third-party access, now a plug-in-free integration solution for metadata is provided. After enabling uploading metadata, the user can obtain the metadata together with the audio and video streams via RTSP, and then selectively overlay the intelligent structured information of metadata on the video.

The figure below shows the effect of displaying the metadata of the radar PTZ camera.



### 6.6.2 API Calling Flow

1. Get the device's capability: `GET /ISAPI/System/capabilities`.
2. Check whether the device supports metadata. When the node `isSupportMetadata` exists in the returned device's system capability and its value is `true`, it indicates that the metadata is supported by the device.
3. Get the metadata type supported by the device: `GET /ISAPI/Streaming/channels/<channelID>/metadata/capabilities`.
4. Enable uploading the metadata: `PUT /ISAPI/Streaming/channels/<channelID>/metadata/<eventType>`. If multiple types of metadata need to be uploaded, you can repeatedly call this API.
5. Check whether the device supports subscribing to the metadata. When the node `isSupportSubscribeType` exists in the returned device's system capability and its value is `true`, it indicates that the metadata is supported by the device. Device can send different types of metadata to different pieces of client software.
6. Subscribe to a type of metadata (Get the streaming URL of a specified type of metadata): `POST /ISAPI/Streaming/channels/<channelID>/metadata/subscribeType?format=json`. The device will return a RTSP address.
7. Use the URL obtained in the previous step to get the RTSP stream, and parse the video and metadata. See the next chapter for details.
8. Use the fixed URL to get the RTSP stream: `rtsp://<IP>:<PORT>/ISAPI/Streaming/channels/<channelID>`. See the next chapter for details.

### 6.6.3 Integration Based on RTSP



1. The client sends a DESCRIBE command, e.g., `DESCRIBE /ISAPI/Streaming/channels/101 RTSP/1.0`. Digest-MD5 authentication should be completed during this step.
2. In the SDP message returned by the device, `a=control:trackID=3` indicates that the trackID of metadata is 3, and the track with trackID=3 needs to be configured during SETUP integration method. `a=rtpmap:107 isapi.metadata/90000` indicates that the RTP payload type corresponding to the metadata is 107 (hexadecimal 0x6B).

3. The client sends a SETUP command to get the SDP message and parse the `trackID`. The trackID returned by different devices has different meanings, for example, trackID=1 indicates video, trackID=2 indicates audio, trackID=3 indicates metadata.
4. When the client sends a RTSP PLAY command to a device, the device starts to push video streams, audio streams, and metadata. The command format is: `PLAY /ISAPI/Streaming/channels/101 RTSP/1.0`.
5. The client receives the RTP packets sent by the device. For divided RTP packets, they should be combined on the client before being parsed.
6. Check the payload in RTP Headers of the packet, if the payload is 107, it is a metadata packet, if not, it is an audio or video packet.
7. The client sends a RTSP TEARDOWN command to stop streaming. Note: This document only introduces the key steps related to metadata during streaming via RTSP. For standard RTSP description, refer to the RTSP document (Real Time Streaming Protocol) on the website [RFC 2326](#).

#### 6.6.4 Sample Message

The following is an interaction example of an request for audio and video streams and metadata via RTSP.

1. The client sends a DESCRIBE request.

```
DESCRIBE /ISAPI/Streaming/channels/101 RTSP/1.0
CSeq:0
Accept:application/sdp
Authorization:
User-Agent:NKPlayer-VSPlayer1.0
```

2. The server responds that the authentication is required.

```
RTSP/1.0 401 Unauthorized
CSeq: 0
WWW-Authenticate: Digest realm="IP Camera(E1142)", nonce="58884d0bc35f98c0cd26f8c0120046c7", stale="FALSE"
Date: Wed, Mar 25 2020 19:44:38 GMT
```

3. The client resends a DESCRIBE request with the authentication information.

```
DESCRIBE /ISAPI/Streaming/channels/101 RTSP/1.0
CSeq:0
Accept:application/sdp
Authorization:Digest username="admin", realm="IP Camera(E1142)", nonce="58884d0bc35f98c0cd26f8c0120046c7", uri="/ISAPI/Streaming/channels/101",
response="72b907d281e7a727faee09f67e594fed"
User-Agent:NKPlayer-VSPlayer1.0
```

4. Authentication completed, the server returns the SDP information.

5. The client sends a SETUP request three times, and the server responds three times correspondingly.

```
SETUP /ISAPI/Streaming/channels/101/trackID=1 RTSP/1.0
CSeq:1
Authorization:Digest username="admin", realm="IP Camera(E1142)", nonce="58884d0bc35f98c0cd26f8c0120046c7", uri="/ISAPI/Streaming/channels/101",
response="d3869bd6535bdeb0ed6bf11a6cbdbeb"
Transport: RTP/AVP/TCP;unicast
User-Agent:NKPPlayer-VSP1aver1.0
```

```
RTSP/1.0 200 OK
CSeq: 1
Session: 1395142895;timeout=60
Transport: RTP/AVP/TCP;unicast;interleaved=0-1:ssrc=6c066f38;mode="play"
Date: Wed, Mar 25 2020 19:44:38 GMT
```

SETUP /ISAPI/Streaming/channels/101/trackID=2 RTSP/1.0  
CSeq:2  
Authorization:Digest username="admin", realm="IP Camera(E1142)", nonce="58884d0bc35f98c0cd26f8c0120046c7", uri="/ISAPI/Streaming/channels/101",  
response="d3869bd46535bdeb0ed6bf11a6cbdbef"  
Session: 1395142895  
Transport: RTP/AVP/TCP;unicast;  
User-Agent:NKPlayer-VPPlayer1.0

```
RTSP/1.0 200 OK
CSeq: 2
Session: 1395142895;timeout=60
Transport: RTP/AVP/TCP;unicast;interleaved=2-3:ssrc=4aa60c85;mode="play"
Date: Wed, Mar 25 2020 19:44:38 GMT
```

SETUP /ISAPI/Streaming/channels/101/trackID=3 RTSP/1.0  
Seq:3  
Authorization:Digest username="admin", realm="IP Camera(E1142)", nonce="58884d0bc35f98c0cd26f8c0120046c7", uri="/ISAPI/Streaming/channels/101",  
response="d3869bd46535bdeb0ed6bf11a6cbdbef"  
Session: 1395142895  
Transport: RTP/AVP/TCP;unicast;  
User-Agent:NKPPlayer-VSPPlayer1.0

```
RTSP/1.0 200 OK
Session: 1395142895;timeout=60
Transport: RTP/AVP/TCP;unicast;interleaved=6-7:ssrc=1c4ccc6a;mode="play"
Date: Wed, Mar 25 2020 19:44:38 GMT
```

6. The client sends a PLAY request.

```
PLAY /ISAPI/Streaming/channels/101 RTSP/1.0
CSeq:4
Authorization:Digest username="admin", realm="IP Camera(E1142)", nonce="58884d0bc35f98c0cd26f8c0120046c7", uri="/ISAPI/Streaming/channels/101",
response="74bb7780cc4b8e896c2ee1c8004aad50"
Session: 1395142895
Range:npt=0.000000-
Scale:1.000
User-Agent:NKPlayer-VSPlayer1.0
```

7. The server starts to push audio and video streams and the metadata.

```
RTSP/1.0 200 OK
CSeq: 4
Session: 1395142895
RTP-Info: url=trackID=1;seq=17176,url=trackID=2;seq=14182,url=trackID=3;seq=53596
Date: Wed, Mar 25 2020 19:44:38 GMT
$..(..C.....Ufw.....$....$..<..C.....Ufw.....#$.....@ .....W.....,$..L.
```

8. The client sends a TEARDOWN request, and the server responds.

```
TEARDOWN /ISAPI/Streaming/channels/101 RTSP/1.0
CSeq:5
Authorization:Digest username="admin", realm="IP Camera(E1142)", nonce="58884d0bc35f98c0cd26f8c0120046c7", uri="/ISAPI/Streaming/channels/101",
response="74bb7780cc4b8e896c2ee1c8004aad50"
Session: 1395142895
User-Agent:NKPlayer-VSPlayer1.0
```

```
RTSP/1.0 200 OK
CSeq: 5
Session: 1395142895
```

## 6.7 Motion Detection

### 6.7.1 Introduction to the Function

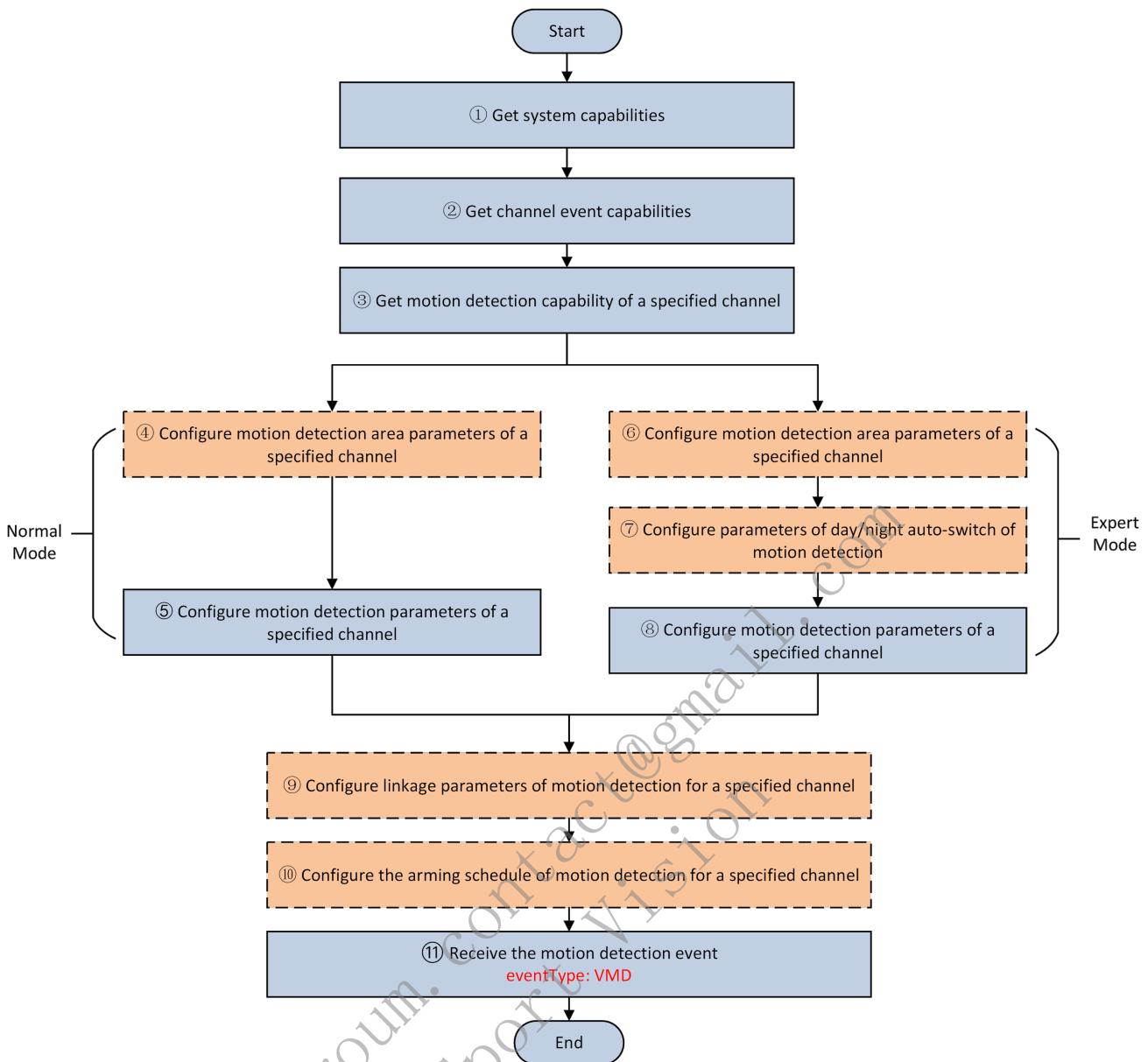
Commonly used in unattended camera recording and automatic alarm, motion detection can help to reduce the cost of manual monitoring for public institutions and companies. Besides, it can greatly improve monitoring efficiency and accuracy by avoiding the errors of manual monitoring due to fatigue after long-time work. For example, it can be applied in home and company theft proof, as well as monitoring and alarm in institutions like medical organizations and nursing homes.

Motion detection has two modes: normal mode and expert mode.

Normal mode: the default mode of motion detection. In this mode, all detected areas only support a same sensitivity.

Expert mode: in this mode, every detected area can have its own detection sensitivity and percentage of objects in area. It supports configuring different sensitivities according to area types. For example, for the motion detection in the company hall, you can set higher detection sensitivity for the gate area to focus on people's entering and exiting.

### 6.7.2 API Calling Flow



1. Get the system capability: `GET /ISAPI/System/capabilities`. If the node `isSupportMotionDetection` is returned and its value is `true`, it indicates that the device supports motion detection.
2. Get the device channel event capability. If the returned `eventType` contains `VMD`, it indicates that the corresponding channel supports motion detection.

Get the event capability of all channels: `GET /ISAPI/Event/channels/capabilities`.

Get the event capability of a specified channel: `GET /ISAPI/Event/channels/<channelID>/capabilities`.

3. Get the motion detection capability of a specified channel (normal mode): `GET /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection/capabilities`.

Check the motion detection parameters (such as function enabling, detection sensitivity) supported by the device. You can check whether the device supports multiple areas via checking whether the node `regionType` contains `region`.

4. (Optional) Configure parameters of the motion detection area (normal mode).

Capability: `GET /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection/layout/capabilities`.

Get: `GET /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection/layout?regionType=<regionType>`. **Note:** If the device supports multiple areas of motion detection, for compatibility, both `<gridMap>` and `<RegionList>` will be returned regardless the `regionType` value in URL.

Configure: `PUT /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection/layout?regionType=<regionType>`. **Note:** If the device supports multiple areas of motion detection, to set multiple areas, you need to input `regionType=<region>` in URL.

5. Configure the motion detection parameters (such as motion detection enabling, detection sensitivity) (normal mode).

Get: `GET /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection`. **Note:** If the device supports multiple areas of motion detection, for compatibility, both `<gridMap>` and `<RegionList>` will be returned regardless the `regionType` value in URL.

Configure: `PUT /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection`. **Note:** If the device supports multiple areas of motion detection, to set multiple areas, you need to input `regionType=<region>` in URL.

6. (Optional) Configure the parameters of motion detection area (expert mode). You can configure different detection conditions such as sensitivity for different areas.

Get: `GET /ISAPI/System/Video/inputs/channels/<channelID>/motionDetectionExt/regions`.

Configure: `PUT /ISAPI/System/Video/inputs/channels/<channelID>/motionDetectionExt/regions`.

7. (Optional) Configure day/night auto switch of motion detection (expert mode). You can configure the time switch schedule of `day` and `night`.

Get: `GET /ISAPI/System/Video/inputs/channels/<channelID>/motionDetectionExt/switch`.

Configure: `PUT /ISAPI/System/Video/inputs/channels/<channelID>/motionDetectionExt/switch`.

8. Configure motion detection conditions (expert mode).

Get: `GET /ISAPI/System/Video/inputs/channels/<channelID>/motionDetectionExt`.

Configure: `PUT /ISAPI/System/Video/inputs/channels/<channelID>/motionDetectionExt`.

9. (Optional) Configure linkage parameters of the motion detection.

Get: `GET /ISAPI/Event/triggers/VMD-<channelID>`.

Configure: `PUT /ISAPI/Event/triggers/VMD-<channelID>`.

Delete: `DELETE /ISAPI/Event/triggers/VMD-<channelID>`.

10. (Optional) Configure the arming schedule of motion detection.

Get: `GET /ISAPI/Event/schedules/motionDetections/VMD_video<channelID>`.

Configure: `PUT /ISAPI/Event/schedules/motionDetections/VMD_video<channelID>`.

11. Receive events reported by device in arming or listening mode: `eventType:VMD`.

#### Remarks:

1. For motion detection (normal mode), the uploaded `VMD` event includes device basic information only, such as channel No. and event time. For details, see 4.3.1 Motion Detection Event (Normal Mode).
2. For motion detection (expert mode), in addition to the device basic information, the uploaded `VMD` event includes the detection area ID, area coordinates and area detection sensitivity. For details, see 4.3.2 Motion Detection Event (Expert Mode).
3. In practical employment, you can select one mode as needed. There are no logical relations between the two modes.

## 6.7.3 Message Example

### 6.7.3.1 Motion Detection Event (Normal Mode)

```
<?xml version="1.0" encoding="UTF-8"?>
<EventNotificationAlert version="1.0" xmlns="urn:psialliance-org">
<ipAddress>10.17.114.252</ipAddress>
<protocolType>HTTP</protocolType>
<macAddress>64:db:8b:6e:4f:08</macAddress>
<channelID>1</channelID>
<dateTime>2018-03-13T19:42:27+08:00</dateTime>
<activePostCount></activePostCount>
<eventType>VMD</eventType>
<eventState>active</eventState>
<eventDescription>Motion alarm</eventDescription>
<channelName>Camera 01</channelName>
<Extensions version="1.0" xmlns="urn:psialliance-org">
<serialNumber xmlns="urn:selfextension:psiaext-ver10-xsd">DS-2DF8236IX-AELW20170718CCWR799562898X</serialNumber>
<eventPush xmlns="urn:selfextension:psiaext-ver10-xsd">VMD&amp;DS-2DF8236IX-AELW20170718CCWR799562898X,2018-01-21T12:50:39+08:00,1,1.0</eventPush>
</Extensions>
</EventNotificationAlert>
```

### 6.7.3.2 Motion Detection Event (Expert Mode)

```
<?xml version="1.0" encoding="UTF-8"?>
<EventNotificationAlert version="1.0" xmlns="urn:psialliance-org">
<ipAddress>10.17.114.252</ipAddress>
<protocolType>HTTP</protocolType>
<macAddress>64:db:8b:6e:4f:08</macAddress>
<channelID>1</channelID>
<dateTime>2018-03-13T20:36:34+08:00</dateTime>
<activePostCount>11</activePostCount>
<eventType>VMD</eventType>
<eventState>active</eventState>
<eventDescription>Motion alarm</eventDescription>
<DetectionRegionList>
<DetectionRegionEntry>
<regionID>1</regionID>
<sensitivityLevel>50</sensitivityLevel>
<RegionCoordinatesList>
<RegionCoordinates>
<positionX>216</positionX>
<positionY>216</positionY>
</RegionCoordinates>
<RegionCoordinates>
<positionX>756</positionX>
<positionY>756</positionY>
</RegionCoordinates>
<RegionCoordinates>
<positionX>756</positionX>
<positionY>756</positionY>
</RegionCoordinates>
<RegionCoordinates>
<positionX>216</positionX>
<positionY>216</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</DetectionRegionEntry>
</DetectionRegionList>
<channelName>Camera 01</channelName>
<Extensions version="1.0" xmlns="urn:psialliance-org">
<serialNumber xmlns="urn:selfextension:psiaext-ver10-xsd">DS-2DF8236IX-AELW20170718CCWR799562898X</serialNumber>
<eventPush xmlns="urn:selfextension:psiaext-ver10-xsd">VMD&amp;DS-2DF8236IX-AELW20170718CCWR799562898X,2018-01-21T12:50:39+08:00,1,1.0</eventPush>
</Extensions>
</EventNotificationAlert>
```

## 6.8 Sync Capturing Manually

### 6.8.1 Introduction to the Function

The client sends a capture command to the device, and the device immediately returns the captured picture data in respond. It is often used in scenarios with high network bandwidth between clients and devices to facilitate the rapid transmission of image data, such as a LAN built on a company campus.

### 6.8.2 API Calling Flow

1. (Optional) The client gets the system capability: GET /ISAPI/System/capabilities, and if `isSupportManualSnapPicture` returns `true`, sync manual capturing shall be supported.

2. (Optional) The client gets sync manual capturing capability for a specified channel: GET  
`/ISAPI/Streaming/channels/<trackStreamID>/picture/capabilities?format=json`.
3. The client controls the specified channel to perform sync manual capturing: GET  
`/ISAPI/Streaming/channels/<trackStreamID>/picture?snapShotImageType=<snapShotImageType>&videoResolutionWidth=<videoResolutionWidth>&videoResolutionHeight=<videoResolutionHeight>&imageQuality=<imageQuality>&x=<x>&y=<y>&height=<height>&width=<width>`, and the device returns the captured binary picture data.

**Remarks:**

- If you only need to capture a JPEG picture, you can jump to step 3.
- If you need to capture pictures other than JPEG or capture in the designated area of the video, you need to perform steps 1 and 2 to determine whether the device supports it. If it does, go to step 3.
- In step 3, the value ranges of `videoResolutionWidth` and `videoResolutionHeight` are the same as `<videoResolutionWidth>` and `<videoResolutionHeight>` returned by GET  
`/ISAPI/Streaming/channels/<trackStreamID>/capabilities`.

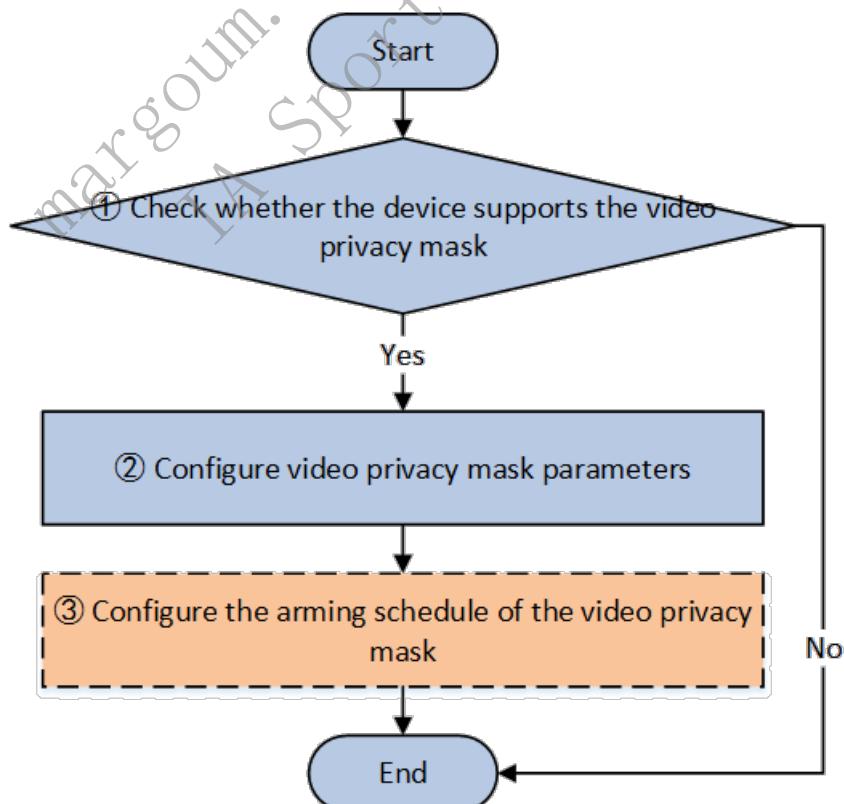
## 6.9 Video Privacy Mask

### 6.9.1 Introduction to the Function

Video privacy mask is to cover certain areas on the live image to protect personal privacy from being live viewed and recorded. You can use the masking method such as a mosaic or black frame as needed to make the area invisible. The device supporting privacy mask can specify the area(s) to be masked permanently on the live image, as a result of which the live view and recorded video during the live view will be influenced and the image cannot be restored. The following image shows the effect of masking an area with the mosaic.

### 6.9.2 API Calling Flow

#### 6.9.2.1 Video Privacy Mask



**1. Check whether the device supports the video privacy mask.** Get the device's system capability: GET  
`/ISAPI/System/capabilities`. If `isSupportPrivacyMask` is returned and its value is true, it indicates that the video privacy mask is supported by the device.

**2. Configure the video privacy mask parameters.** You can specify sensitive area(s) to be masked on the live image (related field: `PrivacyMaskRegionList`), and up to 8 areas are supported. You can also configure the masking type, such as mosaic, black frame, and red frame (related field: `maskType`). Get the video privacy mask capability: `GET /ISAPI/System/Video/inputs/channels/<channelID>/privacyMask/privacyMaskCap`. Configure the video privacy mask parameters: `PUT /ISAPI/System/Video/inputs/channels/<channelID>/privacyMask`. Get the video privacy mask parameters: `GET /ISAPI/System/Video/inputs/channels/<channelID>/privacyMask`.

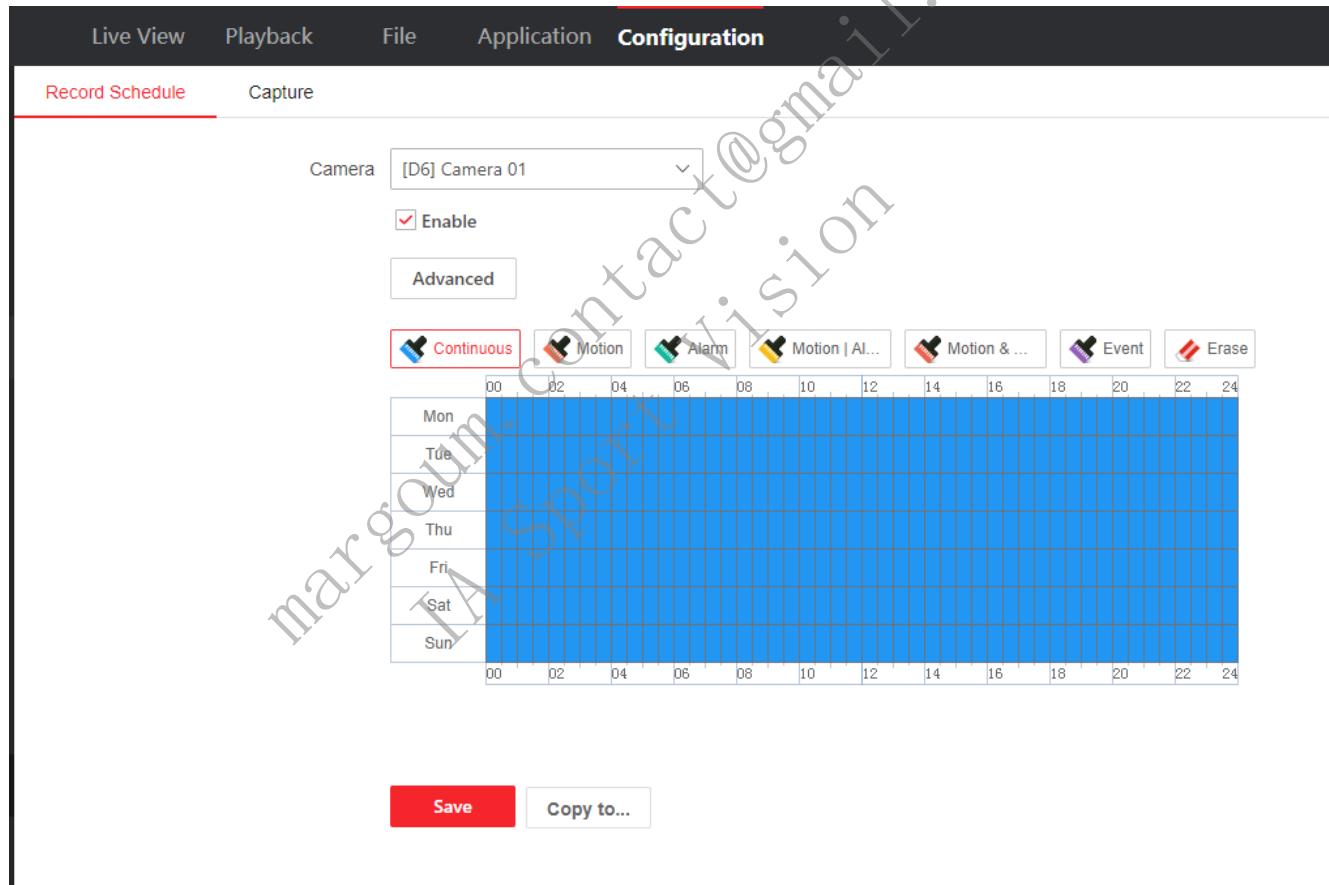
**3. (Optional) Configure the arming schedule of the video privacy mask.** You can configure the privacy mask schedule as needed. If not configured, the privacy mask is performed 24 hours a day by default. Get the video privacy mask schedule: `GET /ISAPI/Event/schedules/privacyMask/privacyMask-<channelID>`. Set the video privacy mask schedule: `PUT /ISAPI/Event/schedules/privacyMask/privacyMask-<channelID>`.

## 7 Video Management

### 7.1 . Scheduled Recording

#### 7.1.1 Introduction to the Function

The web interface for reference is shown below.



#### 7.1.2 API Calling Flow

1. The client software gets the device system capability: `GET /ISAPI/System/capabilities`. When `isSupportScheduledRecordUpload` is returned and its value is `true`, it indicates that scheduled recording is supported.
2. The client software configures schedule parameters of scheduled recording. You can configure recording schedules for all channels or configure the recording schedule for a specified channel.

- Recording schedule management for all channels: Add a recording schedule: `POST /ISAPI/ContentMgmt/record/tracks`. Get all recording schedules parameters: `GET /ISAPI/ContentMgmt/record/tracks`. Set all recording schedules: `PUT /ISAPI/ContentMgmt/record/tracks`.
- Recording schedule management for a specified channel: Get the capability of configuring the recording schedule for a specified channel: `GET /ISAPI/ContentMgmt/record/tracks/<trackStreamID>/capabilities?security=`

```
<security>&iv=<iv>. Get the recording schedule parameters for a specified channel: GET  
/ISAPI/ContentMgmt/record/tracks/<trackStreamID>?security=<security>&iv=<iv>. Set the recording  
schedule for a specified channel: PUT /ISAPI/ContentMgmt/record/tracks/<trackStreamID>?security=  
<security>&iv=<iv>. Delete the recording schedule for a specified channel: DELETE  
/ISAPI/ContentMgmt/record/tracks/<trackStreamID>?security=<security>&iv=<iv>.
```

## 8 People Counting

### 8.1 People Counting Statistics Search

#### 8.1.1 Introduction to the Function

For devices with local storage function, the people counting data will be stored in the local storage after the people counting function is enabled. You can search the local data to get multiple types of reports by setting different conditions.

#### 8.1.2 API Calling Flow

##### 8.1.2.1 Single Channel

1. Get the capability of searching for people counting data of a single channel: GET  
/ISAPI/System/Video/inputs/channels/<channelID>/counting/search/capabilities.
2. Search for people counting data of a single channel: POST  
/ISAPI/System/Video/inputs/channels/<channelID>/counting/search.

##### 8.1.2.2 Multiple Channels

1. Get video capability: GET /ISAPI/System/Video/capabilities. If the node `isSupportMultiChannelCounting` is returned and its value is `true`, it indicates that searching for people counting data of multiple channels is supported.
2. Get the capability of searching for people counting data of multiple channels: GET  
/ISAPI/System/Video/inputs/channels/counting/search/capabilities.
3. Search for the people counting data of multiple channels: POST  
/ISAPI/System/Video/inputs/channels/counting/search.

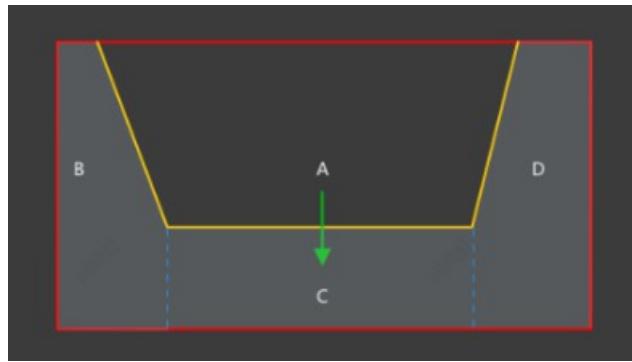
## 8.2 People Flow Direction Analysis

#### 8.2.1 Introduction to the Function

People flow direction analysis is used for counting the people who crosses detection lines and enters the detection area. The flow analysis is valid only when the detection area is quadrilateral and the detection line and the detection area are overlapped only in the start and end points.

The requirements for drawing the rule/detection area of flow direction analysis are as follows:

1. The detection line (in yellow) and the detection area (in red) should have but two overlapped points.
2. The detection line should be perpendicular to the detection direction (along the direction of green arrow).
3. The four points on the detection line respectively are: point 1, 2, 3, 4.
4. Along the detection direction (along the green arrow), draw two parallel lines from point 2 and 3 to intersect the detection area (in red). The area between the two parallel lines is the counting region of area 2 (area C in the figure), and the rest two are area 1 (area B in the figure) and area 3 (area D in the figure).



## 8.2.2 API Calling Flow

The process of people flow direction analysis is the same with that of the real-time people counting. You need to enable the flow direction analysis to receive and parse the flow statistics data.

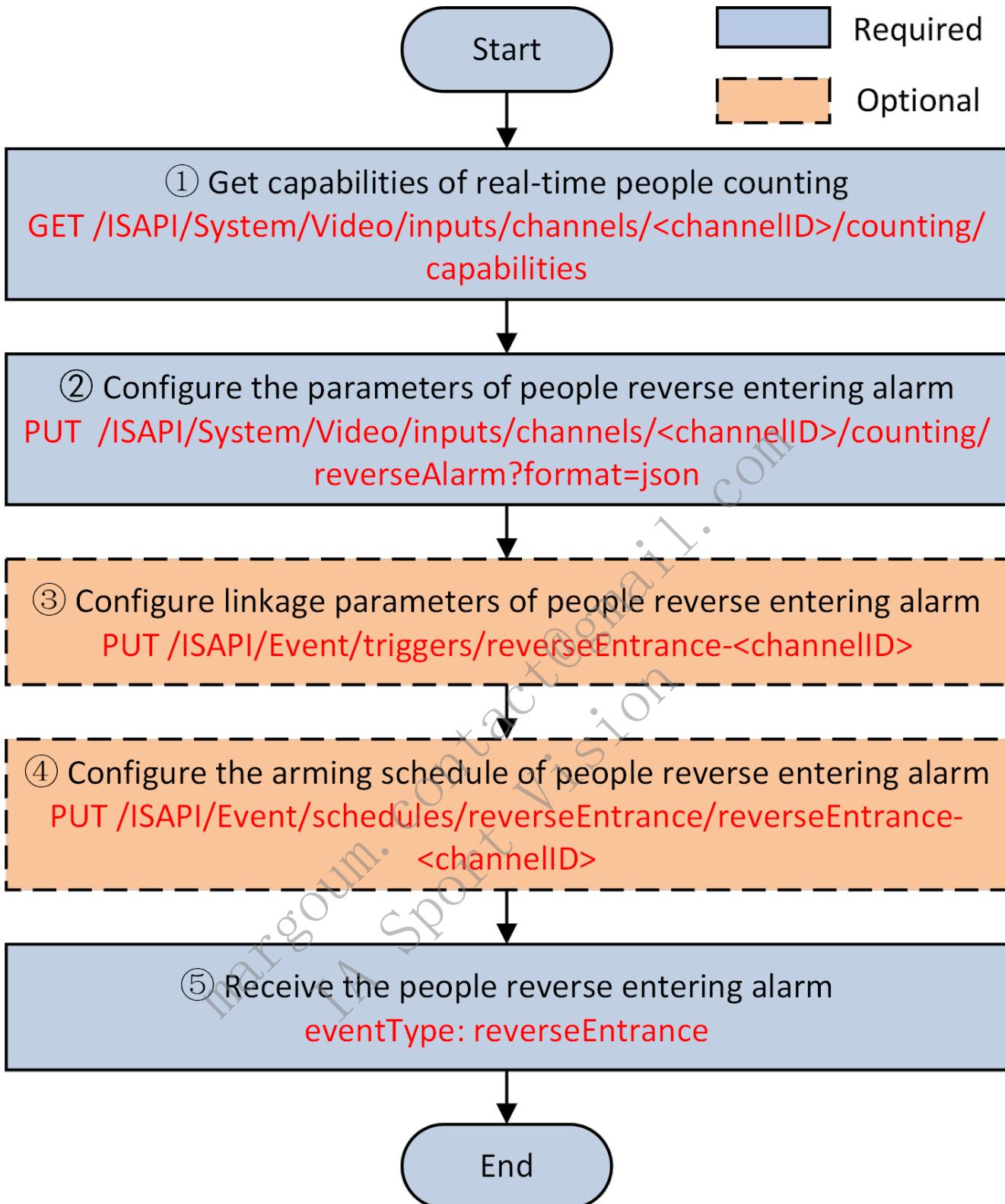
1. Get the capability of people counting: GET `/ISAPI/System/Video/inputs/channels/<channelID>/counting/capabilities`. If `flowDirectionAnalysis` is returned, it indicates that the device supports flow analysis.
2. Configure people counting parameters: PUT `/ISAPI/System/Video/inputs/channels/<channelID>/counting`. Configure detection lines parameters (`CountingLineItemList`) and rule parameters (`CountingRegionList.RuleRegionCoordinatesList`), select scene mode (`countingSceneMode`), enable the flow direction analysis (`flowDirectionAnalysis` is `true`).
3. Receive the event uploaded by devices in arming or listening mode: `PeopleCounting`. Parse `peopleCounting.FlowStatisticsList` to get the flow data.

## 8.3 People Reverse Entering Alarm

### 8.3.1 Introduction to the Function

The people reverse entering alarm will be triggered if a person crosses the detection line from the opposite direction within a specified period. It applies to the circumstance when only single direction passing is allowed within a specified period. For example, if someone exits the workshop within working time, an alarm will be uploaded for notification.

### 8.3.2 API Calling Flow



1. Get the people counting capability: GET

`/ISAPI/System/Video/inputs/channels/<channelID>/counting/capabilities`. If the node `isSupportReverseAlarm` is returned and its value is `true`, it indicates that people reverse entering alarm is supported.

2. Configure parameters of the people reverse entering alarm: PUT

`/ISAPI/System/Video/inputs/channels/<channelID>/counting/reverseAlarm?format=json`.

3. (Optional) Configure linkage parameters of the people reverse entering alarm: PUT  
`/ISAPI/Event/triggers/reverseEntrance-<channelID>`.

4. (Optional) Configure the arming schedule for the people reverse entering alarm: PUT  
`/ISAPI/Event/schedules/reverseEntrance/reverseEntrance-<channelID>`.

5. Receive events reported by device via arming or listening: `reverseEntrance`.

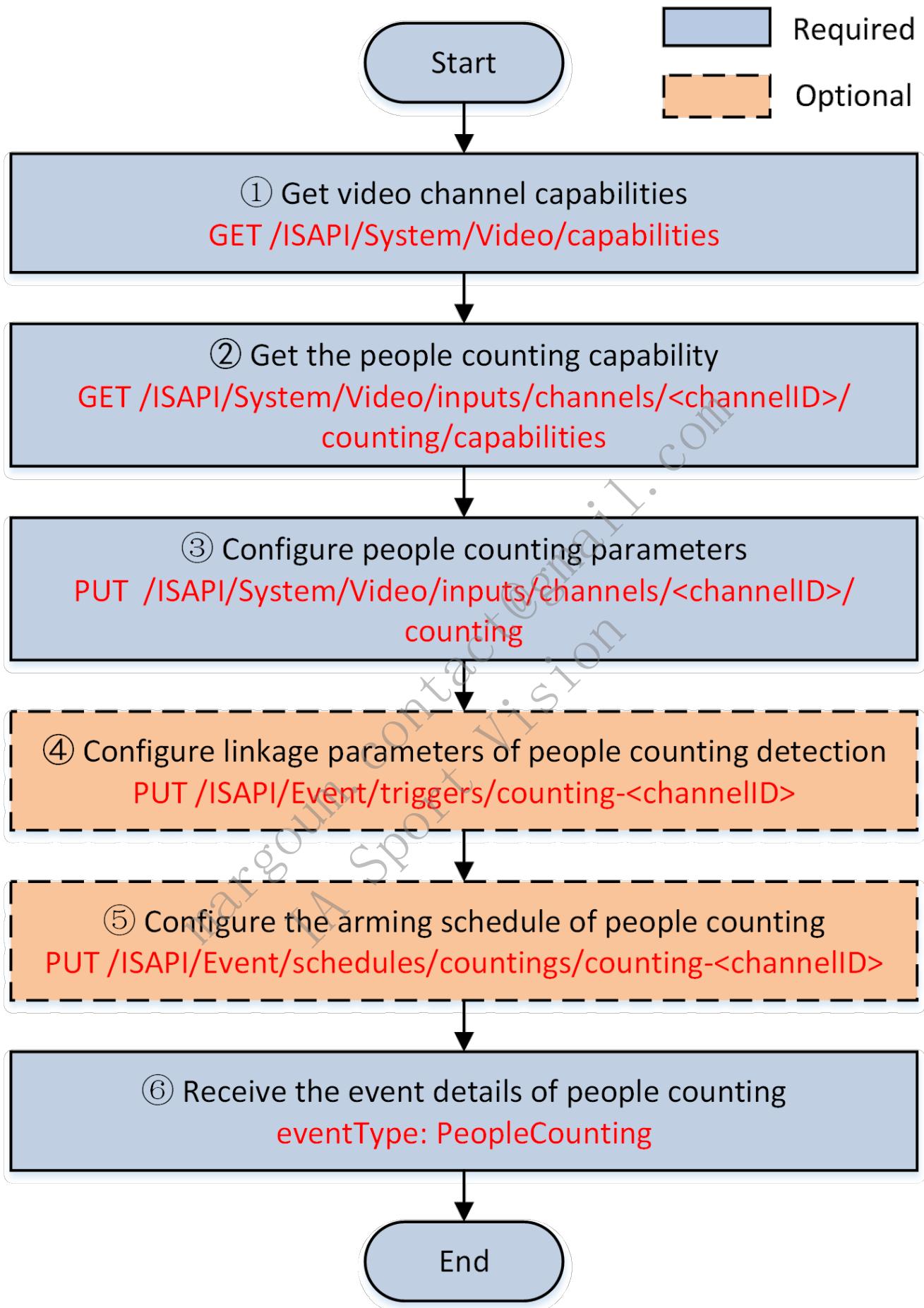
## **8.4 Real-time People Counting**

### **8.4.1 Introduction to the Function**

The function is to count line crossing persons or objects in a specified time period. It is commonly applied in the entrance and exit of the scenic spots, malls, stores, etc.

### **8.4.2 API Calling Flow**

margoum.contact@gmail.com  
IA Sport Vision



- Get video channel capabilities: `GET /ISAPI/System/Video/capabilities`. If the nodes `isSupportCounting` and `isSupportMultiChannelCounting` are returned and the value is true, it indicates that the device supports real-time people counting.
- Get the people counting capability: `GET /ISAPI/System/Video/inputs/channels/<channelID>/counting/capabilities`.

3. Configure people counting parameters: `PUT /ISAPI/System/Video/inputs/channels/<channelID>/counting`.
4. (Optional) Configure linkage rules for people counting detection: `PUT /ISAPI/Event/triggers/counting-<channelID>`.
5. (Optional) Configure the arming schedule for people counting: `PUT /ISAPI/Event/schedules/countings/counting-<channelID>`.
6. Receive events uploaded by device in listening or arming mode: `PeopleCounting`.

#### **Notes:**

- During people counting, considering the height difference between children and adults, you can enable the children detection mode so as to improve the accuracy. a. Get the capability of people counting. If `ChildFilter` is returned, it indicates that the device supports counting the number of children. b. Enable the children counting during the configuration of people counting parameters (set `enabled` of `ChildFilter` to true). c. Other operations are same as above steps.
- The people counting detection event (`PeopleCounting`) supports the following data types: a. Real-time statistics data (`statisticalMethods` is `realTime`): after enabled, persons' entry and exit will trigger the event data uploading, and the data is the real-time people counting statistics from 12:00 A.M. to the current clock on the day. b. Statistics data by period (`statisticalMethods` is `timeRange`): the device will upload the statistics data according to the scheduled uploading interval (`dataUploadCycle`) within a specified period. c. Statistics data triggered by alarm input (`statisticalMethods` is `signalTrigger`): the device is connected to the alarm input signal, and the statistics data will be accumulated and uploaded every time the signal is triggered.
- The dimensions of statistics data are as follows: the number of people entering (`enter`), the number of people exiting (`exit`), the number of people repeating passing (`duplicatePeople`, which is only valid for people entering, and deduplication is invalid for people exiting), the number of people according to different face attributes, including face expression (`FaceExpressionList`), gender (`GenderList`), mask wearing (`MaskList`), etc. See details in the event message.
- If the client software fails to receive the people counting event uploaded by devices due to exceptions, or the connected devices have operated for a while, you can get the history people counting data via URI (POST `/ISAPI/System/Video/inputs/channels/counting/search`).

## **9 Vehicle Recognition**

### **9.1 Motor Vehicle Recognition**

#### **9.1.1 Introduction to the Function**

This function is to detect whether there are vehicles appearing in the image for certain time periods. If a vehicle appears and the set recognition conditions are met, it will be captured, and its related information will be analyzed and uploaded as alarms. The integrator receives the event and analyzes the specific content as needed.

#### **9.1.2. Device Event Upload**

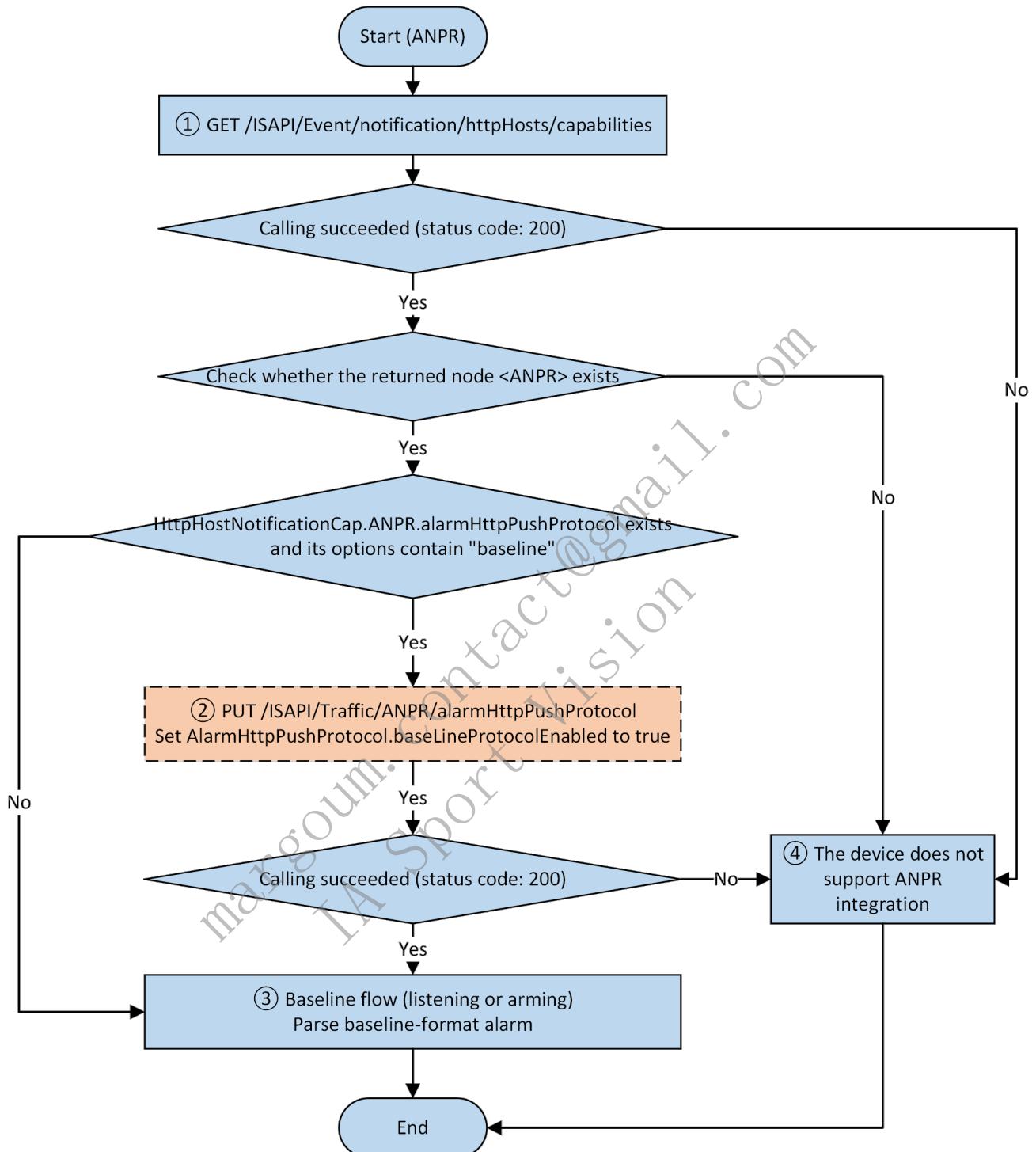
##### **9.1.2.1 Check Whether Device Supports Baseline Vehicle Recognition Event**

The integrator get the capability from device.

1. Get the system capability set via `GET /ISAPI/System/capabilities`: `true` is returned for `<ITCCap>`. `<isSupportVehicleDetection>` by device or `<ITCCap>.isSupportHVTVehicleDetection` is `true`.
2. Get the vehicle capture recognition capability via `GET /ISAPI/ITC/capabilities`: `true` is returned for `<isSupportVehicleDetection>` by device or `<isSupportHVTVehicleDetection>` is `true`.
3. Get the traffic service capability via `/ISAPI/Traffic/capabilities`: `<plateCap>` is returned by device.
4. Get the trigger mode capability set via `GET /ISAPI/ITC/TriggerMode/capabilities`: `<TriggerMode>` is returned.

When any one of the 4 conditions are met, it indicates the vehicle recognition event detection is supported. If none are met, the vehicle recognition event is not supported and please do not proceed to the following steps.

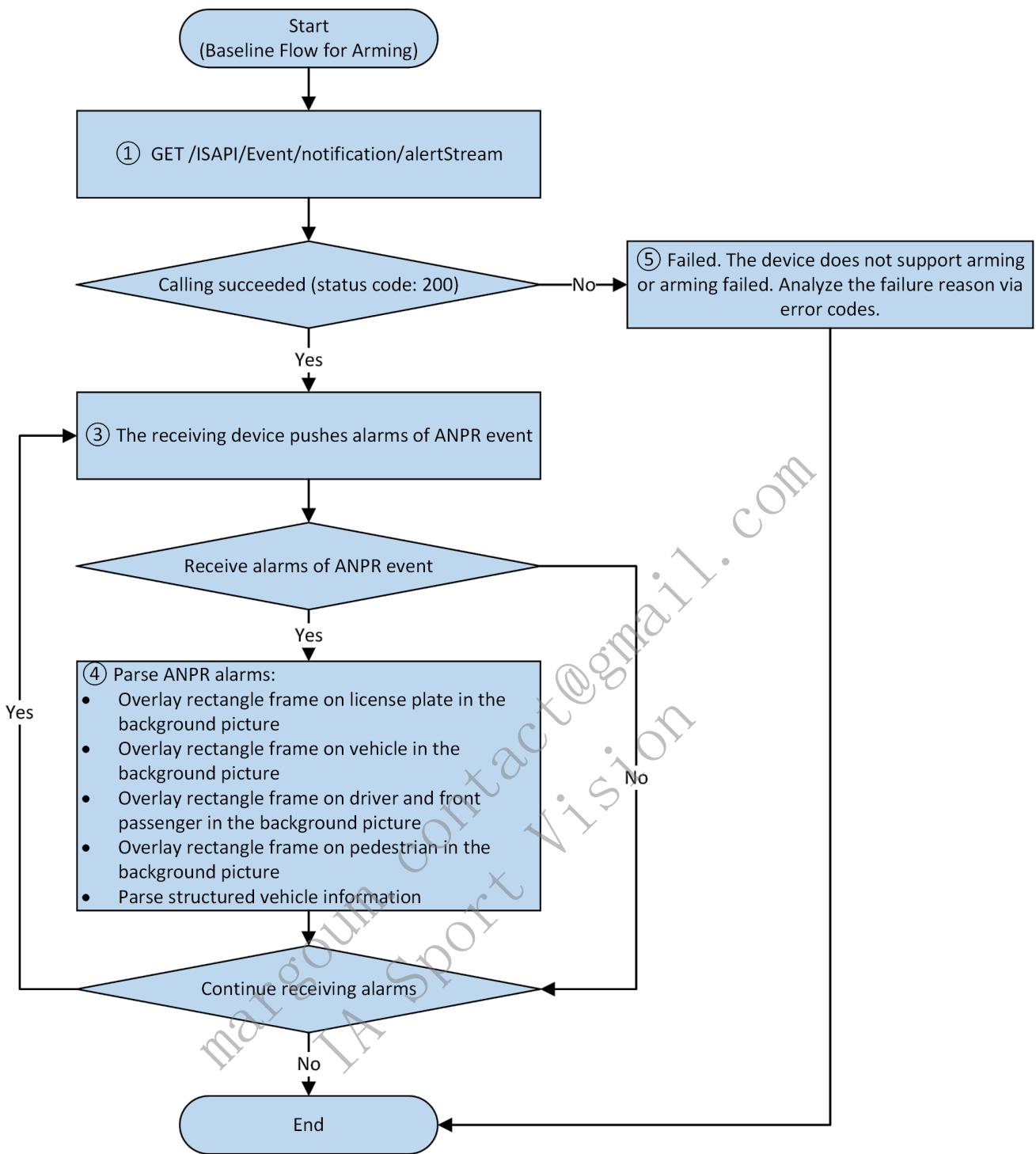
### 9.1.2.2 Check Whether Device Supports Baseline Flow



- (Client) Get the configuration capability of listening host via `GET /ISAPI/Event/notification/httpHosts/capabilities`.
- (Client) Configure the alarm mode switch for device via `PUT /ISAPI/Traffic/ANPR/alarmHttpPushProtocol` and set `AlarmHttpPushProtocol.baseLineProtocolEnabled` to `true`.
- Receive and parse alarms according to the baseline flow. See details in [4.3 Baseline Flow Event Upload](#) and [4.4 Baseline Flow Message Format and Example](#).
- The ANPR function integration is not supported by device.

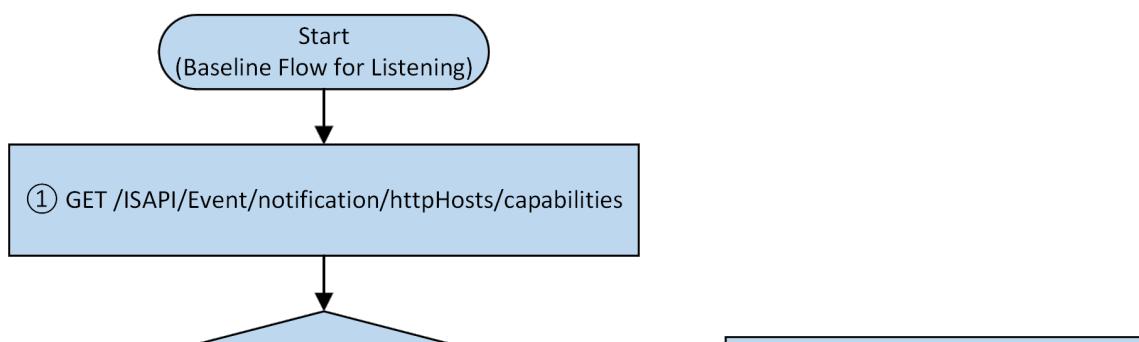
### 9.1.3 Baseline Flow Event Upload

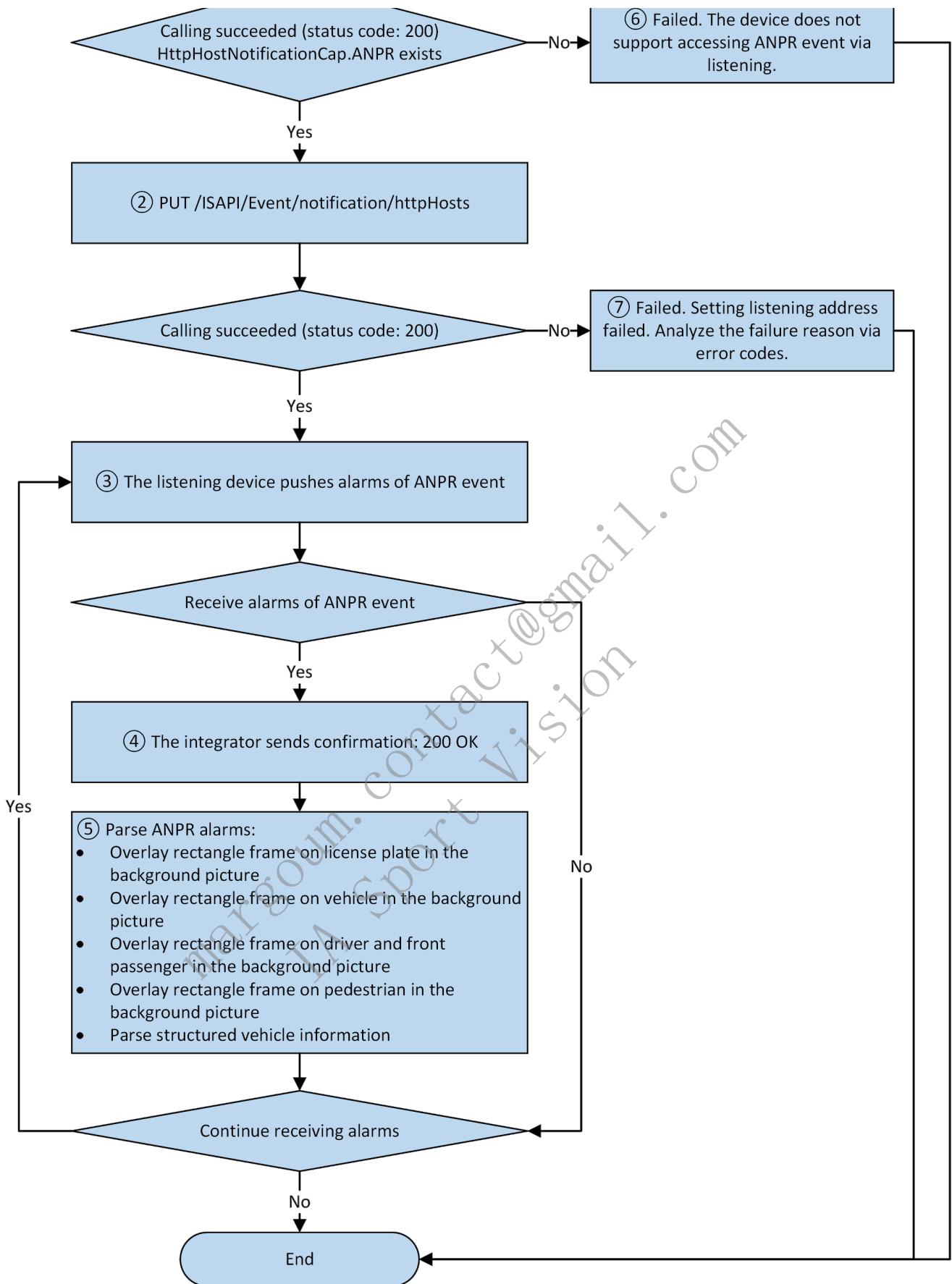
#### 9.1.3.1 Arming Flow



1. Arm the device.
2. Check whether arming succeeded.
3. Receive alarm notifications and check whether `eventType==ANPR`.
4. Parse ANPR alarm content.
5. Arming failed. Analyze the failure reason via error codes.

#### 9.1.3.2 Listening Flow





1. Get the capability set of device HTTP listening host location.
2. Change the HTTP listening host location, IP address, port configuration to the IP address of the server for receiving alarm notifications.
3. Receive alarm notifications and check whether `eventType==ANPR`.
4. The integrator sends confirmation (200 OK) to the device and notify the device of the successful alarm receiving; if the integrator does not reply, the device will consider the alarm notification lost and upload it again.
5. Parse ANPR alarm content.

6. The device does not support accessing ANPR event via listening.
  7. Setting listening address failed. Analyze the failure reason via error codes.

## 9.1.4 Baseline Flow Message Format and Example

#### **9.1.4.1 Baseline Flow Event Upload Method Configuration**

Set `baseLineProtocolEnabled` to `true`,

URI: /ISAPI/Traffic/ANPR/alarmHttpPushProtocol

Method: GET, PUT

## Body:

```
<AlarmHttpPushProtocol version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <baseLineProtocolEnabled>true</baseLineProtocolEnabled>
</AlarmHttpPushProtocol>
```

### 9.1.2.2 Baseline Flow Event Upload Example

Reported by device.

The receiver responds.

HTTP/1.1 200 OK  
Connection: close

**Note:** When there are multiple pictures of the same type, the following naming rule will be applied: add suffix (e.g., "\_1", "\_2") after the extended name. For example, the names of three detection pictures can be detectionPicture\_1.jpg, detectionPicture\_2.jpg, and detectionPicture\_3.jpg.

Attribute Value	Attribute Type Character String
ANPR Alarm XML	anpr.xml
Detection Picture (Background Picture)	detectionPicture.jpg
License Plate Thumbnail	licensePlatePicture.jpg
Driver's Face Matting	pilotPicture.jpg
Passenger's Face Matting	copilotPicture.jpg
Composite Picture	compositePicture.jpg
License Plate Binary Picture	plateBinaryPicture.jpg
Non-Motor Vehicle Matting	nonMotorPicture.jpg
Pedestrian Picture	pedestrianDetectionPicture.jpg
Pedestrian Matting	pedestrianPicture.jpg

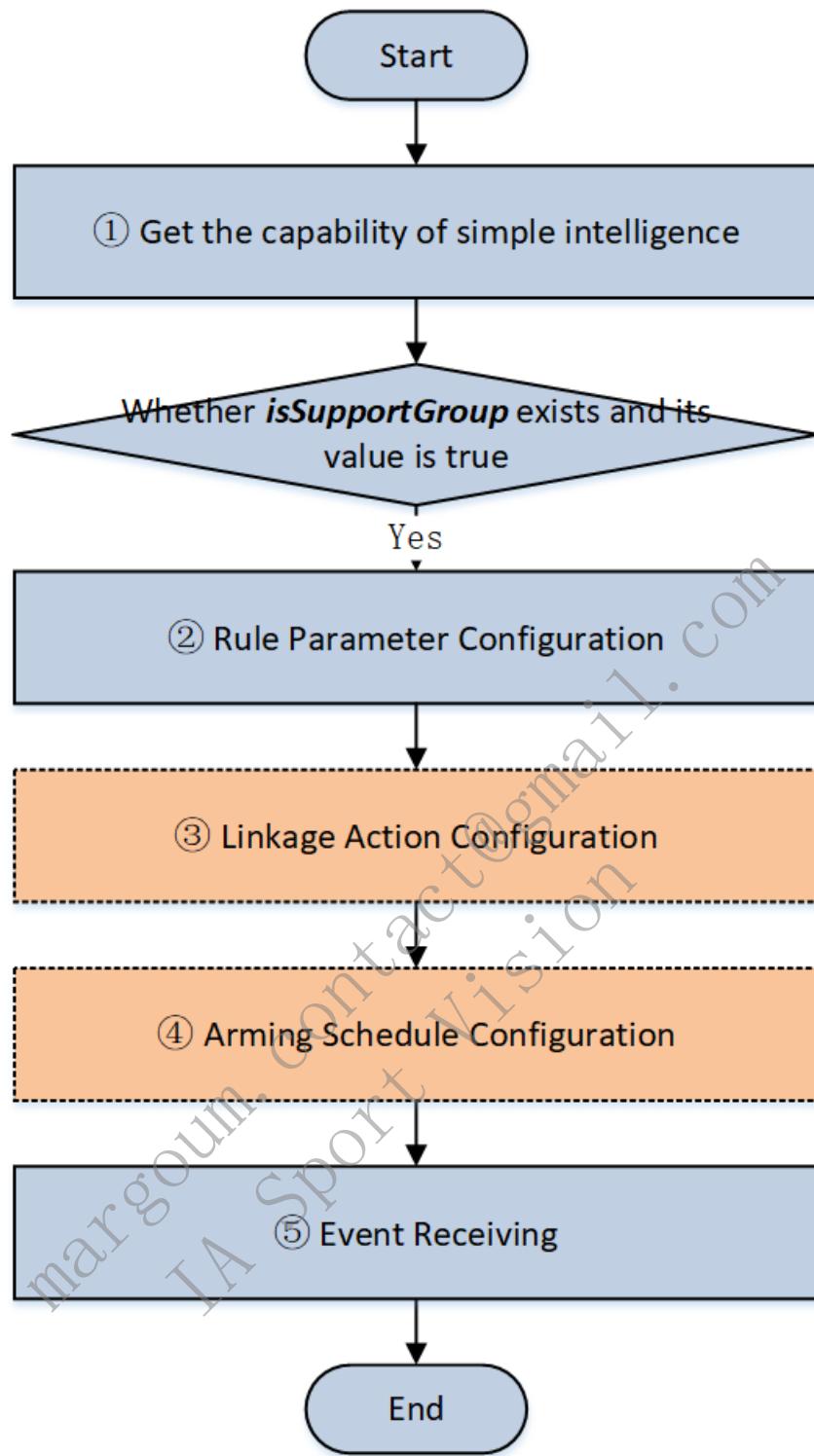
## 10 Abnormal Event Detection

### 10.1 People Gathering

#### 10.1.1 Introduction to the Function

The people gathering function is used for triggering the alarm in the fastest and best way when the people density in the set area exceeds the set threshold. The people gathering function is often used for performing the real-time monitoring of the abnormal behavior of people in the monitoring area, effectively preventing accidents caused by negligence, so as to avoid potential safety accidents and improve the supervision efficiency.

#### 10.1.2 API Calling Flow



### 1. \*\*Get Capability of AI intelligent Search \*\*

Get the `isSupportGroup` in the capability of simple intelligence (related URI: `GET /ISAPI/Smart/capabilities`).

### 2. Rule Parameters Configuration

#### Configure All Channels

Get people gathering parameters (related URI: `GET /ISAPI/Smart/group`).

Configure people gathering parameters (related URI: `PUT /ISAPI/Smart/group`).

#### Configure Specified Channel

Get the capability of people gathering parameters (related URI: `GET /ISAPI/Smart/group/<channelID>/capabilities`).

Get the people gathering parameters of a specified channel (related URI: GET /ISAPI/Smart/group/<channelID>).

Configure the people gathering parameters of a specified channel (related URI: PUT /ISAPI/Smart/group/<channelID>).

### Configure All Areas of Specified Channel

Get the capability of the parameters of all areas for people gathering (related URI: GET /ISAPI/Smart/group/<channelID>/regions/capabilities).

Get the parameters of all areas for people gathering (related URI: GET /ISAPI/Smart/group/<channelID>/regions).

Configure the parameters of all areas for people gathering (related URI: PUT /ISAPI/Smart/group/<channelID>/regions).

### Configure Specified Channel in Specified Channel

Get the capability of the parameters of specified areas for people gathering (related URI: GET /ISAPI/Smart/group/<channelID>/regions/<regionID>/capabilities).

Get the parameters of specified areas for people gathering (related URI: GET /ISAPI/Smart/group/<channelID>/regions/<regionID>).

Configure the parameters of specified areas for people gathering (related URI: PUT /ISAPI/Smart/group/<channelID>/regions/<regionID>). Note: The percentage refers to the percentage of people number accounts for the alarming area. An alarm will be triggered in the system when the percentage is higher than the set percentage. Otherwise, no alarm will be triggered.

## 3. (Optional) Arming Linkage Configuration

Get the capability of event linkage configuration (related URI: GET /ISAPI/Event/triggersCap), whether the message EventTriggersCap contains the node GroupDetectionTriggerCap.

Get the linkage parameters of people gathering (related URI: GET /ISAPI/Event/triggers/group-<channelID>).

Configure linkage parameters of people gathering (related URI: PUT /ISAPI/Event/triggers/group-<channelID>).

## 4. (Optional) Arming Time Configuration

Get the arming schedule of alarm outputs for all channels

Configure the arming schedule of people gathering for all channels (related URI: PUT /ISAPI/Event/schedules/groups).

Get the arming schedule of people gathering for a specified channel (related URI: GET /ISAPI/Event/schedules/groups/group-<channelID>).

Configure the arming schedule of people gathering for a specified channel (related URI: PUT /ISAPI/Event/schedules/groups/group-<channelID>). Note: 1. Supports setting up to 8 time periods in one day on the timeline. 2. Overlapped time between any two time periods is not allowed when setting the arming schedule.

## 5. Event Receiving: "eventType": "group", including the following.

detectionPictureTransType refers to the transmission method of the pictures.

The field value is bkgUrl for the picture URL or the picture encoding data via base64 when the value of detectionPictureTransType is "url" or "base64".

The field value is the picture Content-ID when the value of detectionPictureTransType is "binary".

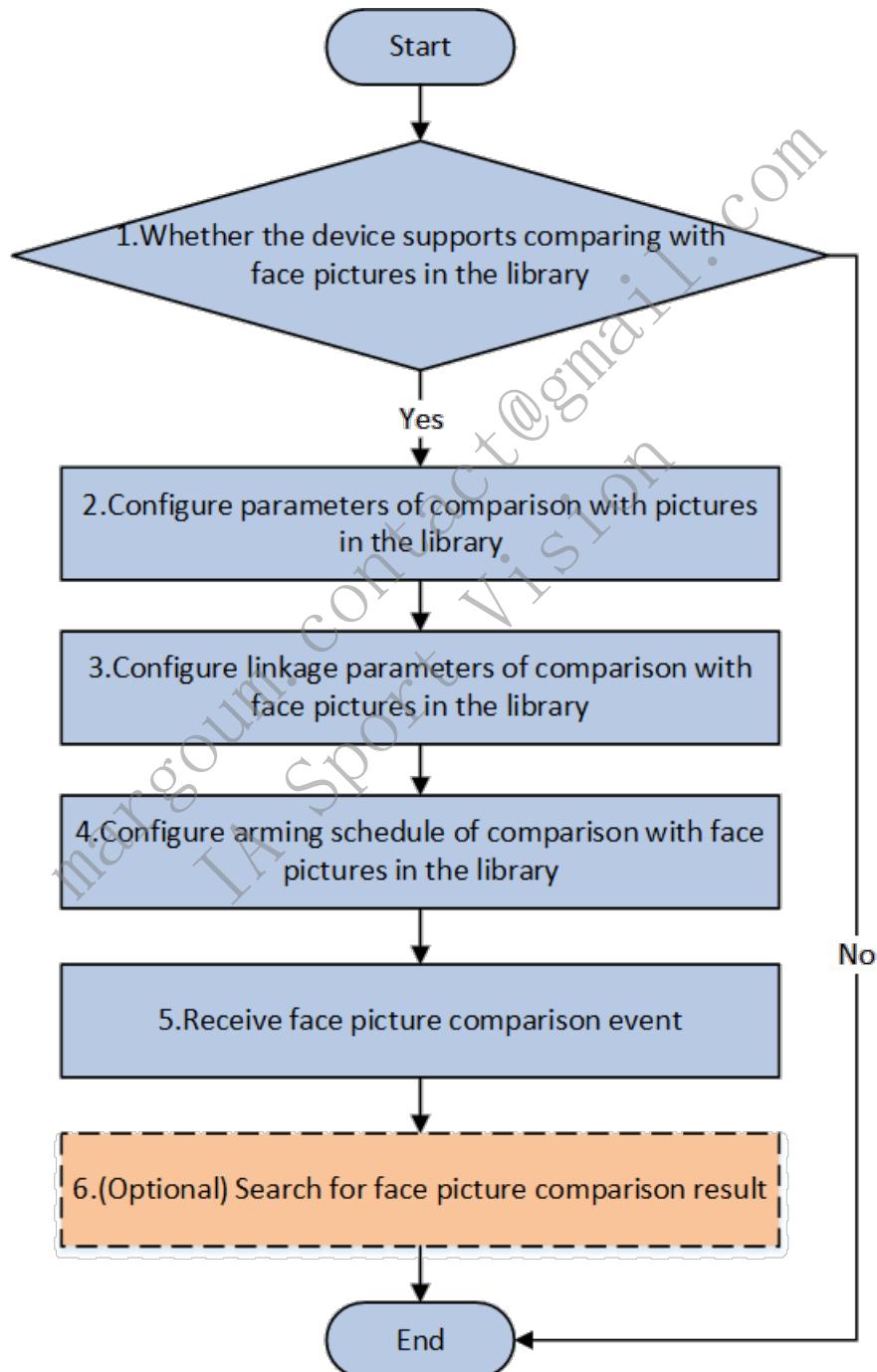
# 11 Face Capture

## 11.1 . Comparison with Pictures in the Library

### 11.1.1 Introduction to the Function

**Note:** This feature depends on the face picture library management (refer to Face Picture Library Management Field for details). Analyze and model the face captured in the camera, then compare the face picture with pictures in the library, and lastly return the comparison similarity.

### 11.1.2 API Calling Flow



#### 1. Whether the device supports comparing with face pictures in the library:

Get the capability node to evaluate whether the device supports comparing pictures with face pictures in the library, `GET /ISAPI/System/capabilities`; If the node `<isSupportFaceContrast>` is returned and its value is `true`, it indicates that the device supports comparing pictures with face pictures in the library.

#### 2. The parameters of comparison with pictures in the library including:

Get the capability of face picture comparison, GET /ISAPI/Intelligent/channels/<channelID>/faceContrast/capabilities. Configure the parameters of face picture comparison, PUT /ISAPI/Intelligent/channels/<channelID>/faceContrast; Get the configuration of face picture comparison, GET /ISAPI/Intelligent/channels/<channelID>/faceContrast; **Note: The rule parameters support configuring the triggering threshold of face similarity event and specifying information including face picture library ID.**

### 3. The linkage parameters of comparison with face pictures in the library:

Two sets of configurations linked to business systems are provided due to the front-end and back-end device differences:

**Linkage action of face picture library:** for front-end devices (e.g., Deepinview), a business system of single channel and multiple face picture libraries. Therefore it supports configuring different arming linkage parameters via each compared face picture library. Call GET /ISAPI/Event/triggersCap to evaluate whether it supports the event linkage configuration capability, it does if <FaceLibTriggerCap> is returned, and the relevant configuration APIs are as follows: Configure the linkage action parameters, PUT /ISAPI/Event/triggers/faceLib-<channelID>/<FDID>; if users have added a customized face picture library ID when creating one, they can also call PUT /ISAPI/Event/triggers/faceLib-<channelID>/<customFaceLibID>?FDType=custom to configure via customFaceLibID; get the linkage action parameters, GET /ISAPI/Event/triggers/faceLib-<channelID>/<FDID>; Customizing the face picture library ID follows the same flow.

**Linkage action of channel:** for back-end devices (e.g., DeepinMind), a business system of multiple channels and single face picture library. Therefore it supports configuring different arming linkage parameters via each video channel. Call GET /ISAPI/Event/triggersCap to evaluate whether it supports the event linkage configuration capability, it does if <FaceContrastTriggerCap> is returned, and the relevant configuration APIs are as follows: Configure the linkage action parameters of face picture comparison, PUT /ISAPI/Event/triggers/faceContrast-<channelID>; get the linkage action parameters of face picture comparison, GET /ISAPI/Event/triggers/faceContrast-<channelID>;

### 4. Arming schedule of comparison with face pictures in the library:

Two sets of arming schedule configurations linked to business systems are provided due to the front-end and back-end device differences: Arming schedule of face picture library: Configure the single-channel arming schedule of face picture library, PUT /ISAPI/Event/schedules/faceLib/<channelID>/<FDID>; if users have added a customized face picture library ID when creating one, they can also call PUT /ISAPI/Event/schedules/faceLib/<channelID>/<FDID>/<customFaceLibID>?FDType=custom to configure via customFaceLibID; get the single-channel arming schedule of face picture library, GET /ISAPI/Event/schedules/faceLib/<channelID>/<FDID>; Customizing the face picture library ID follows the same flow. Arming schedule of channel: Configure the arming schedule of face picture comparison, PUT /ISAPI/Event/schedules/faceContrast-<channelID>; get the arming schedule of face picture comparison, GET /ISAPI/Event/schedules/faceContrast-<channelID>;

### 5. Face picture comparison event:

When the face captured on the camera is compared with pictures in the library, and the similarity has reached the rule parameters including the threshold, the device will upload the face picture comparison event; the event information mainly includes comparison similarity, captured face and its attributes, and the corresponding registered face picture and its attributes, "eventType": "alarmResult".

### 6. Search for face picture comparison result:

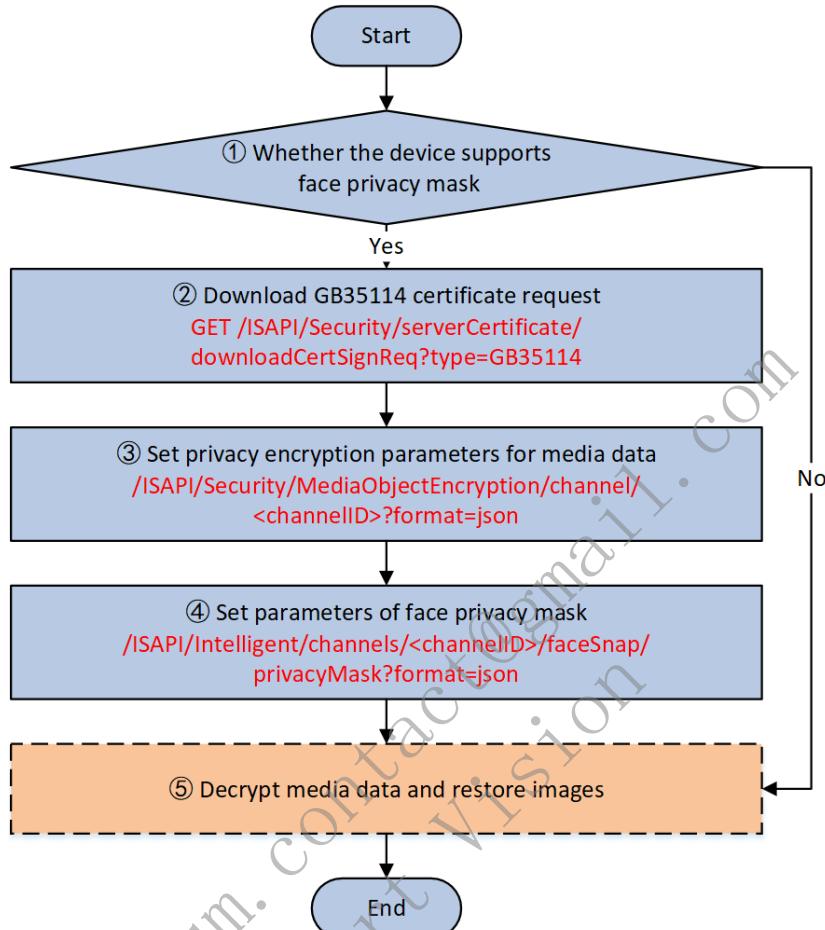
If supported by device, the face picture comparison result will be saved to the device local storage, and you can search for comparison history via the following API: Search for face picture comparison information, GET /ISAPI/Intelligent/FDLib/FCSearch.

## 11.2 Face Capture

### 11.2.1 Introduction to the Function

Face capture is to detect whether face pictures show up in images. If set conditions are met and there are faces showing up, the face pictures will be captured and uploaded as alarms.

### 11.2.2 API Calling Flow



#### 1. Whether the device supports face capture:

Get the system capabilities: `GET /ISAPI/Systemcapabilities`. If the node (its parent node is ) is returned as true, it indicates that the device supports face capture.

#### 2. Whether the device supports multi-scene face capture:

Get the intelligent capabilities of device: `GET /ISAPI/Intelligent/channels/<channelID>/capabilities`. If the node is returned and its value is true, it indicates that the device supports face capture.

#### 3. Set face picture capturing rules of multiple scenes:

If multi-scene capture of a channel is required, you can set the face picture capturing rules for multiple scenes.

Set the face capture rules of a specified scene: `PUT /ISAPI/Intelligent/channels/<channelID>/faceRule/multiScene/<SID>;`

Get the face capture rules of a specified scene: `GET /ISAPI/Intelligent/channels/<channelID>/faceRule/multiScene/<SID>;`

Set the face picture capturing rules of multiple scenes for a specified channel: `PUT /ISAPI/Intelligent/channels/<channelID>/faceRule/multiScene;;`

Get the face picture capturing rules of multiple scenes for a specified channel: `GET /ISAPI/Intelligent/channels/<channelID>/faceRule/multiScene.`

#### 4. Set face picture capturing rules of a specific scene.

If specific-scene capture of a channel is required, you can set the face picture capturing rules.

Get the capabilities of captured face picture rules `GET /ISAPI/Intelligent/channels/<channelID>/faceRule/capabilities;`

Set the face capture rule parameters: `PUT /ISAPI/Intelligent/channels/<channelID>/faceRule;`

Get the face capture rule parameters: `GET /ISAPI/Intelligent/channels/<channelID>/faceRule;`

Get the default parameters of face capture rule: `GET /ISAPI/Intelligent/channels/<channelID>/faceRule?defualt=<defualt>.`

## 5. Set linkage mode of face picture capture.

Set the linkage parameters of face capture: `PUT /ISAPI/Event/triggers/faceSnap-<channelID>;`

Get the linkage parameters of face capture: `GET /ISAPI/Event/triggers/faceSnap-<channelID>;`

Delete the linkage of face capture: `DELETE /ISAPI/Event/triggers/faceSnap-<channelID>.`

## 6. Configure the arming schedule of face capture.

Set the arming schedule of face capture for a specified channel: `PUT /ISAPI/Event/schedules/faceSnap/faceSnap-<channelID>;`

Get the arming schedule of face capture for a specified channel: `GET /ISAPI/Event/schedules/faceSnap/faceSnap-<channelID>;`

Set the arming schedule of face capture: `PUT /ISAPI/Event/schedules/faceSnap;`

Get the arming schedule of face capture: `GET /ISAPI/Event/schedules/faceSnap.`

## 7. Receive face capture and recognition events.

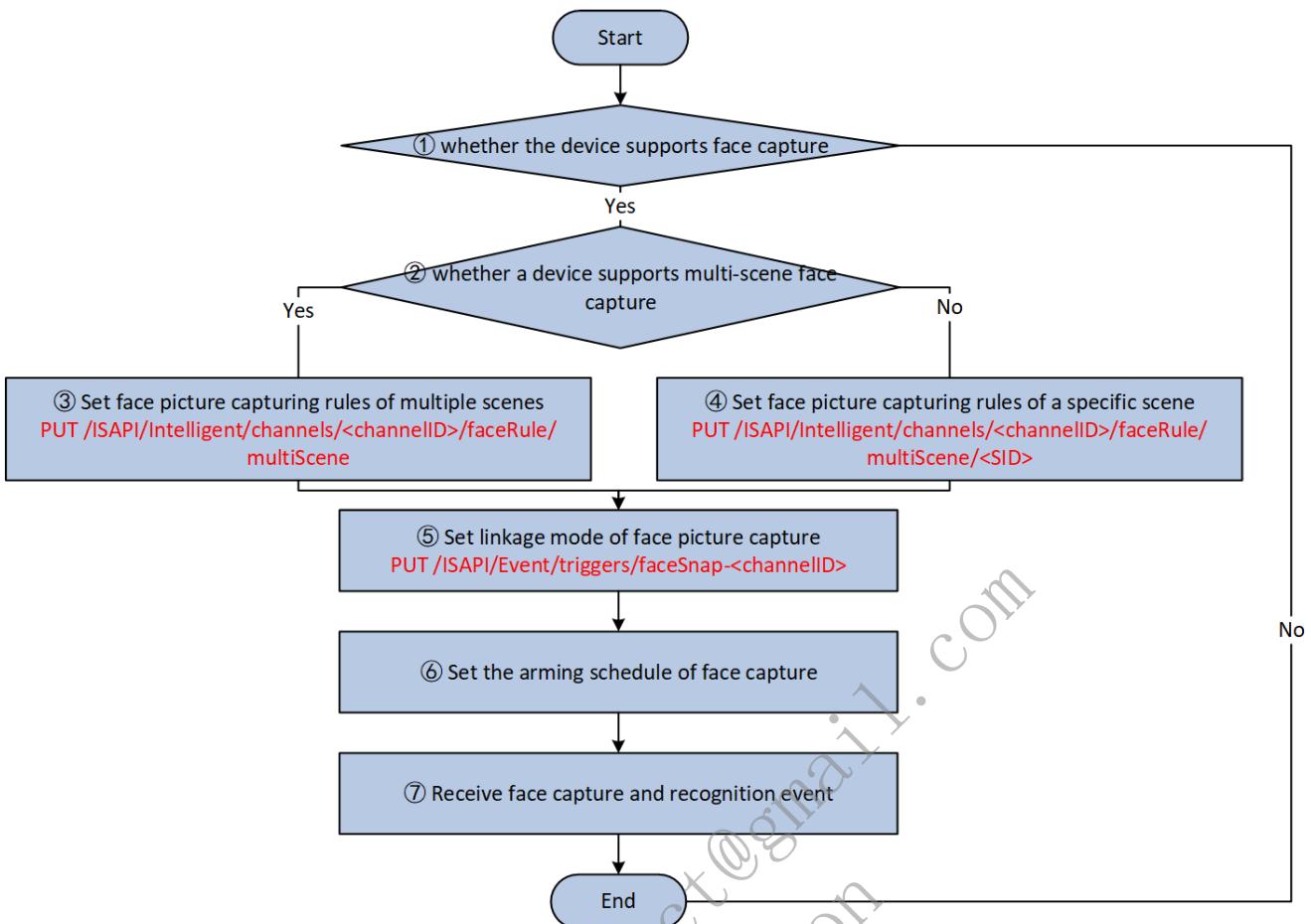
If the wanted face shows up in the set area, face recognition event will be reported. The event information includes face pictures and attribute information, etc. `"eventType": "faceCapture".`

## 11.3 Privacy Mask on Faces

### 11.3.1 Introduction to the Function

Privacy mask on faces is to encrypt privacy contents for face capture and media data (see "Video Privacy Mask" API calling flow); it is the process of privacy masking for all faces of cameras and the encrypted images can be restored. First of all, all faces need to be identified, the original media data of faces in the images and masking block data will be replaced, the original media data of faces will be added to the private data of the live view stream and encrypted, then reported to the user synchronously; and the user can decrypt the private data to get the original media data and restore the face image. **Note: The current image restoration only supports private streams, which needs to be restored by integrating the Playback Library.** The following figure shows the effect of face privacy mask by mosaic processing.

### 11.3.2 API Calling Flow



- 1. Whether the video channel supports face privacy mask.** Get the channel intelligence capabilities: GET `/ISAPI/Intelligent/channels/<channelID>/capabilities`, if the node `<isSupportFacePrivacyMask>` is returned as true, it indicates that the device supports face privacy masking.
- 2. Set privacy encryption parameters in media data.** Users apply the encryption parameters including secondary key `secretKey` and the update time of the primary key `secretKeyUpdateFrequency` to devices, etc. Get the encryption capabilities of media data privacy: GET `/ISAPI/Security/MediaObjectEncryption/channel/<channelID>/capabilities?format=json`. Set the encryption parameters of media data privacy PUT `/ISAPI/Security/MediaObjectEncryption/channel/<channelID>?format=json`. Get the encryption parameters of media data privacy: GET `/ISAPI/Security/MediaObjectEncryption/channel/<channelID>?format=json`.
- 3. Set parameters of face privacy mask.** Users can enable face privacy mask for a channel `enabled`. Get the capability of face privacy masking: GET `/ISAPI/Intelligent/channels/<channelID>/faceSnap/privacyMask/capabilities?format=json`; Set the parameters of face privacy mask: PUT `/ISAPI/Intelligent/channels/<channelID>/faceSnap/privacyMask?format=json`; Get the parameters of face privacy mask: GET `/ISAPI/Intelligent/channels/<channelID>/faceSnap/privacyMask?format=json`;
- 4. (Optional) Decrypt media data and restore images.** The device generates the primary key and updates the primary key periodically according to the primary key update time in the media data privacy encryption parameters, then replaces the original media data with the masking block in the specified sensitive area in the live view screen, after which the images in the sensitive area are encrypted by the primary key and added to the private data of the live view stream, and at the same time the primary key is encrypted by the secondary key and added to the private data of the preview stream. The key is also encrypted by the secondary key and added to the private data of the live view stream. The user can parse the private data in the live view stream and decrypt it with the secondary key to get the primary key, then decrypt the private data with the primary key to get the original media data and restore the image.

**Note:** At present, image restoration is only supported by private streams. You need to get the private data in the live view stream by integrating Playback Library, and get the primary key by decrypting the secondary key, then decrypt the private data by the primary key to get the original media data; and finally call the Playback Library API to restore the image. You can refer to Playback library interface integration manual for playback library related APIs.

## 12 Road Traffic

---

### 12.1 Traffic Enforcement of Vehicles

#### 12.1.1 Introduction to the Function

##### 12.1.1.1 Smart Linkage System for Intelligent Transportation

**Application Scenario:** The smart linkage system of intelligent traffic consists of cameras and traffic speed domes, which support functions of remote monitoring via the network, video servers, and high definition, for traffic violation enforcement, traffic incident analysis, and traffic data statistics. Without people attending, the system can automatically capture and save key information for automatic enforcement, which helps save on labor costs, improve enforcement efficiency, and reduce disputes. The smart linkage system can be widely used in places that require large-scale high-definition monitoring, such as roads, airports, stations, docks, ports, tourist attractions, and streets.

**Enforcement Function:** The traffic enforcement event detection supports the function of violation enforcement, which can automatically detect and capture the evidence of events, including illegal parking, wrong-way driving, illegal U-turning, driving on the lane line, illegal lane change, and motor vehicle on non-motor vehicle lane, on the roads with no more than four lanes. The enforcement results will be uploaded as alarms in real time.

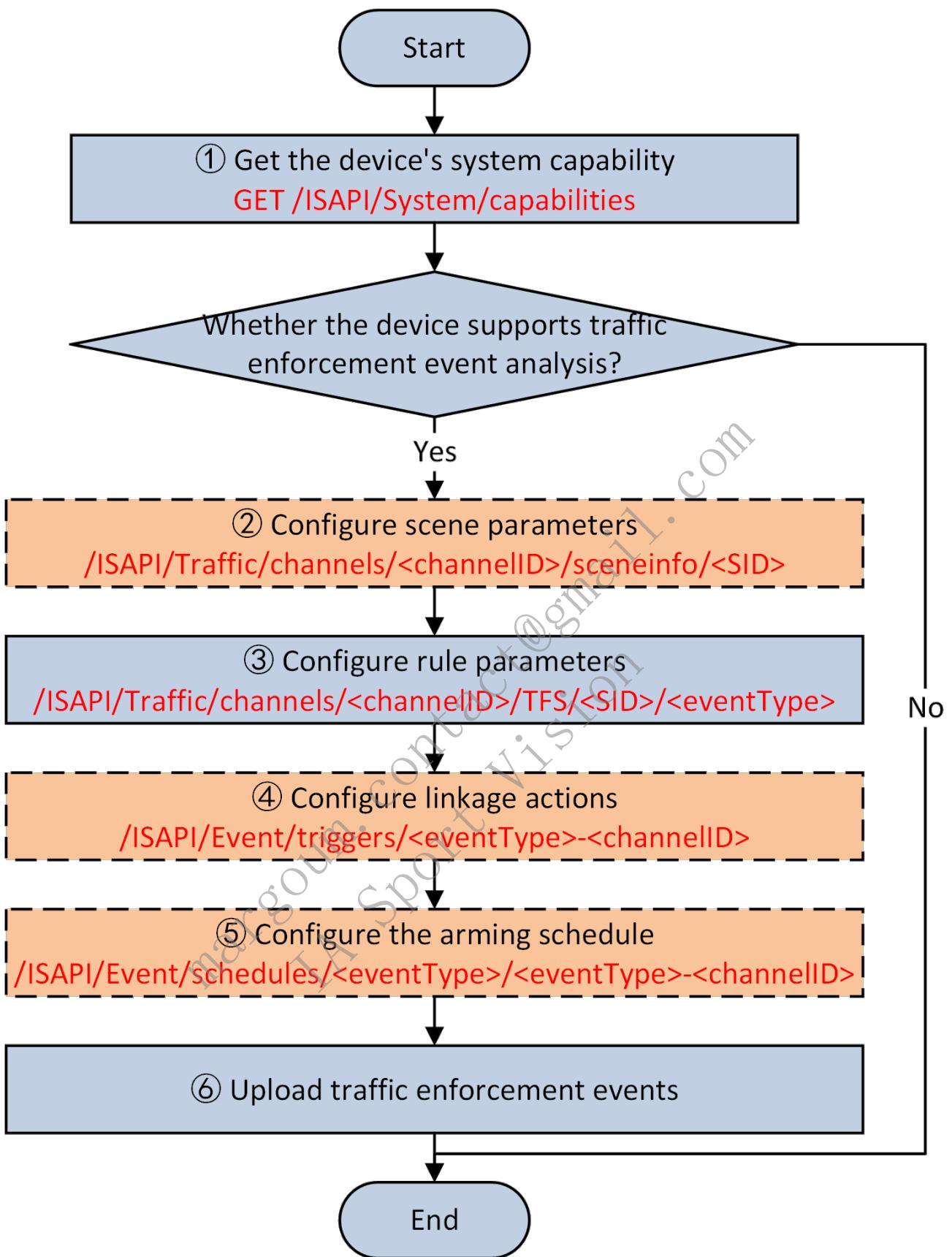
##### 12.1.1.2 Smart Linkage System for Parking Management

**Application Scenario:** The smart linkage system for parking management (hereinafter referred to as "the system") consists of parking speed domes and cameras (hereinafter referred to as "the devices"). The parking speed dome supports functions of remote monitoring via the network, video servers, and high definition. The system is easy to install and use without complex wiring. It focuses on the detection of the parallel parking event during enforcement. Without people attending, the system can automatically capture and save key information for automatic enforcement of the vehicle information in the parking space, which helps save on labor costs.

The devices are mainly used in places like the parking lot.

**Parallel Parking Detection:** The parallel parking detection can detect the parallel parking event of up to four parking spaces during enforcement and upload the vehicle information in the parking space in real time.

#### 12.1.2 API Calling Flow



#### 1. Get the device's system capability:

Get the device's system capability to check the traffic enforcement events supported by the device: GET /ISAPI/System/capabilities. The value of the node TFSEventSupport.opt can be illegalParking (illegal parking detection), wrongDirection (wrong-way driving detection), crossLane (driving on the lane line detection), laneChange (illegal lane change detection), vehicleExist (motor vehicle on non-motor vehicle lane detection), turnRound (illegal U-turning detection), parallelParking (parallel parking detection), notKeepDistance (not keeping vehicle distance detection), notSlowZebraCrossing (not slowing down at zebra crossing detection), overtakeRightSide (overtaking on the right detection), lowSpeed (driving in low speed detection), dragRacing

(street racing detection), changeLaneContinuously (continuous lane change detection), SSharpDriving (slalom driving detection), largeVehicleOccupyLine (lane occupation by large-sized vehicle detection), jamCrossLine (queue jumping detection), blackSmokeVehicle (vehicle emitting black smoke detection), or turnRightStop (large-sized vehicle not slowing down when turning right detection).

## 2. Configure scenes:

Get the capability of configuring basic channel parameters to check whether the node `isEvidenceGettingSupport` exists and its value is true: `GET /ISAPI/Traffic/channels/<channelID>/capability`. If so, it indicates that the device supports configuring scene and rule parameters.

Get the capability of configuring basic scene parameters of a specific channel: `GET /ISAPI/Traffic/channels/<channelID>/sceneinfo/capabilities`.

Set the basic scene parameters for a specific channel: `PUT /ISAPI/Traffic/channels/<channelID>/sceneinfo/<SID>`. If you need to use the scene, the node `SceneInfo.enabled` should be set to true.

## 3. Configure rules:

Get the capability of configuring the **event** rule parameters: `GET /ISAPI/Traffic/channels/<channelID>/TFS/<SID>/XXX/capabilities`.

Get the **event** rule parameters: `GET /ISAPI/Traffic/channels/<channelID>/TFS/<SID>/XXX`.

Set the **event** rule parameters: `PUT /ISAPI/Traffic/channels/<channelID>/TFS/<SID>/XXX`.

### Notes:

- i. The value of `XXX` can be set to: illegalParking (illegal parking detection), wrongDirection (wrong-way driving detection), crossLane (driving on the lane line detection), laneChange (illegal lane change detection), vehicleExist (motor vehicle on non-motor vehicle lane detection), turnRound (illegal U-turning detection), parallelParking (parallel parking detection), notKeepDistance (not keeping vehicle distance detection), notSlowZebraCrossing (not slowing down at zebra crossing detection), overtakeRightSide (overtaking on the right detection), lowSpeed (driving in low speed detection), dragRacing (street racing detection), changeLaneContinuously (continuous lane change detection), SSharpDriving (slalom driving detection), largeVehicleOccupyLine (lane occupation by large-sized vehicle detection), jamCrossLine (queue jumping detection), blackSmokeVehicle (vehicle emitting black smoke detection), or turnRightStop (large-sized vehicle not slowing down when turning right detection).

e.g.: Get the capability of configuring parameters of illegal parking detection: `GET /ISAPI/Traffic/channels/<channelID>/TFS/<SID>/illegalParking/capabilities`.

Get or set parameters of illegal parking detection: `GET/PUT /ISAPI/Traffic/channels/<channelID>/TFS/<SID>/illegalParking`.

- ii. For traffic speed domes, the rule parameters are required. For box cameras, the rule parameters are not required. The scene No. is 1 by default.

## 4. Configure linkage actions:

Check whether the device supports configuring linkage actions of the illegal parking event by calling `GET /ISAPI/Event/triggersCap` and check whether the message `EventTriggersCap` contains the node `IllegalParkingTriggerCap`.

Get the default or configured linkage action of the illegal parking event: `GET /ISAPI/Event/triggers/illegalParking-<channelID>`.

Configure the linkage actions of the illegal parking event: `PUT /ISAPI/Event/triggers/illegalParking-<channelID>`.

**Notes:** Only illegal parking detection supports linkage action configurations. You do not need to configure linkage actions for other events and they will be uploaded to the center by default.

## 5. Configure arming schedules:

Check whether the device supports configuring the arming schedule for the illegal parking detection event by calling `GET /ISAPI/Event/capabilities` and checking whether the message `EventCap` contains the node `isSupportIllegalParking`.

Configure the arming schedule for the illegal parking event: `GET/PUT /ISAPI/Event/schedules/illegalParkings/illegalParking-<channelID>`.

Check whether the device supports configuring the arming schedule for the wrong-way driving detection event by calling `GET /ISAPI/Event/capabilities` and checking whether the message `EventCap` contains the node `isSupportWrongDirection`.

Configure the arming schedule for the wrong-way driving event: `GET/PUT /ISAPI/Event/schedules/wrongDirections/wrongDirection-<channelID>`.

Check whether the device supports configuring the arming schedule for the illegal U-turning event by calling `GET /ISAPI/Event/capabilities` and checking whether the message `EventCap` contains the node `isSupportTurnRound`.

Configure the arming schedule for the illegal U-turning event: `GET/PUT /ISAPI/Event/schedules/turnRounds/turnRound-<channelID>`.

Check whether the device supports configuring the arming schedule for the driving on the lane line event by calling `GET /ISAPI/Event/capabilities` and checking whether the message `EventCap` contains the node `isSupportCrossLane`.

Configure the arming schedule for the driving on the lane line event: `GET/PUT /ISAPI/Event/schedules/crossLanes/crossLane-<channelID>`.

Check whether the device supports configuring the arming schedule for the illegal lane change detection event by calling `GET /ISAPI/Event/capabilities` and checking whether the message `EventCap` contains the node `isSupportLaneChange`.

Configure the arming schedule for the illegal lane change event: `GET/PUT /ISAPI/Event/schedules/laneChanges/laneChange-<channelID>`.

Check whether the device supports configuring the arming schedule for the motor vehicle on non-motor vehicle lane event by calling `GET /ISAPI/Event/capabilities` and checking whether the message `EventCap` contains the node `isSupportVehicleExist`.

Configure the arming schedule for the motor vehicle on non-motor vehicle lane event: `GET/PUT /ISAPI/Event/schedules/vehicleExists/vehicleExist-<channelID>`.

Check whether the device supports configuring the arming schedule for the parallel parking detection event by calling `GET /ISAPI/Event/capabilities` and checking whether the message `EventCap` contains the node `isSupportParallelParking`.

Configure the arming schedule for the parallel parking event: `GET/PUT /ISAPI/Event/schedules/parallelParkings/parallelParking-<channelID>`.

### Notes:

- i. You can determine whether to configure the arming schedule or not. If the arming schedule is not configured, it is all-day arming by default.

- ii. Only the illegal parking detection, wrong-way driving detection, illegal U-turning detection, driving on the lane line detection, illegal lane change detection, motor vehicle on non-motor vehicle lane detection, and parallel parking detection support arming schedule configuration. For other events, configuring the arming schedule is not supported and the default arming schedule is all-day arming.
- 6. Upload the traffic enforcement events** ("eventType": "TFS"). In the event message, TFS.illegalTrafficEvent is the specific traffic violation event, TFS.VehicleInfo is the vehicle information of the traffic violation, and TFS.PlateInfo is the license plate information of the vehicle violation. The detectionPicturesNumber refers to the number of detected violation pictures. The detectionPictureTransType refers to the transmission method of the pictures.

### 12.1.3 Exception Handling

#### 12.1.3.1 Error Codes

statusCode	statusString	subStatusCode	errorCode	errorMsg	Description	Rema
6	Invalid Content	intelligentTrafficMutexWithHighFrames	0x60008014		Disable all functions of traffic incident detection, traffic violation enforcement, and traffic data statistics, or adjust the video frame rate to 50 or below.	
6	Invalid Content	intelligentTrafficMutexWithHighFramesEx	0x60008018		Disable all functions of traffic incident detection, traffic violation enforcement, traffic data statistics, and vehicle detection, or adjust the video frame rate to 50 or below.	

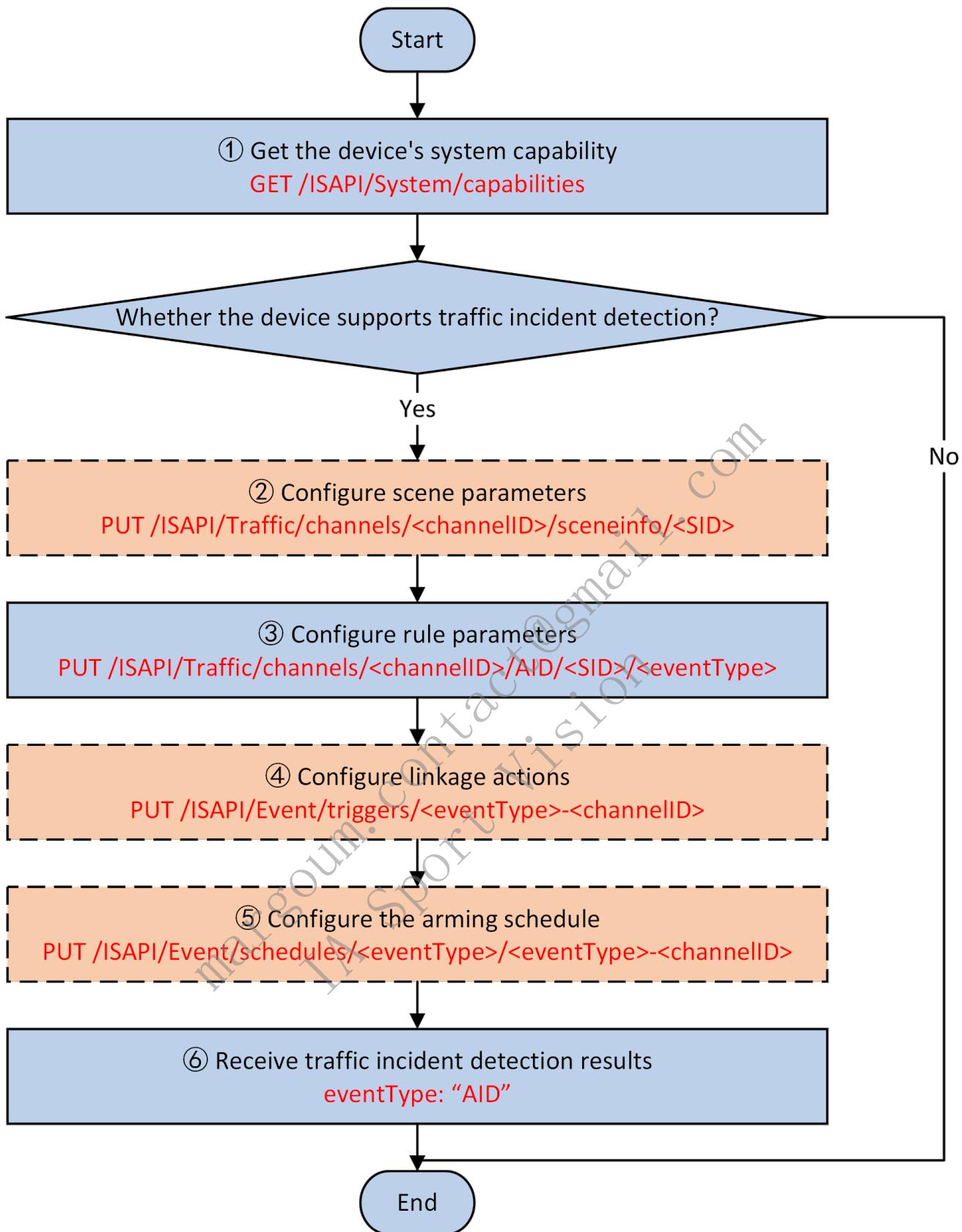
## 12.2 Traffic Incidents of Vehicles

### 12.2.1 Introduction to the Function

**Application Scenario:** The smart linkage system of intelligent traffic consists of cameras and traffic speed domes, which support functions of remote monitoring via the network, video servers, and high definition, for traffic violation enforcement, traffic incident analysis, and traffic data statistics. Without people attending, the system can automatically capture and save key information for automatic enforcement, which helps save on labor costs, improve enforcement efficiency, and reduce disputes. The smart linkage system can be widely used in places that require large-scale high-definition monitoring, such as roads, airports, stations, docks, ports, tourist attractions, and streets.

**Intelligent Traffic Functions:** The AID function can detect traffic incidents that occurred on the roads with no more than four lanes and trigger alarms in real time. Supported incidents include thrown object detection, pedestrian detection, congestion detection, roadblock detection, traffic accident detection, fog detection, etc.

### 12.2.2 API Calling Flow



#### 1. Get the device's system capability:

Get the device's system capability to check the traffic incidents supported by the device: GET /ISAPI/System/capabilities. The value of the node AIDEEventSupport.opt can be abandonedObject (thrown object detection), pedestrian (pedestrian detection), congestion (congestion detection), roadBlock (roadblock detection), construction (construction detection), trafficAccident (traffic incident detection), fogDetection (fog detection), or obstacle (obstacle detection).

#### 2. Configure scenes:

Get the capability of configuring basic channel parameters to check whether the node `isEvidenceGettingSupport` exists and its value is true: `GET /ISAPI/Traffic/channels/<channelID>/capability`. If so, it indicates that the device supports configuring scene and rule parameters.

Get the capability of configuring basic scene parameters of a specific channel: `GET /ISAPI/Traffic/channels/<channelID>/sceneinfo/capabilities`.

Set the basic scene parameters for a specific channel: `PUT /ISAPI/Traffic/channels/<channelID>/sceneinfo/<SID>`. If you need to use the scene, the node `SceneInfo.enabled` should be set to true.

### 3. Configure rules:

Get the capability of configuring the **event** rule parameters: `GET /ISAPI/Traffic/channels/<channelID>/AID/<SID>/XXX/capabilities`.

Get the **event** rule parameters: `GET /ISAPI/Traffic/channels/<channelID>/AID/<SID>/XXX`.

Set the **event** rule parameters: `PUT /ISAPI/Traffic/channels/<channelID>/AID/<SID>/XXX`.

#### Notes:

1. The value of `XXX` can be set to: abandonedObject (thrown object detection), pedestrian (pedestrian detection), congestion (congestion detection), roadBlock (roadblock detection), construction (construction detection), trafficAccident (traffic accident detection), fogDetection (fog detection), or obstacle (obstacle detection).

e.g.: Get the capability of configuring parameters of thrown object detection: `GET /ISAPI/Traffic/channels/<channelID>/AID/<SID>/abandonedObject/capabilities`.

Get or set parameters of thrown object detection: `GET/PUT /ISAPI/Traffic/channels/<channelID>/AID/<SID>/abandonedObject`.

2. For traffic speed domes, the rule parameters are required. For box cameras, the rule parameters are not required. The scene No. is 1 by default.

### 4. Configure linkage actions:

Check whether the device supports configuring linkage actions for the thrown object event by calling `GET /ISAPI/Event/triggersCap` and checking whether the message `EventTriggersCap` contains the node `AbandonedObjectTriggerCap`.

Configure linkage actions for the thrown object event: `GET/PUT /ISAPI/Event/triggers/abandonedObject-<channelID>`.

Check whether the device supports configuring linkage actions for the pedestrian detection event by calling `GET /ISAPI/Event/triggersCap` and checking whether the message `EventTriggersCap` contains the node `PedestrianTriggerCap`.

Configure linkage actions for the pedestrian detection: `GET/PUT /ISAPI/Event/triggers/pedestrian-<channelID>`.

Check whether the device supports configuring linkage actions for the congestion event by calling `GET /ISAPI/Event/triggersCap` and checking whether the message `EventTriggersCap` contains the node `CongestionTriggerCap`.

Configure linkage actions for the congestion event: `GET/PUT /ISAPI/Event/triggers/congestion-<channelID>`.

Check whether the device supports configuring linkage actions for the roadblock detection event by calling `GET /ISAPI/Event/triggersCap` and checking whether the message `EventTriggersCap` contains the node `RoadBlockTriggerCap`.

Configure linkage actions for the roadblock detection event: GET/PUT /ISAPI/Event/triggers/roadBlock-<channelID>.

Check whether the device supports configuring linkage actions for the construction detection event by calling GET /ISAPI/Event/triggersCap and checking whether the message EventTriggersCap contains the node ConstructionTriggerCap.

Configure linkage actions for the construction detection: GET/PUT /ISAPI/Event/triggers/construction-<channelID>.

Check whether the device supports configuring linkage actions for the traffic accident detection event by calling GET /ISAPI/Event/triggersCap and checking whether the message EventTriggersCap contains the node TrafficAccidentTriggerCap.

Configure linkage actions for the traffic accident detection: GET/PUT /ISAPI/Event/triggers/trafficAccident-<channelID>.

#### Notes:

1. You can determine whether to configure the linkage actions or not. If the linkage actions are not configured, the device will upload the event to the center by default.
2. The fog detection and obstacle detection do not support linkage action configuration. The device will upload them to the center by default.

#### 5. Configure arming schedules:

Check whether the device supports configuring the arming schedule for the thrown object event by calling GET /ISAPI/Event/capabilities and checking whether the message EventCap contains the node isSupportAbandonedObject.

Configure the arming schedule for the thrown object event: GET/PUT /ISAPI/Event/schedules/abandonedObjects/abandonedObject-<channelID>.

Check whether the device supports configuring the arming schedule for the pedestrian detection event by calling GET /ISAPI/Event/capabilities and checking whether the message EventCap contains the node isSupportPedestrian.

Configure the arming schedule for the pedestrian detection: GET/PUT /ISAPI/Event/schedules/pedestrians/pedestrian-<channelID>.

Check whether the device supports configuring the arming schedule for the congestion event by calling GET /ISAPI/Event/capabilities and checking whether the message EventCap contains the node isSupportCongestion.

Configure the arming schedule for the congestion event: GET/PUT /ISAPI/Event/schedules/congestions/congestion-<channelID>.

Check whether the device supports configuring the arming schedule for the roadblock detection event by calling GET /ISAPI/Event/capabilities and checking whether the message EventCap contains the node isSupportRoadBlock.

Configure the arming schedule for the roadblock detection event: GET/PUT /ISAPI/Event/schedules/roadBlocks/roadBlock-<channelID>.

Check whether the device supports configuring the arming schedule for the construction detection event by calling GET /ISAPI/Event/capabilities and checking whether the message EventCap contains the node isSupportConstruction.

Configure the arming schedule for the construction detection: GET/PUT /ISAPI/Event/schedules/constructions/construction-<channelID>.

Check whether the device supports configuring the arming schedule for the traffic accident detection event by calling `GET /ISAPI/Event/capabilities` and checking whether the message `EventCap` contains the node `isSupportTrafficAccident`.

Configure the arming schedule for the traffic incident detection: `GET/PUT /ISAPI/Event/schedules/trafficAccidents/trafficAccident-<channelID>`.

Check whether the device supports configuring the arming schedule for the fog detection by calling `GET /ISAPI/Event/capabilities` and checking whether the message `EventCap` contains the node `isSupportFogDetection`.

Configure the arming schedule for the fog detection: `GET/PUT /ISAPI/Event/schedules/fogDetection/fogDetection-<channelID>`.

#### **Notes:**

1. You can determine whether to configure the arming schedule or not. If the arming schedule is not configured, it is all-day arming by default.
2. The obstacle detection does not support arming schedule configuration. The default arming schedule of the device is all-day arming.
6. **Upload the traffic incident detection results** (`"eventType": "AID"`). In the event message, `AID.illegalTrafficEvent` is the specific traffic incident, `AID.illegalTrafficEvent` is the vehicle information, and `AID.PlateInfo` is the license plate information of the vehicle. The `detectionPicturesNumber` refers to the number of pictures in the alarm. The `detectionPictureTransType` refers to the transmission method of the pictures.

### **12.2.3 Exception Handling**

#### **12.2.3.1 Error Codes**

<b>statusCode</b>	<b>statusString</b>	<b>subStatusCode</b>	<b>errorCode</b>	<b>errorMsg</b>	<b>Description</b>	<b>Rema</b>
6	Invalid Content	intelligentTrafficMutexWithHighFrames	0x60008014		Disable all functions of traffic incident detection, traffic violation enforcement, and traffic data statistics, or adjust the video frame rate to 50 or below.	
6	Invalid Content	intelligentTrafficMutexWithHighFramesEx	0x60008018		Disable all functions of traffic incident detection, traffic violation enforcement, traffic data statistics, and vehicle detection, or adjust the video frame rate to 50 or below.	

## 13 Two-Way Audio

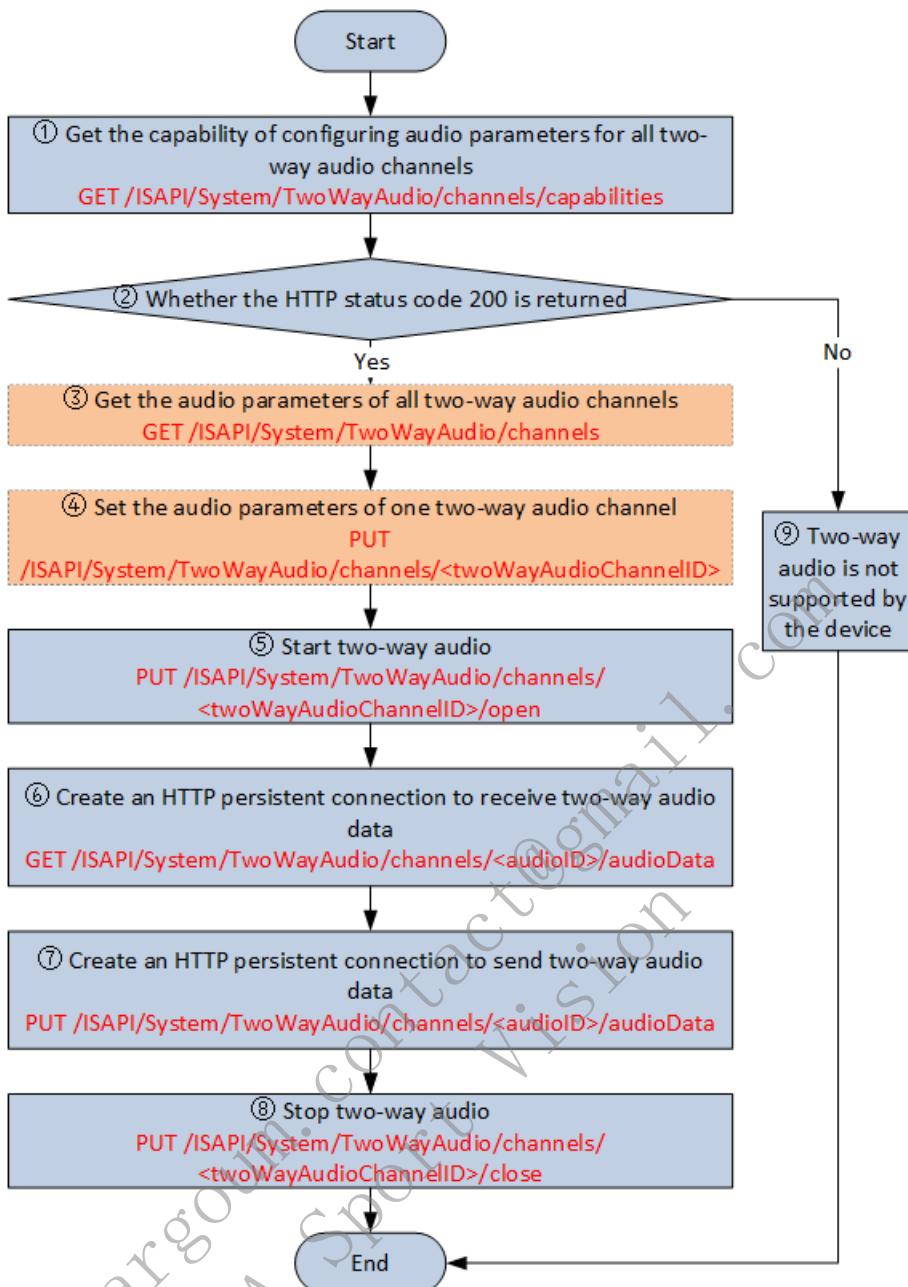
### 13.1 Two-Way Audio

#### 13.1.1 Introduction to the Function

This function is for the two-way audio between the client and device. When you call the API to implement the two-way audio, it requires the client to collect and encode the local audios and to decode the device audio data.

Note: Before starting two-way audio, check whether the way of collecting device audio data, audio encoding format, volume, etc, is correct or not. The audio encoding format of the client should be consistent with the device audio encoding format.

#### 13.1.2 API Calling Flow



1. Get the capability of configuring audio parameters for all two-way audio channels: `GET /ISAPI/System/TwoWayAudio/channels/capabilities`. Check whether the device supports two-way audio. If the HTTP status code `200 OK` is returned, then the device supports two-way audio.
2. Get the audio parameters of all two-way audio channels: `GET /ISAPI/System/TwoWayAudio/channels`. Get the No. of two-way audio channel for response and parsing `<id>`, audio type `<audioCompressionType>`, frame rate `<audioBitRate>`, sampling rate `<audioSamplingRate>` and so on, to ensure the consistent of audio format during the two-way audio process.
3. Set the audio parameters of one two-way audio channel: `PUT /ISAPI/System/TwoWayAudio/channels/<audioID>`. This step is optional, and the `<audioID>` is the channel No. `<id>` responded and parsed in step 2.
4. Start two-way audio: `PUT /ISAPI/System/TwoWayAudio/channels/<audioID>/open`.
5. Create an HTTP persistent connection to receive the two-way audio data: `GET /ISAPI/System/TwoWayAudio/channels/<audioID>/audioData`.
6. Create an HTTP persistent connection to send the two-way audio data: `PUT /ISAPI/System/TwoWayAudio/channels/<audioID>/audioData`.
7. When the two-way audio ends, the client disables the persistent connections in steps 6 and 7 and sends the message of stopping the two-way audio: `PUT /ISAPI/System/TwoWayAudio/channels/<audioID>/close`.

Note:

- Two-way audio over ISAPI supports digest authentication.

- It supports the two-way audio mode or sending only mode (the platform only call `PUT /ISAPI/System/TwoWayAudio/channels/<audioID>/audioData` to send the audio data to devices).
- The `<audioID>` in step 3, 4, 5, 6, and 7 is the two-way audio channel No. which starts from 1. It can be parsed from the `<id>` in the device response message of the step 2.
- The audio type `<audioCompressionType>`, frame rate `<audioBitRate>`, and sampling rate `<audioSamplingRate>` can be parsed from the device response message of the step 2 for audio playing and collection in the step 5 and 6.
- The step 6 and 7 use persistent connections. In the HTTP request headers, you don't need to set `Content-Length`, but need to set the `Connection: keep-alive` and `Content-Type: application/octet-stream`.
- When the encoding format of audio data in the step 5 and 6 is AAC/MP2L2/MP3, the audio data length is variable, so the frame header should be lengthened by 4 bytes. When the encoding format is G.722.1/G.711alaw/G.711ulaw/G.726/G.729/G.729a/G.729b/PCM/ADPCM/Opus, the audio data length is fixed, there is no need to lengthen the frame header. For example, if the content length of a frame of MP3 audio data is 576 bytes and the header length is 4 bytes, the big end data is `0x00000240`, the total length is 580 bytes.
- The damaged audio file in the AC3 format is usually incompatible when using. Currently, the audio algorithm library does not support this format.

Statistics of audio types with fixed lengths: |Audio Type|Fixed Length|Frame Interval|-----|-----|-----|  
 |G.722.1|80|40ms| |G.711alaw|160/320|20ms/40ms| |G.711ulaw|160/320|20ms/40ms| |G.726|80|40ms| |G.729|10|10ms|  
 |G.729a|10|10ms| |G.729b|10|10ms| |PCM|1920|None| |ADPCM|80|20ms| |Opus|32/64|20ms/40ms|

### 13.1.3 Exception Handling

#### 13.1.3.1 Error Codes

statusCode	statusString	subStatusCode	errorCode	errorMsg	Description	Re
4	Invalid Operation	twoWayAudioInProgressPleaseWait	0x40002068	Two-way audio in progress...Please wait.	Two-way audio in progress...Please wait.	The operation is all TV at in progress...Please wait.

### 13.1.4 Message Format and Example

#### 13.1.4.1 Receive Two-Way Audio Data

```
GET /ISAPI/System/TwoWayAudio/channels/1/audioData HTTP/1.1
Host: 10.17.115.128
Connection: keep-alive
Content-Type: application/octet-stream

HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: application/octet-stream
```

//The following are the two-way audio data sent by the device.

#### 13.1.4.2 Send Two-Way Audio Data

```

PUT /ISAPI/System/TwoWayAudio/channels/1/audioData HTTP/1.1
Host: 10.17.115.128
Connection: keep-alive
Content-Type: application/octet-stream

//The following are the two-way audio data sent by the client.

```

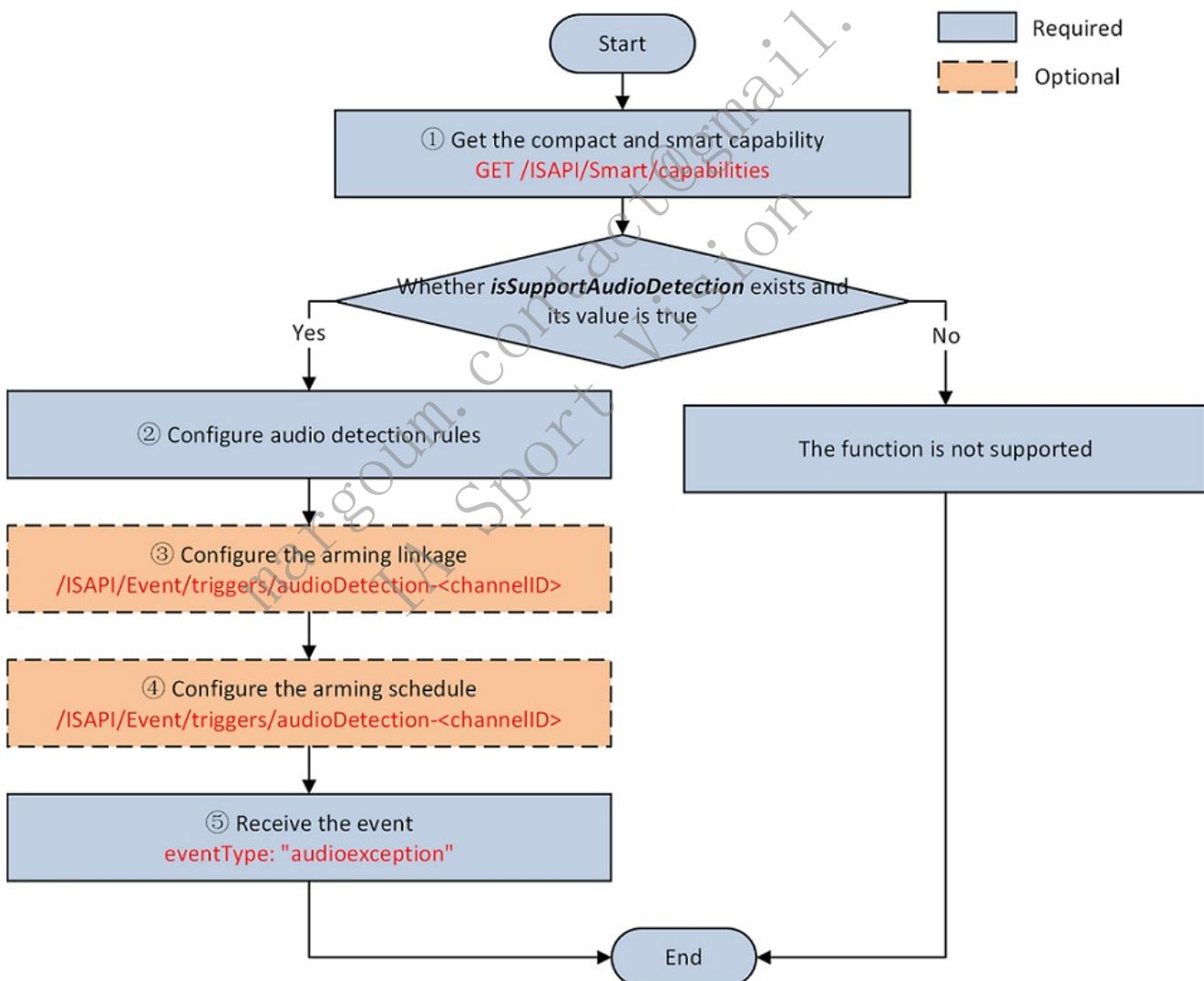
## 14 Audio Recognition

### 14.1 Audio Detection

#### 14.1.1 Introduction to the Function

Detects the sound change, including the sudden increase or decrease of sound intensity and audio loss, of devices or the surrounding environment. Audio loss detection is used to timely detect audio loss situations, that are due to the audio configuration error or pickup damage.

#### 14.1.2 API Calling Flow



##### 1. Get the compact and smart capability.

Get the compact and smart capability: `GET /ISAPI/Smart/capabilities`. Check whether the `isSupportAudioDetection` is returned with true value.

##### 2. Configure rules.

**Configure rules for all channels:**

Get audio detection parameters of all channels: GET /ISAPI/Smart/AudioDetection/channels.

Set audio detection parameters of all channels: PUT /ISAPI/Smart/AudioDetection/channels.

#### **Configure rules for a specified channel:**

Get the capability of audio detection parameters of a specified channel: GET  
/ISAPI/Smart/AudioDetection/channels/<channelID>/capabilities.

Get audio detection parameters of a specified channel: GET  
/ISAPI/Smart/AudioDetection/channels/<channelID>.

Set audio detection parameters of a specified channel: PUT  
/ISAPI/Smart/AudioDetection/channels/<channelID>.

#### **3. (Optional) Configure arming linkages.**

Get the event linkage configuration capability: GET /ISAPI/Event/triggersCap. Check if AudioExceptionCap is contained in EventTriggersCap.

Get the audio detection linkage parameters: GET /ISAPI/Event/triggers/audioDetection-<channelID>.

Set the audio detection linkage parameters: PUT /ISAPI/Event/triggers/audioDetection-<channelID>.

#### **4. (Optional) Configure the arming schedule.**

Get the audio detection arming schedule for all channels: GET /ISAPI/Event/schedules/audioDetections.

Set the audio detection arming schedule for all channels: PUT /ISAPI/Event/schedules/audioDetections.

Get the audio detection arming schedule for a specified channel: GET  
/ISAPI/Event/schedules/audioDetections/audioDetection-<channelID>.

Set the audio detection arming schedule for a specified channel: PUT  
/ISAPI/Event/schedules/audioDetections/audioDetection-<channelID>.

Notes:

- i. Up to eight time periods can be configured on the timeline for one day.
- ii. Time of each two time periods cannot be overlapped.

#### **5. Receive the event.**

The event type (eventType) is audioexception, and it contains alarmType (the alarm type, including audio loss, sudden decrease of sound intensity, and sudden increase of sound intensity).

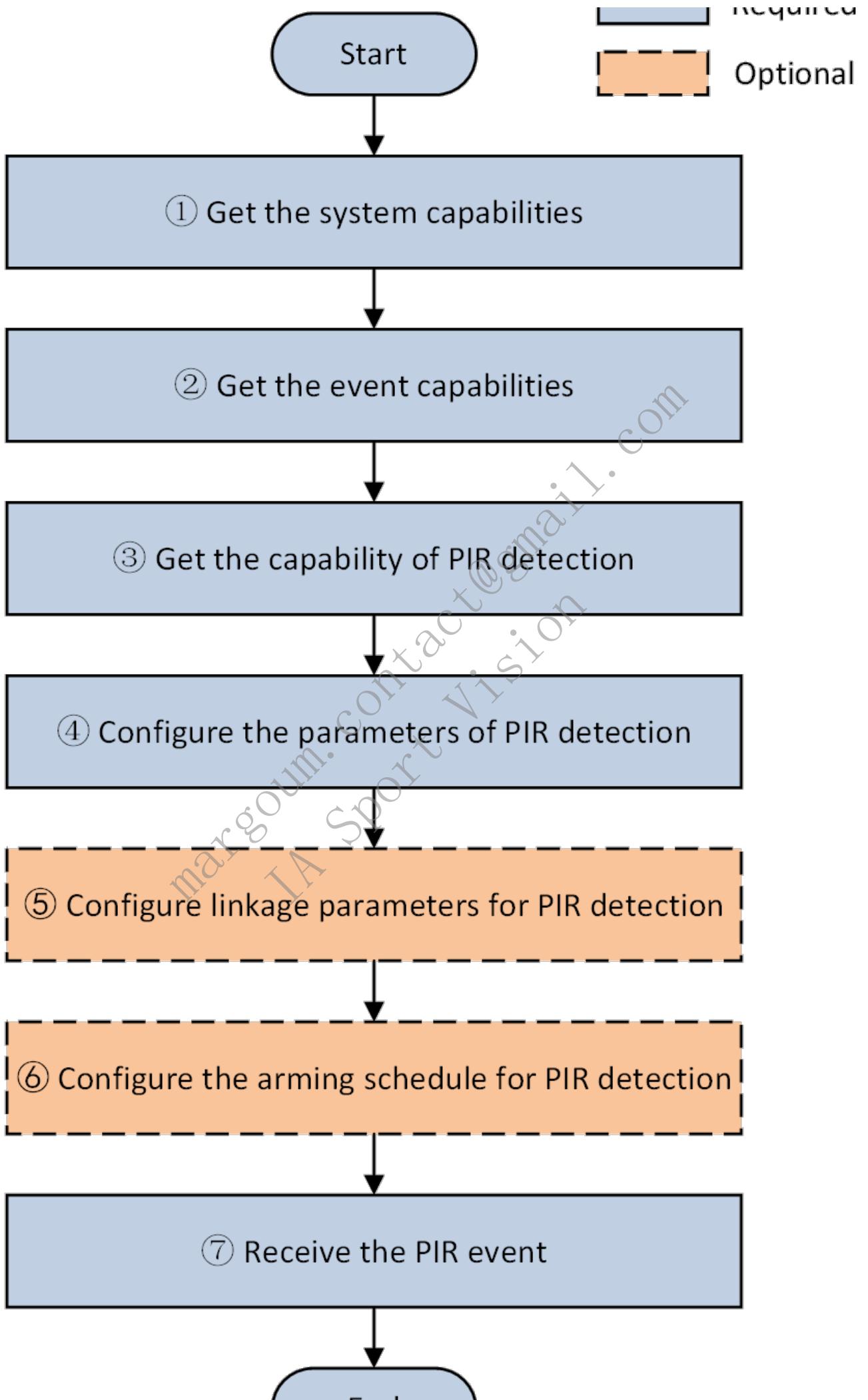
## **15 Human Body Detection and PIR Sensing**

### **15.1 PIR Detection**

#### **15.1.1 Introduction to the Function**

PIR work by looking for the changes in IR energy that occur with movement. Every person, animal, and object gives off varying amounts of IR energy. When a person, animal, or object comes within the specified area, the device will detect the change in IR energy via PIR and upload an alarm to the Client. Currently, it is mainly used to detect the movement of animals or persons in a specified area, like detecting animals movement in some wild areas, people's entering or exiting office buildings at night, etc.

#### **15.1.2 API Calling Flow**



1. Get the capability to check whether the device supports PIR detection: GET /ISAPI/System/capabilities. If the node `isSupportPIR` is returned, it indicates that the device supports PIR detection.
2. Get the event capability of a specified channel or all channels: /ISAPI/Event/channels/<channelID>/capabilities or GET /ISAPI/Event/channels/capabilities. If the returned `eventType` contains PIR, it indicates that the corresponding channel supports PIR detection.
3. Get the PIR detection parameters capability: GET /ISAPI/WLAlarm/PIR/capabilities.
4. Configure PIR detection parameters (including PIR detection enabling, detection sensitivity, whether to enable picture or video upload).

Get: GET /ISAPI/WLAlarm/PIR.

Configure: PUT /ISAPI/WLAlarm/PIR.

5. (Optional) Configure linkage parameters for PIR detection.

Get: GET /ISAPI/Event/triggers/PIR.

Configure: PUT /ISAPI/Event/triggers/PIR.

6. (Optional) Configure arming schedule of PIR detection.

Get: GET /ISAPI/Event/schedules/PIR.

Configure: PUT /ISAPI/Event/schedules/PIR.

7. Receive events reported by device in arming or listening mode: PIR. Event contents include the pictures captured when an event occurs and related UUID (unique ID of the event).

#### Remarks:

If it is configured to enable PIR detection video upload in step 4, the device will store the corresponding video if an event occurs. Due to the real-time of the alarm and the data transmission efficiency, the device will not upload the video to the Client along with the PIR event. Instead, it will store the complete video in the specified storage server first and then upload the corresponding video URL to the Client via `relationVideo` event. PIR event will be uploaded to the Client together with its UUID (unique ID of the event), which corresponds to the `linkUUID` of the `relationVideo` event. If the `UUID` and `linkUUID` share one value, it indicates that the corresponding event and video both belong to one alarm.

## 16 API Reference

### 16.1 Device (General)

#### 16.1.1 Device Exception Detection

##### 16.1.1.1 Get the linkage parameters of HDD error

###### Request URL

GET /ISAPI/Event/triggers/diskerror

###### Query Parameter

None

###### Request Message

None

###### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->diskerror
  </id>
  <eventType>
    <!--ro, req, enum, subType:string-->diskerror
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subType:object-->
    <EventTriggerNotification>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->test
      </id>
      <notificationMethod>
        <!--ro, req, enum, subType:string-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--ro, opt, enum, subType:string-->beginning
      </notificationRecurrence>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

### 16.1.1.2 Set the linkage parameters of HDD error

#### Request URL

PUT /ISAPI/Event/triggers/diskerror

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->diskerror
  </id>
  <eventType>
    <!--req, enum, subType:string-->diskerror
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
    <EventTriggerNotification>
      <!--opt, object-->
      <id>
        <!--req, string-->test
      </id>
      <notificationMethod>
        <!--req, enum, subType:string-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--opt, enum, subType:string-->beginning
      </notificationRecurrence>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.1.1.3 Delete the linkage of HDD error

#### Request URL

DELETE /ISAPI/Event/triggers/diskerror

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.1.1.4 Get the linkage parameters of full HDD

#### Request URL

GET /ISAPI/Event/triggers/diskfull

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->diskfull
  </id>
  <eventType>
    <!--ro, req, enum, subType:string-->diskfull
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->test
  </eventDescription>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subType:object-->
    <EventTriggerNotification>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->test
      </id>
      <notificationMethod>
        <!--ro, req, enum, subType:string-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--ro, opt, enum, subType:string-->beginning
      </notificationRecurrence>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

### 16.1.1.5 Set the linkage parameters of full HDD

#### Request URL

PUT /ISAPI/Event/triggers/diskfull

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->diskfull
  </id>
  <eventType>
    <!--req, enum, subType:string-->diskfull
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
    <EventTriggerNotification>
      <!--opt, object-->
      <id>
        <!--req, string-->test
      </id>
      <notificationMethod>
        <!--req, enum, subType:string-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--opt, enum, subType:string-->beginning
      </notificationRecurrence>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.1.1.6 Delete the linkage of full HDD

#### Request URL

DELETE /ISAPI/Event/triggers/diskfull

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.1.1.7 Delete the linkage of illegal login

#### Request URL

DELETE /ISAPI/Event/triggers/illaccess

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.1.1.8 Set the linkage parameters of illegal login

#### Request URL

PUT /ISAPI/Event/triggers/illaccess

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->illaccess
  </id>
  <eventType>
    <!--req, enum, subType:string-->illaccess
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
    <EventTriggerNotification>
      <!--opt, object-->
      <id>
        <!--req, string-->test
      </id>
      <notificationMethod>
        <!--req, enum, subType:string-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--opt, enum, subType:string-->beginning
      </notificationRecurrence>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.1.1.9 Get the linkage parameters of illegal login

#### Request URL

GET /ISAPI/Event/triggers/illaccess

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
    <id>
        <!--ro, req, string-->illaccess
    </id>
    <eventType>
        <!--ro, req, enum, subType:string-->illaccess
    </eventType>
    <eventDescription>
        <!--ro, opt, string-->test
    </eventDescription>
    <videoInputChannelID>
        <!--ro, opt, string-->1
    </videoInputChannelID>
    <dynVideoInputChannelID>
        <!--ro, opt, string-->1
    </dynVideoInputChannelID>
    <EventTriggerNotificationList>
        <!--ro, opt, array, subType:object-->
        <EventTriggerNotification>
            <!--ro, opt, object-->
            <id>
                <!--ro, req, string-->test
            </id>
            <notificationMethod>
                <!--ro, req, enum, subType:string-->FTP
            </notificationMethod>
            <notificationRecurrence>
                <!--ro, opt, enum, subType:string-->beginning
            </notificationRecurrence>
        </EventTriggerNotification>
    </EventTriggerNotificationList>
</EventTrigger>
```

### 16.1.1.10 Get the linkage parameters of IP address conflict event

#### Request URL

GET /ISAPI/Event/triggers/ipconflict

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->ipconflict
  </id>
  <eventType>
    <!--ro, req, enum, subType:string-->ipconflict
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subType:object-->
  </EventTriggerNotificationList>
</EventTrigger>

```

### 16.1.1.11 Delete the linkage of IP address conflict event

#### Request URL

DELETE /ISAPI/Event/triggers/ipconflict

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.1.1.12 Set the linkage parameters of IP address conflict event

#### Request URL

PUT /ISAPI/Event/triggers/ipconflict

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->ipconflict
  </id>
  <eventType>
    <!--req, enum, subType:string-->ipconflict
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
  </EventTriggerNotificationList>
</EventTrigger>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.1.1.13 Get the linkage parameters of network disconnection

#### Request URL

GET /ISAPI/Event/triggers/nicbroken

#### Query Parameter

None

#### Request Message

None

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->nicbroken
  </id>
  <eventType>
    <!--ro, req, enum, subType:string-->nicbroken
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subType:object-->
  </EventTriggerNotificationList>
</EventTrigger>

```

#### 16.1.1.14 Set the linkage parameters of network disconnection

##### Request URL

PUT /ISAPI/Event/triggers/nicbroken

##### Query Parameter

None

##### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->nicbroken
  </id>
  <eventType>
    <!--req, enum, subType:string-->nicbroken
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
  </EventTriggerNotificationList>
</EventTrigger>
```

##### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

#### 16.1.1.15 Delete the linkage of network disconnection

##### Request URL

DELETE /ISAPI/Event/triggers/nicbroken

##### Query Parameter

None

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.1.1.16 HDD error

#### EventType:diskerror

```

<?xml version="1.0" encoding="UTF-8"?>

<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{opt, string, protocolVersion}-->
  <ipAddress>
    <!--ro, req, string, IPv4 address of the device that triggers the alarm-->172.6.64.7
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPv6 address of the device that triggers the alarm-->1080:0:0:0:8:800:200C:417A
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, communication port No. of the device that triggers the alarm-->80
  </portNo>
  <protocol>
    <!--ro, opt, enum, transmission communication protocol type, subType:string, desc:when ISAPI protocol is transmitted via HCNetSDK, the channel No. is the video channel No. of private protocol. When ISAPI protocol is transmitted via EZ protocol, the channel No. is the video channel No. of EZ protocol. When ISAPI protocol is transmitted via ISUP, the channel No. is the video channel No. of ISUP-->HTTP
  </protocol>
  <macAddress>
    <!--ro, opt, string, MAC address-->01:17:24:45:D9:F4
  </macAddress>
  <dynChannelID>
    <!--ro, opt, string, digital channel No.-->test
  </dynChannelID>
  <channelID>
    <!--ro, opt, int, channel No. of the device that triggers the alarm, desc:video channel No. that triggers the alarm-->1
  </channelID>
  <dateTime>
    <!--ro, req, datetime, alarm trigger time-->2004-05-03T17:30:08+08:00
  </dateTime>
  <activePostCount>
    <!--ro, opt, int, times that the same alarm has been uploaded, desc:event triggering frequency-->1
  </activePostCount>
  <eventType>
    <!--ro, req, enum, event type, subType:string, desc:"diskerror" (HDD Error)-->diskerror
  </eventType>
  <eventState>
    <!--ro, req, enum, event status, subType:string, desc:for durative event: "active" (valid), "inactive" (invalid)-->active
  </eventState>
  <eventDescription>
    <!--ro, req, string, event description, desc:HDD Error-->diskerror
  </eventDescription>
  <channelName>
    <!--ro, opt, string, channel name, range:[1,64]-->test
  </channelName>
  <deviceID>
    <!--ro, opt, string, device ID, desc:it should be returned for ISUP alarms, e.g., test0123 (Ehome2.0, Ehome4.0, and ISUP5.0)-->12345
  </deviceID>
  <diskNo>
    <!--ro, opt, int, HDD number-->1
  </diskNo>
  <HDList>
    <!--ro, opt, array, HDD List, subType:object-->
    <HDD>
      <!--ro, opt, object, HDD-->
      <id>
        <!--ro, req, int, index, desc:it starts from 1-->1
      </id>
      <diskNumber>
        <!--ro, req, int, HDD No.-->1
      </diskNumber>
    </HDD>
  </HDList>
</EventNotificationAlert>

```

### 16.1.1.17 Invalid access

#### EventType:illaccess

```
<?xml version="1.0" encoding="UTF-8"?>

<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, event parameters, attr:version{opt, string, protocolVersion}-->
    <ipAddress>
        <!--ro, req, string, IPv4 address of the alarm device-->172.6.64.7
    </ipAddress>
    <ipv6Address>
        <!--ro, opt, string, IPv6 address of the alarm device-->1080:0:0:0:8:800:200C:417A
    </ipv6Address>
    <portNo>
        <!--ro, opt, int, port No. of the alarm device-->80
    </portNo>
    <protocol>
        <!--ro, opt, enum, transfer protocol type, subType:string, desc:transfer protocol type: "HTTP", "HTTPS", "EHome". The value should be "HTTP" when ISAPI protocol is transmitted via EZVIZ protocol. The value should be "EHome" when ISAPI protocol is transmitted via ISUP-->HTTP
    </protocol>
    <macAddress>
        <!--ro, opt, string, MAC address-->01:17:24:45:D9:F4
    </macAddress>
    <dynChannelID>
        <!--ro, opt, string, digital channel No.-->test
    </dynChannelID>
    <channelID>
        <!--ro, opt, int, No. of the channel that triggers an alarm, desc:No. of the video channel that triggers an alarm-->1
    </channelID>
    <dateTime>
        <!--ro, req, datetime, alarm time-->2004-05-03T17:30:08+08:00
    </dateTime>
    <activePostCount>
        <!--ro, opt, int, the uploading times of the same alarm, desc:event frequency-->1
    </activePostCount>
    <eventType>
        <!--ro, req, enum, event type, subType:string, desc:"illaccess" (invalid access)-->illaccess
    </eventType>
    <eventState>
        <!--ro, req, enum, event status, subType:string, desc:for persistent events; "active" (valid), "inactive" (invalid)-->active
    </eventState>
    <eventDescription>
        <!--ro, req, string, event description, desc:illaccess-invalid access-->illaccess
    </eventDescription>
    <channelName>
        <!--ro, opt, string, channel name, range:[1,64]-->test
    </channelName>
    <deviceID>
        <!--ro, opt, string, device ID, desc:it should be returned for ISUP alarms, e.g., test0123 (Ehome2.0, Ehome4.0, and ISUP5.0)-->12345
    </deviceID>
    <username>
        <!--ro, opt, string, Locked user name, range:[0,32], desc:the number of attempts by current user name has reached Limit. The user name and IPv4 address may be locked at the same time-->test
    </username>
    <ipV4Address>
        <!--ro, opt, string, Locked IPv4 address, desc:the number of attempts by current IPv4 address has reached Limit. The user name and Ipv4 address may be locked at the same time-->test
    </ipV4Address>
    <lockStatus>
        <!--ro, opt, enum, Locking status, subType:string, desc:; "unlocked", "locked"-->unlocked
    </lockStatus>
    <retryLoginTime>
        <!--ro, opt, int, remaining attempts, range:[0,65535], unit:分钟, desc:it is valid when the value of lockStatus is "unlocked"-->0
    </retryLoginTime>
    <lockedTime>
        <!--ro, opt, int, Locking duration (minutes), range:[0,65535], unit:分钟, desc:it is valid when the value of lockStatus is "locked"-->0
    </lockedTime>
    <accessInfoList>
        <!--ro, opt, array, invalid access information list, subType:object-->
        <accessInfo>
            <!--ro, opt, object, invalid access information-->
            <ipV4Address>
                <!--ro, opt, string, Invalid access IP address-->test
            </ipV4Address>
            <dateTime>
                <!--ro, opt, datetime, invalid access time-->2004-05-03T17:30:08+08:00
            </dateTime>
        </accessInfo>
    </accessInfoList>
</EventNotificationAlert>
```

### 16.1.1.18 Network disconnected

#### EventType:nicbroken

```

<?xml version="1.0" encoding="UTF-8"?>

<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, event information, attr:version{opt, string, protocolVersion}-->
  <ipAddress>
    <!--ro, req, string, IPv4 address of the device that triggers an alarm-->172.6.64.7
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPv6 address of the device that triggers an alarm-->1080:0:0:0:8:800:200C:417A
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, port No. of the device that triggers an alarm-->80
  </portNo>
  <protocol>
    <!--ro, opt, enum, communication protocol types, subType:string, desc:transmission communication protocol type: "HTTP", "HTTPS", "EHome" (the value should be "HTTP" when ISAPI protocol is transmitted via EZ protocol, and the value should be "EHome" when ISAPI protocol is transmitted via ISUP)-->HTTP
  </protocol>
  <macAddress>
    <!--ro, opt, string, MAC address-->01:17:24:45:D9:F4
  </macAddress>
  <dynChannelID>
    <!--ro, opt, string, digital channel No.-->test
  </dynChannelID>
  <channelID>
    <!--ro, opt, int, channel No. of the device that triggers an alarm, desc:video channel No. that triggers the alarm-->1
  </channelID>
  <dateTime>
    <!--ro, req, datetime, alarm trigger time-->2004-05-03T17:30:08+08:00
  </dateTime>
  <activePostCount>
    <!--ro, opt, int, the uploading times of the same alarm, desc:event triggering frequency-->1
  </activePostCount>
  <eventType>
    <!--ro, req, string, event type, desc:"nicbroken" (network disconnected)-->nicbroken
  </eventType>
  <eventState>
    <!--ro, req, enum, event status, subType:string, desc:for durative event: "active" (valid), "inactive" (invalid)-->active
  </eventState>
  <eventDescription>
    <!--ro, req, string, event description, desc:"nicbroken" (network disconnected)-->nicbroken
  </eventDescription>
  <channelName>
    <!--ro, opt, string, channel name, range:[1,64]-->test
  </channelName>
  <deviceID>
    <!--ro, opt, string, device ID, desc:it should be returned for ISUP alarms, e.g., test0123 (Ehome2.0, Ehome4.0, and ISUP5.0)-->12345
  </deviceID>
  <DevicePositionInfo>
    <!--ro, opt, object, device location, desc:the current location of the device when triggering the event, which is mainly for on-board device-->
    <divisionEW>
      <!--ro, opt, enum, eastern/western hemisphere, subType:string, desc:"E" (eastern hemisphere), "W" (western hemisphere)-->E
    </divisionEW>
    <longitude>
      <!--ro, opt, int, longitude, desc:degree*3600*100 + minute*60*100 + second*100-->100
    </longitude>
    <divisionNS>
      <!--ro, opt, enum, southern/northern hemisphere, subType:string, desc:"S" (southern hemisphere), "N" (northern hemisphere)-->N
    </divisionNS>
    <latitude>
      <!--ro, opt, int, latitude, desc:degree*3600*100 + minute*60*100 + second*100-->100
    </latitude>
    <direction>
      <!--ro, opt, int, vehicle direction, desc:actual direction (degree) (due north as 0 degree, calculated clockwise)-->100
    </direction>
    <speed>
      <!--ro, opt, int, speed, desc:cm/h-->100
    </speed>
    <height>
      <!--ro, opt, int, height, unit:cm, desc:cm-->1
    </height>
  </DevicePositionInfo>
</EventNotificationAlert>

```

## 16.1.2 Event Subscription

### 16.1.2.1 Get the alarm/event subscription capability

#### Request URL

GET /ISAPI/Event/notification/subscribeEventCap

#### Query Parameter

None

#### Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<SubscribeEventCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, picture uploading modes of all events which contain pictures, attr:version{req, string, protocolVersion}-->
    <format opt="xml,json">
        <!--ro, opt, string, attr:opt{req, string}-->xml
    </format>
    <heartbeat min="1" max="180">
        <!--ro, opt, int, heartbeat interval time, range:[1,180], unit:s, attr:min{req, int},max{req, int}-->1
    </heartbeat>
    <channelMode opt="all,list">
        <!--ro, opt, enum, channel subscription mode, subType:string, attr:opt{req, string}, desc:"all" (subscribe to all channels), "List" (subscribe to channels according to channel list)-->list
    </channelMode>
    <eventMode opt="all,list">
        <!--ro, opt, enum, event subscription mode, subType:string, attr:opt{req, string}, desc:"all" (subscribe to all alarms/events), "List" (subscribe to specified alarms/events)-->list
    </eventMode>
    <EventList>
        <!--ro, opt, array, event type List for subscription, subType:object, desc:this node is valid when eventMode is "List"-->
        <Event>
            <!--ro, opt, object, subscription of a specified alarm/event-->
            <type>
                <!--ro, req, enum, event type, subType:string, desc:refer to event type List (eventType): "ADAS"(advanced driving assistance system), "ADASAlarm" (advanced driving assistance alarm), "AID"(traffic incident detection), "ANPR"(automatic number plate recognition), "AccessControllerEvent" (access controller event), "CDStatus" (CD burning status), "DBD"(driving behavior detection) "GPSUpload" (GPS information upload), "HFPD"(frequently appeared person detection), "IO"(I/O alarm), "IOTD" (IoT device detection), "LES" (Logistics scanning event), "LFD"(rarely appeared person detection), "PALMismatch" (video standard mismatch), "PIR", "PeopleCounting" (people counting), "PeopleNumChange" (people number change detection), "Standup"(standing up detection), "TMA"(thermometry alarm), "TMPA"(temperature measurement pre-alarm), "VMD"(motion detection), "abnormalAcceleration", "abnormalDriving", "advReachHeight", "alarmResult", "attendance", "attendedBaggage", "audioAbnormal", "audioException", "behaviorResult"(abnormal event detection), "blindSpotDetection"(blind spot detection alarm), "cardMatch", "changedStatus", "collision", "containerDetection", "crowdSituationAnalysis", "databaseException", "defocus"(defocus detection), "diskInformat"(disk unformatted), "diskError", "diskfull", "driverConditionMonitor"(driver status monitoring alarm); "emergencyAlarm", "faceCapture", "faceSnapModeling", "facedetection", "failDown"(People Falling Down), "faultAlarm", "fielddetection"(intrusion detection), "fireDetection", "fireEscapeDetection", "flowOverrun", "framesPeopleCounting", "getUp"(getting up detection), "group" (people gathering), "hdBadBlock"(HDD bad sector detection event), "hdImpact"(HDD impact detection event), "heatmap"(heat map alarm), "highHTTemperature"(HDD high temperature detection event), "highTempAlarm"(HDD high temperature alarm), "hotSpore"(hot spare exception), "illAccess"(invalid access), "ipcTransferAbnormal", "ipConflict"(IP address conflicts), "keyPersonGetUp"(key person getting up detection), "LeavePosition"(absence detection), "linedetection"(line crossing detection), "listSyncException"(list synchronization exception), "Loitering"(loitering detection), "lowHTTemperature"(HDD low temperature detection event), "mixedTargetDetection"(multi-target-type detection), "modelError", "nicbroken"(network disconnected), "nodeOffline"(node disconnected), "nonPoliceIntrusion", "overSpeed"(overspeed alarm), "overtimeTarry"(staying overtime detection), "parking"(parking detection), "peopleNumChange", "peopleNumCounting", "personAbnormalAlarm"(person ID exception alarm), "personDensityDetection", "personQueueCounting", "personQueueDetection", "personQueueRealTime"(real-time data of people queuing-up detection), "personQueueTime"(waiting time detection), "playCellphone"(playing mobile phone detection), "pocException"(video exception), "poe"(POE power exception), "policeAbsent", "radarAlarm", "radarFieldDetection", "radarLineDetection", "radarPerimeterRule"(radar rule data), "radarTargetDetection", "radarVideoDetection"(radar-assisted target detection), "raidException", "rapidMove", "reachHeight"(climbing detection), "recordCycleAbnormal"(insufficient recording period), "recordException", "regionEntrance", "regionExiting", "retention"(people overstay detection), "rollOver", "running"(people running), "safetyHelmetDetection"(hard hat detection), "scenecangedetection", "sensorAlarm"(angular acceleration alarm), "severeHDFailure"(HDD major fault detection), "shelteralarm"(video tampering alarm), "shipsDetection", "sitQuietly"(sitting detection), "smokeAndFireDetection", "smokeDetection", "softIO", "spacingChange"(distance exception), "sysStorFull"(storing full alarm of cluster system), "takingElevatorDetection"(elevator electric moped detection), "targetCapture", "temperature"(temperature difference alarm), "thermometry"(temperature alarm), "thirdPartyException", "toiletTarry"(in-toilet overtime detection), "tollCodeInfo"(QR code information report), "tossing"(thrown object detection), "unattendedBaggage", "vehicleMatchResult"(uploading list alarms), "vehicleRecognizeResult", "versionAbnormal"(cluster version exception), "videoException", "videoloss", "violationAlarm", "violentMotion"(violent motion detection), "yardTarry"(playground overstay detection), "AccessControllerEvent", "IDCardInfoEvent", "FaceTemperatureMeasurementEvent", "QRCodeEvent"(QR code event of access control), "CertificateCaptureEvent"(person ID capture comparison event), "UncertificateCompareEvent", "ConsumptionAndTransactionRecordEvent", "ConsumptionEvent", "TFS" (traffic enforcement event), "TransactionRecordEvent", "HealthInfoSyncQuery" (health information search event), "SetMealQuery"(searching consumption set meals), "ConsumptionStatusQuery"(searching the consumption status), "certificateRevocation" (certificate expiry), "humanBodyComparison" (human body comparison), "regionTargetNumberCounting" (regional target statistics)-->mixedTargetDetection
            </type>
            <channels>
                <!--ro, opt, string, channel information linked to event, desc:it supports multiple channels, which are separated by commas-->1,2,3,4
            </channels>
        </Event>
    </EventList>
    <isSupportUnSubscribeEvent>
        <!--ro, opt, bool-->true
    </isSupportUnSubscribeEvent>
</SubscribeEventCap>
```

## 16.1.3 HEOP

### 16.1.3.1 Get the application program information list

#### Request URL

GET /ISAPI/Custom/OpenPlatform/App

#### Query Parameter

None

#### Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<AppList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, application program information list, subType:object, attr:version{req, string, protocolVersion}-->
  <App>
    <!--ro, opt, object, application program information-->
    <id>
      <!--ro, req, int, index (it is not AppID), range:[0,128]-->1
    </id>
    <packageName>
      <!--ro, req, string, software package name to be displayed, range:[0,128]-->test
    </packageName>
    <vendor>
      <!--ro, req, string, manufacturer name of the third-party application program, range:[0,128]-->test
    </vendor>
    <softVersion>
      <!--ro, req, string, software version number, range:[0,128]-->test
    </softVersion>
    <runStatus>
      <!--ro, req, enum, program running status, subType:string, desc:"true" (running), "false" (stopped)-->true
    </runStatus>
    <memory>
      <!--ro, opt, int, memory size occupied by the application program, range:[0,1024], step:1, unit:MB-->1
    </memory>
    <flash>
      <!--ro, opt, int, flash size occupied by the application program, range:[0,1024], step:1, unit:MB-->1
    </flash>
    <bStartedOnce>
      <!--ro, opt, bool, whether the application program has been started, desc:"true" (yes), "false" (no)-->true
    </bStartedOnce>
    <appPackageUUID>
      <!--ro, opt, string, the UUID of the application program package, range:[0,128]-->test
    </appPackageUUID>
    <caculation>
      <!--ro, opt, float, computing power used by the application program. It should be accurate to one decimal place, desc:unit: TOPS-->0.000
    </caculation>
    <intelligentMemory>
      <!--ro, opt, int, intelligent memory size occupied by the application program, unit:Byte-->0
    </intelligentMemory>
  </App>
</AppList>
```

### 16.1.3.2 Get Information of an Application Program

#### Request URL

GET /ISAPI/Custom/OpenPlatform/App/<AppID>

#### Query Parameter

Parameter Name	Parameter Type	Description
AppID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<App xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, application program information list, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, int, application program ID-->
  </id>
  <packageName>
    <!--ro, req, string, packet name for display-->test
  </packageName>
  <vendor>
    <!--ro, req, string, vendor name of third-party application program-->test
  </vendor>
  <softVersion>
    <!--ro, req, string, software version number-->test
  </softVersion>
  <runStatus>
    <!--ro, req, enum, program running status, subType:string, desc:"true" (running), "false" (stopped)-->true
  </runStatus>
  <memory>
    <!--ro, opt, int, memory size occupied by the application program, unit:MB-->1
  </memory>
  <flash>
    <!--ro, opt, int, flash size occupied by the application program, unit:MB-->1
  </flash>
  <bStartedOnce>
    <!--ro, opt, bool, whether the application program has been started:, desc:"true" (yes), "false" (no)-->true
  </bStartedOnce>
  <appPackageUUID>
    <!--ro, opt, string, the UUID of the application program package, range:[0,128]-->test
  </appPackageUUID>
  <calculation>
    <!--ro, float, computing power size occupied by the application program, desc:The number is accurate to one decimal place (e.g., 0.1)-->0.1
  </calculation>
  <intelligentMemory>
    <!--ro, opt, int, intelligent memory size occupied by the application program-->0
  </intelligentMemory>
</App>

```

### 16.1.3.3 Delete Specified Application Program

#### Request URL

DELETE /ISAPI/Custom/OpenPlatform/App/<AppID>/delete?userData=<userData>

#### Query Parameter

Parameter Name	Parameter Type	Description
AppID	string	--
userData	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL, range:[0,1024]-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, describe the error reason in detail, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
  <description>
    <!--ro, opt, string, range:[0,1024]-->badXmlFormat
  </description>
</ResponseStatus>

```

### 16.1.3.4 Set permission of the third-party application program

## Request URL

PUT /ISAPI/Custom/OpenPlatform/App/<AppID>/permission?format=json&engineID=<engineID>

## Query Parameter

Parameter Name	Parameter Type	Description
AppID	string	--
engineID	string	--

## Request Message

```
{
  "PermissionList": {
    /*req, object, list of permissions*/
    "Stream": {
      /*req, object*/
      "enabled": true
      /*req, bool*/
    },
    "configAuth": {
      /*req, object, configure the permission for free authentication*/
      "enabled": true
      /*req, bool, enable the permission to get streams*/
    },
  },
}
```

## Response Message

```
{
  "statusCode": 1,
  /*ro, opt, int, status code, desc:1 (succeeded). It is required when an error occurred*/
  "statusString": "ok",
  /*ro, opt, string, status description, range:[1,64], desc:"ok" (succeeded). It is required when an error occurred*/
  "subStatusCode": "ok",
  /*ro, opt, string, sub status code, range:[1,64], desc:"ok" (succeeded). It is required when an error occurred*/
  "errorCode": 1,
  /*ro, opt, int, error code, desc:when the value of statusCode is not 1, it corresponds to subStatusCode*/
  "errorMsg": "ok"
  /*ro, opt, string, error information, desc:this field is required when the value of statusCode is not 1*/
}
```

## 16.1.3.5 Get permissions of the third-party application program

### Request URL

GET /ISAPI/Custom/OpenPlatform/App/<AppID>/permission?format=json&engineID=<engineID>

### Query Parameter

Parameter Name	Parameter Type	Description
AppID	string	--
engineID	string	--

### Request Message

None

### Response Message

```
{
    "PermissionList": {
        /*ro, req, object, permission information*/
        "Stream": {
            /*ro, req, object, streaming permission*/
            "enabled": true
            /*ro, req, bool, enable the permission to get streams*/
        },
        "configAuth": {
            /*ro, req, object, configure the permission for free authentication*/
            "enabled": true
            /*ro, req, bool, enable the permission for free authentication*/
        }
    }
}
```

### 16.1.3.6 Get the HEOP capability supported by device

#### Request URL

GET /ISAPI/Custom/OpenPlatform/capabilities

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenPlatformCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, HEOP capability set supported by device, attr:version{req, string, protocolVersion}-->
    <isSupportOpenPlatform>
        <!--ro, req, bool, whether the device supports HEOP (Hikvision Embedded Open Platform)-->true
    </isSupportOpenPlatform>
</OpenPlatformCap>
```

### 16.1.3.7 Get Application Program Page Information

#### Request URL

GET /ISAPI/Custom/OpenPlatform/pageitem

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<pageItemList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, array, page information list of application program, subType:object, attr:version{req, string, protocolVersion}-->
    <pageItem>
        <!--ro, opt, object, page information of application program-->
        <id>
            <!--ro, req, int, page No. of application program, range:[0,128], step:1-->1
        </id>
        <appName>
            <!--ro, req, string, the third-party application program name, range:[0,128]-->test
        </appName>
        <htmlPath>
            <!--ro, opt, string, html page path of the third-party application program, range:[0,128]-->test
        </htmlPath>
    </pageItem>
</pageItemList>
```

### 16.1.3.8 Get HEOP Remaining Resources

#### Request URL

GET /ISAPI/Custom/OpenPlatform/resource

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<resource xmlns="http://www.isapi.org/ver20/XMLSchema" version="1.0">
  <!--ro, req, object, remaining resource information, attr:version{req, string, protocolVersion}-->
  <totalMemory>
    <!--ro, opt, int, the size of all available memory for open platform, unit:MB-->0
  </totalMemory>
  <restMemory>
    <!--ro, opt, int, the size of remaining available memory for open platform, unit:MB-->0
  </restMemory>
  <totalFlash>
    <!--ro, opt, int, the size of all available memory for open platform, unit:MB-->0
  </totalFlash>
  <restFlash>
    <!--ro, opt, int, the size of remaining available flash for open platform, unit:MB-->0
  </restFlash>
  <totalIntelligentMemory>
    <!--ro, opt, int, the size of all available intelligent memory for open platform, unit:MB, desc:The physical address of intelligent memory is continuous, which is used to speed up processing.-->0
  </totalIntelligentMemory>
  <restIntelligentMemory>
    <!--ro, opt, int, the size of remaining available intelligent memory for open platform, unit:MB-->0
  </restIntelligentMemory>
</resource>
```

### 16.1.3.9 Delete the third-party server information

#### Request URL

DELETE /ISAPI/OrgPreview/Channels/<channelID>/Server

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (Device Busy), 2 (Device Error), 3 (Device Operation), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.1.3.10 Set address and port parameters of third-party server

#### Request URL

PUT /ISAPI/OrgPreview/Channels/<channelID>/Server

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Server xmlns="http://www.isapi.org/ver20/XMLSchema" version="1.0">
  <!--req, object, the address and port parameters of third-party server, attr:version{req, string, protocolVersion}-->
  <addressingFormatType>
    <!--req, enum, address type, subType:string, desc:ipaddress; hostname;-->ipaddress
  </addressingFormatType>
  <ipAddress>
    <!--opt, string, IPv4 address, range:[1,32]-->192.168.1.112
  </ipAddress>
  <portNo>
    <!--opt, int, port No., range:[1,65535], desc:port No.-->74
  </portNo>
</Server>
```

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.1.3.11 Get the address and port No. of the third-party server

#### Request URL

GET /ISAPI/OrgPreview/Channels/<channelID>/Server

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Server xmlns="http://www.isapi.org/ver20/XMLSchema" version="1.0">
  <!--req, object, the address and port number of the third-party server, attr:version{req, string, protocolVersion}-->
  <addressingFormatType>
    <!--ro, req, enum, address type, subType:string, desc:"ipaddress", "hostname"-->ipaddress
  </addressingFormatType>
  <ipAddress>
    <!--ro, opt, string, IPv4 address, range:[1,32]-->192.168.1.112
  </ipAddress>
  <portNo>
    <!--ro, opt, int, Port No., range:[1,65535], desc:Port No.-->74
  </portNo>
</Server>
```

### 16.1.4 Low Power Consumption

#### 16.1.4.1 Get the linkage triggering parameters of scheduled wakeup event

##### Request URL

GET /ISAPI/Event/triggers/arouse-<channelID>/notifications

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

None

##### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotification xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
</EventTriggerNotification>
```

#### 16.1.4.2 Set the linkage triggering parameters of scheduled wakeup event

##### Request URL

PUT /ISAPI/Event/triggers/arouse-<channelID>/notifications

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotification xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
</EventTriggerNotification>
```

##### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.1.5 Port Settings

#### 16.1.5.1 Get the serial port capability of the device

##### Request URL

GET /ISAPI/System/Serial/capabilities

##### Query Parameter

None

##### Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<SerialCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, range of RS-485 serial port numbers supported by the device, attr:version{req, string, protocolVersion}-->
  <rs485PortNums>
    <!--ro, opt, int, the maximum number of RS-485 serial ports supported by the device-->0
  </rs485PortNums>
  <supportRS232Config>
    <!--ro, opt, bool, whether the device supports configuring parameters of RS-232 serial ports-->true
  </supportRS232Config>
  <rs422PortNums>
    <!--ro, opt, int, the maximum number of RS-422 serial ports supported by the device-->0
  </rs422PortNums>
  <rs232PortNums>
    <!--ro, opt, int, the maximum number of RS-232 serial ports supported by the device-->0
  </rs232PortNums>
</SerialCap>
```

### 16.1.5.2 Get the list of serial ports supported by the device

#### Request URL

GET /ISAPI/System/Serial/ports?permissionController=<indexID>

#### Query Parameter

Parameter Name	Parameter Type	Description
indexID	string	--

#### Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<SerialPortList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, see details in the message of XML_SerialPort, attr:version{req, string, protocolVersion}-->
  <SerialPort>
    <!--ro, opt, object, port-->
    <id>
      <!--ro, req, int-->0
    </id>
    <enabled>
      <!--ro, opt, bool, whether to enable the function-->true
    </enabled>
    <serialPortType>
      <!--ro, opt, enum, subType:string-->RS485
    </serialPortType>
    <duplexMode>
      <!--ro, opt, enum, subType:string-->half
    </duplexMode>
    <baudRate>
      <!--ro, opt, enum, subType:int-->2400
    </baudRate>
    <dataBits>
      <!--ro, opt, int-->6
    </dataBits>
    <parityType>
      <!--ro, opt, enum, subType:string-->none
    </parityType>
    <stopBits>
      <!--ro, opt, enum, subType:string-->1
    </stopBits>
    <workMode>
      <!--ro, opt, enum, work mode, subType:string, desc:work mode-->console
    </workMode>
    <flowCtrl>
      <!--ro, opt, enum, subType:string-->none
    </flowCtrl>
  </SerialPort>
</SerialPortList>
```

### 16.1.6 System Maintenance

#### 16.1.6.1 Get the storage capability of the device

## Request URL

GET /ISAPI/ContentMgmt/capabilities

## Query Parameter

None

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<RacmCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, device storage capability, attr:version{req, string, protocolVersion}-->
  <nasNums>
    <!--ro, opt, int, supported number of NAS devices, desc:related URI: /ISAPI/ContentMgmt/Storage/nas-->0
  </nasNums>
  <pictureSearchType>
    opt="AllEvent,CMR,MOTION,ALARM,EDR,ALARMANDMOTION,Command,pir,wlsensor,callhelp,facedetection,FieldDetection,scenedetection,LineDetection,
regionEntrance,regionExiting,loitering,group,rapidMove,parking,unattendedBaggage,attendedBaggage,vehicleDetection,manual,manualSnapShot,playSnapShot,safetyH
elmet,
personQueueCounting,personQueueTime,violentMotion,leavePosition,advReachHeight,peopleNumChange,mixedTargetDetection,illegalParking,pedestrian,construction,r
oadblock,
abandonedObject,trafficAccident,fogDetection,smokeDetection,personDensity,failDown,accessController,videoInterCom,GJD,Luminite,OPTEX,securityControlPanel,ge
tUp,toiletTarry,
playCellphone,vehicleMonitor,vibrationDetection,running,retention,spacingChange,yardTarry,keyPersonGetUp,sitQuietly,standUp,plateRecognition,smoking,onThePh
one,fatigueDriving,
notLookingStraightAhead,leaveThePost,seatBelt,sunglasses,identityAbnormal,ForwardCollisionWarning,LaneDepartureWarning,BlindSpotDetection,HeadwayMonitoringW
arning,
PedestrianCollisionWarning,SuddenSpeedUp,SuddenSpeedDown,SuddenLeftTurn,SuddenRightTurn,Turnover,NotYieldToPedestrian,unregisteredStreetVendor,stallOutsideS
hop,stallOccupyingRoad, illegalHeap,illegalParkingofNon-
MotorVehicle,illegalOutdoorAdvertisement,packGarbage,stallUnderUmbrella,dustbinOverflow,exposeGarbage,hangClothingAlongStreet,allPic,ATMPanel,
ATMSurround,ATMSafetyCabin,loitering,smokeDetectAlarm,temperatureIntervalMeasurement,AIOP,tail,abnormalFace,normalFace,multiFace,callFace,sunglasses
Face,humanEnter,operateOverTime,stickUp,installScanner,objectPlacedOrTaken,fakeCard,securityChecking,pothole,crack,roadMaint,imageCapture,methaneConcentrati
onException,methaneLightIntensityException,bannerDetection,variableDataEvent,shipChannelAbnormal,radarVideoDetection,smokeAndFireDetection,smokeDetection,se
nsorAlarm,alarmResult,AIOP_Video,AIOP_Polling_Snap,swipePatrol,BatteryCarInfoEvent,indoorPasswayBlock,detectorTemperatureAlarm,fireNoRegulation,detectorMoti
onAlarm,detectorShelterAlarm,detectorRunningStatus,faceCapture,UVPhotonDetection,DevPowerStatusReport,HFPD, audioAnalysis, IndoorPasswayRecovery, BatteryCarRec
overy,DetectorTemperatureRecovery,RadarFallenObjectsOnTracks,VegetationSpectrumDetection,personDensityDetection,channelClearEvent,channelOccupied,channelPassi
ngEvent,FallenObjectsOnTracksEvent,audioexception,CarWindow"
    <!--ro, opt, enum, picture search type, subType:string, attr:opt{req, string}, desc:"AllEvent", "CMR", "MOTION", "ALARM", "EDR", "ALARMANDMOTION",
"pir:PIR", "wlsensor", "callhelp", "facedetection", "FieldDetection", "scenedetection", "LineDetection", "regionEntrance", "regionExiting",
"Loitering", "group", "rapidMove", "parking", "unattendedBaggage", "attendedBaggage", "vehicleDetection", "manual", "manualSnapShot", "playSnapShot",
"safetyHelmet", "personQueueCounting", "personQueueTime", "violentMotion", "LeavePosition", "advReachHeight", "peopleNumChange", "mixedTargetDetection",
"illegalParking", "pedestrian", "construction", "roadblock", "evidence", "crosslane", "vehicLeexist", "Lanechange", "wrongdirection", "congestion",
"turnround", "parallelParking", "abandonedObject", "trafficAccident", "fogDetection", "smokeDetection", "personDensity", "failDown", "accessController",
"videoInterCom", "GJD", "Luminite", "OPTEX", "securityControlPanel", "getUp", "toiletTarry", "playCellphone", "vehicleMonitor", "running", "retention",
"spacingChange", "yardTarry", "keyPersonGetUp", "sitQuietly", "standUp", "plateRecognition", "smoking", "onThePhone", "fatigueDriving",
"notLookingStraightAhead", "LeaveThePost", "seatbelt", "sunglasses", "identityAbnormal", "ForwardCollisionWarning", "LaneDepartureWarning",
"BlindspotDetection", "HeadwayMonitoringWarning", "PedestrianCollisionWarning", "SuddenSpeedUp", "SuddenSpeedDown", "SuddenLeftTurn", "SuddenRightTurn",
"Turnover", "NotYieldToPedestrian;allPic", "ATMPanel", "ATMSurround", "ATMSafetyCabin", "smokeDetectAlarm", "temperatureIntervalMeasurement",
"AIOP", "tail", "abnormalFace", "normalFace;callFace", "sunglassesFace", "humanEnter", "operateOverTime", "stickup", "installScanner",
"objectPlacedOrTaken", "fakeCard", "securityChecking", "pothole", "crack", "roadMaint", "imageCapture", "ReID", "thermalVehicleDetection",
"methaneConcentrationException", "methaneLightIntensityException", "bannerDetection", "waterQualityDetection", "waterLevelDetection", "parallelParking",
"methaneConcentrationException", "variableDataEvent", "VOCsGasDetection", "shipChannelAbnormal", "radarVideoDetection", "AIDfogDetection:AID",
"smokeAndFireDetection", "smokeDetection", "sensorAlarm", "GasTankDetection", "alarmResult", "AIOP_Video", "AIOP_Polling_Snap", "swipePatrol",
"BatteryCarInfoEvent", "indoorPasswayBlock", "detectorTemperatureAlarm", "fireNoRegulation", "detectorMotionAlarm", "detectorShelterAlarm",
"detectorRunningStatus", "faceCapture", "UVPhotonDetection", "DevPowerStatusReport", "HFPD"-->AllEvent
  </pictureSearchType>
  <recordSearchType>
    opt="AllEvent,CMR,MOTION,ALARM,EDR,ALARMANDMOTION,Command,pir,wlsensor,callhelp,facedetection,FieldDetection,scenedetection,LineDetection,
regionEntrance,regionExiting,loitering,group,rapidMove,parking,unattendedBaggage,attendedBaggage,vehicleDetection,manual,manualSnapShot,playSnapShot,safetyH
elmet,
failDown,personDensityDetection,smokeDetection,vibrationDetection,unregisteredStreetVendor,stallOutsideShop,stallOccupyingRoad,illegalHeap,illegalParkingofN
on-MotorVehicle,
illegalOutdoorAdvertisement,packGarbage,stallUnderUmbrella,dustbinOverflow,exposeGarbage,hangClothingAlongStreet,ATMPanel,ATMSurround,ATMSafetyCabin
,temperatureIntervalMeasurement,AIOP,securityChecking,scheduledRecordUpload,variableDataEvent,smokeAndFireDetection,smokeDetection,AIOP_Video,AIOP_Polling_S
nap,BatteryCarInfoEvent,indoorPasswayBlock,detectorTemperatureAlarm,fireNoRegulation,detectorMotionAlarm,detectorShelterAlarm,detectorRunningStatus,faceCapt
ure,UVPhotonDetection,DevPowerStatusReport,ChannelClearEvent,channelOccupied,channelPassingEvent"
    <!--ro, opt, enum, video search type, subType:string, attr:opt{req, string}, desc:"AllEvent", "CMR", "MOTION", "ALARM", "EDR", "ALARMANDMOTION", "pir",
"wlsensor", "callhelp", "facedetection", "FieldDetection", "scenedetection", "LineDetection", "regionEntrance", "regionExiting", "Loitering", "group",
"rapidMove", "parking", "unattendedBaggage", "attendedBaggage", "vehicleDetection", "manual", "manualSnapShot", "playSnapShot", "safetyHelmet",
"personDensityDetection", "smokeDetection", "ATMPanel", "ATMSurround", "ATMSafetyCabin", "temperatureIntervalMeasurement", "AIOP",
"securityChecking", "scheduledRecordUpload", "thermalVehicleDetection", "waterQualityDetection", "waterLevelDetection", "methaneConcentrationException",
"variableDataEvent", "smokeAndFireDetection", "smokeDetection", "sensorAlarm", "parallelParking", "parallelParkingContinuousRecord", "GasTankDetection",
"AIOP_Video", "AIOP_Polling_Snap", "BatteryCarInfoEvent", "indoorPasswayBlock", "detectorTemperatureAlarm", "fireNoRegulation", "detectorMotionAlarm",
"detectorShelterAlarm", "detectorRunningStatus", "faceCapture", "UVPhotonDetection", "DevPowerStatusReport"-->AllEvent
  </recordSearchType>
  <isSupportRacmChannelsCap opt="true,false">
    <!--ro, opt, bool, whether the device supports the Racm capability by channel, attr:opt{opt, string}, desc:related URI:
/ISAPI/ContentMgmt/channels/<ID>/capabilities-->true
  </isSupportRacmChannelsCap>
  <SecurityLog>
    <!--ro, opt, object, security Log-->
    <isSupportSecurityLog>
      <!--ro, opt, bool, whether the device supports the security Log-->true
    </isSupportSecurityLog>
    <isSupportLogServer>
      <!--ro, opt, bool, whether it supports security log server configuration-->true
    </isSupportLogServer>
  </SecurityLog>

```

```

</isSupportLogServer>
<isSupportLogServerTest>
  <!--ro, opt, bool, whether it supports security log server test-->true
</isSupportLogServerTest>
<SecurityLogTypeList>
  <!--ro, req, array, List of security log types, subType:object-->
  <SecurityLogType>
    <!--ro, opt, object, List of log types-->
    <primaryType>
      <!--ro, req, enum, main Log type, subType:string, desc:"Event", "Operation", "Other" (other types), "ALL" (all types)-->Event
    </primaryType>
    <secondaryType opt="test">
      <!--ro, req, enum, sub type, subType:string, attr:opt{req, string}, desc:see other types on the appendix. "all" (all types)-->all
    </secondaryType>
  </SecurityLogType>
</SecurityLogTypeList>
<SecurityLog>
</RacmCap>

```

### 16.1.6.2 Get device storage capability by channel

#### Request URL

GET /ISAPI/ContentMgmt/channels/<channelID>/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RacmChannelsCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, storage capability of a device, attr:version{req, string, protocolVersion}-->
  <pictureSearchType
opt="AllEvent,CMR,MOTION,ALARM,EDR,ALARMANDMOTION,Command,pic,wlsensor,callhelp,facedetection,FieldDetection,scenechangedetection,LineDetection,regionEntrance,regionExiting,loitering,group,rapidMove,parking,unattendedBaggage,attendedBaggage,vehicleDetection,manual,manualSnapShot,playSnapShot,allPic,methaneConcentrationException,methaneLightIntensityException,bannerDetection,AIOpenPlatform">
    <!--ro, opt, enum, picture search type, subType:string, attr:opt{req, string}, desc:"AllEvent", "CMR", "MOTION", "ALARM", "EDR", "ALARMANDMOTION", "Command", "pic", "wlsensor", "callhelp", "facedetection", "FieldDetection", "scenechangedetection", "LineDetection", "regionEntrance", "regionExiting", "Loitering", "group", "rapidMove", "parking", "unattendedBaggage", "attendedBaggage", "vehicleDetection", "manual", "manualSnapShot", "playSnapShot", "allPic", "methaneConcentrationException", "methaneLightIntensityException", "bannerDetection", "waterQualityDetection", "waterLevelDetection", "parallelParking", "waterLevelAndRainfall", "AIOpenPlatform", "VOCsGasDetection", "angleDeviationDetection"-->AllEvent
  </pictureSearchType>
</RacmChannelsCap>

```

### 16.1.6.3 Get the system security capability

#### Request URL

GET /ISAPI/Security/capabilities?username=<userName>

#### Query Parameter

Parameter Name	Parameter Type	Description
userName	string	user name

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<SecurityCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, system security capability, attr:version{req, string, protocolVersion}-->
  <supportUserNums>
    <!--ro, opt, int, supported maximum number of users-->0
  </supportUserNums>
  <userBondIpNums>
    <!--ro, opt, int, supported maximum number of IP addresses that can be bound-->0
  </userBondIpNums>
  <userBondMacNums>

```

```

<!--ro, opt, int, supported maximum number of MAC addresses that can be bound-->0
</userBondMacNums>
<isSupCertificate>
    <!--ro, opt, bool, whether the device supports authentication-->true
</isSupCertificate>
<issupIllegalLoginLock>
    <!--ro, opt, bool, whether the device supports Locking Login-->true
</issupIllegalLoginLock>
<isSupportOnlineUser>
    <!--ro, opt, bool, whether the device supports the online user configuration-->true
</isSupportOnlineUser>
<isSupportAnonymous>
    <!--ro, opt, bool, whether the device supports anonymous Login-->true
</isSupportAnonymous>
<isSupportStreamEncryption>
    <!--ro, opt, bool, whether the device supports stream encryption-->true
</isSupportStreamEncryption>
<securityVersion opt="1,2">
    <!--ro, opt, int, encryption capability set, attr:opt{req, string}, desc:the encryption capability of each version consists of two parts: encryption algorithm and the range of encrypted nodes currently 1 refers to AES128 encryption and 2 refers to AES256 encryption, the range of encrypted nodes is described in each protocol-->1
    </securityVersion>
    <keyIterateNum>
        <!--ro, opt, int, secret key iteration times, dep:or,{$.SecurityCap.securityVersion,eq,1},{$.SecurityCap.securityVersion,eq,2}, desc:this node depends on the node securityVersion, the range is between 100 and 1000-->100
    </keyIterateNum>
    <isSupportUserCheck>
        <!--ro, opt, bool, whether the device supports verifying the Login password when editing (editing/adding/deleting) user parameters, dep:or, {$.SecurityCap.securityVersion,eq,0},{$.SecurityCap.securityVersion,eq,1}, desc:it is an added capability, which indicates that whether supporting the login password verification for editing/adding/deleting user parameters, this node depends on the node securityVersion, which means that it is only valid for the versions that support encrypting the sensitive information-->true
        </isSupportUserCheck>
    <isSupportSecurityQuestionConfig>
        <!--ro, opt, bool, whether the device supports answering security questions, desc:related URI: /ISAPI/Security/questionConfiguration-->true
    </isSupportSecurityQuestionConfig>
    <supportSecurityNode opt="wirelessServer,wirelessDial">
        <!--ro, opt, string, additional node for supporting sensitive information encryption, attr:opt{req, string}, desc:in the earlier version of the solution for encrypting the sensitive information via ISAPI (i.e., the solution for transmitting plaintext via ISAPI), some URIs containing sensitive information are left out now these URIs need to be encrypted for transmission, this capability node is added to be compatible with old devices
            if wirelessServer in this node is returned, it indicates that the following sensitive information should be encrypted
            /ISAPI/System/Network/interfaces/<ID>/wirelessServer/capabilities, /ISAPI/System/Network/interfaces/<ID>/wirelessServer, if wirelessDial in this node is returned, it indicates that the following sensitive information should be encrypted
            /ISAPI/System/Network/WirelessDial/Interfaces,
        /ISAPI/System/Network/WirelessDial/Interfaces/<ID>-->wirelessServer
    </supportSecurityNode>
    <SecurityLimits>
        <!--ro, opt, object, capability of configuring security limit parameters-->
        <LoginPasswordLenLimit min="1" max="16">
            <!--ro, opt, string, length limit of the user's login password, attr:min{req, int},max{req, int}-->1
        </LoginPasswordLenLimit>
        <SecurityAnswerLenLimit min="1" max="128">
            <!--ro, opt, string, length limit of the security question's answer, attr:min{req, int},max{req, int}-->1
        </SecurityAnswerLenLimit>
    </SecurityLimits>
    <RSAKeyLength opt="512,1024,2048" def="2048">
        <!--ro, opt, enum, HTTPS certificate length, subType:int, attr:opt{req, string},def{req, string}, desc:512, 1024, 2048-->2048
    </RSAKeyLength>
    <isSupportONVIFUserManagement>
        <!--ro, opt, bool, whether the device supports user management of Open Network Video Interface Protocol-->true
    </isSupportONVIFUserManagement>
    <WebCertificateCap>
        <!--ro, opt, object, HTTP authentication capability, desc:if this node is not returned, it indicates that device supports basic and digest authentication-->
        <CertificateType opt="basic,digest,digest/basic">
            <!--ro, req, enum, certificate type, subType:string, attr:opt{req, string}, desc:"basic" (basic authentication), "digest" (digest authentication), "digest/basic" (digest/basic authentication)-->basic
        </CertificateType>
        <SecurityAlgorithm>
            <!--ro, opt, object, security algorithm, dep:or,{$.SecurityCap.WebCertificateCap.CertificateType,eq,digest}, {$.SecurityCap.WebCertificateCap.CertificateType,eq,digest/basic}-->
            <algorithmType opt="MD5,SHA256,MD5/SHA256">
                <!--ro, opt, enum, algorithm type, subType:string, attr:opt{req, string}, desc:"MD5", "SHA256", "MD5/SHA256"-->MD5
            </algorithmType>
        </SecurityAlgorithm>
    </WebCertificateCap>
    <isSupportConfigFileImport>
        <!--ro, opt, bool, whether the device supports importing the configuration file, desc:true (support), this node is not returned (not support)-->true
    </isSupportConfigFileImport>
    <isSupportConfigFileExport>
        <!--ro, opt, bool, whether the device supports exporting the configuration file, desc:true (support), this node is not returned (not support)-->true
    </isSupportConfigFileExport>
    <cfgFileSecretKeyLenLimit min="0" max="16">
        <!--ro, opt, int, length limit of the configuration file's verification key, attr:min{req, int},max{req, int}-->1
    </cfgFileSecretKeyLenLimit>
    <isIrreversible>
        <!--ro, opt, bool, whether the device supports irreversible password storage, desc:If this function is not supported, the plaintext password of the user information will be stored in the device; otherwise, the password will be hashed for storage in the device.-->true
    </isIrreversible>
    <salt>
        <!--ro, opt, string, the specific salt used by the user to log in-->test
    </salt>
    <isSupportOnvifInfo>
        <!--ro, opt, bool, whether the device supports getting the Open Network Video Interface Protocol-->true (support), this node is not returned (not
    </isSupportOnvifInfo>

```

```

<!--ro, opt, bool, whether the device supports getting the open network video interface protocol, desc: true (support), this node is not returned (not support)-->true
</isSupportOnvifInfo>
<isSupportGB35114Certificate>
    <!--ro, opt, bool, whether supports GB35114 protocol and CA/SIP authentication certificate-->true
</isSupportGB35114Certificate>
<isSupportDeviceCertificatesManagement>
    <!--ro, opt, bool, whether the device supports device certificate management, desc: if this node is not returned, it indicates that device does not support this function-->true
</isSupportDeviceCertificatesManagement>
<isSupportSIPCertificatesManagement>
    <!--ro, opt, bool, whether the device supports SIP certificate management, desc: this node is returned if the device supports this function and not returned if device does not support this function, related URI: /ISAPI/Security/SIPCertificate, additional node: <SIPCertificatesManagementCap-->true
</isSupportSIPCertificatesManagement>
<isSupportDeviceSelfSignCertExport>
    <!--ro, opt, bool, whether the device supports exporting self-signed certificate-->true
</isSupportDeviceSelfSignCertExport>
<isSupportSecurityEmail>
    <!--ro, opt, bool, whether the device supports configuring the security email, desc: if this node is not returned, it indicates that device does not support this function-->true
</isSupportSecurityEmail>
<maxIllegalLoginTimes min="3" max="20" def="5">
    <!--ro, opt, int, maximum failed illegal login attempts, range:[3,20], attr:min{req, int},max{req, int},def{req, int}, desc: this node is valid when the value of the node issupportIllegalLoginLock is true and will be returned when the maximum failed attempts of illegally logging in to the device reach the limit-->3
</maxIllegalLoginTimes>
<SecurityAdvanced>
    <!--ro, opt, object, security advanced parameters-->
<noOperationEnabled>
    <!--ro, req, bool, whether to enable the control timeout when there is no operation-->true
</noOperationEnabled>
<noOperationTime min="1" max="60" def="15">
    <!--ro, req, int, control timeout when there is no operation, range:[1,60], unit:min, attr:min{req, int},max{req, int},def{req, int}-->15
</noOperationTime>
<SecurityAdvanced>
<LoginLinkNum>
    <!--ro, opt, object, number of controls Logging in at the same time-->
<maxLinkNum min="1" max="128" def="50">
    <!--ro, req, int, maximum number of controls Logging in at the same time, range:[1,128], attr:min{req, int},max{req, int},def{req, int}-->50
</maxLinkNum>
</LoginLinkNum>
<isSupportEncryptCertificate>
    <!--ro, opt, bool, whether the device supports certificate encryption, desc: /ISAPI/Security/deviceCertificate-->true
</isSupportEncryptCertificate>
<maxIllegalLoginLockTime min="1" max="120" def="30.00">
    <!--ro, opt, int, lock time range, range:[1,120], unit:min, attr:min{req, int},max{req, int},def{req, int}-->1
</maxIllegalLoginLockTime>
<isSupportCertificateCustomID>
    <!--ro, opt, bool, whether the device supports using the user's custom ID to configure the certificate-->true
</isSupportCertificateCustomID>
</SecurityCap>

```

#### 16.1.6.4 Get the device's system capability

##### Request URL

GET /ISAPI/System/capabilities

##### Query Parameter

None

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<DeviceCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, system capability of the device, attr:version{opt, string, protocolVersion}-->
    <isSupportSystemMaintain>
        <!--ro, opt, bool, whether the device supports system maintenance, desc: for traffic devices, this node is required and must be true.-->true
    </isSupportSystemMaintain>
    <isSupportReboot>
        <!--ro, opt, bool, whether the device supports rebooting, desc: for traffic devices, this node is required and must be true.-->true
    </isSupportReboot>
    <isSupportFactoryReset>
        <!--ro, opt, bool, whether the device supports restoring to default settings, desc: for traffic devices, this node is required and must be true.-->true
    </isSupportFactoryReset>
    <isSupportUpdatefirmware>
        <!--ro, opt, bool, whether the device supports upgrading, desc: for traffic devices, this node is required and must be true.-->true
    </isSupportUpdatefirmware>
    <SysCap>
        <!--ro, opt, object, system capability-->
        <isSupportDst>
            <!--ro, opt, bool, whether the device supports DST (Daylight Saving Time)-->true
        </isSupportDst>
        <NetworkCap>

```

```

<!--ro, opt, object, network capability, desc:related URI: /ISAPI/System/Network/capabilities-->
<isSupportWireless>
    <!--ro, req, bool, whether the device supports wireless network-->true
</isSupportWireless>
<isSupportPPPoE>
    <!--ro, req, bool, whether the device supports PPPoE (Point to Point Protocol over Ethernet)-->true
</isSupportPPPoE>
<isSupportBond>
    <!--ro, req, bool, whether the device supports NIC (Network Interface Card) bonding-->true
</isSupportBond>
<isSupport802_1x>
    <!--ro, req, bool, whether the device supports 802.1x protocol-->true
</isSupport802_1x>
<isSupportNtp>
    <!--ro, opt, bool, whether the device supports NTP (Network Time Protocol)-->true
</isSupportNtp>
<isSupportFtp>
    <!--ro, opt, bool, whether the device supports FTP (File Transfer Protocol)-->true
</isSupportFtp>
<isSupportUpnp>
    <!--ro, opt, bool, whether the device supports UPnP (Universal Plug and Play ) protocol-->true
</isSupportUpnp>
<isSupportDdns>
    <!--ro, opt, bool, whether the device supports DDNS (Dynamic Domain Name System) service-->true
</isSupportDdns>
<isSupportHttps>
    <!--ro, opt, bool, whether the device supports HTTPS (Hypertext Transfer Protocol Secure)-->true
</isSupportHttps>
<SnmpCap>
    <!--ro, opt, object, SNMP (Simple Network Management Protocol) capability-->
<isSupport>
    <!--ro, req, bool, whether the device supports SNMP-->true
</isSupport>
</SnmpCap>
<isSupportExtNetCfg>
    <!--ro, opt, bool, whether the device supports configuring extended network parameters-->true
</isSupportExtNetCfg>
<isSupportIPFilter>
    <!--ro, opt, bool, whether the device supports IP filtering-->true
</isSupportIPFilter>
<isSupportEZVIZ>
    <!--ro, opt, bool, whether the device supports EZ protocol-->true
</isSupportEZVIZ>
<isSupportEhome>
    <!--ro, opt, bool, whether it supports ISUP functions: true (support),this node is not returned (not support)-->true
</isSupportEhome>
<isSupportWirelessDial>
    <!--ro, opt, bool, whether the device supports wireless dial-up protocol-->true
</isSupportWirelessDial>
<isSupportWirelessServer>
    <!--ro, opt, bool, whether the device supports wireless server-->true
</isSupportWirelessServer>
<isSupportWPS>
    <!--ro, opt, bool, whether the device supports Wi-Fi Protected Setup-->true
</isSupportWPS>
<isWirelessMutexWithWirelessServer>
    <!--ro, opt, bool, whether the device supports mutual exclusion of wireless access and wireless service-->true
</isWirelessMutexWithWirelessServer>
<isSupportMACFilter>
    <!--ro, opt, bool, whether the device supports physical address filtering-->true
</isSupportMACFilter>
<isSupportRFIDData>
    <!--ro, opt, bool, whether the device supports RFID collection configuration-->true
</isSupportRFIDData>
<isSupportwifiProbeSSID>
    <!--ro, opt, bool, whether the device supports configuration of Wi-Fi probe SSID-->true
</isSupportwifiProbeSSID>
<isSupportWifiProbe>
    <!--ro, opt, bool, whether the device supports Wi-Fi probe-->true
</isSupportWifiProbe>
<verificationCode min="1" max="1">
    <!--ro, opt, string, verification code, attr:min{opt, string},max{opt, string}-->test
</verificationCode>
<VerificationCodeModification>
    <!--ro, opt, object, device verification code configuration-->
<verificationCodeType opt="normal,empty">
    <!--ro, opt, string, verification code type, attr:opt{opt, string}-->true
</verificationCodeType>
<isSupportDeclarationURL>
    <!--ro, opt, bool, whether the device supports the hyperlink to the Terms of Service-->true
</isSupportDeclarationURL>
<isSupportPrivacyPolicyURL>
    <!--ro, opt, bool, whether the device supports the hyperlink to the Privacy Policy-->true
</isSupportPrivacyPolicyURL>
<verificationCodeModify opt="true,false">
    <!--ro, opt, bool, whether the device verification code has been modified, attr:opt{opt, string}-->true
</verificationCodeModify>
<Hyperlinks>
    <!--ro, opt, object, hyperlink configuration-->
<declarationURL>
    <!--ro, opt, string, hyperlink to the Terms of Service-->test
</declarationURL>
<privacyPolicyURL>
    <!--ro, opt, string, hyperlink to the Privacy Policy-->test

```

```

        ... ro, opt, bool, whether the device supports the privacy policy, please
    </privacyPolicyURL>
</HyperLinks>
</VerificationCodeModification>
<isSupportIntegrate>
    <!--ro, opt, bool, whether the device supports access protocol configuration-->true
</isSupportIntegrate>
<isSupportPlatformAccess>
    <!--ro, opt, bool, whether the device supports platform access-->true
</isSupportPlatformAccess>
<isSupportGetLinkSocketIP>
    <!--ro, opt, bool, whether the device supports getting the SocketIP of the current Link-->true
</isSupportGetLinkSocketIP>
<isSupportWebSocket>
    <!--ro, opt, bool, whether the device supports WebSocket-->true
</isSupportWebSocket>
<isSupportWebSockets>
    <!--ro, opt, bool, whether the device supports WebSocketS-->true
</isSupportWebSockets>
<isSupportVideoImgDB>
    <!--ro, opt, bool, whether the device supports image and video Library configuration-->true
</isSupportVideoImgDB>
<isSupportDynamicHostName>
    <!--ro, opt, bool, whether the device supports DDNS (Dynamic Domain Name System) configuration-->true
</isSupportDynamicHostName>
<isSupportEmailEncrypt>
    <!--ro, opt, bool, whether the device supports email encryption configuration-->true
</isSupportEmailEncrypt>
<isSupportIntranetSegment>
    <!--ro, opt, bool, whether the device supports LAN segment configuration, desc:if this function is supported, this node will be returned and its value is true; otherwise, this node will be returned and its value is true or not be returned. Related URI: /ISAPI/System/Network/intranetSegment/capabilities?format=json-->true
</isSupportIntranetSegment>
</NetworkCap>
<IOCap>
    <!--ro, opt, object, I/O capability, desc:related URI: /ISAPI/System/IO/capabilities-->
<IOInputPortNums>
    <!--ro, opt, int, number of local input ports of the device-->true
</IOInputPortNums>
<IOOutputPortNums>
    <!--ro, opt, int, number of local output ports of the device-->true
</IOOutputPortNums>
<isSupportStrobeLamp>
    <!--ro, opt, bool, whether the device supports configuring the output of the gate's light, desc:related URI: /ISAPI/System/IO/outputs/strobeLampConf-->true
</isSupportStrobeLamp>
</IOCap>
<SerialCap>
    <!--ro, opt, object, range of RS-485 serial port numbers supported by the device-->
<rs485PortNums>
    <!--ro, opt, int, the maximum number of RS-485 serial ports supported by the device-->true
</rs485PortNums>
<supportRS232Config>
    <!--ro, opt, bool, whether the device supports configuring parameters of RS-232 serial ports-->true
</supportRS232Config>
<rs422PortNums>
    <!--ro, opt, int, the maximum number of RS-422 serial ports supported by the device-->true
</rs422PortNums>
<rs232PortNums>
    <!--ro, opt, int, the maximum number of RS-232 serial ports supported by the device-->true
</rs232PortNums>
</SerialCap>
<VideoCap>
    <!--ro, opt, object, video encoding capability, desc:related URI: /ISAPI/System/Video/capabilities-->
<videoInputPortNums>
    <!--ro, opt, int, number of video input ports-->true
</videoInputPortNums>
<videoOutputPortNums>
    <!--ro, opt, int, number of video output ports-->true
</videoOutputPortNums>
<isSupportHeatmap>
    <!--ro, opt, bool, whether the device supports heat map-->true
</isSupportHeatmap>
<isSupportCounting>
    <!--ro, opt, bool, whether the device supports people counting-->true
</isSupportCounting>
<isSupportPicture>
    <!--ro, opt, bool, whether the device supports OSD picture overlay-->true
</isSupportPicture>
<isSupportBinocularPreviewSwitch>
    <!--ro, opt, bool, whether the device supports switching live view of the dual-lens camera-->true
</isSupportBinocularPreviewSwitch>
<isSupportCalibCheck>
    <!--ro, opt, bool, whether the device supports calibration verification-->true
</isSupportCalibCheck>
<isSupportPIP>
    <!--ro, opt, bool, whether the device supports PIP-->true
</isSupportPIP>
</VideoCap>
<AudioCap>
    <!--ro, opt, object, audio encoding capability, desc:related URL: /ISAPI/System/Audio/capabilities-->
<audioInputNums>
    <!--ro, req, int, number of audio inputs-->1
</audioInputNums>

```

```

<audioOutputNums>
    <!--ro, req, int, number of audio outputs-->1
</audioOutputNums>
</AudioCap>
<isSupportExternalDevice>
    <!--ro, opt, bool, whether the device supports connecting to peripherals: true (support), desc:related URI: /ISAPI/System/externalDevice/capabilities-->true
</isSupportExternalDevice>
<isSupportSubscribeEvent>
    <!--ro, opt, bool, whether the device supports subscribing to events, desc:related URI: /ISAPI/Event/notification/subscribeEventCap-->true
</isSupportSubscribeEvent>
<isSupportDiagnosedData>
    <!--ro, opt, bool, whether the device supports exporting the device diagnosis data, desc:true (support), this node is not returned (not support). related URI: /ISAPI/System/diagnosedData.-->true
</isSupportDiagnosedData>
<isSupportMetadata>
    <!--ro, opt, bool, whether the device supports metadata, desc:related URI: /ISAPI/Streaming/channels/<ID>/metadata/capabilities-->true
</isSupportMetadata>
<supportSmartOverlapChannles opt="1">
    <!--ro, opt, bool, whether the device supports configuring the smart event stream, attr:opt{opt, string}, desc:If this function is supported, this node and the corresponding channel ID will be returned; otherwise, this node will not be returned-->true
</supportSmartOverlapChannles>
</SysCap>
<voicetalkNums>
    <!--ro, opt, int, number of two-way audio channels-->2
</voicetalkNums>
<isSupportSnapshot>
    <!--ro, opt, bool, whether the device supports capturing pictures-->true
</isSupportSnapshot>
<SecurityCap>
    <!--ro, opt, object, encryption capability set-->
<supportUserNums>
    <!--ro, opt, int, supported maximum number of users-->1
</supportUserNums>
<userBondIpNums>
    <!--ro, opt, int, supported maximum number of IP addresses that can be bound-->1
</userBondIpNums>
<userBondMacNums>
    <!--ro, opt, int, supported maximum number of MAC addresses that can be bound-->1
</userBondMacNums>
<isSupCertificate>
    <!--ro, opt, bool, whether the device supports authentication-->true
</isSupCertificate>
<issupIllegalLoginLock>
    <!--ro, opt, bool, whether the device supports locking login-->true
</issupIllegalLoginLock>
<isSupportOnlineUser>
    <!--ro, opt, bool, whether the device supports the online user configuration-->true
</isSupportOnlineUser>
<isSupportAnonymous>
    <!--ro, opt, bool, whether the device supports anonymous login-->true
</isSupportAnonymous>
<isSupportStreamEncryption>
    <!--ro, opt, bool, whether the device supports stream encryption-->true
</isSupportStreamEncryption>
<securityVersion opt="1,2,3,4,7">
    <!--ro, opt, string, encryption capability, attr:opt{opt, string}, desc:the encryption capability of each version consists of two parts: encryption algorithm and the range of encrypted nodes. Currently 1 refers to AES128 encryption and 2 refers to AES256 encryption. The range of encrypted nodes is described in each protocol-->test
</securityVersion>
<keyIterateNum>
    <!--ro, opt, int, iteration times, desc:the value is usually between 100 and 1000-->100
</keyIterateNum>
<isSupportUserCheck>
    <!--ro, opt, bool, whether the device supports verifying the login password when editing (editing/adding/deleting) user parameters, desc:this node depends on the node <securityVersion>, which means that it is only valid for the versions that support encrypting the sensitive information-->true
</isSupportUserCheck>
<isSupportSecurityQuestionConfig>
    <!--ro, opt, bool, whether the device supports answering security questions-->true
</isSupportSecurityQuestionConfig>
<supportSecurityNode opt="wirelessServer,wirelessDial">
    <!--ro, opt, string, additional node for supporting sensitive information encryption, attr:opt{opt, string}, desc:in the earlier version of the solution for encrypting the sensitive information via ISAPI (i.e., the solution for transmitting plaintext via ISAPI), some URIs containing sensitive information are left out. Now these URIs need to be encrypted for transmission. This capability node is added to be compatible with old devices-->test
</supportSecurityNode>
<SecurityLimits>
    <!--ro, opt, object, capability of configuring security limit parameters-->
<LoginPasswordLenLimit min="1" max="16">
    <!--ro, opt, string, Length limit of the user's Login password, attr:min{opt, string},max{opt, string}-->test
</LoginPasswordLenLimit>
<SecurityAnswerLenLimit min="1" max="128">
    <!--ro, opt, string, Length limit of the security question's answer, attr:min{opt, string},max{opt, string}-->test
</SecurityAnswerLenLimit>
</SecurityLimits>
<RSAKeyLength opt="512,1024,2048" def="2048">
    <!--ro, opt, enum, HTTPS certificate length, subType:string, attr:opt{opt, string},def{opt, string}, desc:512, 1024, 2048-->2048
</RSAKeyLength>
<isSupportONVIFUserManagement>
    <!--ro, opt, bool, whether the device supports ONVIF user management-->true
</isSupportONVIFUserManagement>
<WebCertificateCap>
    <!--ro, opt, object, HTTP authentication capability-->
<CertificateType opt="basic,digest,digest/basic">
    <!--ro, opt, string, certificate type: basic (authentication) attr:opt{opt, string}-->test

```

```

<!--ro, req, string, certificate type. basic (authentication), user.password, string-->test
</CertificateType>
<SecurityAlgorithm>
    <!--ro, opt, object, encryption algorithm, dep:or,{$.DeviceCap.SecurityCap.WebCertificateCap.CertificateType,eq,digest/basic}, 
    {$.DeviceCap.SecurityCap.WebCertificateCap.CertificateType,eq,digest/basic}-->
    <algorithmType opt="MD5,SHA256,MDS/SHA256">
        <!--ro, opt, string, algorithm type, attr:opt{req, string}-->test
        </algorithmType>
    </SecurityAlgorithm>
</WebCertificateCap>
<isSupportConfigfileImport>
    <!--ro, opt, bool, whether the device supports importing the configuration file-->true
</isSupportConfigfileImport>
<isSupportConfigfileExport>
    <!--ro, opt, bool, whether the device supports exporting the configuration file-->true
</isSupportConfigfileExport>
<cfgFileSecretKeyLenLimit min="0" max="16">
    <!--ro, opt, string, Length limit of the configuration file's verification key, attr:min{opt, string},max{opt, string}-->0
</cfgFileSecretKeyLenLimit>
<isIrreversible>
    <!--ro, opt, bool, whether the device supports irreversible password storage, desc:if this function is not supported, the plaintext password of the 
    user information will be stored in the device; otherwise, the password will be hashed for storage in the device-->true
</isIrreversible>
<salt>
    <!--ro, opt, string, the specific salt used by the user to log in-->test
</salt>
<isSupportOnvifInfo>
    <!--ro, opt, bool, whether the device supports getting the ONVIF protocol information-->true
</isSupportOnvifInfo>
<isSupportDeviceCertificatesManagement>
    <!--ro, opt, bool, whether the device supports certificate management-->true
</isSupportDeviceCertificatesManagement>
<isSupportSIPCertificatesManagement>
    <!--ro, opt, bool, whether the device supports SIP certificate management-->true
</isSupportSIPCertificatesManagement>
<isSupportDeviceSelfSignCertExport>
    <!--ro, opt, bool, whether the device supports exporting the self-signed certificate-->true
</isSupportDeviceSelfSignCertExport>
<isSupportSecurityEmail>
    <!--ro, opt, bool, whether the device supports configuring the security email-->true
</isSupportSecurityEmail>
<isSupportGB35114Certificate>
    <!--ro, opt, bool, N/A-->true
</isSupportGB35114Certificate>
<maxIllegalLoginTimes min="3" max="20" def="5">
    <!--ro, opt, string, maximum failed illegal login attempts, attr:min{opt, string},max{opt, string},def{opt, string}, desc:This node is valid when the 
    value of the node <issupportIllegalLoginLock> is true and will be returned when the maximum failed attempts of illegally logging in to the device reach the 
    limit.-->1
</maxIllegalLoginTimes>
<SecurityAdvanced>
    <!--ro, opt, object, advanced security configuration-->
    <noOperationEnabled>
        <!--ro, req, bool, whether to enable the control timeout when there is no operation-->true
    </noOperationEnabled>
    <noOperationTime min="1" max="60" def="15">
        <!--ro, req, int, control timeout when there is no operation, unit:分钟, attr:min{opt, string},max{opt, string},def{opt, string}-->1
    </noOperationTime>
</SecurityAdvanced>
<LoginLinkNum>
    <!--ro, opt, object, number of users Logging in at the same time-->
    <maxLinkNum min="1" max="128" def="50">
        <!--ro, req, int, number of controls Logging in at the same time, attr:min{opt, string},max{opt, string},def{opt, string}-->64
    </maxLinkNum>
</LoginLinkNum>
<isSupportEncryptCertificate>
    <!--ro, opt, bool, whether the device supports certificate encryption, desc:/ISAPI/Security/deviceCertificate-->true
</isSupportEncryptCertificate>
<maxIllegalLoginLockTime min="1" max="120" def="30.00">
    <!--ro, opt, int, Lock time range, range:[1,120], unit:min, attr:min{req, int},max{req, int},def{req, int}-->1
</maxIllegalLoginLockTime>
<isSupportCertificateCustomID>
    <!--ro, opt, bool, whether the device supports using the user's custom ID to configure the certificate-->true
</isSupportCertificateCustomID>
</SecurityCap>
<EventCap>
    <!--ro, opt, object, event capability-->
    <isSupportHDFull>
        <!--ro, opt, bool, whether the device supports HDD full alarm-->true
    </isSupportHDFull>
    <isSupportHDError>
        <!--ro, opt, bool, whether the device supports HDD error alarm-->true
    </isSupportHDError>
    <isSupportNicBroken>
        <!--ro, opt, bool, whether the device supports the network disconnected event-->true
    </isSupportNicBroken>
    <isSupportIpConflict>
        <!--ro, opt, bool, whether the device supports the IP address conflict event-->true
    </isSupportIpConflict>
    <isSupportIILAccess>
        <!--ro, opt, bool, whether the device supports the illegal login event-->true
    </isSupportIILAccess>
    <isSupportViException>
        <!--ro, opt, bool, whether the device supports video input exception (only for analog channels)-->true
    </isSupportViException>

```

```

<isSupportViMismatch>
    <!--ro, opt, bool, whether the device supports the event of video standard mismatch-->true
</isSupportViMismatch>
<isSupportRecordException>
    <!--ro, opt, bool, whether the device supports the recording exception event-->true
</isSupportRecordException>
<isSupportTriggerFocus>
    <!--ro, opt, bool, whether the device supports defocus exception-->true
</isSupportTriggerFocus>
<isSupportMotionDetection>
    <!--ro, opt, bool, whether the device supports motion detection-->true
</isSupportMotionDetection>
<isSupportVideoLoss>
    <!--ro, opt, bool, whether the device supports video Loss events-->true
</isSupportVideoLoss>
<isSupportTamperDetection>
    <!--ro, opt, bool, whether the device supports tampering detection-->true
</isSupportTamperDetection>
<isSupportFaceContrast>
    <!--ro, opt, bool, whether the device supports face picture comparison events-->true
</isSupportFaceContrast>
<isSupportFaceLib>
    <!--ro, opt, bool, Front-end devices support Linkage by face picture library., desc:related URI: /ISAPI/Event/schedules/facelib/<ID>/<FDID>-->true
</isSupportFaceLib>
<isSupportStudentsStoodUp>
    <!--ro, opt, bool, whether the device supports the student standing up detection event-->true
</isSupportStudentsStoodUp>
<isSupportFramesPeopleCounting>
    <!--ro, opt, bool, whether the device supports the regional people counting event-->true
</isSupportFramesPeopleCounting>
<isSupportFaceSnap>
    <!--ro, opt, bool, whether the device supports face capture events-->true
</isSupportFaceSnap>
<isSupportPersonDensityDetection>
    <!--ro, opt, bool, whether the device supports people density detection-->true
</isSupportPersonDensityDetection>
<isSupportPersonQueueDetection>
    <!--ro, opt, bool, whether the device supports queue detection events-->true
</isSupportPersonQueueDetection>
<isSupportTeacherBehaviorDetect>
    <!--ro, opt, bool, whether the device supports teacher's behavior detection-->true
</isSupportTeacherBehaviorDetect>
<isSupportCityManagement>
    <!--ro, opt, bool, whether the device supports the city management function-->true
</isSupportCityManagement>
<isSupportMixedTargetDetection>
    <!--ro, opt, bool, whether the device supports multi-target-type detection events-->true
</isSupportMixedTargetDetection>
<isSupportFaceSnapModeling>
    <!--ro, opt, bool, whether the device supports face modeling events-->true
</isSupportFaceSnapModeling>
<isSupportIntersectionAnalysis>
    <!--ro, opt, bool, whether the device supports the intersection analysis-->true
</isSupportIntersectionAnalysis>
<isSupportLuma>
    <!--ro, opt, bool, VQD brightness exception-->true
</isSupportLuma>
<isSupportChroma>
    <!--ro, opt, bool, VQD image color cast-->true
</isSupportChroma>
<isSupportSnow>
    <!--ro, opt, bool, VQD snow noise-->true
</isSupportSnow>
<isSupportStreak>
    <!--ro, opt, bool, VQD stripe noise-->true
</isSupportStreak>
<isSupportFreeze>
    <!--ro, opt, bool, VQD image freeze-->true
</isSupportFreeze>
<isSupportSigLose>
    <!--ro, opt, bool, VQD signal Loss-->true
</isSupportSigLose>
<isSupportClarity>
    <!--ro, opt, bool, VQD resolution exception-->true
</isSupportClarity>
<isSupportJitter>
    <!--ro, opt, bool, VQD image flutter-->true
</isSupportJitter>
<isSupportBlock>
    <!--ro, opt, bool, VQD video tampering-->true
</isSupportBlock>
<isSupportFlowers>
    <!--ro, opt, bool, VQD blurred image alarm-->true
</isSupportFlowers>
<isSupportNoise>
    <!--ro, opt, bool, VQD blurred screen-->true
</isSupportNoise>
<isSupportGhost>
    <!--ro, opt, bool, VQD image noise-->true
</isSupportGhost>
<isSupportPurple>
    <!--ro, opt, bool, VQD purple fringing-->true
</isSupportPurple>
<isSupportVibration>
    <!--ro, opt, bool, VQD vibration-->true
</isSupportVibration>

```

```

<!--SupportICR>
<!--ro, opt, bool, VOD ICP-->true
</isSupportICR>
<isSupportProtectiveFilm>
<!--ro, opt, bool, VOD protective film unremoved-->true
</isSupportProtectiveFilm>
</EventCap>
<RacmCap>
<!--ro, opt, object, UI before picture search-->
<nasNums>
<!--ro, opt, int, supported number of NAS devices, desc:related URI: /ISAPI/ContentMgmt/Storage/nas-->1
</nasNums>
<pictureSearchType
opt="AllEvent,CMR,MOTION,ALARM,EDR,ALARMANDMOTION,Command,pir,wlsensor,callhelp,facedetection,FieldDetection,scenechangedetection,LineDetection,
regionEntrance,regionExiting,loitering,group,rapidMove,parking,unattendedBaggage,attendedBaggage,vehicleDetection,manual,manualSnapShot,playSnapShot,safetyH
elmet,
personQueueCounting,personQueueTime,violentMotion,leavePosition,advReachHeight,peopleNumChange,mixedTargetDetection,illegalParking,pedestrian,construction,r
oadblock,
abandonedObject,trafficAccident,fogDetection,smokeDetection,personDensity,failDown,accessController,videoInterCom,GJD,Luminite,OPTEX,securityControlPanel,ge
tUp,toiletTarry,
playCellphone,vehicleMonitor,vibrationDetection,running,retention,spacingChange,yardTarry,keyPersonGetUp,sitQuietly,standUp,plateRecognition,smoking,onThePh
one,fatigueDriving,
notLookingStraightAhead,leaveThePost,seatBelt,sunglasses,identityAbnormal,ForwardCollisionWarning,LaneDepartureWarning,BlindSpotDetection,HeadwayMonitoringW
arning,
PedestrianCollisionWarning,SuddenSpeedUp,SuddenSpeedDown,SuddenLeftTurn,SuddenRightTurn,Turnover,NotYieldToPedestrian,unregisteredStreetVendor,stallOutsideS
hop,stallOccupyingRoad, illegalHeap,illegalParkingofNon-
MotorVehicle,illegalOutdoorAdvertisement,packGarbage,stallUnderUmbrella,dustbinOverflow,exposeGarbage,hangClothingAlongStreet,allPic,ATMPanel,
ATMSurround,ATMFace,ATMSafetyCabin,loitering,smokeDetectAlarm,alarmResult">
<!--ro, opt, string, picture search type, attr:opt{opt, string}-->test
</pictureSearchType>
<recordSearchType
opt="AllEvent,CMR,MOTION,ALARM,EDR,ALARMANDMOTION,Command,pir,wlsensor,callhelp,facedetection,FieldDetection,scenechangedetection,LineDetection,
regionEntrance,regionExiting,loitering,group,rapidMove,parking,unattendedBaggage,attendedBaggage,vehicleDetection,manual,manualSnapShot,playSnapShot,safetyH
elmet,
failDown,personDensityDetection,smokeDetection,vibrationDetection,unregisteredStreetVendor,stallOutsideShop,stallOccupyingRoad,illegalHeap,illegalParkingofN
on-MotorVehicle,
illegalOutdoorAdvertisement,packGarbage,stallUnderUmbrella,dustbinOverflow,exposeGarbage,hangClothingAlongStreet,ATMPanel,ATMSurround,ATMFace,ATMSafetyCabin
">
<!--ro, opt, string, video search type, attr:opt{opt, string}, desc:"LLEvent" (all events), "CMR" (capture by schedule), "MOTION" (capture motion
detection pictures), "ALARM" (capture alarm pictures), "EDR" (capture alarm pictures or motion detection pictures), "ALARMANDMOTION" (capture alarm pictures
and motion detection pictures), "Command", "pir" (pir alarm), "wlsensor" (wireless alarm), "callhelp" (panic alarm), "facedetection" (face detection),
"FieldDetection" (intrusion detection), "scenechangedetection" (scene change detection), "LineDetection" (line crossing detection)-->test
</recordSearchType>
<isSupportRacmChannelsCap>
<!--ro, opt, bool, whether the device supports the Racm capability by channel, desc:related URI: /ISAPI/ContentMgmt/channels/<ID>/capabilities-->true
</isSupportRacmChannelsCap>
<SecurityLog>
<!--ro, opt, object, device's capability of security log-->
<isSupportSecurityLog>
<!--ro, opt, bool, whether the device supports the security log-->true
</isSupportSecurityLog>
<isSupportLogServer>
<!--ro, opt, bool, whether it supports security log server configuration-->true
</isSupportLogServer>
<isSupportLogServerTest>
<!--ro, opt, bool, whether it supports security log server test-->true
</isSupportLogServerTest>
<SecurityLogTypeList>
<!--ro, req, array, List of security log types, subType:object-->
<SecurityLogType>
<!--ro, opt, object, security log type-->
<primaryType>
<!--ro, req, string, main log type, desc:"Event", "Operation", "Other", "ALL"-->test
</primaryType>
<secondaryType opt="all">
<!--ro, req, string, minor log type, attr:opt{opt, string}, desc:for other types, refer to the appendix-->test
</secondaryType>
</SecurityLogType>
</SecurityLogTypeList>
</SecurityLog>
</RacmCap>
<SmartCap>
<!--ro, opt, object, compact and smart capability, desc:/ISAPI/Smart/capabilities-->
<isSupportROI>
<!--ro, opt, bool, whether the device supports ROI encoding configuration-->true
</isSupportROI>
<isSupportFaceDetect>
<!--ro, opt, bool, whether the device supports face detection-->true
</isSupportFaceDetect>
<isSupportIntelliTrace>
<!--ro, opt, bool, whether the device supports smart tracking-->true
</isSupportIntelliTrace>
<isSupportFieldDetection>
<!--ro, opt, bool, whether the device supports intrusion detection-->true
</isSupportFieldDetection>
<isSupportDefocusDetection>
<!--ro, opt, bool, whether the device supports defocus detection-->true
</isSupportDefocusDetection>
<isSupportAudioDetection>
<!--ro, opt, bool, whether the device supports audio detection-->true
</isSupportAudioDetection>
<isSupportSceneChangeDetection>
<!--ro, opt, bool, whether the device supports scene change detection-->true
</isSupportSceneChangeDetection>

```

```

<isSupportLineDetection>
    <!--ro, opt, bool, whether the device supports line crossing detection-->true
</isSupportLineDetection>
<isSupportRegionEntrance>
    <!--ro, opt, bool, whether the device supports region entrance detection-->true
</isSupportRegionEntrance>
<isSupportRegionExiting>
    <!--ro, opt, bool, whether the device supports region exiting detection-->true
</isSupportRegionExiting>
<isSupportLoitering>
    <!--ro, opt, bool, whether the device supports loitering detection-->true
</isSupportLoitering>
<isSupportGroup>
    <!--ro, opt, bool, whether the device supports people gathering detection-->true
</isSupportGroup>
<isSupportRapidMove>
    <!--ro, opt, bool, whether the device supports fast moving detection-->true
</isSupportRapidMove>
<isSupportParking>
    <!--ro, opt, bool, whether the device supports parking detection-->true
</isSupportParking>
<isSupportUnattendedBaggage>
    <!--ro, opt, bool, whether the device supports unattended baggage detection-->true
</isSupportUnattendedBaggage>
<isSupportAttendedBaggage>
    <!--ro, opt, bool, whether the device supports object removal detection-->true
</isSupportAttendedBaggage>
<isSupportPeopleDetection>
    <!--ro, opt, bool, whether the device supports human body detection-->true
</isSupportPeopleDetection>
<isSupportSmartCalibration>
    <!--ro, opt, bool, whether the device supports calibration filter size-->true
</isSupportSmartCalibration>
<isSupportStorageDetection>
    <!--ro, opt, bool, whether the device supports health monitoring storage-->true
</isSupportStorageDetection>
<isSupportChannelResource>
    <!--ro, opt, bool, whether the device supports channel resources-->true
</isSupportChannelResource>
</SmartCap>
<TestCap>
    <!--ro, opt, object, test capability-->
<isSupportEmailTest>
    <!--ro, opt, bool, whether the device supports email test-->true
</isSupportEmailTest>
</TestCap>
<PanoramaCap>
    <!--ro, opt, object, capability of panoramic channels-->
<isSupportGeneratePanorama>
    <!--ro, opt, bool, whether the device supports generating panoramic images-->true
</isSupportGeneratePanorama>
<isSupportPanoramaPosition3D>
    <!--ro, opt, bool, whether the device supports panoramic 3D position-->true
</isSupportPanoramaPosition3D>
<isSupportPreset>
    <!--ro, opt, bool, whether the device supports configuring panoramic preset-->true
</isSupportPreset>
</PanoramaCap>
<WLAlarmCap>
    <!--ro, opt, object, wireless alarm capability-->
<isSupportTeleControl>
    <!--ro, opt, bool, whether the device supports remote control-->true
</isSupportTeleControl>
<isSupportPIR>
    <!--ro, opt, bool, whether the device supports PIR configuration-->true
</isSupportPIR>
<isSupportWL Sensors>
    <!--ro, opt, bool, N/A-->true
</isSupportWL Sensors>
<isSupportCallHelp>
    <!--ro, opt, bool, whether the device supports calling for help-->true
</isSupportCallHelp>
<WL SensorsNum>
    <!--ro, opt, int, N/A-->1
</WL SensorsNum>
</WLAlarmCap>
<isSupportGIS>
    <!--ro, opt, bool, whether the device supports GIS, desc:related URI: /ISAPI/GIS/channels-->true
</isSupportGIS>
<isSupportCompass>
    <!--ro, opt, bool, whether the device supports compass configuration, desc:related URI: /ISAPI/Compass/channels/<ID>/capabilities-->true
</isSupportCompass>
<isSupportRoadInfoOverlays>
    <!--ro, opt, bool, whether the device supports overlaying the lane information-->true
</isSupportRoadInfoOverlays>
<isSupportFaceCaptureStatistics>
    <!--ro, opt, bool, whether the device supports face capture statistics, desc:related URI:
/ISAPI/Intelligent/channels/<ID>/faceCaptureStatistics/search-->true
</isSupportFaceCaptureStatistics>
<isSupportElectronicsEnlarge>
    <!--ro, opt, bool, whether the device supports electronics enlarge-->true
</isSupportElectronicsEnlarge>
<isSupportRemoveStorage>

```

```

<!--ro, opt, bool, whether the device supports remote storage-->true
</isSupportRemoveStorage>
</isSupportGPSLabelTracking>
    <!--ro, opt, bool, whether the device supports GPS tag tracking, desc:related URI:/ISAPI/GIS/channels/<ID>/GPSLabelTracking?format=json-->true
</isSupportGPSLabelTracking>
</isSupportFirmwareVersionInfo>
    <!--ro, opt, bool, whether the device supports displaying the firmware version information-->true
</isSupportFirmwareVersionInfo>
</isSupportSetupCalibration>
    <!--ro, opt, bool, whether the device supports configuring calibration-->true
</isSupportSetupCalibration>
</isSupportLaserSpotManual>
    <!--ro, opt, bool, whether the device supports configuring Laser Light spot-->true
</isSupportLaserSpotManual>
</isSupportTraffic>
    <!--ro, opt, bool, traffic service capability-->true
</isSupportTraffic>
</isSupportPicInfoOverlap>
    <!--ro, opt, bool, whether the device supports overlaying pictures and texts-->true
</isSupportPicInfoOverlap>
</isSupportChannelEventCap>
    <!--ro, opt, bool, whether the device supports getting the event capability by channel, desc:related URI: /ISAPI/Event/channels/<ID>/capabilities-->true
</isSupportChannelEventCap>
</isSupportGPSCalibration>
    <!--ro, opt, bool, whether the device supports the GPS calibration function-->true
</isSupportGPSCalibration>
<supportSnapshotChannel opt="1">
    <!--ro, opt, string, the capability supported by the capture schedule channel, attr:opt{opt, string}-->test
</supportSnapshotChannel>
</isSupportActiveMulticast>
    <!--ro, opt, bool, whether the device supports multicast configuration (/ISAPI/Streaming/channels/<ID>/capabilities)-->true
</isSupportActiveMulticast>
</isSupportChannelEventListCap>
    <!--ro, opt, bool, it indicates that whether the device supports getting the event capability of all channels of the device-->true
</isSupportChannelEventListCap>
</VCAResourceChannelsCap>
    <!--ro, opt, object, it indicates that the device supports the independent switch of intelligent resources by channel-->
<ChannelsList>
    <!--ro, opt, array, channel list, subType:object-->
    <channelsID>
        <!--ro, req, int, No. of channels supported by the device-->1
    </channelsID>
</ChannelsList>
</VCAResourceChannelsCap>
</isSupportAutoMaintenance>
    <!--ro, opt, bool, whether the device supports automatic maintenance, desc:related URI: /ISAPI/System/autoMaintenance/capabilities?format=json-->true
</isSupportAutoMaintenance>
</isSupportRtspOverHTTPS>
    <!--ro, opt, bool, supports tspOverHTTPS-->true
</isSupportRtspOverHTTPS>
</isSupportAUXInfoCap>
    <!--ro, opt, bool, it indicates that whether the device supports getting the capability of getting the attributes of device's all channels-->true
</isSupportAUXInfoCap>
</regionClip>
    <!--ro, opt, object, the channels and streams that supports area cropping-->
    <channels opt="101">
        <!--ro, opt, string, the channels and streams that supports area cropping, attr:opt{opt, string}, desc:101: channel No.*100+stream type-->test
    </channels>
</regionClip>
</isSupportISUPHttpPassthrough>
    <!--ro, opt, bool, whether the device supports ISUPV5.0http transparent transmission-->true
</isSupportISUPHttpPassthrough>
</isSupportSnapshotAsync>
    <!--ro, opt, bool, whether the device supports capturing pictures asynchronously (the picture URL will be returned), desc:related URI:
/ISAPI/Streaming/channels/<ID>/picture/async?format=json-->true
</isSupportSnapshotAsync>
</isSupportVehicleMonitor>
    <!--ro, opt, bool, whether the device supports intelligent arming and tracking of vehicles, desc:related URI:
/ISAPI/Traffic/channels/<ID>/vehicleMonitor/capabilities?format=json-->true
</isSupportVehicleMonitor>
</isSupportManualVehicleMonitor>
    <!--ro, opt, bool, whether the device supports manual arming and tracking of vehicles, desc:related URI:
/ISAPI/Traffic/channels/<ID>/vehicleMonitor/manual/capabilities?format=json-->true
</isSupportManualVehicleMonitor>
</isSupportCalibrationStream>
    <!--ro, opt, bool, supported capability of calibration stream, desc:related URI: /ISAPI/Streaming/calibStream/capabilities?format=json-->true
</isSupportCalibrationStream>
</isSupportConvergenceCloud>
    <!--ro, opt, bool, whether the device supports the URI for EZ interconnection channels, desc:URI for capability of all channels:
/ISAPI/Event/convergenceCloud/channels/capabilities?format=json; URI for capability of specified channel:
/ISAPI/Event/convergenceCloud/channels/<ID>/capabilities?format=json-->true
</isSupportConvergenceCloud>
<characteristicCode min="1" max="128">
    <!--ro, opt, string, device attribute code, range:[1,128], attr:min{opt, string},max{opt, string}, desc:related URI:
/ISAPI/System/deviceInfo/characteristicCode?format=json-->test
</characteristicCode>
</isSupportSIMCardStatus>
    <!--ro, opt, bool, whether the device supports getting the information of SIM card status, desc:related URI: /ISAPI/System/SIMCardStatus?format=json-->true
</isSupportSIMCardStatus>
</OpenPlatformCap>
    <!--ro, opt, object, detailed HEOP capability supported by the device, desc:related URI: /ISAPI/Custom/OpenPlatform/capabilities-->
</isSupportOpenPlatform>
    <!--ro, req, bool, whether the device supports HEOP (Embedded Open Platform)-->true

```

```

... ro, opt, bool, whether the device supports URI (Embedded open / close, etc.) -->
</isSupportOpenPlatform>
</OpenPlatformCap>
<manualEvent opt="fielddetection,linedetection,ANPR,faceCapture,mixedTargetDetection,studentsStoodUp,headRiseRate,teacherBehavior">
    <!--ro, opt, string, whether the device supports URI for manual triggering event, attr:opt=req, string), desc:related URI:>
</ISAPI/Intelligent/channels/<ID>/manualEvent/<eventType>?format=json-->test
</manualEvent>
<isSupportGetBoxDoorDetectionCfg>
    <!--ro, opt, bool, whether the device supports getting chassis door status detection configuration, desc:POST /ISAPI/System/GetBoxDoorDetectionCfg?
format=json-->true
</isSupportGetBoxDoorDetectionCfg>
<isSupportModifyBoxDoorDetectionCfg>
    <!--ro, opt, bool, whether the device supports modifying chassis door status configuration, desc:PUT /ISAPI/System/ModifyBoxDoorDetectionCfg?
format=json-->true
</isSupportModifyBoxDoorDetectionCfg>
<isSupportSmartBoxStatus>
    <!--ro, opt, bool, whether the device supports intelligent chassis status, desc:/ISAPI/System/SmartBoxStatus?format=json-->true
</isSupportSmartBoxStatus>
<isSupportBatteryStatusCtrl>
    <!--ro, opt, bool, whether the device supports battery status event upload control parameters, desc:/ISAPI/System/BatteryStatusCtrl?format=json-->true
</isSupportBatteryStatusCtrl>
<isSupportBoxDoorDetectionCfg>
    <!--ro, opt, bool, whether the device supports configuring chassis door status detection, desc:/ISAPI/System/BoxDoorDetectionCfg?format=json-->true
</isSupportBoxDoorDetectionCfg>
<isSupportBoxDoorStatus>
    <!--ro, opt, bool, whether the device supports chassis door status, desc:/ISAPI/System/BoxDoorStatus?format=json-->true
</isSupportBoxDoorStatus>
<isSupportLightningProtectionStatus>
    <!--ro, opt, bool, whether the device supports Lightning protection status, desc:/ISAPI/System/LightningProtectionStatus?format=json-->true
</isSupportLightningProtectionStatus>
<isSupportGetPowerPortWorkParamList>
    <!--ro, opt, bool, whether the device supports getting the working parameter List of power supply interface, desc:POST
/ISAPI/System/GetPowerPortWorkParamList?format=json-->true
</isSupportGetPowerPortWorkParamList>
</DeviceCap>

```

### 16.1.6.5 Export device configuration file

#### Request URL

GET /ISAPI/System/configurationData?secretkey=<secretkey>

#### Query Parameter

Parameter Name	Parameter Type	Description
security	string	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	string	The initialization vector, and it is required when security is 1 or 2.
secretkey	string	The verification key, it is provided by the upper-layer. It should be encrypted for exporting and recorded for importing.

#### Request Message

None

#### Response Message

None

### 16.1.6.6 Get the device attribute code

#### Request URL

GET /ISAPI/System/deviceInfo/characteristicCode?format=json

#### Query Parameter

Parameter Name	Parameter Type	Description
security	string	--
iv	string	--

## Request Message

None

## Response Message

```
{  
    "Characteristic": {  
        /*ro, req, object, attribute code, desc:attribute code*/  
        "code": "+/  
        /*ro, opt, string, device attribute code, range:[1,128], desc:device attribute code*/  
    }  
}
```

## 16.1.6.7 Set restoring devices' default parameters

### Request URL

PUT /ISAPI/System/factoryReset?mode=<mode>&childDevID=<devIndex>&loginPassword=<loginPassword>

### Query Parameter

Parameter Name	Parameter Type	Description
mode	enum	--
devIndex	string	--
loginPassword	string	--
security	string	--
iv	string	--

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>  
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">  
    <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->  
    <requestURL>  
        <!--ro, req, string, request URL-->null  
    </requestURL>  
    <statusCode>  
        <!--ro, req, enum, status code, subType:int, desc:@ (OK), 1 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0  
    </statusCode>  
    <statusString>  
        <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK  
    </statusString>  
    <subStatusCode>  
        <!--ro, req, string, sub status code, desc:sub status code-->OK  
    </subStatusCode>  
</ResponseStatus>
```

## 16.1.6.8 Get the network service capability

### Request URL

GET /ISAPI/System/Network/capabilities

### Query Parameter

None

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>  
<NetworkCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">  
    <!--ro, req, object, get the network service capability, attr:version{int, string, protocolVersion}-->
```

```

<!-- ro, req, object, get the network service capability, deviceStatus,opt, setting, protocolSetting -->
<isSupportWirelessisSupportWirelessisSupportPPPoEisSupportPPPoEisSupportBondisSupportBondisSupport802_1xisSupport802_1xisSupportNtpisSupportNtpisSupportFtpisSupportFtpisSupportUpnpisSupportUpnpisSupportDdnsisSupportDdnsisSupportHttpsisSupportHttpsSnmpCapisSupportisSupportSnmpCapisSupportExtNetCfgisSupportExtNetCfgisSupportIPFilterisSupportIPFilterisSupportEZVIZisSupportEZVIZisSupportEhomeisSupportEhomeisSupportWirelessDialisSupportWirelessDialisSupportWirelessServerisSupportWirelessServerisSupportWPSisSupportWPSisWirelessMutexWithWirelessServerisWirelessMutexWithWirelessServerisSupportMACFilterisSupportMACFilterisSupportRFIDDataisSupportRFIDDataisSupportwifiProbeSSIDisSupportwifiProbeSSIDisSupportwifiProbeisSupportwifiProbeverificationCode min="1" max="10"verificationCodeVerificationCodeModificationverificationCodeType opt="normal,empty"verificationCodeTypeisSupportDeclarationURLisSupportDeclarationURLisSupportPrivacyPolicyURLisSupportPrivacyPolicyURLverificationCodeModify opt="true,false"verificationCodeModifyhyperlinksdeclarationURLdeclarationURLprivacyPolicyURL

```

```

</privacyPolicyURL>
</Hyperlinks>
</VerificationCodeModification>
<isSupportIntegrate>
    <!--ro, opt, bool, whether the device supports access protocol configuration-->true
</isSupportIntegrate>
<isSupportPlatformAccess>
    <!--ro, opt, bool, whether the device supports platform access-->true
</isSupportPlatformAccess>
<isSupportGetLinkSocketIP>
    <!--ro, opt, bool, whether the device supports getting the SocketIP of the current Link-->true
</isSupportGetLinkSocketIP>
<isSupportWebSocket>
    <!--ro, opt, bool, whether the device supports WebSocket-->true
</isSupportWebSocket>
<isSupportWebSockets>
    <!--ro, opt, bool, whether the device supports WebSockets-->true
</isSupportWebSockets>
<isSupportVideoImgDB>
    <!--ro, opt, bool, whether the device supports image and video library configuration-->true
</isSupportVideoImgDB>
<isSupportDynamicHostName>
    <!--ro, opt, bool, whether the device supports DDNS (Dynamic Domain Name System) configuration-->true
</isSupportDynamicHostName>
<isSupportEmailEncrypt>
    <!--ro, opt, bool, whether the device supports email encryption configuration-->true
</isSupportEmailEncrypt>
<isSupportIntranetSegment>
    <!--ro, opt, bool, whether the device supports LAN segment configuration, desc:if this function is supported, this node will be returned and its value is true; otherwise, this node will be returned and its value is false or not be returned. Related URI: /ISAPI/System/Network/intranetSegment/capabilities?format=json-->true
</isSupportIntranetSegment>
</NetworkCap>

```

### 16.1.6.9 Set SSH parameters

#### Request URL

PUT /ISAPI/System/Network/ssh?readerID=<readerID>&type=<type>

#### Query Parameter

Parameter Name	Parameter Type	Description
readerID	string	--
type	enum	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<SSH xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--wo, opt, object, attr:version{opt, string, protocolVersion}-->
    <enabled>
        <!--wo, req, bool, whether to enable the function-->true
    </enabled>
    <port>
        <!--wo, opt, int-->22
    </port>
</SSH>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status description: OK,Device Busy,Device Error,Invalid Operation,Invalid XML Format,Invalid XML Content,Reboot,Additional Error, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
      <statusString>
        <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
      </subStatusCode>
    </statusCode>
  </statusCode>
</ResponseStatus>

```

### 16.1.6.10 Reboot device

#### Request URL

PUT /ISAPI/System/reboot?childDevID=<devIndex>&module=<module>

#### Query Parameter

Parameter Name	Parameter Type	Description
devIndex	string	--
module	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status description: OK,Device Busy,Device Error,Invalid Operation,Invalid XML Format,Invalid XML Content,Reboot,Additional Error, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
      <statusString>
        <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
      </subStatusCode>
    </statusCode>
  </statusCode>
</ResponseStatus>

```

### 16.1.6.11 Get software service parameters

#### Request URL

GET /ISAPI/System/Software/channels/<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<SoftwareService xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, software service, attr:version{req, string, protocolVersion}-->
</SoftwareService>

```

### 16.1.6.12 Set software service parameters

#### Request URL

PUT /ISAPI/System/Software/channels/<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<SoftwareService xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, software service, attr:version{req, string, protocolVersion}-->
</SoftwareService>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.1.6.13 Get device status

#### Request URL

GET /ISAPI/System/status

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<DeviceStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, device status, attr:version{opt, string, protocolVersion}-->
  <currentDeviceTime>
    <!--ro, opt, datetime, device current time-->1970-01-01T00:00:00+08:00
  </currentDeviceTime>
  <deviceUpTime>
    <!--ro, opt, int, device running time, unit:s-->1200
  </deviceUpTime>
  <TemperatureList>
    <!--ro, opt, array, temperature data list, subType:object-->
    <Temperature>
      <!--ro, opt, object, temperature data-->
      <tempSensorDescription>
        <!--ro, req, string, description-->test
      </tempSensorDescription>
    </Temperature>
  </TemperatureList>
</DeviceStatus>

```

```

<temperature>
  <!--ro, req, float, temperature value-->1.000
</temperature>
</Temperature>
</TemperatureList>
<CPUList>
  <!--ro, opt, array, CPU data list, subType:object-->
<CPU>
  <!--ro, opt, object, CPU data-->
<cpuDescription>
  <!--ro, req, string, description-->test
</cpuDescription>
<cpuUtilization>
  <!--ro, req, int, CPU usage, range:[0,100]-->1
</cpuUtilization>
</CPU>
</CPUList>
<MemoryList>
  <!--ro, opt, array, memory list, subType:object-->
<Memory>
  <!--ro, opt, object, memory-->
<memoryDescription>
  <!--ro, req, string, description-->test
</memoryDescription>
<memoryUsage>
  <!--ro, req, float, used memory size, unit:MB-->1.0
</memoryUsage>
<memoryAvailable>
  <!--ro, req, float, available memory size, unit:MB-->1.0
</memoryAvailable>
</Memory>
</MemoryList>
<CameraList>
  <!--ro, opt, array, camera list, subType:object-->
<Camera>
  <!--ro, opt, object, camera-->
<zoomReverseTimes>
  <!--ro, req, int, zoom reverse times-->0
</zoomReverseTimes>
<zoomTotalSteps>
  <!--ro, req, int, zoom total steps-->0
</zoomTotalSteps>
<focusReverseTimes>
  <!--ro, req, int, focus reverse times-->0
</focusReverseTimes>
<focusTotalSteps>
  <!--ro, req, int, focus total steps-->0
</focusTotalSteps>
<irisShiftTimes>
  <!--ro, req, int, iris shift times-->0
</irisShiftTimes>
<irisTotalSteps>
  <!--ro, req, int, iris total steps-->0
</irisTotalSteps>
<icrShiftTimes>
  <!--ro, req, int, ICR shift times-->0
</icrShiftTimes>
<icrTotalSteps>
  <!--ro, req, int, ICR total steps-->0
</icrTotalSteps>
<lensInitTimes>
  <!--ro, req, int, Lens Initialization times-->0
</lensInitTimes>
<cameraRunTotalTime>
  <!--ro, req, int, camera total running time-->0
</cameraRunTotalTime>
</Camera>
</CameraList>
<DomeInfoList>
  <!--ro, opt, array, PTZ information list, subType:object-->
<DomeInfo>
  <!--ro, opt, object, PTZ information-->
<domeRunTotalTime>
  <!--ro, opt, int, the combination of running time at -20° below/40° above/-20°- 40°-->1
</domeRunTotalTime>
<runTimeUnderNegativetwenty>
  <!--ro, opt, int, running time at -20° below-->1
</runTimeUnderNegativetwenty>
<runTimeBetweenNtwentypforty>
  <!--ro, opt, int, running time at -20°- 40°-->1
</runTimeBetweenNtwentypforty>
<runtimeOverPositiveforty>
  <!--ro, opt, int, running time at 40° above-->1
</runtimeOverPositiveforty>
<panTotalRounds>
  <!--ro, opt, int, pan rounds-->1
</panTotalRounds>
<tiltTotalRounds>
  <!--ro, opt, int, tilt rounds-->1
</tiltTotalRounds>
<heatState>
  <!--ro, opt, enum, heater status, subType:int, desc:0 (temperature control), 1 (remain open), 2 (remain closed)-->1
</heatState>
<fanStatus>

```

```

<!--translate>
<!--ro, opt, enum, fan status, subType:int, desc:@0 (temperature control), 1 (remain open), 2 (remain closed)-->1
</FanState>
<panFrecRecord>
<!--ro, opt, int, pan calling times-->1
</panFrecRecord>
<tiltFrecRecord>
<!--ro, opt, int, tilt calling times-->1
</tiltFrecRecord>
</DomeInfo>
</DomeInfoList>
</totalRebootCount>
<!--ro, opt, int, device reboot times-->1
</totalRebootCount>
<AlarmInfoList>
<!--ro, opt, array, alarm information list, subType:object, desc:add-->
<AlarmInfo>
<!--ro, opt, object, alarm information-->
<channelId>
<!--ro, req, int, channel No.-->1
</channelId>
<FaceAlarm>
<!--ro, opt, object, face picture alarm-->
<faceSnapRecvAlarm>
<!--ro, req, int, alarm times of DSP received via face picture-->1
</faceSnapRecvAlarm>
<faceSnapSendAlarm>
<!--ro, req, int, alarm times of face picture sent to the platform-->1
</faceSnapSendAlarm>
</FaceAlarm>
<MixTargetAlarm>
<!--ro, opt, object, multi-target-type detection alarm-->
<mixTargetRecvAlarm>
<!--ro, req, int, alarm times of DSP received via multi-target-type detection-->1
</mixTargetRecvAlarm>
<mixTargetSendAlarm>
<!--ro, req, int, times of face picture alarm sent to the platform-->1
</mixTargetSendAlarm>
</MixTargetAlarm>
</AlarmInfo>
</AlarmInfoList>
</DeviceStatus>

```

### 16.1.6.14 Get the video analysis task capability

#### Request URL

GET /ISAPI/System/Video/capabilities

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<VideoCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <videoInputPortNums>
    <!--ro, opt, int, number of video input ports-->0
  </videoInputPortNums>
  <videoOutputPortNums>
    <!--ro, opt, int, number of video output ports-->0
  </videoOutputPortNums>
  <isSupportHeatmap>
    <!--ro, opt, bool, whether the device supports heat map-->true
  </isSupportHeatmap>
  <isSupportCounting>
    <!--ro, opt, bool, whether the device supports people counting-->true
  </isSupportCounting>
  <isSupportPicture>
    <!--ro, opt, bool, whether the device supports OSD picture overlay, desc:related URI: /ISAPI/System/Video/inputs/channels/<channelID>/image/picture-->true
  </isSupportPicture>
  <isSupportBinocularPreviewSwitch>
    <!--ro, opt, bool, whether the device supports switching Live view of the dual-Lens camera-->true
  </isSupportBinocularPreviewSwitch>
  <isSupportCalibCheck>
    <!--ro, opt, bool, whether the device supports calibration verification-->true
  </isSupportCalibCheck>
  <isSupportPIP>
    <!--ro, opt, bool-->true
  </isSupportPIP>
  <OSDLanguage opt="GBK,EUC-KR,Hebrew" def="GBK">
    <!--ro, opt, enum, subType:string, attr:opt{req, string},def{req, string}-->GBK
  </OSDLanguage>
  <isSupportVideoOutputMode>
    <!--ro, opt, bool, related URI: /ISAPI/System/Video/outputs/mode/capabilities?format=json, desc:related URI: /ISAPI/System/Video/outputs/mode/capabilities?format=json-->true
  </isSupportVideoOutputMode>
</VideoCap>

```

### 16.1.6.15 Get the wireless alarm capabilities

#### Request URL

GET /ISAPI/WLAlarm/capabilities

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<WLAlarmCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <isSupportTeleControl>
    <!--ro, opt, bool-->true
  </isSupportTeleControl>
  <isSupportPIR>
    <!--ro, opt, bool-->true
  </isSupportPIR>
  <isSupportWL Sensors>
    <!--ro, opt, bool-->true
  </isSupportWL Sensors>
  <isSupportCallHelp>
    <!--ro, opt, bool-->true
  </isSupportCallHelp>
  <WL SensorsNum>
    <!--ro, opt, int-->0
  </WL SensorsNum>
</WLAlarmCap>

```

### 16.1.7 System Upgrade

#### 16.1.7.1 Get the device upgrade progress

#### Request URL

GET /ISAPI/System/upgradeStatus?format=json

#### Query Parameter

None

### Request Message

None

### Response Message

```
{  
    "requestURL": "/ISAPI/Streaming/channels/1",  
    /*ro, opt, string, request URL*/  
    "statusCode": "test",  
    /*ro, req, string, status code*/  
    "statusString": "test",  
    /*ro, req, string, status description*/  
    "subStatusCode": "test",  
    /*ro, req, string, sub status code*/  
    "errorCode": 1,  
    /*ro, opt, int, This field is required when the value of statusCode is not 1, and it corresponds to subStatusCode.*/  
    "errorMsg": "ok",  
    /*ro, opt, string, This field is required when the value of statusCode is not 1. Detailed error description of a certain parameter can be provided*/  
    "upgrading": "TRUE",  
    /*ro, opt, string, whether the device is upgrading: "TRUE" (upgrading), "FALSE" (not in upgrading)*/  
    "percent": 22,  
    /*ro, opt, int, upgrade progress (% complete)*/  
    "idList": [  
        /*ro, opt, array, ID list, subType:object*/  
        {  
            "id": "test",  
            /*ro, req, string, analysis unit ID*/  
            "percent": 22,  
            /*ro, opt, int, upgrade progress (% complete)*/  
            "status": "test"  
            /*ro, opt, string, "backingUp" (backing up upgrade)*/  
        }  
    ]  
}
```

## 16.1.7.2 Get the device upgrading status and progress

### Request URL

GET /ISAPI/System/upgradeStatus?type=<Type>&childDevID=<devIndex>

### Query Parameter

Parameter Name	Parameter Type	Description
Type	string	--
devIndex	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>  
<upgradeStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">  
    <!--ro, req, object, upgrade status and result, attr:version[req, string, protocolVersion]-->  
    <upgrading>  
        <!--ro, req, bool, upgrade status-->true  
    </upgrading>  
    <percent>  
        <!--ro, req, int, upgrade progress (% complete), range:[0,100]-->1  
    </percent>  
</upgradeStatus>
```

## 16.1.8 Time Settings

### 16.1.8.1 Get device time synchronization management parameters

### Request URL

GET /ISAPI/System/time

### Query Parameter

None

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Time xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, time management, attr:version{opt, string, protocolVersion}-->
  <timeMode>
    <!--ro, req, enum, time synchronization mode, subType:string, desc:"NTP" (NTP time synchronization), "manual" (manual time synchronization), "satellite" (satellite time synchronization), "platform" (platform time synchronization), "NONE" (time synchronization is not allowed or no time synchronization source), "GB28181" (GB28181 time synchronization)-->NTP
    </timeMode>
    <localTime>
      <!--ro, opt, string, local time, range:[0,256], dep:and,{$.Time.timeMode,eq,manual}-->2019-02-28T10:50:44+08:00
    </localTime>
    <timeZone>
      <!--ro, opt, string, time zone, range:[0,256], dep:and,{$.Time.timeMode,eq,manual},{$.Time.timeMode,eq,NTP}-->CST-8:00:00DST00:30:00,M4.1.0/02:00:00,M10.5.0/02:00:00
    </timeZone>
  </Time>
```

### 16.1.8.2 Set device time synchronization management parameters

#### Request URL

PUT /ISAPI/System/time

#### Query Parameter

None

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Time xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, time management, attr:version{opt, string, protocolVersion}-->
  <timeMode>
    <!--req, enum, time synchronization mode, subType:string, desc:"NTP" (NTP time synchronization), "manual" (manual time synchronization), "satellite" (satellite time synchronization), "platform" (platform time synchronization), "NONE" (time synchronization is not allowed or no time synchronization source), "GB28181" (GB28181 time synchronization)-->NTP
    </timeMode>
    <localTime>
      <!--opt, string, Local time, range:[0,256], dep:and,{$.Time.timeMode,eq,manual}-->2019-02-28T10:50:44+08:00
    </localTime>
    <timeZone>
      <!--opt, string, time zone, range:[0,256], dep:and,{$.Time.timeMode,eq,manual},{$.Time.timeMode,eq,NTP}-->CST-8:00:00DST00:30:00,M4.1.0/02:00:00,M10.5.0/02:00:00
    </timeZone>
  </Time>
```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code-->OK
  </subStatusCode>
  <FailedNodeInfoList>
    <!--ro, opt, object, information list of failed nodes, desc:for the manual time synchronization of central analysis cluster, this field is returned if time synchronization failed-->
    <FailedNodeInfo>
      <!--ro, opt, object, information of failed nodes-->
      <nodeID>
        <!--ro, req, string, node ID, range:[0,64]-->test
      </nodeID>
      <nodeIP>
        <!--ro, req, string, node IP, range:[0,20]-->test
      </nodeIP>
      <reason>
        <!--ro, opt, string, reason why the node failed to synchronize time, range:[0,128]-->test
      </reason>
    </FailedNodeInfo>
  </FailedNodeInfoList>
</ResponseStatus>

```

### 16.1.8.3 Get the capability of device time synchronization management

#### Request URL

GET /ISAPI/System/time/capabilities

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Time xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, time management capability set, attr:version{opt, string, protocolVersion}-->
  <timeMode opt="NTP,manual,satellite,SDK,28181,ONVIF,ALL(任何支持的校时方式都允许校时),NONE(不允校时或无校时源),platform">
    <!--ro, req, enum, time synchronization mode, subType:string, attr:opt{opt, string}, desc:"NTP" (NTP time synchronization), "manual" (manual time synchronization), "satellite" (satellite time synchronization), "platform" (platform time synchronization), "NONE" (time synchronization is not allowed or no time synchronization source), "GB28181" (GB28181 time synchronization)-->NTP
  </timeMode>
  <localTime min="0" max="256">
    <!--ro, opt, string, local time, range:[0,256], attr:min{opt, string},max{opt, string}-->test
  </localTime>
  <timeZone min="0" max="256">
    <!--ro, opt, string, time zone, range:[0,256], attr:min{opt, string},max{opt, string}-->test
  </timeZone>
  <satelliteInterval min="0" max="3600">
    <!--ro, opt, int, satellite time synchronization interval, step:1, unit:min, attr:min{opt, string},max{opt, string}, desc:unit: minute-->60
  </satelliteInterval>
  <timeType opt="local,UTC">
    <!--ro, opt, enum, time type, subType:string, attr:opt{opt, string}, desc:"local" (Local time), "UTC" (UTC time)-->local
  </timeType>
</Time>

```

### 16.1.8.4 Set device local time

#### Request URL

PUT /ISAPI/System/time/localTime

#### Query Parameter

None

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code description-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.1.8.5 Set parameters of all NTP servers

#### Request URL

PUT /ISAPI/System/time/ntpServers

#### Query Parameter

None

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<NTPServerList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, NTP server information list, subType:object, attr:version{opt, string, protocolVersion}-->
  <NTPServer>
    <!--opt, object, NTP server information-->
    <id>
      <!--req, string, ID-->1
    </id>
    <addressingFormatType>
      <!--req, enum, NTP server address type, subType:string, desc:"ipaddress" (IP address), "hostname" (domain name)-->hostname
    </addressingFormatType>
    <hostName>
      <!--opt, string, NTP server domain name, range:[1,64]-->12345
    </hostName>
    <portNo>
      <!--opt, int, port No., range:[1,65535], desc:the default port No. is 123-->123
    </portNo>
    <synchronizeInterval>
      <!--opt, int, time synchronization interval, range:[1,10800], unit:min-->1440
    </synchronizeInterval>
  </NTPServer>
</NTPServerList>
```

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->/ISAPI/xxxx
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.1.8.6 Get the parameters of a specific NTP (Network Time Protocol) server

#### Request URL

GET /ISAPI/System/time/ntpServers

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<NTPServerList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, NTP server information List, subType:object, attr:version{req, string, protocolVersion}-->
  <NTPServer>
    <!--ro, opt, object, NTP server information-->
    <id>
      <!--ro, req, string, ID-->1
    </id>
    <addressingFormatType>
      <!--ro, req, enum, NTP server address type, subType:string, desc:"ipaddress" (IP address), "hostname" (domain name)-->hostname
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, NTP server domain name, range:[1,64]-->12345
    </hostName>
    <portNo>
      <!--ro, opt, int, port No., range:[1,65535], desc:the default port No. is 123-->123
    </portNo>
    <synchronizeInterval>
      <!--ro, opt, int, time synchronization interval, range:[1,10800], unit:min-->1440
    </synchronizeInterval>
  </NTPServer>
</NTPServerList>
```

### 16.1.8.7 Set the parameters of a NTP server

#### Request URL

PUT /ISAPI/System/time/ntpServers/<NTPServerID>

#### Query Parameter

Parameter Name	Parameter Type	Description
NTPServerID	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<NTPServer xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, NTP server information, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string, ID-->1
  </id>
  <addressingFormatType>
    <!--req, enum, IP address type of NTP server, subType:string, desc:"ipaddress" (IP address), "hostname" (domain name)-->hostname
  </addressingFormatType>
  <hostName>
    <!--opt, string, NTP server domain name地址去掉, range:[1,64]-->12345
  </hostName>
  <portNo>
    <!--opt, int, port No., range:[1,65535], step:1, desc:port No.-->1
  </portNo>
  <synchronizeInterval>
    <!--opt, int, time synchronization interval, range:[1,10800], step:1, unit:min, desc:NTP time synchronization interval, unit: minute-->1440
  </synchronizeInterval>
</NTPServer>
```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, desc:sub status code-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, desc:sub status code-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

### 16.1.8.8 Get the parameters of a NTP server

#### Request URL

GET /ISAPI/System/time/ntpServers/<NTPServerID>

#### Query Parameter

Parameter Name	Parameter Type	Description
NTPServerID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<NTPServer xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, NTP server information, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string, ID-->1
  </id>
  <addressingFormatType>
    <!--ro, req, enum, IP address type of NTP server, subType:string, desc:"ipaddress" (IP address), "hostname" (domain name)-->hostname
  </addressingFormatType>
  <hostName>
    <!--ro, opt, string, NTP server domain name, range:[1,64]-->12345
  </hostName>
  <ipAddress>
    <!--ro, opt, string, IPv4 address, range:[1,32], desc:IPv4 address-->192.168.1.112
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPv6 address, range:[1,128], desc:IPv6 address-->1030:C9B4:FF12:48AA:1A2B
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, port No., range:[1,65535], step:1, desc:port No.-->1
  </portNo>
  <synchronizeInterval>
    <!--ro, opt, int, time synchronization interval, range:[1,10800], step:1, unit:min, desc:NTP time synchronization interval, unit: minute-->1440
  </synchronizeInterval>
</NTPServer>

```

### 16.1.8.9 Get the configuration capability of parameters of a specific NTP (Network Time Protocol) server

#### Request URL

GET /ISAPI/System/time/ntpServers/capabilities

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<NTPServerList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, NTP server information list, subType:object, attr:version{opt, string, protocolVersion}-->
  <NTPServer>
    <!--ro, opt, object, NTP server information-->
    <id>
      <!--ro, req, string, ID-->1
    </id>
    <addressingFormatType opt="ipaddress,hostname">
      <!--ro, req, enum, NTP server address type, subType:string, attr:opt{req, string}, desc:"ipaddress" (IP address), "hostname" (domain name)-->hostname
    </addressingFormatType>
    <hostName min="1" max="64">
      <!--ro, opt, string, NTP server domain name, range:[1,64], attr:min{req, int},max{req, int}-->12345
    </hostName>
    <portNo min="1" max="65535">
      <!--ro, opt, int, port No., range:[1,65535], attr:min{req, int},max{req, int}, desc:the default port No. is 123-->123
    </portNo>
    <synchronizeInterval min="1" max="10800">
      <!--ro, opt, int, time synchronization interval, unit:min, attr:min{req, int},max{req, int}-->1440
    </synchronizeInterval>
  </NTPServer>
</NTPServerList>

```

### 16.1.8.10 Execute a test to check if the NTP server is available

#### Request URL

POST /ISAPI/System/time/ntpServers/test

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>

<NPTTestDescription xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <addressingFormatType>
    <!--req, enum, "ipaddress,hostname", subType:string, desc:"ipaddress" (IP address), "hostname" (domain name)-->hostname
  </addressingFormatType>
  <hostName>
    <!--opt, string-->test
  </hostName>
  <ipAddress>
    <!--opt, string, IPv4 address-->192.168.1.112
  </ipAddress>
  <ipv6Address>
    <!--opt, string, IPv6 address-->1030::C9B4:FF12:48AA:1A2B
  </ipv6Address>
  <portNo>
    <!--req, int, port No.-->1
  </portNo>
</NPTTestDescription>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<NPTTestResult xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <errorDescription>
    <!--ro, req, string, error description-->test
  </errorDescription>
</NPTTestResult>

```

### 16.1.8.11 Set the device time type

#### Request URL

PUT /ISAPI/System/time/timeType?format=json

#### Query Parameter

None

#### Request Message

```
{
    "TimeType": {
        /*req, object, time type information, desc:time type information*/
        "type": "UTC",
        /*req, enum, time type, subType:string, desc:"UTC, local"*/
    }
}
```

## Response Message

```
{
    "statusCode": 1,
    /*ro, opt, int, status code, desc:1 (succeeded). It is required when an error occurred*/
    "statusString": "ok",
    /*ro, opt, string, status description, range:[1,64], desc:"ok" (succeeded). It is required when an error occurred*/
    "subStatusCode": "ok",
    /*ro, opt, string, sub status code, range:[1,64], desc:"ok" (succeeded). It is required when an error occurred*/
    "errorCode": 1,
    /*ro, opt, int, error code, desc:it is required when the value of statusCode is not 1, and it corresponds to subStatusCode*/
    "errorMsg": "ok"
    /*ro, opt, string, error information, desc:this field is required when the value of statusCode is not 1*/
}
```

## 16.1.8.12 Get the device time type

### Request URL

GET /ISAPI/System/time/timeType?format=json

### Query Parameter

None

### Request Message

None

### Response Message

```
{
    "TimeType": {
        /*ro, req, object*/
        "type": "UTC",
        /*ro, req, enum, value: local,UTC, subType:string, desc:"UTC,local"*/
    }
}
```

## 16.1.8.13 Get time zone

### Request URL

GET /ISAPI/System/time/timeZone

### Query Parameter

None

### Request Message

None

### Response Message

None

## 16.1.8.14 Set time zone

### Request URL

PUT /ISAPI/System/time/timeZone

### Query Parameter

None

### Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code-->OK
  </subStatusCode>
  </ResponseStatus>
```

## 16.2 Video (General)

### 16.2.1 Backlight Compensation

#### 16.2.1.1 Set the BLC (Backlight Compensation) parameters of a specific channel

##### Request URL

PUT /ISAPI/Image/channels/<channelID>/BLC

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<BLC xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, BLC area, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool, whether to enable BLC-->true
  </enabled>
  <BLCRRegionList>
    <!--opt, array, List of BLC areas, subType:object-->
    <BLCRRegion>
      <!--opt, object, BLC area-->
      <id>
        <!--req, int, area No.-->1
      </id>
      <RegionCoordinatesList>
        <!--req, array, coordinate List, subType:object, range:[3,10]-->
        <RegionCoordinates>
          <!--opt, object, coordinates, desc:the origin is the Lower-Left corner of the screen-->
          <positionX>
            <!--req, int, X-coordinate, range:[0,704]-->1
          </positionX>
          <positionY>
            <!--req, int, Y-coordinate, range:[0,576]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </BLCRRegion>
  </BLCRRegionList>
</BLC>
```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

### 16.2.1.2 Get the BLC (Backlight Compensation) parameters of a specific channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/BLC

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<BLC xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, BLC mode: "UP, DOWN, LEFT, RIGHT, CENTER, MULTI-AREA, Region", attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
  <BLRegionList>
    <!--ro, opt, array, subType:object-->
    <BLRegion>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, int-->1
      </id>
      <RegionCoordinatesList>
        <!--ro, req, array, subType:object, range:[3,10]-->
        <RegionCoordinates>
          <!--ro, opt, object, coordinates, desc:the origin is the lower-left corner of the screen-->
          <positionX>
            <!--ro, req, int, X-coordinate, range:[0,704]-->1
          </positionX>
          <positionY>
            <!--ro, req, int, Y-coordinate, range:[0,576]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </BLRegion>
  </BLRegionList>
</BLC>

```

### 16.2.1.3 Get the backlight compensation (in day mode) parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/BLC/day

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<BLC xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
  <BLRegionList>
    <!--ro, opt, array, subType:object-->
    <BLCRegion>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, int-->1
      </id>
      <RegionCoordinatesList>
        <!--ro, req, array, subType:object, range:[3,10]-->
        <RegionCoordinates>
          <!--ro, opt, object-->
          <positionX>
            <!--ro, req, int, range:[0,704]-->1
          </positionX>
          <positionY>
            <!--ro, req, int, range:[0,576]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </BLCRegion>
  </BLRegionList>
</BLC>
```

### 16.2.1.4 Set the backlight compensation (in day mode) parameters of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/BLC/day

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<BLC xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool-->true
  </enabled>
  <BLRegionList>
    <!--opt, array, subType:object-->
    <BLCRegion>
      <!--opt, object-->
      <id>
        <!--req, int-->1
      </id>
      <RegionCoordinatesList>
        <!--req, array, subType:object, range:[3,10]-->
        <RegionCoordinates>
          <!--opt, object-->
          <positionX>
            <!--req, int, range:[0,704]-->1
          </positionX>
          <positionY>
            <!--req, int, range:[0,576]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </BLCRegion>
  </BLRegionList>
</BLC>
```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.1.5 Get the backlight compensation (in night mode) parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/BLC/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<BLC xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
  <BLCRegionList>
    <!--ro, opt, array, subType:object-->
    <BLCRegion>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, int-->1
      </id>
      <RegionCoordinatesList>
        <!--ro, req, array, subType:object, range:[3,10]-->
        <RegionCoordinates>
          <!--ro, opt, object-->
          <positionX>
            <!--ro, req, int, range:[0,704]-->1
          </positionX>
          <positionY>
            <!--ro, req, int, range:[0,576]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </BLCRegion>
  </BLCRegionList>
</BLC>

```

### 16.2.1.6 Set the backlight compensation (in night mode) parameters of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/BLC/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<BLC xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool-->true
  </enabled>
  <BLCRegionList>
    <!--opt, array, subType:object-->
    <BLCRegion>
      <!--opt, object-->
      <id>
        <!--req, int-->1
      </id>
      <RegionCoordinatesList>
        <!--req, array, subType:object, range:[3,10]-->
        <RegionCoordinates>
          <!--opt, object-->
          <positionX>
            <!--req, int, range:[0,704]-->1
          </positionX>
          <positionY>
            <!--req, int, range:[0,576]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </BLCRegion>
  </BLCRegionList>
</BLC>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.1.7 Get the highlight compensation (in auto mode) parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/HLC

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<HLC xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
  <HLCLevel>
    <!--ro, opt, int-->1
  </HLCLevel>
</HLC>

```

### 16.2.1.8 Set the highlight compensation (in auto mode) parameters of a specified channel

## Request URL

PUT /ISAPI/Image/channels/<channelID>/HLC

## Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<HLC xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool-->true
  </enabled>
  <HLCLevel>
    <!--opt, int-->1
  </HLCLevel>
</HLC>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

## 16.2.1.9 Set the highlight compensation (in day mode) parameters of a specified channel

### Request URL

PUT /ISAPI/Image/channels/<channelID>/HLC/day

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<HLC xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool-->true
  </enabled>
  <HLCLevel>
    <!--opt, int-->1
  </HLCLevel>
</HLC>
```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.1.10 Get the highlight compensation (in day mode) parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/HLC/day

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<HLC xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, opt, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool-->true
  </enabled>
  <HLCLevel>
    <!--req, opt, int-->1
  </HLCLevel>
</HLC>

```

### 16.2.1.11 Set the highlight compensation (in night mode) parameters of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/HLC/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<HLC xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool-->true
  </enabled>
  <HLCLevel>
    <!--opt, int-->1
  </HLCLevel>
</HLC>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.1.12 Get the highlight compensation (in night mode) parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/HLC/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<HLC xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, opt, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
  <HLCLevel>
    <!--ro, opt, int-->1
  </HLCLevel>
</HLC>

```

### 16.2.1.13 Set the wide dynamic range (WDR) parameters in auto mode of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/WDR

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<WDR xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--req, enum, subType:string-->auto
  </mode>
  <WDRLevel>
    <!--opt, int, range:[1,100]-->50
  </WDRLevel>
</WDR>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.1.14 Get the WDR (Wide Dynamic Range) parameters in auto mode of a specific channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/WDR

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<WDR xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--ro, req, enum, WDR mode, subType:string, desc:WDR mode-->auto
  </mode>
  <WDRLevel>
    <!--ro, opt, int, WDR Level, range:[1,100], desc:WDR Level-->50
  </WDRLevel>
</WDR>

```

### 16.2.1.15 Get the wide dynamic range (WDR) parameters in day mode of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/WDR/day

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<WDR xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--ro, req, enum, subType:string-->open
  </mode>
  <WDRLevel>
    <!--ro, opt, int-->1
  </WDRLevel>
</WDR>

```

### 16.2.1.16 Set the wide dynamic range (WDR) parameters in day mode of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/WDR/day

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<WDR xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--req, enum, subType:string-->open
  </mode>
  <WDRLevel>
    <!--opt, int-->1
  </WDRLevel>
</WDR>
```

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->0
  </subStatusCode>
</ResponseStatus>
```

### 16.2.1.17 Get the wide dynamic range (WDR) parameters in night mode of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/WDR/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<WDR xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--ro, req, enum, subType:string-->open
  </mode>
  <WDRLevel>
    <!--ro, opt, int-->1
  </WDRLevel>
</WDR>
```

### 16.2.1.18 Set the wide dynamic range (WDR) parameters in night mode of a specified channel

## Request URL

PUT /ISAPI/Image/channels/<channelID>/WDR/night

## Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<WDR xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--req, enum, subType:string-->open
  </mode>
  <WDRLevel>
    <!--opt, int-->1
  </WDRLevel>
</WDR>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

## 16.2.2 Exposure Settings

### 16.2.2.1 Set the exposure mode for a specified channel

## Request URL

PUT /ISAPI/Image/channels/<channelID>/exposure

## Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Exposure xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, exposure mode, attr:version{req, string, protocolVersion}-->
  <ExposureType>
    <!--req, enum, exposure type, subType:string, desc:exposure type-->auto
  </ExposureType>
  <OverexposeSuppress>
    <!--opt, object-->
    <enabled>
      <!--req, bool-->true
    </enabled>
  </OverexposeSuppress>
</Exposure>
```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

### 16.2.2.2 Get the exposure mode of a specific channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/exposure?parameterType=<parameterType>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Exposure xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, exposure mode, attr:version{req, string, protocolVersion}-->
  <ExposureType>
    <!--ro, req, enum, Lens type, subType:string, desc:lens type-->auto
  </ExposureType>
  <OverexposeSuppress>
    <!--ro, opt, object-->
    <enabled>
      <!--ro, req, bool-->true
    </enabled>
  </OverexposeSuppress>
</Exposure>

```

### 16.2.2.3 Set the exposure time parameters of a specific channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/shutter

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Shutter xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <ShutterLevel>
    <!--opt, enum, exposure Level, subType:string, desc:"1/1", "1/2", "1/3", "1/6", "1/12", "1/25", "1/50", "1/75", "1/100", "1/120", "1/125", "1/150", "1/175", this node is valid when exposure type is "shutter first"-->1/1
  </ShutterLevel>
</Shutter>

```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.2.4 Get the exposure time parameters of a specific channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/shutter?parameterType=<parameterType>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Shutter xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <ShutterLevel>
    <!--ro, opt, enum, exposure Level, subType:string, desc:"1/1", "1/2", "1/3", "1/6", "1/12", "1/25", "1/50", "1/75", "1/100", "1/120", "1/125", "1/150", "1/175", this node is valid when exposure type is "shutter first"-->1/1
  </ShutterLevel>
</Shutter>
```

## 16.2.3 Focus Settings

### 16.2.3.1 Get the focus parameters of the specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/focusConfiguration

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FocusConfiguration xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, focus configuration, attr:version{req, string, protocolVersion}-->
  <focusStyle>
    <!--ro, req, enum, focus mode, subType:string, desc:"AUTO" (automatic focus), "MANUAL" (manual focus), "SEMIAUTOMATIC" (semi-automatic focus)-->AUTO
  </focusStyle>
</FocusConfiguration>

```

### 16.2.3.2 Set the focus parameters for a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/focusConfiguration

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FocusConfiguration xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, focus configuration, attr:version{req, string, protocolVersion}-->
  <focusStyle>
    <!--req, enum, focus mode, subType:string, desc:"auto" (automatic focus), "manual" (manual focus), "semiautomatic" (semiautomatic focus)-->AUTO
  </focusStyle>
</FocusConfiguration>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.3.3 Get the image capability of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/imageCap

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <isSupportRegionalExposure>
    <!--ro, opt, bool-->true
  </isSupportRegionalExposure>
  <isSupportRegionalFocus>
    <!--ro, opt, bool-->true
  </isSupportRegionalFocus>
  <isSupportManualBackGroundCorrect>
    <!--ro, opt, bool-->true
  </isSupportManualBackGroundCorrect>
  <isSupportManualShutterCorrect>
    <!--ro, opt, bool-->true
  </isSupportManualShutterCorrect>
</ImageCap>

```

## 16.2.4 Image Adjustment Settings

### 16.2.4.1 Get image rotate mode of a specific channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/corridor

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<corridor xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, mode, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool, whether to enable the function-->true
  </enabled>
</corridor>

```

### 16.2.4.2 Set parameters of the image rotate mode for a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/corridor

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<corridor xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, mode, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool, whether to enable-->true
  </enabled>
</corridor>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

#### 16.2.4.3 Set the display parameters of a specific channel

##### Request URL

PUT /ISAPI/Image/channels/<channelID>/displayParamSwitch

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<DisplayParamSwitch xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, display parameters, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--req, enum, mode, subType:string, desc:mode-->disable
  </mode>
  <Month>
    <!--opt, object, dep:and, ${$.DisplayParamSwitch.mode, eq, month}-->
    <PeriodList size="4">
      <!--req, array, subType:object, attr:size{req, int}-->
      <Period>
        <!--opt, object-->
        <id>
          <!--req, string-->test
        </id>
        <TimeRangeList size="4">
          <!--req, array, subtype:object, attr:size{req, int}-->
        </TimeRangeList>
      </Period>
    </PeriodList>
  </Month>
</DisplayParamSwitch>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

#### 16.2.5 Image Adjustment

### 16.2.5.1 Get image processing parameters of all channels

#### Request URL

GET /ISAPI/Image/channels

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, array, subType:object, attr:version{req, string, protocolVersion}-->
    <ImageChannel>
        <!--ro, req, object-->
        <id>
            <!--ro, req, int-->1
        </id>
        <enabled>
            <!--ro, req, bool-->true
        </enabled>
        <videoInputID>
            <!--ro, req, int-->1
        </videoInputID>
        <ImageFlip>
            <!--ro, opt, object-->
            <enabled>
                <!--ro, req, bool-->true
            </enabled>
        </ImageFlip>
        <ImageFreeze>
            <!--ro, opt, object-->
            <enabled>
                <!--ro, req, bool-->true
            </enabled>
        </ImageFreeze>
        <proportionalpan>
            <!--ro, opt, object-->
            <enabled>
                <!--ro, req, bool-->true
            </enabled>
        </proportionalpan>
        <WDR>
            <!--ro, opt, object-->
            <WDRLevel>
                <!--ro, opt, int, range:[1,100]-->1
            </WDRLevel>
        </WDR>
        <BLC>
            <!--ro, opt, object-->
            <enabled>
                <!--ro, req, bool-->true
            </enabled>
            <BLCRegionList>
                <!--ro, opt, array, subType:object-->
                <BLCRegion>
                    <!--ro, opt, object-->
                    <id>
                        <!--ro, req, int, range:[1,10]-->1
                    </id>
                    <RegionCoordinatesList>
                        <!--ro, req, array, subType:object, range:[0,4]-->
                        <RegionCoordinates>
                            <!--ro, opt, object-->
                            <positionX>
                                <!--ro, req, int, range:[0,704]-->1
                            </positionX>
                            <positionY>
                                <!--ro, req, int, range:[0,576]-->1
                            </positionY>
                        </RegionCoordinates>
                    </RegionCoordinatesList>
                </BLCRegion>
            </BLCRegionList>
        </BLC>
        <NoiseReduce>
            <!--ro, opt, object-->
            <mode>
                <!--ro, req, enum, subType:string-->close
            </mode>
            <GeneralMode>
                <!--ro, opt, object-->
                <generalLevel>
                    <!--ro, req, int, range:[0,100]-->1
                </generalLevel>
            </GeneralMode>
        </NoiseReduce>
    </ImageChannel>
</ImageChannelList>
```

```

        <!--ro, opt, object-->
</NoiseReduce>
<Exposure>
    <!--ro, opt, object-->
    <ExposureType>
        <!--ro, req, enum, subType:string-->auto
    </ExposureType>
    <OverexposeSuppress>
        <!--ro, opt, object-->
        <enabled>
            <!--ro, req, bool-->true
        </enabled>
    </OverexposeSuppress>
</Exposure>
<Sharpness>
    <!--ro, opt, object-->
    <SharpnessLevel1>
        <!--ro, req, int, range:[0,100]-->1
    </SharpnessLevel1>
</Sharpness>
<Shutter>
    <!--ro, opt, object-->
    <ShutterLevel>
        <!--ro, opt, enum, subType:string-->1/1
    </ShutterLevel>
</Shutter>
<powerLineFrequency>
    <!--ro, opt, object-->
    <powerLineFrequencyMode>
        <!--ro, opt, enum, subType:string-->50hz
    </powerLineFrequencyMode>
</powerLineFrequency>
<Gain>
    <!--ro, opt, object-->
    <GainLevel>
        <!--ro, opt, string, range:[0,10]-->test
    </GainLevel>
</Gain>
<Color>
    <!--ro, opt, object-->
    <brightnessLevel>
        <!--ro, opt, int, range:[0,100]-->0
    </brightnessLevel>
    <contrastLevel>
        <!--ro, opt, int, range:[0,100]-->0
    </contrastLevel>
    <saturationLevel>
        <!--ro, opt, int, range:[0,100]-->0
    </saturationLevel>
    <grayScale>
        <!--ro, opt, object-->
        <grayScaleMode>
            <!--ro, opt, enum, subType:string-->indoor
        </grayScaleMode>
    </grayScale>
</Color>
<IrCutFilter>
    <!--ro, opt, object-->
    <IrCutFilterType>
        <!--ro, opt, enum, subType:string-->auto
    </IrCutFilterType>
</IrCutFilter>
<PTZ>
    <!--ro, opt, object-->
    <enabled>
        <!--ro, opt, bool-->true
    </enabled>
</PTZ>
<HLC>
    <!--ro, opt, object-->
    <enabled>
        <!--ro, req, bool-->true
    </enabled>
    <HLCLevel>
        <!--ro, opt, int, range:[0,10]-->0
    </HLCLevel>
</HLC>
<corridor>
    <!--ro, opt, object-->
    <enabled>
        <!--ro, req, bool-->true
    </enabled>
</corridor>
<Dehaze>
    <!--ro, opt, object-->
    <DehazeMode>
        <!--ro, opt, enum, subType:string-->open
    </DehazeMode>
</Dehaze>
<CaptureMode>
    <!--ro, opt, object-->
    <mode>
        <!--ro, req, enum, subType:string-->close
    </mode>

```

```
</mode>
</CaptureMode>
</ImageChannel>
</ImageChannellist>
```

### 16.2.5.2 Set image processing parameters of all channels

#### Request URL

PUT /ISAPI/Image/channels

#### Query Parameter

None

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageChannellist xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--req, array, subType:object, attr:version{req, string, protocolVersion}-->
    <ImageChannel>
        <!--req, object-->
        <id>
            <!--req, int-->1
        </id>
        <enabled>
            <!--req, bool-->true
        </enabled>
        <videoInputID>
            <!--req, int-->1
        </videoInputID>
        <imageFlip>
            <!--opt, object-->
            <enabled>
                <!--req, bool-->true
            </enabled>
        </imageFlip>
        <imageFreeze>
            <!--opt, object-->
            <enabled>
                <!--req, bool-->true
            </enabled>
        </imageFreeze>
        <proportionalpan>
            <!--opt, object-->
            <enabled>
                <!--req, bool-->true
            </enabled>
        </proportionalpan>
        <WDR>
            <!--opt, object-->
            <WDRLevel>
                <!--opt, int, range:[1,100]-->1
            </WDRLevel>
        </WDR>
        <BLC>
            <!--opt, object-->
            <enabled>
                <!--req, bool-->true
            </enabled>
            <BLCRegionList>
                <!--opt, array, subType:object-->
                <BLCRegion>
                    <!--opt, object-->
                    <id>
                        <!--req, int, range:[1,10]-->1
                    </id>
                    <RegionCoordinatesList>
                        <!--req, array, subType:object, range:[0,4]-->
                        <RegionCoordinates>
                            <!--opt, object-->
                            <positionX>
                                <!--req, int, range:[0,704]-->1
                            </positionX>
                            <positionY>
                                <!--req, int, range:[0,576]-->1
                            </positionY>
                        </RegionCoordinates>
                    </RegionCoordinatesList>
                </BLCRegion>
            </BLCRegionList>
        </BLC>
        <noiseReduce>
            <!--opt, object-->
            <mode>
                <!--req, enum, subType:string-->close
            </mode>
            <GeneralMode>
                <!--opt, object-->
```

```
<generalLevel>
  <!--req, int, range:[0,100]-->1
</generalLevel>
</GeneralMode>
</NoiseReduce>
<Exposure>
  <!--opt, object-->
<ExposureType>
  <!--req, enum, subType:string-->auto
</ExposureType>
<OverexposeSuppress>
  <!--opt, object-->
<enabled>
  <!--req, bool-->true
</enabled>
</OverexposeSuppress>
</Exposure>
<Sharpness>
  <!--opt, object-->
<SharpnessLevel>
  <!--req, int, range:[0,100]-->1
</SharpnessLevel>
</Sharpness>
<Shutter>
  <!--opt, object-->
<ShutterLevel>
  <!--opt, enum, subType:string-->1/1
</ShutterLevel>
</Shutter>
<powerLineFrequency>
  <!--opt, object-->
<powerLineFrequencyMode>
  <!--opt, enum, subType:string-->50hz
</powerLineFrequencyMode>
</powerLineFrequency>
<Gain>
  <!--opt, object-->
<GainLevel>
  <!--opt, string, range:[0,10]-->test
</GainLevel>
</Gain>
<Color>
  <!--opt, object-->
<brightnessLevel>
  <!--opt, int, range:[0,100]-->0
</brightnessLevel>
<contrastLevel>
  <!--opt, int, range:[0,100]-->0
</contrastLevel>
<saturationLevel>
  <!--opt, int, range:[0,100]-->0
</saturationLevel>
<grayScale>
  <!--opt, object-->
<grayScaleMode>
  <!--opt, enum, subType:string-->indoor
</grayScaleMode>
</grayScale>
</Color>
<IrCutFilter>
  <!--opt, object-->
<IrCutFilterType>
  <!--opt, enum, subType:string-->auto
</IrCutFilterType>
</IrCutFilter>
<PTZ>
  <!--opt, object-->
<enabled>
  <!--opt, bool-->true
</enabled>
</PTZ>
<HLC>
  <!--opt, object-->
<enabled>
  <!--req, bool-->true
</enabled>
<HLCLevel>
  <!--opt, int, range:[0,10]-->0
</HLCLevel>
</HLC>
<corridor>
  <!--opt, object-->
<enabled>
  <!--req, bool-->true
</enabled>
</corridor>
<Dehaze>
  <!--opt, object-->
<DehazeMode>
  <!--opt, enum, subType:string-->open
</DehazeMode>
</Dehaze>
<CaptureMode>
  <!--opt, object-->
```

```

<!--opt, object-->
<mode>
    <!--req, enum, subType:string-->close
</mode>
</CaptureMode>
</ImageChannel>
</ImageChannellist>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
    <requestURL>
        <!--ro, req, string-->null
    </requestURL>
    <statusCode>
        <!--ro, req, enum, subType:int-->0
    </statusCode>
    <statusString>
        <!--ro, req, enum, subType:string-->OK
    </statusString>
    <subStatusCode>
        <!--ro, req, string-->OK
    </subStatusCode>
</ResponseStatus>

```

### 16.2.5.3 Set image parameters of a specific channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--req, object, attr:version{req, string, protocolVersion}-->
    <id>
        <!--req, int-->1
    </id>
    <enabled>
        <!--req, bool, whether to enable the function-->true
    </enabled>
    <videoInputID>
        <!--req, int-->1
    </videoInputID>
    <ImageFlip>
        <!--opt, object-->
        <enabled>
            <!--req, bool, whether to enable the function-->true
        </enabled>
    </ImageFlip>
    <ImageFreeze>
        <!--opt, object-->
        <enabled>
            <!--req, bool, whether to enable the function-->true
        </enabled>
    </ImageFreeze>
    <proportionalpan>
        <!--opt, object-->
        <enabled>
            <!--req, bool, whether to enable the function-->true
        </enabled>
    </proportionalpan>
    <WDR>
        <!--opt, object-->
        <mode>
            <!--req, enum, subType:string-->open
        </mode>
        <WDRLevel>
            <!--opt, int, range:[1,100]-->1
        </WDRLevel>
    </WDR>
    <BLC>
        <!--opt, object-->
        <enabled>
            <!--req, bool, whether to enable the function-->true
        </enabled>

```

```

</enabled>
<BLCRegionList>
  <!--opt, array, subType:object-->
  <BLCRegion>
    <!--opt, object-->
    <id>
      <!--req, int, area No., range:[1,10]-->1
    </id>
    <RegionCoordinatesList>
      <!--req, array, subType:object, range:[0,4], desc:rectangle-->
      <RegionCoordinates>
        <!--opt, object, coordinates, desc:the origin is the lower-left corner of the screen-->
        <positionX>
          <!--req, int, X-coordinate, range:[0,704]-->1
        </positionX>
        <positionY>
          <!--req, int, Y-coordinate, range:[0,576]-->1
        </positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
  </BLCRegion>
</BLCRegionList>
</BLC>
<NoiseReduce>
  <!--opt, object-->
  <mode>
    <!--req, enum, subType:string-->close
  </mode>
<GeneralMode>
  <!--opt, object-->
  <generalLevel>
    <!--req, int, range:[0,100]-->1
  </generalLevel>
</GeneralMode>
</NoiseReduce>
<WhiteBalance>
  <!--opt, object, white balance-->
  <WhiteBalanceStyle>
    <!--opt, enum, subType:string-->auto
  </WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--opt, int, range:[0,100]-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--opt, int, range:[0,100]-->1
  </WhiteBalanceBlue>
</WhiteBalance>
<Exposure>
  <!--opt, object-->
  <ExposureType>
    <!--req, enum, subType:string-->auto
  </ExposureType>
  <OverexposeSuppress>
    <!--opt, object-->
    <enabled>
      <!--req, bool, whether to enable the function-->true
    </enabled>
  </OverexposeSuppress>
</Exposure>
<Sharpness>
  <!--opt, object-->
  <SharpnessLevel>
    <!--req, int, range:[0,100]-->1
  </SharpnessLevel>
</Sharpness>
<Shutter>
  <!--opt, object-->
  <ShutterLevel>
    <!--opt, enum, subType:string-->1/1
  </ShutterLevel>
</Shutter>
<powerLineFrequency>
  <!--opt, object-->
  <powerLineFrequencyMode>
    <!--opt, enum, subType:string-->50hz
  </powerLineFrequencyMode>
</powerLineFrequency>
<Gain>
  <!--opt, object, gain-->
  <GainLevel>
    <!--opt, string, range:[0,10]-->test
  </GainLevel>
</Gain>
<Color>
  <!--opt, object-->
  <brightnessLevel>
    <!--opt, int, brightness, range:[0,100]-->0
  </brightnessLevel>
  <contrastLevel>
    <!--opt, int, contrast, range:[0,100]-->0
  </contrastLevel>
  <saturationLevel>
    <!--opt, int, saturation, range:[0,100]-->0
  </saturationLevel>

```

```

</saturationLevel>
<grayScale>
    <!--opt, object-->
    <grayScaleMode>
        <!--opt, enum, subType:string-->indoor
    </grayScaleMode>
</grayScale>
</Color>
<IrcutFilter>
    <!--opt, object-->
    <IrcutFilterType>
        <!--opt, enum, subType:string-->auto
    </IrcutFilterType>
    <nightToDayFilterLevel>
        <!--opt, enum, subType:string-->low
    </nightToDayFilterLevel>
    <nightToDayFilterTime>
        <!--opt, int, range:[0,10]-->1
    </nightToDayFilterTime>
</EventTrigger>
    <!--opt, object-->
    <eventType>
        <!--req, enum, event type, subType:string-->IO
    </eventType>
    <IrcutFilterAction>
        <!--req, enum, subType:string-->day
    </IrcutFilterAction>
</EventTrigger>
</IrcutFilter>
<PTZ>
    <!--opt, object, PTZ-->
    <enabled>
        <!--opt, bool, whether to enable the function-->true
    </enabled>
</PTZ>
<HLC>
    <!--opt, object-->
    <enabled>
        <!--req, bool, whether to enable the function-->true
    </enabled>
    <HLCLevel>
        <!--opt, int, range:[0,10]-->0
    </HLCLevel>
</HLC>
<corridor>
    <!--opt, object-->
    <enabled>
        <!--req, bool, whether to enable the function-->true
    </enabled>
</corridor>
<Dehaze>
    <!--opt, object-->
    <DehazeMode>
        <!--opt, enum, subType:string-->open
    </DehazeMode>
</Dehaze>
<CaptureMode>
    <!--opt, object, capture mode-->
    <mode>
        <!--req, enum, subType:string-->close
    </mode>
</CaptureMode>
<SupplementLight>
    <!--opt, object-->
    <supplementLightMode>
        <!--opt, enum, mode of supplement light, subType:string-->mixed
    </supplementLightMode>
    <irLightBrightness>
        <!--opt, int, brightness of IR supplement light, range:[0,100], desc:this node is valid when supplementLightMode is set to "mixed" and mixedLightBrightnessRegulatMode is set to "manual"-->80
    </irLightBrightness>
    <mixedLightBrightnessRegulatMode>
        <!--opt, enum, brightness adjustment mode of mixed supplement light, subType:string, desc:"manual" (in this mode, you need to set the parameters irLightBrightness and brightnessLimit), "auto" (brightness-related parameters are not required). This node is valid when supplementLightMode is set to "mixed"-->auto
    </mixedLightBrightnessRegulatMode>
    <whiteLightBrightness>
        <!--opt, int, light brightness (regardless of far or near light), range:[0,100], desc:it is shared by different supplement light modes, including "mixed", "whiteLight", "colorVluWhiteLight", "irlight", and "dualLight", when the adjustment mode of the supplement light is "auto"-->80
    </whiteLightBrightness>
</SupplementLight>
<DigitalZoom>
    <!--opt, object-->
    <ZoomRatio>
        <!--opt, enum, subType:string-->1x
    </ZoomRatio>
</DigitalZoom>
<Palettes>
    <!--opt, object-->
    <mode>
        <!--opt, enum, subType:string-->WhiteHot
    </mode>
</Palettes>
<aserl ioh>

```

```

<!-- opt, object-->
<mode>
  <!-- opt, enum, subType:string-->auto
</mode>
</LaserLight>
</ImageChannel>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.5.4 Get image configuration capability of a specific channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, temperature range capability, see XML_Cap_tempRange for details, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, int-->1
  </id>
  <enabled>
    <!--ro, req, bool, whether to enable the function-->true
  </enabled>
  <videoInputID>
    <!--ro, req, int-->1
  </videoInputID>
  <FocusConfiguration>
    <!--ro, opt, object-->
    <focusStyle opt="AUTO,MANUAL,SEMIAUTOMATIC">
      <!--ro, req, string, attr:opt{req, string}-->test
    </focusStyle>
  </FocusConfiguration>
  <ImageFlip>
    <!--ro, opt, object-->
    <enabled opt="true,false">
      <!--ro, req, bool, whether to enable the function, attr:opt{req, string}-->true
    </enabled>
    <ImageFlipStyle opt="LEFTRIGHT,UPDOWN,CENTER,AUTO">
      <!--ro, opt, string, type, attr:opt{req, string}-->LEFTRIGHT
    </ImageFlipStyle>
  </ImageFlip>
  <WDR>
    <!--ro, opt, object-->
    <mode opt="open,close,auto,SDR">
      <!--ro, req, enum, WDR mode: "open,close,auto", subType:string, attr:opt{req, string}, desc:WDR mode: "open,close,auto"-->[open
    </mode>
    <WDRLevel min="1" max="100">
      <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
    </WDRLevel>
  </WDR>
</ImageChannel>

```

```

<!-->
<!--ro, opt, object-->
<enabled opt="true,false">
  <!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<BLMode opt="UP,DOWN,LEFT,RIGHT,CENTER,MULTI-AREA,Region,AUTO">
  <!--ro, opt, enum, subType:string, attr:opt{req, string}-->UP
</BLMode>
</BLC>
<NoiseReduce>
  <!--ro, opt, object-->
  <mode opt="close,general,advanced">
    <!--ro, req, enum, mode, subType:string, attr:opt{req, string}, desc:WDR mode: "open,close,auto"-->close
  </mode>
<GeneralMode>
  <!--ro, opt, object, mode-->
  <generalLevel min="0" max="10">
    <!--ro, req, int, attr:min{req, int},max{req, int}-->1
  </generalLevel>
</GeneralMode>
<AdvancedMode>
  <!--ro, opt, object-->
  <FrameNoiseReduceLevel min="0" max="10">
    <!--ro, req, int, attr:min{req, int},max{req, int}-->1
  </FrameNoiseReduceLevel>
  <InterFrameNoiseReduceLevel min="0" max="10">
    <!--ro, req, int, attr:min{req, int},max{req, int}-->1
  </InterFrameNoiseReduceLevel>
</AdvancedMode>
<NoiseReduce>
<WhiteBalance>
  <!--ro, opt, object, white balance-->
  <WhiteBalanceStyle opt="manual, auto1, auto2, locked, outdoor, indoor, daylightLamp, sodiumlight, autotrace, once, autooutdoor, autosodiumlight, mercurylight, auto, incandescentlight, warmLight, naturalLight">
    <!--ro, req, string, attr:opt{req, string}-->test
  </WhiteBalanceStyle>
  <WhiteBalanceRed min="0" max="10">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue min="0" max="10">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
  </WhiteBalanceBlue>
</WhiteBalance>
<Exposure>
  <!--ro, opt, object-->
  <ExposureType opt="auto,IrisFirst,ShutterFirst,gainFirst,manual,piris,T5280-PQ1,T5289-PQ1,T1140-PQ1,T2712-PQ1,HV1250P-MPIR,pc,pIris-General,HZ1140P-8MPIRA,HZ1135P-8MPSGA,MV1550P-12MPIR,MZ1545P-12MP,HZ0618P-4MPSG,MZ1050P-12MP">
    <!--ro, req, string, attr:opt{req, string}-->test
  </ExposureType>
<OverexposeSuppress>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, whether to enable the function, attr:opt{req, string}-->true
  </enabled>
</OverexposeSuppress>
</Exposure>
<Sharpness>
  <!--ro, opt, object-->
  <SharpnessLevel min="0" max="100">
    <!--ro, req, int, attr:min{req, int},max{req, int}-->1
  </SharpnessLevel>
</Sharpness>
<Shutter>
  <!--ro, opt, object-->
  <ShutterLevel opt="1/25,1/50,1/75,1/100,1/125,1/150,1/175,1/200,1/225,1/250,1/300,1/400,1/500,1/750,1/1000,1/2000,1/4000,1/10000,1/100000">
    <!--ro, opt, string, attr:opt{req, string}-->test
  </ShutterLevel>
</Shutter>
<powerLineFrequency>
  <!--ro, opt, object-->
  <powerLineFrequencyMode opt="50hz,60hz">
    <!--ro, opt, string, mode, attr:opt{req, string}-->test
  </powerLineFrequencyMode>
</powerLineFrequency>
<Gain>
  <!--ro, opt, object, gain-->
  <GainLevel min="0" max="10">
    <!--ro, opt, string, attr:min{req, int},max{req, int}-->test
  </GainLevel>
</Gain>
<Color>
  <!--ro, opt, object, image adjustment capability-->
  <brightnessLevel min="0" max="10">
    <!--ro, opt, int, brightness, attr:min{req, int},max{req, int}-->0
  </brightnessLevel>
  <contrastLevel min="0" max="10">
    <!--ro, opt, int, contrast, attr:min{req, int},max{req, int}-->0
  </contrastLevel>
  <saturationLevel min="0" max="10">
    <!--ro, opt, int, saturation, attr:min{req, int},max{req, int}-->0
  </saturationLevel>
  <grayScale>
    <!--ro, opt, object, gray scale-->
    <grayScaleMode opt="indoor,outdoor">

```

```

<!--ro, opt, string, gray scale mode, attr:opt{req, string}-->test
</grayScaleMode>
</grayScale>
</Color>
<IrcutFilter>
<!--ro, opt, object-->
<IrcutFilterType opt="auto,day,night,schedule,eventTrigger,videoAuto,IrcutFilterType,darkFighterX,darkFighterXAuto,darkFighterXSchedule,colorVuAuto">
    <!--ro, opt, enum, subType:string, attr:opt{req, string}-->auto
</IrcutFilterType>
<nightToDayFilterLevel opt="low,normal,high">
    <!--ro, opt, string, attr:opt{req, string}-->test
</nightToDayFilterLevel>
<nightToDayFilterTime min="0" max="10">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
</nightToDayFilterTime>
<Schedule>
    <!--ro, opt, object-->
    <scheduleType opt="day,night">
        <!--ro, req, enum, subType:string, attr:opt{req, string}-->day
    </scheduleType>
    <TimeRange>
        <!--ro, req, object, time period-->
        <beginTime>
            <!--ro, req, time, start time-->00:00:00+08:00
        </beginTime>
        <endTime>
            <!--ro, req, time, end time-->00:00:00+08:00
        </endTime>
    </TimeRange>
    </Schedule>
</IrcutFilter>
<PTZ>
    <!--ro, opt, object, PTZ-->
    <enabled opt="true,false">
        <!--ro, opt, bool, whether to enable the function, attr:opt{req, string}-->true
    </enabled>
</PTZ>
<HLC>
    <!--ro, opt, object-->
    <enabled opt="true,false">
        <!--ro, req, bool, whether to enable the function, attr:opt{req, string}-->true
    </enabled>
    <HLCLevel min="0" max="10">
        <!--ro, opt, int, attr:min{req, int},max{req, int}-->0
    </HLCLevel>
</HLC>
<corridor>
    <!--ro, opt, object-->
    <enabled opt="true,false">
        <!--ro, req, bool, whether to enable the function, attr:opt{req, string}-->true
    </enabled>
</corridor>
<SupplementLight>
    <!--ro, opt, object-->
    <supplementLightMode
opt="mixed,whiteLight,colorVuWhiteLight,irLight,dualLight,close,eventIntelligence,multiAzimuthMixed,multiAzimuthWhiteLight,multiAzimuthIrLight,panoramicWhiteLight,panoramicIrLight">
        <!--ro, opt, enum, mode of supplement light, subType:string, attr:opt{req, string}-->mixed
    </supplementLightMode>
    <irLightBrightness min="0" max="100">
        <!--ro, opt, int, brightness of IR supplement light, range:[0,100], attr:min{req, int},max{req, int}, desc:this node is valid when supplementLightMode is set to "mixed" and mixedLightBrightnessRegulatMode is set to "manual"-->80
    </irLightBrightness>
    <mixedLightBrightnessRegulatMode opt="manual,auto">
        <!--ro, opt, enum, brightness adjustment mode of mixed supplement light, subType:string, attr:opt{req, string}, desc:"manual" (in this mode, you need to set the parameters irlightBrightness and brightnessLimit), "auto" (brightness-related parameters are not required). This node is valid when supplementLightMode is set to "mixed"-->auto
    </mixedLightBrightnessRegulatMode>
    <whiteLightBrightness min="0" max="100">
        <!--ro, opt, int, Light brightness (regardless of far or near light), range:[0,100], attr:min{req, int},max{req, int}, desc:it is shared by different supplement light modes, including "mixed", "whiteLight", "colorVuWhiteLight", "irLight", and "dualLight", when the adjustment mode of the supplement light is "auto"-->80
    </whiteLightBrightness>
    <isAutoModeBrightnessCfg>
        <!--ro, opt, bool-->true
    </isAutoModeBrightnessCfg>
    <SupplementLight>
    <isSupportLaserSpotManual>
        <!--ro, opt, bool-->true
    </isSupportLaserSpotManual>
</ImageChannel>

```

### 16.2.5.5 Set the image adjustment parameters in auto mode of a specific channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/color

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Color xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{opt, string, protocolVersion}-->
  <brightnessLevel>
    <!--opt, int, brightness-->24
  </brightnessLevel>
  <contrastLevel>
    <!--opt, int, contrast-->22
  </contrastLevel>
  <saturationLevel>
    <!--opt, int, saturation, dep:and,{$.Color.nightMode,eq,true}-->33
  </saturationLevel>
  <grayScale>
    <!--opt, object, gray scale-->
    <grayScaleMode>
      <!--opt, enum, gray scale mode, subType:string, desc:"indoor", "outdoor"-->outdoor
    </grayScaleMode>
  </grayScale>
</Color>
```

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->/ISAPI/xxxx
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.5.6 Set the image adjustment (in day mode) parameters of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/color/day

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Color xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <brightnessLevel>
    <!--opt, int-->24
  </brightnessLevel>
  <contrastLevel>
    <!--opt, int-->22
  </contrastLevel>
  <saturationLevel>
    <!--opt, int, dep:and, {$Color.nightMode, eq, true}-->33
  </saturationLevel>
  <grayScale>
    <!--opt, object-->
    <grayScaleMode>
      <!--opt, enum, subType:string-->indoor
    </grayScaleMode>
  </grayScale>
</Color>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->0K
  </subStatusCode>
</ResponseStatus>

```

### 16.2.5.7 Get the image adjustment (in day mode) parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/color/day

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Color xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <brightnessLevel>
    <!--ro, opt, int-->24
  </brightnessLevel>
  <contrastLevel>
    <!--ro, opt, int-->22
  </contrastLevel>
  <saturationLevel>
    <!--ro, opt, int, dep:and, {$Color.nightMode, eq, true}-->33
  </saturationLevel>
  <grayScale>
    <!--ro, opt, object-->
    <grayScaleMode>
      <!--ro, opt, enum, subType:string-->indoor
    </grayScaleMode>
  </grayScale>
</Color>

```

### 16.2.5.8 Get the image adjustment (in night mode) parameters of a specified channel

## Request URL

GET /ISAPI/Image/channels/<channelID>/color/night

## Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Color xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <brightnessLevel>
    <!--ro, opt, int-->24
  </brightnessLevel>
  <contrastLevel>
    <!--ro, opt, int-->22
  </contrastLevel>
  <saturationLevel>
    <!--ro, opt, int, dep:and,{$.Color.nightMode,eq,true}-->33
  </saturationLevel>
  <grayScale>
    <!--ro, opt, object-->
    <grayScaleMode>
      <!--ro, opt, enum, subType:string-->indoor
    </grayScaleMode>
  </grayScale>
</Color>
```

## 16.2.5.9 Set the image adjustment (in night mode) parameters of a specified channel

### Request URL

PUT /ISAPI/Image/channels/<channelID>/color/night

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Color xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <brightnessLevel>
    <!--opt, int-->24
  </brightnessLevel>
  <contrastLevel>
    <!--opt, int-->22
  </contrastLevel>
  <saturationLevel>
    <!--opt, int, dep:and,{$.Color.nightMode,eq,true}-->33
  </saturationLevel>
  <grayScale>
    <!--opt, object-->
    <grayScaleMode>
      <!--opt, enum, subType:string-->indoor
    </grayScaleMode>
  </grayScale>
</Color>
```

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.5.10 Get the image adjustment parameters in auto mode of a specific channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/color?parameterType=<parameterType>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Color xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{opt, string, protocolVersion}-->
  <brightnessLevel>
    <!--ro, opt, int, brightness-->24
  </brightnessLevel>
  <contrastLevel>
    <!--ro, opt, int, contrast-->22
  </contrastLevel>
  <saturationLevel>
    <!--ro, opt, int, saturation, dep:and, {$ .Color.nightMode, eq, true}-->33
  </saturationLevel>
  <grayScale>
    <!--ro, opt, object-->
    <grayScaleMode>
      <!--ro, opt, enum, gray scale mode, subType:string, desc:"indoor", "outdoor"-->outdoor
    </grayScaleMode>
  </grayScale>
</Color>

```

### 16.2.5.11 Set the image general mode parameters of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/common

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	video channel No.

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, image signal processing parameters, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, int, ID-->1
  </id>
  <enabled>
    <!--req, bool, whether to enable-->true
  </enabled>
  <videoInputID>
    <!--req, int, video channel ID-->1
  </videoInputID>
  <Imageflip>
    <!--opt, object, image flipping parameters-->
    <enabled>
      <!--req, bool, whether to enable-->true
    </enabled>
  </Imageflip>
  <Exposure>
    <!--opt, object, exposure mode-->
    <ExposureType>
      <!--req, enum, lens type, subType:string, desc:"auto", "IrisFirst", "ShutterFirst", "manual", "pIris", "T5280-PQ1", "T5289-PQ1", "T1140-PQ1", "T2712-PQ1", "HV1250P-MPIR", "pc", "pIris-General", "H21135P-8MPIRA", "H21135P-8MPSGA", "MV1550P-12MPIR", "MZ1545P-12MP", "H20618P-4MPMSG", "MZ1050P-12MP"-->auto
    </ExposureType>
    <OverexposeSuppress>
      <!--opt, object, overexposure suppression-->
      <enabled>
        <!--req, bool, whether to enable-->true
      </enabled>
    </OverexposeSuppress>
  </Exposure>
  <powerLineFrequency>
    <!--opt, object, video standard-->
    <powerLineFrequencyMode>
      <!--opt, enum, video standard mode, subType:string, desc:"50hz" (50 HZ), "60hz" (60 HZ)-->50hz
    </powerLineFrequencyMode>
  </powerLineFrequency>
  <Color>
    <!--opt, object, color-->
    <saturationLevel>
      <!--opt, int, saturation, range:[0,100]-->0
    </saturationLevel>
  </Color>
  <IrcutFilter>
    <!--opt, object, IR cut filter (ICR) configurations-->
    <IrcutFilterType>
      <!--opt, enum, IR cut filter (ICR) type, subType:string, desc:"auto", "day", "night", "schedule", "eventTrigger", "videoAuto", "darkFighterX", "darkFighterXAuto", "darkFighterXSchedule"-->auto
    </IrcutFilterType>
    <nightToDayFilterLevel>
      <!--opt, enum, level of switching from night to day, subType:string, desc:"Low", "normal", "high"-->low
    </nightToDayFilterLevel>
    <nightToDayFilterTime>
      <!--opt, int, time of switching from night to day, range:[0,10]-->1
    </nightToDayFilterTime>
    <EventTrigger>
      <!--opt, object, event linkage parameters-->
      <eventType>
        <!--req, enum, event type, subType:string, desc:"IO", "VMD"-->IO
      </eventType>
      <IrcutFilterAction>
        <!--req, enum, IR cut filter (ICR) operation, subType:string, desc:"day", "night"-->day
      </IrcutFilterAction>
    </EventTrigger>
  </IrcutFilter>
  <corridor>
    <!--opt, object, Lens corridor mode-->
    <enabled>
      <!--req, bool, whether to enable-->true
    </enabled>
  </corridor>
  <CaptureMode>
    <!--opt, object, capture mode-->
    <mode>
      <!--req, enum, mode, subType:string, desc:"close", "1920*1080@30fps"-->close
    </mode>
  </CaptureMode>
</ImageChannel>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.5.12 Get the image general mode parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/common

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	video channel No.

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, image signal processing parameters, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, int, ID-->1
  </id>
  <enabled>
    <!--ro, req, bool, whether to enable-->true
  </enabled>
  <videoInputID>
    <!--ro, req, int, video channel ID-->1
  </videoInputID>
  <ImageFlip>
    <!--ro, opt, object, image flipping parameters-->
    <enabled>
      <!--ro, req, bool, whether to enable-->true
    </enabled>
  </ImageFlip>
  <Exposure>
    <!--ro, opt, object, exposure mode-->
    <ExposureType>
      <!--ro, req, enum, Lens type, subType:string, desc:"auto", "IrisFirst", "ShutterFirst", "gainFirst", "manual", "pIris", "T5280-PQ1", "T5289-PQ1", "T1140-PQ1", "T2712-PQ1", "HV1250P-MPIR", "pc", "pIris-General", "HZ1140P-8MPIRA", "HZ1135P-8MPSGA", "MV1550P-12MPIR", "MZ1545P-12MP", "HZ0618P-4MPG", "MZ1050P-12MP"-->auto
    </ExposureType>
    <OverexposeSuppress>
      <!--ro, opt, object, overexposure suppression-->
      <enabled>
        <!--ro, req, bool, whether to enable-->true
      </enabled>
    </OverexposeSuppress>
  </Exposure>
  <powerLineFrequency>
    <!--ro, opt, object, video standard-->
    <powerLineFrequencyMode>
      <!--ro, opt, enum, video standard mode, subType:string, desc:"50hz" (50 Hz), "60hz" (60 Hz)-->50hz
    </powerLineFrequencyMode>
  </powerLineFrequency>
  <Color>
    <!--ro, opt, object, color-->
    <saturationLevel>
      <!--ro, opt, int, saturation, range:[0,100]-->0
    </saturationLevel>
  </Color>
  <IrcutFilter>
    <!--ro, opt, object, IR cut filter (ICR) configurations-->
    <IrcutFilterType>
      <!--ro, opt, enum, IR cut filter (ICR) type, subType:string, desc:"auto", "day", "night", "schedule", "eventTrigger", "videoAuto", "darkFighterX", "darkFighterXAuto", "darkFighterXSchedule"-->auto
    </IrcutFilterType>
    <nightToDayFilterLevel>
      <!--ro, opt, enum, Level of switching from night to day, subType:string, desc:"Low", "normal", "high"-->low
    </nightToDayFilterLevel>
    <nightToDayFilterTime>
      <!--ro, opt, int, time of switching from night to day, range:[0,10]-->1
    </nightToDayFilterTime>
    <EventTrigger>
      <!--ro, opt, object, event linkage parameters-->
      <eventType>
        <!--ro, req, enum, event type, subType:string, desc:"IO", "VMD"-->IO
      </eventType>
      <IrcutFilterAction>
        <!--ro, req, enum, IR cut filter (ICR) operation, subType:string, desc:"day", "night"-->day
      </IrcutFilterAction>
    </EventTrigger>
  </IrcutFilter>
  <corridor>
    <!--ro, opt, object, lens corridor mode-->
    <enabled>
      <!--ro, req, bool, whether to enable-->true
    </enabled>
  </corridor>
  <CaptureMode>
    <!--ro, opt, object, capture mode-->
    <mode>
      <!--ro, req, enum, mode, subType:string, desc:"close", "1920*1080@30fps"-->close
    </mode>
  </CaptureMode>
</ImageChannel>

```

### 16.2.5.13 Get the capability of configuring image general mode parameters

#### Request URL

GET /ISAPI/Image/channels/<channelID>/common/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

None

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, int-->1
  </id>
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
  <videoInputID>
    <!--ro, req, int-->1
  </videoInputID>
  <ImageFlip>
    <!--ro, opt, object-->
    <enabled opt="true,false">
      <!--ro, req, bool, attr:opt{req, string}-->true
    </enabled>
    <ImageFlipStyle opt="LEFTRIGHT,UPDOWN,CENTER,AUTO">
      <!--ro, opt, string, attr:opt{req, string}-->LEFTRIGHT
    </ImageFlipStyle>
  </ImageFlip>
  <Exposure>
    <!--ro, opt, object-->
    <ExposureType opt="auto,IrisFirst,ShutterFirst,gainFirst,manual,pIris,T5280-PQ1,T5289-PQ1,T1140-PQ1,T2712-PQ1,HV1250P-MPIR,pc,pIris-General,HZ1140P-8MPIRA,HZ1135P-8MPSGA,MV1550P-12MPIR,MZ1545P-12MP,HZ0618P-4MPSG,MZ1050P-12MP">
      <!--ro, req, string, attr:opt{req, string}-->test
    </ExposureType>
    <OverexposeSuppress>
      <!--ro, opt, object-->
      <enabled opt="true,false">
        <!--ro, req, bool, attr:opt{req, string}-->true
      </enabled>
    </OverexposeSuppress>
  </Exposure>
  <powerLineFrequency>
    <!--ro, opt, object-->
    <powerLineFrequencyMode opt="50hz,60hz">
      <!--ro, opt, string, attr:opt{req, string}-->test
    </powerLineFrequencyMode>
  </powerLineFrequency>
  <Color>
    <!--ro, opt, object-->
  </Color>
  <IrcutFilter>
    <!--ro, opt, object-->
    <IrcutFilterType opt="auto,day,night,schedule,eventTrigger,videoAuto,IrcutFilterType,darkFighterX,darkFighterXAuto,darkFighterXSchedule,colorVuAuto">
      <!--ro, opt, enum, subType:string, attr:opt{req, string}-->auto
    </IrcutFilterType>
    <nightToDayFilterLevel opt="low,normal,high">
      <!--ro, opt, string, attr:opt{req, string}-->test
    </nightToDayFilterLevel>
    <nightToDayFilterTime min="0" max="10">
      <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
    </nightToDayFilterTime>
    <Schedule>
      <!--ro, opt, object-->
      <scheduleType opt="day,night">
        <!--ro, req, enum, subType:string, attr:opt{req, string}-->day
      </scheduleType>
      <TimeRange>
        <!--ro, req, object-->
        <beginTime>
          <!--ro, req, time-->00:00:00+08:00
        </beginTime>
        <endTime>
          <!--ro, req, time-->00:00:00+08:00
        </endTime>
      </TimeRange>
    </Schedule>
  </IrcutFilter>
  <corridor>
    <!--ro, opt, object-->
    <enabled opt="true,false">
      <!--ro, req, bool, attr:opt{req, string}-->true
    </enabled>
  </corridor>
  <SupplementLight>
    <!--ro, opt, object-->
    <supplementLightMode
      <!--mixed white light color white light in light dual light close event to all license multi timer mixed multi timer white light multi timer white light non normal light
      &nt>"mixed white light color white light in light dual light close event to all license multi timer mixed multi timer white light multi timer white light non normal light
      &lt;/supplementLightMode>
  </SupplementLight>
</ImageChannel>

```

```

opt=mixed,whiteLight,colorWarmWhiteLight,infrared,audibleLight,close,eventIntelligence,multizoommixed,multizoomwhiteLight,multizoominfrared,persondetection
eLight,panoramicIrLight">
    <!--ro, opt, enum, subType:string, attr:opt{req, string}-->mixed
</supplementLightMode>
<irLightBrightness min="0" max="100">
    <!--ro, opt, int, range:[0,100], attr:min{req, int},max{req, int}-->80
</irLightBrightness>
<mixedLightBrightnessRegulatMode opt="manual,auto">
    <!--ro, opt, enum, subType:string, attr:opt{req, string}-->auto
</mixedLightBrightnessRegulatMode>
<whiteLightBrightness min="0" max="100">
    <!--ro, opt, int, range:[0,100], attr:min{req, int},max{req, int}-->80
</whiteLightBrightness>
<isAutoModeBrightnessCfg>
    <!--ro, opt, bool-->true
</isAutoModeBrightnessCfg>
</SupplementLight>
</ImageChannel>

```

### 16.2.5.14 Get the day mode of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/day

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	video channel No.

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, image signal processing parameters, attr:version{req, string, protocolVersion}-->
    <id>
        <!--ro, req, int, ID-->1
    </id>
    <enabled>
        <!--ro, req, bool, whether to enable-->true
    </enabled>
    <videoInputID>
        <!--ro, req, int, video channel ID-->1
    </videoInputID>
    <HDR>
        <!--ro, opt, object, wide dynamic range (WDR)-->
        <mode>
            <!--ro, req, enum, subType:string, desc:"open" (enable), "close" (disable), "auto" (automatic), "SDR"-->open
        </mode>
        <WDRLevel>
            <!--ro, opt, int, WDR Level, range:[1,100]-->1
        </WDRLevel>
    </HDR>
    <BLC>
        <!--ro, opt, object, BLC area-->
        <enabled>
            <!--ro, req, bool, whether to enable-->true
        </enabled>
        <BLCRegionList>
            <!--ro, opt, array, List of BLC areas, subType:object-->
            <BLCRegion>
                <!--ro, opt, object, BLC area-->
                <id>
                    <!--ro, req, int, area No., range:[1,10]-->1
                </id>
                <RegionCoordinatesList>
                    <!--ro, req, array, coordinate List, subType:object, range:[0,4], desc:rectangle-->
                    <RegionCoordinates>
                        <!--ro, opt, object, coordinates, desc:the origin is the Lower-Left corner of the screen-->
                        <positionX>
                            <!--ro, req, int, x-coordinate, range:[0,704]-->1
                        </positionX>
                        <positionY>
                            <!--ro, req, int, y-coordinate, range:[0,576]-->1
                        </positionY>
                    </RegionCoordinates>
                </RegionCoordinatesList>
            </BLCRegion>
        </BLCRegionList>
    </BLC>
    <NoiseReduce>
        <!--ro, opt, object, noise reduction parameters-->
    
```

```

<mode>
  <!--ro, req, enum, mode, subType:string, desc:"close", "general", "advanced"-->close
</mode>
<GeneralMode>
  <!--ro, opt, object, general mode-->
  <generalLevel>
    <!--ro, req, int, Level, range:[0,100]-->1
  </generalLevel>
</GeneralMode>
</NoiseReduce>
<WhiteBalance>
  <!--ro, opt, object, white balance-->
  <WhiteBalanceStyle>
    <!--ro, opt, enum, white balance type, subType:string, desc:"auto" (automatic), "auto1" (automatic 1), "auto2" (automatic 2), "manual", "indoor", "outdoor", "autotrace" (automatic tracking), "once" (one time), "sodiumlight" (sodium Light), "mercurylight" (mercury Light), "customColorTemperature" (custom color temperature)-->auto
  </WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--ro, opt, int, white balance red, range:[0,100]-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--ro, opt, int, white balance blue, range:[0,100]-->1
  </WhiteBalanceBlue>
</WhiteBalance>
<Sharpness>
  <!--ro, opt, object, sharpness-->
  <SharpnessLevel>
    <!--ro, req, int, sharpness level, range:[0,100]-->1
  </SharpnessLevel>
</Sharpness>
<Color>
  <!--ro, opt, object, color-->
  <brightnessLevel>
    <!--ro, opt, int, brightness, range:[0,100]-->0
  </brightnessLevel>
  <contrastLevel>
    <!--ro, opt, int, contrast, range:[0,100]-->0
  </contrastLevel>
  <saturationLevel>
    <!--ro, opt, int, saturation, range:[0,100]-->0
  </saturationLevel>
  <grayScale>
    <!--ro, opt, object, gray scale-->
    <grayScaleMode>
      <!--ro, opt, enum, gray scale mode, subType:string, desc:"indoor", "outdoor"-->indoor
    </grayScaleMode>
  </grayScale>
</Color>
<HLC>
  <!--ro, opt, object, HLC-->
  <enabled>
    <!--ro, req, bool, whether to enable-->true
  </enabled>
  <HLCLevel>
    <!--ro, opt, int, HLC Level, range:[0,10]-->0
  </HLCLevel>
</HLC>
<Dehaze>
  <!--ro, opt, object, defog-->
  <DehazeMode>
    <!--ro, opt, enum, defog mode, subType:string, desc:"open", "close", "auto"-->open
  </DehazeMode>
</Dehaze>
</ImageChannel>

```

### 16.2.5.15 Set the day mode of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/day

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	video channel No.

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, image signal processing parameters, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, int, ID-->1
  </id>
  <enabled>
    <!--req, bool, whether to enable-->true
  </enabled>

```

```

</enabled>
<videoInputID>
    <!--req, int, video channel ID-->1
</videoInputID>
<WDR>
    <!--opt, object, wide dynamic range (WDR)-->
    <mode>
        <!--req, enum, subType:string, desc:"open" (enable), "close" (disable), "auto" (automatic), "SDR"-->open
    </mode>
    <WDRLevel>
        <!--opt, int, WDR Level, range:[1,100]-->1
    </WDRLevel>
</WDR>
<BLC>
    <!--opt, object, BLC area-->
    <enabled>
        <!--req, bool, whether to enable-->true
    </enabled>
    <BLCRegionList>
        <!--opt, array, List of BLC areas, subType:object-->
        <BLCRegion>
            <!--opt, object, BLC area-->
            <id>
                <!--req, int, area No., range:[1,10]-->1
            </id>
            <RegionCoordinatesList>
                <!--req, array, coordinate List, subType:object, range:[0,4], desc:rectangle-->
                <RegionCoordinates>
                    <!--opt, object, coordinates, desc:the origin is the lower-left corner of the screen-->
                    <positionX>
                        <!--req, int, x-coordinate, range:[0,704]-->1
                    </positionX>
                    <positionY>
                        <!--req, int, y-coordinate, range:[0,576]-->1
                    </positionY>
                </RegionCoordinates>
            </RegionCoordinatesList>
        </BLCRegion>
    </BLCRegionList>
</BLC>
<NoiseReduce>
    <!--opt, object, noise reduction parameters-->
    <mode>
        <!--req, enum, mode, subType:string, desc:"close", "general", "advanced"-->close
    </mode>
    <GeneralMode>
        <!--opt, object, mode-->
        <generalLevel>
            <!--req, int, Level, range:[0,100]-->1
        </generalLevel>
    </GeneralMode>
</NoiseReduce>
<WhiteBalance>
    <!--opt, object, white balance-->
    <WhiteBalanceStyle>
        <!--opt, enum, white balance type, subType:string, desc:"auto" (automatic), "auto1" (automatic 1), "auto2" (automatic 2), "manual", "indoor", "outdoor", "autotrace" (automatic tracking), "once" (one time), "sodiumlight" (sodium Light), "mercurylight" (mercury Light), "customColorTemperature" (custom color temperature)-->auto
    </WhiteBalanceStyle>
    <WhiteBalanceRed>
        <!--opt, int, white balance red, range:[0,100]-->1
    </WhiteBalanceRed>
    <WhiteBalanceBlue>
        <!--opt, int, white balance blue, range:[0,100]-->1
    </WhiteBalanceBlue>
</WhiteBalance>
<Sharpness>
    <!--opt, object, sharpness-->
    <SharpnessLevel>
        <!--req, int, sharpness Level, range:[0,100]-->1
    </SharpnessLevel>
</Sharpness>
<Color>
    <!--opt, object, color-->
    <brightnessLevel>
        <!--opt, int, brightness, range:[0,100]-->0
    </brightnessLevel>
    <contrastLevel>
        <!--opt, int, contrast, range:[0,100]-->0
    </contrastLevel>
    <saturationLevel>
        <!--opt, int, saturation, range:[0,100]-->0
    </saturationLevel>
    <grayScale>
        <!--opt, object, gray scale-->
        <grayScaleMode>
            <!--opt, enum, gray scale mode, subType:string, desc:"indoor", "outdoor"-->indoor
        </grayScaleMode>
    </grayScale>
</Color>
<HLC>
    <!--opt, object, HLC-->
    <enabled>
        <!--req, bool, whether to enable-->true
    </enabled>

```

```

</enabled>
<HLCLevel>
  <!--opt, int, HLC Level, range:[0,10]-->0
</HLCLevel>
</HLC>
<Dehaze>
  <!--opt, object, defog-->
<DehazeMode>
  <!--opt, enum, defog mode, subType:string, desc:"open", "close", "auto"-->open
</DehazeMode>
</Dehaze>
</ImageChannel>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device_Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.5.16 Get the day mode capability

#### Request URL

GET /ISAPI/Image/channels/<channelID>/day/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, int-->1
  </id>
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
  <videoInputID>
    <!--ro, req, int-->1
  </videoInputID>
  <WDR>
    <!--ro, opt, object-->
    <mode opt="open,close,auto,SDR">
      <!--ro, req, enum, subType:string, attr:opt{req, string}-->open
    </mode>
    <WDRLevel min="1" max="100">
      <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
    </WDRLevel>
  </WDR>
  <BLC>
    <!--ro, opt, object-->
    <enabled opt="true,false">
      <!--ro, req, bool, attr:opt{req, string}-->true
    </enabled>
    <BLCMode opt="UP,DOWN,LEFT,RIGHT,CENTER,MULTI-AREA,Region,AUTO">
      <!--ro, opt, enum, subType:string, attr:opt{req, string}-->UP
    </BLCMode>
  </BLC>
  <NoiseReduce>

```

```

<!--ro, opt, object-->
<mode opt="close,general,advanced">
  <!--ro, req, enum, subType:string, attr:opt{req, string}-->close
</mode>
<GeneralMode>
  <!--ro, opt, object-->
  <generalLevel min="0" max="10">
    <!--ro, req, int, attr:min{req, int},max{req, int}-->1
  </generalLevel>
</GeneralMode>
<AdvancedMode>
  <!--ro, opt, object-->
  <FrameNoiseReduceLevel min="0" max="10">
    <!--ro, req, int, attr:min{req, int},max{req, int}-->1
  </FrameNoiseReduceLevel>
  <InterFrameNoiseReduceLevel min="0" max="10">
    <!--ro, req, int, attr:min{req, int},max{req, int}-->1
  </InterFrameNoiseReduceLevel>
</AdvancedMode>
</NoiseReduce>
<WhiteBalance>
  <!--ro, opt, object-->
  <WhiteBalanceStyle opt="manual, auto1, auto2, locked, outdoor, indoor, daylightLamp, sodiumlight, autotrace, onece, autooutdoor, autosodiumlight, mercurylight, auto, incandescentlight, warmlight, naturallight">
    <!--ro, req, string, attr:opt{req, string}-->test
  </WhiteBalanceStyle>
  <WhiteBalanceRed min="0" max="10">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue min="0" max="10">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
  </WhiteBalanceBlue>
</WhiteBalance>
<Sharpness>
  <!--ro, opt, object-->
  <SharpnessLevel min="0" max="100">
    <!--ro, req, int, attr:min{req, int},max{req, int}-->1
  </SharpnessLevel>
</Sharpness>
<Shutter>
  <!--ro, opt, object-->
  <ShutterLevel opt="1/25,1/50,1/75,1/100,1/125,1/150,1/175,1/200,1/225,1/250,1/300,1/400,1/500,1/750,1/1000,1/2000,1/4000,1/10000,1/100000">
    <!--ro, opt, string, attr:opt{req, string}-->test
  </ShutterLevel>
</Shutter>
<Color>
  <!--ro, opt, object-->
  <brightnessLevel min="0" max="10">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->0
  </brightnessLevel>
  <contrastLevel min="0" max="10">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->0
  </contrastLevel>
  <saturationLevel min="0" max="10">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->0
  </saturationLevel>
  <grayScale>
    <!--ro, opt, object-->
    <grayScaleMode opt="indoor,outdoor">
      <!--ro, opt, string, attr:opt{req, string}-->test
    </grayScaleMode>
  </grayScale>
</Color>
<HLC>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <HLClevel min="0" max="10">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->0
  </HLClevel>
</HLC>
</ImageChannel>

```

### 16.2.5.17 Get day/night auto switch parameters of a specific channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/IrcutFilter

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<IrcutFilter xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <IrcutFilterType>
    <!--ro, opt, enum, day/night auto switch mode, subType:string, desc:"auto, day, night, schedule, eventTrigger, darkFighterX, darkFighterXAuto, darkFighterXSchedule"-->auto
    </IrcutFilterType>
    <nightToDayFilterLevel>
      <!--ro, opt, enum, Level of switching night to day, subType:string, desc:"Low, normal, high"-->low
    </nightToDayFilterLevel>
    <nightToDayFilterTime>
      <!--ro, opt, int, time interval of switching night to day-->1
    </nightToDayFilterTime>
    <EventTrigger>
      <!--ro, opt, object-->
      <eventType>
        <!--ro, req, enum, event type, subType:string, desc:event type-->IO
      </eventType>
      <IrcutFilterAction>
        <!--ro, req, enum, "day, night", subType:string, desc:"day", "night"-->day
      </IrcutFilterAction>
    </EventTrigger>
  </IrcutFilter>
```

## 16.2.5.18 Set day/night auto switch parameters of a specific channel

### Request URL

PUT /ISAPI/Image/channels/<channelID>/IrcutFilter

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<IrcutFilter xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, IrcutFilter, attr:version{req, string, protocolVersion}-->
  <IrcutFilterType>
    <!--opt, enum, IrcutFilter type, subType:string, desc:"auto", "day", "night", "schedule" , "eventTrigger", "videoAuto", "darkFighterX", "darkFighterXAuto", "darkFighterXSchedule", "colorVuAuto"-->auto
    </IrcutFilterType>
    <nightToDayFilterLevel>
      <!--opt, enum, Level of switching night to day, subType:string, desc:"Low", "normal", "high"-->low
    </nightToDayFilterLevel>
    <nightToDayFilterTime>
      <!--opt, int, time interval of switching night to day-->1
    </nightToDayFilterTime>
    <EventTrigger>
      <!--opt, object, event trigger-->
      <eventType>
        <!--req, enum, event type, subType:string, desc:"IO", "VID"-->IO
      </eventType>
      <IrcutFilterAction>
        <!--req, enum, "day,night", subType:string, desc:"day", "night"-->day
      </IrcutFilterAction>
    </EventTrigger>
  </IrcutFilter>
```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0-OK, 1-OK, 2-Device Busy, 3-Device Error, 4-Invalid Operation, 5-Invalid XML Format, 6-Invalid XML Content, 7-Reboot Required-->0
    <statusCode>
      <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, desc:sub status code-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, desc:sub status code-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

### 16.2.5.19 Get the night mode parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, int-->1
  </id>
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
  <videoInputID>
    <!--ro, req, int-->1
  </videoInputID>
  <WDR>
    <!--ro, opt, object-->
    <mode>
      <!--ro, req, enum, subType:string-->open
    </mode>
    <WDRLevel>
      <!--ro, opt, int, range:[1,100]-->1
    </WDRLevel>
  </WDR>
  <BLC>
    <!--ro, opt, object-->
    <enabled>
      <!--ro, req, bool-->true
    </enabled>
    <BLRegionList>
      <!--ro, opt, array, subType:object-->
      <BLRegion>
        <!--ro, opt, object-->
        <id>
          <!--ro, req, int, range:[1,10]-->1
        </id>
        <RegionCoordinatesList>
          <!--ro, req, array, subType:object, range:[0,4]-->
          <RegionCoordinates>
            <!--ro, opt, object-->
            <positionX>
              <!--ro, req, int, range:[0,704]-->1
            </positionX>
            <positionY>
              <!--ro, req, int, range:[0,576]-->1
            </positionY>
          </RegionCoordinates>
        </RegionCoordinatesList>
      </BLRegion>
    </BLRegionList>
  </BLC>
</ImageChannel>

```

```

</BLC>
<NoiseReduce>
<!--ro, opt, object-->
<mode>
  <!--ro, req, enum, subType:string-->close
</mode>
<GeneralMode>
  <!--ro, opt, object-->
  <generalLevel>
    <!--ro, req, int, range:[0,100]-->1
  </generalLevel>
</GeneralMode>
</NoiseReduce>
<WhiteBalance>
  <!--ro, opt, object-->
  <WhiteBalanceStyle>
    <!--ro, opt, enum, subType:string-->auto
  </WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--ro, opt, int, range:[0,100]-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--ro, opt, int, range:[0,100]-->1
  </WhiteBalanceBlue>
</WhiteBalance>
<Sharpness>
  <!--ro, opt, object-->
  <SharpnessLevel>
    <!--ro, req, int, range:[0,100]-->1
  </SharpnessLevel>
</Sharpness>
<Color>
  <!--ro, opt, object-->
  <brightnessLevel>
    <!--ro, opt, int, range:[0,100]-->0
  </brightnessLevel>
  <contrastLevel>
    <!--ro, opt, int, range:[0,100]-->0
  </contrastLevel>
  <saturationLevel>
    <!--ro, opt, int, range:[0,100]-->0
  </saturationLevel>
  <grayScale>
    <!--ro, opt, object-->
    <grayScaleMode>
      <!--ro, opt, enum, subType:string-->indoor
    </grayScaleMode>
  </grayScale>
</Color>
<HLC>
  <!--ro, opt, object-->
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
  <HLCLevel>
    <!--ro, opt, int, range:[0,10]-->0
  </HLCLevel>
</HLC>
<Dehaze>
  <!--ro, opt, object-->
  <DehazeMode>
    <!--ro, opt, enum, subType:string-->open
  </DehazeMode>
</Dehaze>
</ImageChannel>

```

### 16.2.5.20 Set the night mode parameters of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, int-->1
  </id>
  <enabled>
    <!--req, bool-->true
  </enabled>

```

```

<!--req, pool-->true
</enabled>
<videoInputID>
    <!--req, int-->1
</videoInputID>
<WDR>
    <!--opt, object-->
    <mode>
        <!--req, enum, subType:string-->open
    </mode>
    <WDRLevel>
        <!--opt, int, range:[1,100]-->1
    </WDRLevel>
</WDR>
<BLC>
    <!--opt, object-->
    <enabled>
        <!--req, bool-->true
    </enabled>
    <BLCRegionList>
        <!--opt, array, subType:object-->
        <BLCRegion>
            <!--opt, object-->
            <id>
                <!--req, int, range:[1,10]-->1
            </id>
            <RegionCoordinatesList>
                <!--req, array, subType:object, range:[0,4]-->
                <RegionCoordinates>
                    <!--opt, object-->
                    <positionX>
                        <!--req, int, range:[0,704]-->1
                    </positionX>
                    <positionY>
                        <!--req, int, range:[0,576]-->1
                    </positionY>
                </RegionCoordinates>
                </RegionCoordinatesList>
            </BLCRegion>
        </BLCRegionList>
    </BLC>
    <NoiseReduce>
        <!--opt, object-->
        <mode>
            <!--req, enum, subType:string-->close
        </mode>
        <GeneralMode>
            <!--opt, object-->
            <generalLevel>
                <!--req, int, range:[0,100]-->1
            </generalLevel>
        </GeneralMode>
    </NoiseReduce>
    <WhiteBalance>
        <!--opt, object-->
        <WhiteBalanceStyle>
            <!--opt, enum, subType:string-->auto
        </WhiteBalanceStyle>
        <WhiteBalanceRed>
            <!--opt, int, range:[0,100]-->1
        </WhiteBalanceRed>
        <WhiteBalanceBlue>
            <!--opt, int, range:[0,100]-->1
        </WhiteBalanceBlue>
    </WhiteBalance>
    <Sharpness>
        <!--opt, object-->
        <SharpnessLevel>
            <!--req, int, range:[0,100]-->1
        </SharpnessLevel>
    </Sharpness>
    <Color>
        <!--opt, object-->
        <brightnessLevel>
            <!--opt, int, range:[0,100]-->0
        </brightnessLevel>
        <contrastLevel>
            <!--opt, int, range:[0,100]-->0
        </contrastLevel>
        <saturationLevel>
            <!--opt, int, range:[0,100]-->0
        </saturationLevel>
    </Color>
    <grayScale>
        <!--opt, object-->
        <grayScaleMode>
            <!--opt, enum, subType:string-->indoor
        </grayScaleMode>
    </grayScale>
</Color>
<HLC>
    <!--opt, object-->
    <enabled>
        <!--req, bool-->true
    </enabled>

```

```

<HLCLevel>
  <!--opt, int, range:[0,10]-->0
</HLCLevel>
</HLC>
<Dehaze>
  <!--opt, object-->
<DehazeMode>
  <!--opt, enum, subType:string-->open
</DehazeMode>
</Dehaze>
</ImageChannel>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.5.21 Get the capability of configuring the night mode parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/night/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, int-->1
  </id>
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
  <videoInputID>
    <!--ro, req, int-->1
  </videoInputID>
  <WDR>
    <!--ro, opt, object-->
    <mode>
      <!--ro, req, enum, subType:string-->auto
    </mode>
    <WDRLevel>
      <!--ro, opt, int, range:[1,100]-->50
    </WDRLevel>
  </WDR>
  <BLC>
    <!--ro, opt, object-->
    <enabled>
      <!--ro, req, bool-->true
    </enabled>
    <BLCMode>
      <!--ro, opt, enum, subType:string-->UP
    </BLCMode>
  </BLC>
  <NoiseReduce>
    <!--ro, opt, object-->
    <mode>
      <!--ro, req, enum, subType:string-->close
    </mode>
    <GeneralMode>
      <!--ro, opt, object-->
      <generalLevel>
        <!--ro, req, int-->1
      </generalLevel>
    </GeneralMode>
    <AdvancedMode>
      <!--ro, opt, object-->
      <FrameNoiseReduceLevel>
        <!--ro, req, int-->1
      </FrameNoiseReduceLevel>
      <InterFrameNoiseReduceLevel>
        <!--ro, req, int-->1
      </InterFrameNoiseReduceLevel>
    </AdvancedMode>
  </NoiseReduce>
  <Sharpness>
    <!--ro, opt, object-->
    <SharpnessLevel>
      <!--ro, req, int-->1
    </SharpnessLevel>
  </Sharpness>
  <Color>
    <!--ro, opt, object-->
    <brightnessLevel>
      <!--ro, opt, int-->24
    </brightnessLevel>
    <contrastLevel>
      <!--ro, opt, int-->22
    </contrastLevel>
    <saturationLevel>
      <!--ro, opt, int, dep:and, ${.Color.nightMode},eq,true-->33
    </saturationLevel>
    <grayScale>
      <!--ro, opt, object-->
      <grayScaleMode>
        <!--ro, opt, enum, subType:string-->outdoor
      </grayScaleMode>
    </grayScale>
  </Color>
  <HLC>
    <!--ro, opt, object-->
    <enabled>
      <!--ro, req, bool-->true
    </enabled>
    <HLCLevel>
      <!--ro, opt, int-->1
    </HLCLevel>
  </HLC>
</ImageChannel>

```

#### 16.2.5.22 Reset the image parameters of a specific channel

**Request URL**

PUT /ISAPI/Image/channels/<channelID>/reset

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:error reason description in detail-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.5.23 Restore the image parameters of a specific channel to default settings

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/restore

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->/ISAPI/xxxx
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.5.24 Set the sharpness control parameters of a specific channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/sharpness

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Sharpness xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, sharpness, attr:version{opt, string, protocolVersion}-->
  <SharpnessLevel>
    <!--req, int, sharpness level-->1
  </SharpnessLevel>
</Sharpness>
```

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->/ISAPI/xxxx
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6
(Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format",
"Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.5.25 Get the sharpness control (in day mode) parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/sharpness/day

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Sharpness xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <SharpnessLevel>
    <!--ro, opt, int-->1
  </SharpnessLevel>
</Sharpness>
```

### 16.2.5.26 Set the sharpness control (in day mode) parameters of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/sharpness/day

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Sharpness xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <SharpnessLevel>
    <!--opt, int-->1
  </SharpnessLevel>
</Sharpness>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->请求的URL
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->test
  </subStatusCode>
</ResponseStatus>
```

## 16.2.5.27 Set the sharpness control (in night mode) parameters of a specified channel

### Request URL

PUT /ISAPI/Image/channels/<channelID>/sharpness/night

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Sharpness xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <SharpnessLevel>
    <!--opt, int-->1
  </SharpnessLevel>
</Sharpness>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->请求的URL
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->test
  </subStatusCode>
</ResponseStatus>
```

## 16.2.5.28 Get the sharpness control (in night mode) parameters of a specified channel

### Request URL

GET /ISAPI/Image/channels/<channelID>/sharpness/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Sharpness xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <SharpnessLevel>
    <!--ro, opt, int-->1
  </SharpnessLevel>
</Sharpness>
```

### 16.2.5.29 Get the sharpness control parameters of a specific channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/sharpness?parameterType=<parameterType>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Sharpness xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, sharpness, attr:version{opt, string, protocolVersion}-->
  <SharpnessLevel>
    <!--ro, req, int, sharpness Level-->1
  </SharpnessLevel>
</Sharpness>
```

### 16.2.5.30 Get the image processing parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>?parameterType=<parameterType>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, int-->1
  </id>
```

```
<!-- ro, req, object -->
</id>
<enabled>
  <!-- ro, req, bool-->true
</enabled>
<videoInputID>
  <!-- ro, req, int-->1
</videoInputID>
<ImageFlip>
  <!-- ro, opt, object-->
<enabled>
  <!-- ro, req, bool-->true
</enabled>
</ImageFlip>
<ImageFreeze>
  <!-- ro, opt, object-->
<enabled>
  <!-- ro, req, bool-->true
</enabled>
</ImageFreeze>
<proportionalpan>
  <!-- ro, opt, object-->
<enabled>
  <!-- ro, req, bool-->true
</enabled>
</proportionalpan>
<WDR>
  <!-- ro, opt, object-->
<mode>
  <!-- ro, req, enum, subType:string-->open
</mode>
<WDRLevel>
  <!-- ro, opt, int, range:[1,100]-->1
</WDRLevel>
</WDR>
<BLC>
<enabled>
  <!-- ro, req, bool-->true
</enabled>
<BLRegionList>
  <!-- ro, opt, array, subType:object-->
<BLRegion>
  <!-- ro, opt, object-->
<id>
  <!-- ro, req, int, range:[1,10]-->1
</id>
<RegionCoordinatesList>
  <!-- ro, req, array, subType:object, range:[0,4]-->
<RegionCoordinates>
  <!-- ro, opt, object-->
<positionX>
  <!-- ro, req, int, range:[0,704]-->1
</positionX>
<positionY>
  <!-- ro, req, int, range:[0,576]-->1
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</BLRegion>
</BLRegionList>
</BLC>
<NoiseReduce>
  <!-- ro, opt, object-->
<mode>
  <!-- ro, req, enum, subType:string-->close
</mode>
<GeneralMode>
  <!-- ro, opt, object-->
<generalLevel>
  <!-- ro, req, int, range:[0,100]-->1
</generalLevel>
</GeneralMode>
</NoiseReduce>
<WhiteBalance>
  <!-- ro, opt, object-->
<WhiteBalanceStyle>
  <!-- ro, opt, enum, subType:string-->auto
</WhiteBalanceStyle>
<WhiteBalanceRed>
  <!-- ro, opt, int, range:[0,100]-->1
</WhiteBalanceRed>
<WhiteBalanceBlue>
  <!-- ro, opt, int, range:[0,100]-->1
</WhiteBalanceBlue>
</WhiteBalance>
<Exposure>
  <!-- ro, opt, object-->
<ExposureType>
  <!-- ro, req, enum, subType:string-->auto
</ExposureType>
<OverexposeSuppress>
  <!-- ro, opt, object-->
<enabled>
```

```

        <!--ro, req, bool-->true
    </enabled>
</OverexposeSuppress>
</Exposure>
<Sharpness>
    <!--ro, opt, object-->
<SharpnessLevel>
    <!--ro, req, int, range:[0,100]-->1
</SharpnessLevel>
</Sharpness>
<Shutter>
    <!--ro, opt, object-->
<ShutterLevel>
    <!--ro, opt, enum, subType:string-->1/1
</ShutterLevel>
</Shutter>
<powerLineFrequency>
    <!--ro, opt, object-->
<powerLineFrequencyMode>
    <!--ro, opt, enum, subType:string-->50hz
</powerLineFrequencyMode>
</powerLineFrequency>
<Gain>
    <!--ro, opt, object-->
<GainLevel>
    <!--ro, opt, string, range:[0,10]-->test
</GainLevel>
</Gain>
<Color>
    <!--ro, opt, object-->
<brightnessLevel>
    <!--ro, opt, int, range:[0,100]-->0
</brightnessLevel>
<contrastLevel>
    <!--ro, opt, int, range:[0,100]-->0
</contrastLevel>
<saturationLevel>
    <!--ro, opt, int, range:[0,100]-->0
</saturationLevel>
<grayScale>
    <!--ro, opt, object-->
<grayScaleMode>
    <!--ro, opt, enum, subType:string-->indoor
</grayScaleMode>
</grayScale>
</Color>
<IrCutFilter>
    <!--ro, opt, object-->
<IrCutFilterType>
    <!--ro, opt, enum, subType:string-->auto
</IrCutFilterType>
<nightToDayFilterLevel>
    <!--ro, opt, enum, subType:string-->low
</nightToDayFilterLevel>
<nightToDayFilterTime>
    <!--ro, opt, int, range:[0,10]-->1
</nightToDayFilterTime>
<EventTrigger>
    <!--ro, opt, object-->
<eventType>
    <!--ro, req, enum, subType:string-->IO
</eventType>
<IrCutFilterAction>
    <!--ro, req, enum, subType:string-->day
</IrCutFilterAction>
</EventTrigger>
</IrCutFilter>
<PTZ>
    <!--ro, opt, object-->
<enabled>
    <!--ro, opt, bool-->true
</enabled>
</PTZ>
<HLC>
    <!--ro, opt, object-->
<enabled>
    <!--ro, req, bool-->true
</enabled>
<HLCLevel>
    <!--ro, opt, int, range:[0,10]-->0
</HLCLevel>
</HLC>
<corridor>
    <!--ro, opt, object-->
<enabled>
    <!--ro, req, bool-->true
</enabled>
</corridor>
<Dehaze>
    <!--ro, opt, object-->
<DehazeMode>
    <!--ro, opt, enum, subType:string-->open
</DehazeMode>
</Dehaze>

```

```

</VideoSize>
<CaptureMode>
<!--ro, opt, object-->
<mode>
<!--ro, req, enum, subType:string-->close
</mode>
</CaptureMode>
<SupplementLight>
<!--ro, opt, object-->
<supplementLightMode>
<!--ro, opt, enum, subType:string-->mixed
</supplementLightMode>
<irLightBrightness>
<!--ro, opt, int, range:[0,100]-->80
</irLightBrightness>
<mixedLightBrightnessRegulatMode>
<!--ro, opt, enum, subType:string-->auto
</mixedLightBrightnessRegulatMode>
<whiteLightBrightness>
<!--ro, opt, int, range:[0,100]-->80
</whiteLightBrightness>
</SupplementLight>
<DigitalZoom>
<!--ro, opt, object-->
<ZoomRatio>
<!--ro, opt, enum, subType:string-->1x
</ZoomRatio>
</DigitalZoom>
<Palettes>
<!--ro, opt, object-->
<mode>
<!--ro, opt, enum, subType:string-->WhiteHot
</mode>
</Palettes>
<LaserLight>
<!--ro, opt, object-->
<mode>
<!--ro, opt, enum, subType:string-->auto
</mode>
</LaserLight>
</ImageChannel>

```

### 16.2.5.31 Get the complete video and image configuration capability of all channels

#### Request URL

GET /ISAPI/Image/channels/capabilities

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
<!--ro, req, object, video and image capability, attr:version{req, string, protocolVersion}-->
<id>
<!--ro, req, int, index-->0
</id>
<enabled>
<!--ro, req, bool, whether to enable-->true
</enabled>
<videoInputID>
<!--ro, req, int, video input channel No.-->0
</videoInputID>
<ImageFlip>
<!--ro, opt, object-->
<enabled opt="true,false">
<!--ro, opt, bool, attr:opt{req, string}-->true
</enabled>
<ImageFlipStyle opt="LEFTRIGHT,UPDOWN,CENTER,AUTO">
<!--ro, opt, enum, subType:string, attr:opt{req, string}-->LEFTRIGHT
</ImageFlipStyle>
</ImageFlip>
<WDR>
<!--ro, opt, object-->
<enabled opt="true,false">
<!--ro, opt, bool, attr:opt{req, string}-->true
</enabled>
<WDRLevel>
<!--ro, opt, int, range:[1,100], attr:opt{req, string}-->&gt; 50
</WDRLevel>

```

opt="0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100">

```

</WDR>
<WDRExt>
<!--ro, opt, object-->
<mode opt="open,close,auto,SDR">
  <!--ro, req, enum, mode, subType:string, attr:opt{req, string}, desc:WDR mode: "open,close,auto"-->auto
</mode>
<WDRLevel min="0" max="100">
  <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
</WDRLevel>
</WDRExt>
<BLC>
<!--ro, opt, object-->
<enabled opt="true,false">
  <!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<BLCMode opt="UP,DOWN,LEFT,RIGHT,CENTER,MULTI-AREA,Region,AUTO">
  <!--ro, opt, enum, subType:string, attr:opt{req, string}-->UP
</BLCMode>
</BLC>
<WhiteBlance>
<!--ro, opt, object, White Balance-->
</WhiteBlance>
<Exposure>
<!--ro, opt, object-->
<ExposureType opt="auto,manual,IrisFirst,ShutterFirst,gainFirst,pIris,T5280-PQ1,T5289-PQ1,T1140-PQ1,T2712-PQ1,HV1250P-MPIR,pc,pIris-General,HZ1140P-8MPIRA,HZ1135P-8MPSGA,MV1550P-12MPIR,MZ1545P-12MP,H20618P-4MPSG,MZ1050P-12MP">
  <!--ro, req, enum, subType:string, attr:opt{req, string}-->auto
</ExposureType>
<OverexposeSuppress>
<!--ro, opt, object-->
<enabled opt="true,false">
  <!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
</OverexposeSuppress>
</Exposure>
<Sharpness>
<!--ro, opt, object-->
<SharpnessLevel min="0" max="100">
  <!--ro, req, int, attr:min{req, int},max{req, int}-->1
</SharpnessLevel>
</Sharpness>
<powerLineFrequency>
<!--ro, opt, object-->
<powerLineFrequencyMode opt="50hz,60hz">
  <!--ro, opt, enum, subType:string, attr:opt{req, string}-->50hz
</powerLineFrequencyMode>
</powerLineFrequency>
<Color>
<!--ro, opt, object, image color-->
<brightnessLevel min="0" max="100">
  <!--ro, opt, int, brightness, attr:min{req, int},max{req, int}-->24
</brightnessLevel>
<contrastLevel min="0" max="100">
  <!--ro, opt, int, contrast, attr:min{req, int},max{req, int}-->22
</contrastLevel>
<saturationLevel min="0" max="100">
  <!--ro, opt, int, saturation, dependency:{$.Color.nightMode,eq,true}, attr:min{req, int},max{req, int}-->33
</saturationLevel>
<grayScale>
<!--ro, opt, object, gray scale-->
<grayScaleMode opt="outdoor,indoor">
  <!--ro, opt, enum, gray scale mode, subType:string, attr:opt{req, string}, desc:indoor; outdoor-->outdoor
</grayScaleMode>
</grayScale>
</Color>
<IrcutFilter>
<!--ro, opt, object-->
<IrcutFilterType opt="auto,day,night,schedule,eventTrigger,videoAuto,darkFighterX,darkFighterXAuto,darkFighterXSchedule">
  <!--ro, opt, enum, subType:string, attr:opt{req, string}-->auto
</IrcutFilterType>
</IrcutFilter>
<HLC>
<!--ro, opt, object-->
<enabled opt="true,false">
  <!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<HLCLevel min="0" max="10">
  <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
</HLCLevel>
</HLC>
<corridor>
<!--ro, opt, object, mode-->
<enabled opt="true,false">
  <!--ro, req, bool, whether to enable, attr:opt{req, string}-->true
</enabled>
</corridor>
</ImageChannel>

```

## 16.2.6 Image Enhancement

### 16.2.6.1 Get defog mode parameters of a specific channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/dehaze

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Dehaze xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, "open,close,auto", attr:version{req, string, protocolVersion}-->
  <DehazeMode>
    <!--ro, opt, enum, defog mode, subType:string, desc:"open", "close", "auto"-->open
  </DehazeMode>
  <DehazeLevel>
    <!--ro, opt, int, defog Level-->1
  </DehazeLevel>
</Dehaze>
```

### 16.2.6.2 Set defog mode parameters of a specific channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/dehaze

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Dehaze xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, "open,close,auto", attr:version{req, string, protocolVersion}-->
  <DehazeMode>
    <!--opt, enum, defog mode, subType:string, desc:"open", "close", "auto"-->open
  </DehazeMode>
  <DehazeLevel>
    <!--opt, int, defog Level-->1
  </DehazeLevel>
</Dehaze>
```

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.6.3 Set the defogging mode (in day mode) parameters of a specified channel

## Request URL

PUT /ISAPI/Image/channels/<channelID>/dehaze/day

## Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Dehaze xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <DehazeMode>
    <!--opt, enum, subType:string-->open
  </DehazeMode>
  <DehazeLevel>
    <!--opt, int-->1
  </DehazeLevel>
</Dehaze>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

## 16.2.6.4 Get the defogging mode (in day mode) parameters of a specified channel

### Request URL

GET /ISAPI/Image/channels/<channelID>/dehaze/day

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Dehaze xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <DehazeMode>
    <!--ro, opt, enum, subType:string-->open
  </DehazeMode>
  <DehazeLevel>
    <!--ro, opt, int-->1
  </DehazeLevel>
</Dehaze>
```

## 16.2.6.5 Get the defogging mode (in night mode) parameters of a specified channel

### Request URL

GET /ISAPI/Image/channels/<channelID>/dehaze/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Dehaze xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <DehazeMode>
    <!--ro, opt, enum, subType:string-->open
  </DehazeMode>
  <DehazeLevel>
    <!--ro, opt, int-->1
  </DehazeLevel>
</Dehaze>
```

### 16.2.6.6 Set the defogging mode (in night mode) parameters of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/dehaze/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Dehaze xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <DehazeMode>
    <!--opt, enum, subType:string-->open
  </DehazeMode>
  <DehazeLevel>
    <!--opt, int-->1
  </DehazeLevel>
</Dehaze>
```

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.6.7 Set the picture gain parameters of a specific channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/gain

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Gain xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, gain, attr:version{req, string, protocolVersion}-->
  <GainLevel>
    <!--opt, int, gain Level-->1
  </GainLevel>
</Gain>
```

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

#### 16.2.6.8 Get the picture gain parameters of a specific channel

##### Request URL

GET /ISAPI/Image/channels/<channelID>/gain?parameterType=<parameterType>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Gain xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, gain, attr:version{req, string, protocolVersion}-->
  <GainLevel>
    <!--ro, opt, int, gain Level-->1
  </GainLevel>
</Gain>
```

#### 16.2.6.9 Get the capability of image distortion correction of a specific channel

##### Request URL

GET /ISAPI/Image/channels/<channelID>/lensDistortionCorrection/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<LensDistortionCorrection xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, the capability of image distortion correction, attr:version{req, string, protocolVersion}-->
  <enabled opt="true,false">
    <!--ro, req, bool, whether to enable the function, attr:opt{req, string}-->true
  </enabled>
</LensDistortionCorrection>
```

## 16.2.6.10 Set the DNR (Digital Noise Reduction) parameters in auto mode of a specific channel

### Request URL

PUT /ISAPI/Image/channels/<channelID>/noiseReduce

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<NoiseReduce xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, noise reduction parameters, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--req, enum, mode, subType:string, desc:"close", "general", "advanced"-->close
  </mode>
  <GeneralMode>
    <!--opt, object, mode-->
    <generalLevel>
      <!--req, int, level-->1
    </generalLevel>
  </GeneralMode>
</NoiseReduce>
```

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

## 16.2.6.11 Get the DNR (Digital Noise Reduction) parameters in auto mode of a specific channel

### Request URL

GET /ISAPI/Image/channels/<channelID>/noiseReduce

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<NoiseReduce xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, noise reduction parameters, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--ro, req, enum, mode, subType:string, desc:"close", "general", "advanced"-->close
  </mode>
  <GeneralMode>
    <!--ro, opt, object, this node is valid only when <mode> is "general"-->
    <generalLevel>
      <!--ro, req, int-->1
    </generalLevel>
  </GeneralMode>
</NoiseReduce>
```

### 16.2.6.12 Set the digital noise reduction (DNR) parameters in day mode of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/noiseReduce/day

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<NoiseReduce xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--req, enum, subType:string-->close
  </mode>
  <GeneralMode>
    <!--opt, object, dep:and,{$.NoiseReduce.mode,eq,general}-->
    <generalLevel>
      <!--req, int, range:[0,100]-->0
    </generalLevel>
  </GeneralMode>
</NoiseReduce>
```

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.6.13 Get the digital noise reduction (DNR) parameters in day mode of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/noiseReduce/day

## Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<NoiseReduce xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--ro, req, enum, subType:string-->close
  </mode>
  <GeneralMode>
    <!--ro, opt, object, dep:and, ${$.NoiseReduce.mode, eq, general}-->
    <generalLevel>
      <!--ro, req, int, range:[0,100]-->0
    </generalLevel>
  </GeneralMode>
</NoiseReduce>
```

## 16.2.6.14 Set the digital noise reduction (DNR) parameters in night mode a specified channel

### Request URL

PUT /ISAPI/Image/channels/<channelID>/noiseReduce/night

## Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<NoiseReduce xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--req, enum, subType:string-->close
  </mode>
  <GeneralMode>
    <!--opt, object, dep:and, ${$.NoiseReduce.mode, eq, general}-->
    <generalLevel>
      <!--req, int, range:[0,100]-->0
    </generalLevel>
  </GeneralMode>
</NoiseReduce>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

## 16.2.6.15 Get the digital noise reduction (DNR) parameters in day mode of a specified channel

### Request URL

GET /ISAPI/Image/channels/<channelID>/noiseReduce/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<NoiseReduce xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--ro, req, enum, subType:string-->close
  </mode>
  <GeneralMode>
    <!--ro, opt, object, dep:and, ${$.NoiseReduce.mode,eq,general}-->
    <generalLevel>
      <!--ro, req, int, range:[0,100]-->0
    </generalLevel>
  </GeneralMode>
</NoiseReduce>
```

## 16.2.7 Image Light Supplement

### 16.2.7.1 Set the day/night mode parameters of a specific channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/ISPMODE

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ISPMODE xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, ISP mode, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--opt, enum, "auto,schedule", subType:string, desc:"auto", "schedule"-->schedule
  </mode>
  <Schedule>
    <!--opt, object-->
    <scheduleType>
      <!--req, enum, "day,night", subType:string, desc:"day", "night"-->day
    </scheduleType>
    <TimeRange>
      <!--req, object, time range-->
      <beginTime>
        <!--req, time, start time-->00:00:00+08:00
      </beginTime>
      <endTime>
        <!--req, time, end time-->00:00:00+08:00
      </endTime>
    </TimeRange>
  </Schedule>
</ISPMODE>
```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0-OK, 1-OK, 2-Device Busy, 3-Device Error, 4-Invalid Operation, 5-Invalid XML Format, 6-Invalid XML Content, 7-Reboot Required-->0
    <statusCode>
      <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, desc:sub status code-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, desc:sub status code-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

### 16.2.7.2 Get the ISP mode of a specific channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/ISPMODE

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ISPMODE xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, ISP mode, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--ro, opt, enum, mode, subType:string, desc:"auto", "schedule"-->schedule
  </mode>
  <Schedule>
    <!--ro, opt, object, schedule information-->
    <scheduleType>
      <!--ro, req, enum, schedule type, subType:string, desc:"day", "night"-->day
    </scheduleType>
    <TimeRange>
      <!--ro, req, object, time range-->
      <beginTime>
        <!--ro, req, time, start time-->00:00:00+08:00
      </beginTime>
      <endTime>
        <!--ro, req, time, end time-->00:00:00+08:00
      </endTime>
    </TimeRange>
  </Schedule>
</ISPMODE>

```

### 16.2.7.3 Set the laser supplement light parameters of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/laserLight

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<LaserLight xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <node>
    <!--req, enum, subType:string-->auto
  </node>
  <brightnessLevel>
    <!--opt, int, range:[0,255], dep:and,{$.LaserLight.mode,eq,manual}-->0
  </brightnessLevel>
  <laserangle>
    <!--opt, int, range:[1,36], dep:and,{$.LaserLight.mode,eq,manual}-->1
  </laserangle>
</LaserLight>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.7.4 Get the laser supplement light parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/laserLight

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<LaserLight xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <node>
    <!--ro, enum, subType:string-->auto
  </node>
  <brightnessLevel>
    <!--ro, opt, int, range:[0,255], dep:and,{$.LaserLight.mode,eq,manual}-->0
  </brightnessLevel>
  <laserangle>
    <!--ro, opt, int, range:[1,36], dep:and,{$.LaserLight.mode,eq,manual}-->1
  </laserangle>
</LaserLight>

```

### 16.2.7.5 Set the mounting scenario mode of a camera

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/mountingScenario

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<MountingScenario xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--req, enum, "indoor,outdoor,day,night,morning,nightfall,mode1,mode2,mode3,mode4,highway,road2,faceSnap,backLight,frontLight", "LowIllumination"-Low
    illumination,"backLight"-back Light,"frontLight"-front Light,"faceSnap"-face picture capture, subType:string, desc:"indoor", "outdoor", "day", "night",
    "morning", "nightfall", "mode1", "mode2", "mode3", "mode4", "street", "LowIllumination"-Low illumination, "custom1", "custom2", "normal", "road",
    "faceSnap"-face picture capture, "highway", "road2", "backLight"-back Light, "frontLight"-front Light, "facenight", "fastRoadAndLight", "fastRoadNoLight",
    "normalRoadAndLight", "normalRoadNoLight", "cityRoadAndBright"-->indoor
  </mode>
</MountingScenario>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0-OK, 1-OK, 2-Device Busy, 3-Device Error, 4-Invalid Operation, 5-Invalid XML Format, 6-Invalid XML
    Content, 7-Reboot Required-->0
    </statusCode>
    <statusString>
      <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format",
      "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, desc:sub status code-->OK
    </subStatusCode>
  </ResponseStatus>
```

### 16.2.7.6 Get the capability of configurations of mounting scenario mode

#### Request URL

GET /ISAPI/Image/channels/<channelID>/mountingScenario/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<MountingScenario xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    opt="indoor,outdoor,day,night,morning,nightfall,mode1,mode2,mode3,mode4,street,lowIllumination,custom1,custom2,normal,road,faceSnap,highway,road2,backlight,
    frontlight, facenight, fastRoadAndLight, fastRoadNoLight, normalRoadAndLight, normalRoadNoLight"
    <!--ro, req, enum, Mode, subType:string, attr:opt{req, string}, desc:"lowIllumination"-low illumination, "backlight"-back Light, "frontlight"-front
    Light, "faceSnap"-face picture capture-->indoor
  </mode>
</MountingScenario>
```

### 16.2.7.7 Get the mounting scenario mode of a camera

#### Request URL

GET /ISAPI/Image/channels/<channelID>/mountingScenario?parameterType=<parameterType>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<MountingScenario xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <mode>
    <!--ro, req, enum, "indoor,outdoor,day,night,morning,nightfall,mode1,mode2,mode3,mode4,highway,road2,faceSnap,backlight,frontlight", "LowIllumination"-Low illumination,"backlight"-back Light,"frontLight"-front Light,"faceSnap"-face picture capture, subType:string, desc:"indoor", "outdoor", "day", "night", "morning", "nightfall", "mode1", "mode2"," mode3", "mode4", "highway", "road2", "faceSnap", "backlight", "frontlight", "LowIllumination"-Low illumination,"backlight"-back Light, "frontlight"-front Light, "faceSnap"-face picture capture-->indoor
  </mode>
</MountingScenario>
```

## 16.2.8 Manual Capture

### 16.2.8.1 Get the manual capture result information of a specified channel

#### Request URL

GET /ISAPI/Streaming/channels/<trackStreamID>/picture?snapShotImageType=<snapShotImageType>&videoResolutionWidth=<videoResolutionWidth>&videoResolutionHeight=<videoResolutionHeight>&imageQuality=<imageQuality>&x=<x>&y=<y>&hight=<hight>&width=<width>&maxSnapResolution=<maxSnapResolution>

#### Query Parameter

Parameter Name	Parameter Type	Description
trackStreamID	string	--
snapShotImageType	string	--
videoResolutionWidth	string	--
videoResolutionHeight	string	--
imageQuality	string	--
x	string	--
y	string	--
hight	string	--
width	string	--
maxSnapResolution	enum	--

#### Request Message

None

#### Response Message

None

## 16.2.9 Meta Data

### 16.2.9.1 Get all metadata parameters of the specified channel

#### Request URL

GET /ISAPI/Streaming/channels/<channelID>/metadata

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<MetadataCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, metadata configurations, attr:version{req, string, protocolVersion}-->
    <MetadataList>
        <!--ro, req, array, metadata information list, subType:object-->
        <Metadata>
            <!--ro, req, object, metadata information-->
            <type>
                <!--ro, req, enum, event types that support metadata, subType:string, desc:"thermometry" (real-time temperature measurement), "fireDetection" (fire detection), "shipsDetection" (ship detection), "behaviorAnalysis" (abnormal event detection), "ANPR" (automatic number plate recognition), "personQueue" (queue management, including people queuing-up detection and waiting time detection), "faceSnap" (face capture), "mixedTargetDetection" (multi-target-detection), "AIOP" (AIOP detection), "radarDetection" (radar detection), "personalTrack" (personal tracking)-->thermometry
            </type>
            <enable>
                <!--ro, req, bool, whether to enable uploading metadata-->true
            </enable>
        </Metadata>
    </MetadataList>
</MetadataCfg>
```

#### 16.2.9.2 Set all metadata parameters for a specified channel

#### Request URL

PUT /ISAPI/Streaming/channels/<channelID>/metadata

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<MetadataCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, object, metadata configurations, attr:version{req, string, protocolVersion}-->
    <MetadataList>
        <!--req, array, metadata information list, subType:object-->
        <Metadata>
            <!--req, object, metadata information-->
            <type>
                <!--req, enum, event types that support metadata, subType:string, desc:"thermometry" (real-time temperature measurement), "fireDetection" (fire detection), "shipsDetection" (ship detection), "behaviorAnalysis" (abnormal event detection), "ANPR" (automatic number plate recognition), "personQueue" (queue management, including people queuing-up detection and waiting time detection), "faceSnap" (face capture), "mixedTargetDetection" (multi-target-detection), "AIOP" (AIOP detection)-->thermometry
            </type>
            <enable>
                <!--req, bool, whether to enable uploading metadata-->true
            </enable>
        </Metadata>
    </MetadataList>
</MetadataCfg>
```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL, range:[0,1024]-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
    <description>
      <!--ro, opt, string, range:[0,1024]-->badXmlFormat
    </description>
  </statusCode>
</ResponseStatus>

```

### 16.2.9.3 Get metadata configuration capabilities of the specified channel

#### Request URL

GET /ISAPI/Streaming/channels/<channelID>/metadata/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<MetadataCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, metadata configurations, attr:version{req, string, protocolVersion}-->
  <MetadataList size="10">
    <!--ro, req, array, metadata information list, subType:object, attr:size{req, int}-->
    <Metadata>
      <!--ro, req, object, metadata information-->
      <type>
        <!--ro, req, enum, event types that support metadata, subType:string, desc:"thermometry" (real-time temperature measurement), "fireDetection" (fire detection), "shipsDetection" (ship detection), "behaviorAnalysis" (abnormal event detection), "ANPR" (automatic number plate recognition), "personQueue" (queue management, including people queuing-up detection and waiting time detection), "faceSnap" (face capture), "mixedTargetDetection" (multi-target-detection), "AIOP" (AIOP detection)-->thermometry
      </type>
      <enable>
        <!--ro, req, bool, whether to enable uploading metadata-->true
      </enable>
    </Metadata>
  </MetadataList>
  <isSupportSubscribeType>
    <!--ro, opt, bool, whether the device supports subscribing to the metadata, desc:this field is returned when the device supports subscribing to the metadata. By default, the subscription of metadata is according to channels-->true
  </isSupportSubscribeType>
</MetadataCfg>

```

### 16.2.10 Motion Detection

#### 16.2.10.1 Get arming schedule of motion detection of all channels

#### Request URL

GET /ISAPI/Event/schedules/motionDetections

#### Query Parameter

None

#### Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<MotionDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, arming time list, subType:object, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--ro, opt, object-->
    <id>
      <!--ro, req, string, ID, desc:ID-->VMD_video1
    </id>
    <eventType>
      <!--ro, opt, enum, event type, subType:string, desc:event type-->VMD
    </eventType>
    <videoInputChannelID>
      <!--ro, opt, string, video input channel ID-->1
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--ro, req, array, arming time list, subType:object, attr:size{opt, int}-->
      <TimeBlock>
        <!--ro, opt, object, arming time-->
        <dayOfWeek>
          <!--ro, opt, enum, days of the week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->1
        </dayOfWeek>
        <TimeRange>
          <!--ro, req, object, time range-->
          <beginTime>
            <!--ro, req, time, start time-->10:00:00
          </beginTime>
          <endTime>
            <!--ro, req, time, end time-->10:00:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>
</MotionDetectionScheduleList>
```

### 16.2.10.2 Set arming schedule of motion detection of all channels

#### Request URL

PUT /ISAPI/Event/schedules/motionDetections

#### Query Parameter

None

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<MotionDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, arming time list, subType:object, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--opt, object-->
    <id>
      <!--req, string, No., desc>No.-->VMD_video1
    </id>
    <eventType>
      <!--opt, enum, event type, subType:string, desc:"VMD" (motion detection)-->VMD
    </eventType>
    <videoInputChannelID>
      <!--opt, string, video input channel ID-->1
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--req, array, arming time list, subType:object, attr:size{opt, int}-->
      <TimeBlock>
        <!--opt, object, arming time-->
        <dayOfWeek>
          <!--req, enum, days of the week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->1
        </dayOfWeek>
        <TimeRange>
          <!--req, object, time range-->
          <beginTime>
            <!--req, time, start time-->10:00:00
          </beginTime>
          <endTime>
            <!--req, time, end time-->10:00:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>
</MotionDetectionScheduleList>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.10.3 Get the arming schedule of motion detection for a specified channel

#### Request URL

GET /ISAPI/Event/schedules/motionDetections/VMD\_video<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->VMD_video01
  </id>
  <eventType>
    <!--ro, opt, string-->VMD
  </eventType>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <timeBlockList size="8">
    <!--ro, req, array, subType:object, attr:size{opt, int}-->
    <TimeBlock>
      <!--ro, opt, object-->
      <dayOfWeek>
        <!--ro, opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--ro, req, object-->
        <beginTime>
          <!--ro, req, time-->10:00:00
        </beginTime>
        <endTime>
          <!--ro, req, time-->10:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </timeBlockList>
</Schedule>
```

### 16.2.10.4 Set the arming schedule of motion detection for a specified channel

#### Request URL

PUT /ISAPI/Event/schedules/motionDetections/VMD\_video<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->VMD_video1
  </id>
  <eventType>
    <!--opt, string-->VMD
  </eventType>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--req, array, subType:object, attr:size{opt, int}-->
    <TimeBlock>
      <!--opt, object-->
      <dayOfWeek>
        <!--opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--req, object-->
        <beginTime>
          <!--req, time-->10:00:00
        </beginTime>
        <endTime>
          <!--req, time-->10:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>
```

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.10.5 Delete the linkage of motion detection for a specified channel

#### Request URL

DELETE /ISAPI/Event/triggers/VMD-<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

## 16.2.10.6 Set the linkage triggering parameters of motion detection for a specified channel

### Request URL

PUT /ISAPI/Event/triggers/VMD-<channelID>/notifications

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotification xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
</EventTriggerNotification>

```

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

## 16.2.10.7 Get the linkage triggering parameters of motion detection for a specified channel

### Request URL

GET /ISAPI/Event/triggers/VMD-<channelID>/notifications

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotification xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
</EventTriggerNotification>

```

### 16.2.10.8 Get the linkage parameters of motion detection for a specified channel

#### Request URL

GET /ISAPI/Event/triggers/VMD-<channelID>?triggersEnabledStatus=<triggersEnabledStatus>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
triggersEnabledStatus	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->1
  </id>
  <eventType>
    <!--ro, req, string-->VMD
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subType:object-->
    <EventTriggerNotification>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->test
      </id>
      <notificationMethod>
        <!--ro, req, enum, subType:string-->[email
      </notificationMethod>
      <notificationRecurrence>
        <!--ro, opt, enum, subType:string-->beginning
      </notificationRecurrence>
      <videoInputID>
        <!--ro, opt, string-->1
      </videoInputID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

### 16.2.10.9 Set the linkage parameters of motion detection for a specified channel

#### Request URL

PUT /ISAPI/Event/triggers/VMD-<channelID>?triggersEnabledStatus=<triggersEnabledStatus>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
triggersEnabledStatus	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->1
  </id>
  <eventType>
    <!--req, enum, subType:string-->VMD
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
    <EventTriggerNotification>
      <!--opt, object-->
      <id>
        <!--req, string-->test
      </id>
      <notificationMethod>
        <!--req, enum, subType:string-->email
      </notificationMethod>
      <notificationRecurrence>
        <!--opt, enum, subType:string-->beginning
      </notificationRecurrence>
      <videoInputID>
        <!--opt, string-->1
      </videoInputID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.10.10 Set the motion detection parameters of a specific channel

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<MotionDetection xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, motion detection parameters, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool, whether to enable motion detection-->true
  </enabled>
  <enableHighlight>
    <!--opt, bool, whether to enable dynamic analysis-->true
  </enableHighlight>
  <samplingInterval>
    <!--opt, int, frame interval, range:[0,2]-->0
  </samplingInterval>
  <startTriggerTime>
    <!--opt, int, start time, range:[0,500], unit:ms-->0
  </startTriggerTime>
  <endTriggerTime>
    <!--opt, int, end time, range:[0,500], unit:ms-->0
  </endTriggerTime>
  <regionType>
    <!--req, enum, region type, subType:string, desc:"grid", "roi" (region of interest), "none", "region" (multiple regions)-->grid
  </regionType>
  <Grid>
    <!--opt, object, grid-->
    <rowGranularity>
      <!--req, int, total number of rows, range:[0,18]-->0
    </rowGranularity>
    <columnGranularity>
      <!--req, int, total number of columns, range:[0,22]-->0
    </columnGranularity>
  </Grid>
  <MotionDetectionLayout>
    <!--opt, object, layout configuration-->
    <sensitivityLevel>
      <!--req, int, sensitivity, range:[0,100]-->0
    </sensitivityLevel>
    <layout>
      <!--opt, object, layout information-->
      <gridMap>
        <!--opt, string, grid map, desc:the image is divided in to N × N grids, and motion detection can be enabled (value: 1) or disabled (value 0) for each grid. Each four horizontal grids consist of a hexadecimal number, e.g., the corresponding hexadecimal number of "0011" is 3-->test
        </gridMap>
      <RegionList>
        <!--opt, array, region List, subType:object-->
        <Region>
          <!--opt, object, region information-->
          <id>
            <!--req, string, region ID, range:[1,4]-->test
          </id>
          <RegionCoordinatesList>
            <!--opt, array, region coordinate list, subType:object, range:[0,4], desc:rectangle-->
            <RegionCoordinates>
              <!--opt, object, region coordinates, desc:the origin is the lower-left corner of the screen-->
              <positionX>
                <!--req, int, X-coordinate, range:[0,1000]-->0
              </positionX>
              <positionY>
                <!--req, int, Y-coordinate, range:[0,1000]-->0
              </positionY>
            </RegionCoordinates>
          </RegionCoordinatesList>
        </Region>
      </RegionList>
    </layout>
    <targetType>
      <!--opt, enum, target classification, subType:string, desc:"human", "vehicle"-->human
    </targetType>
  </MotionDetectionLayout>
</MotionDetection>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL, range:[0,1024]-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
  <description>
    <!--ro, opt, string, custom error information description, range:[0,1024], desc:the detailed information of custom error returned by device applications, which is used for fast debugging-->badXmlFormat
  </description>
</ResponseStatus>

```

### 16.2.10.11 Get the motion detection parameters of a specific channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<MotionDetection xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, motion detection parameters, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool, whether to enable motion detection-->true
  </enabled>
  <enableHighlight>
    <!--ro, opt, bool, whether to enable dynamic analysis-->true
  </enableHighlight>
  <samplingInterval>
    <!--ro, opt, int, frame interval, range:[0,2]-->0
  </samplingInterval>
  <startTriggerTime>
    <!--ro, opt, int, start time, range:[0,500], unit:ms-->0
  </startTriggerTime>
  <endTriggerTime>
    <!--ro, opt, int, end time, range:[0,500], unit:ms-->0
  </endTriggerTime>
  <regionType>
    <!--ro, req, enum, region type, subType:string, desc:"grid", "roi" (region of interest), "none", "region" (multiple regions)-->grid
  </regionType>
  <Grid>
    <!--ro, opt, object, grid-->
    <rowGranularity>
      <!--ro, req, int, total number of rows, range:[0,18]-->0
    </rowGranularity>
    <columnGranularity>
      <!--ro, req, int, total number of columns, range:[0,22]-->0
    </columnGranularity>
  </Grid>
  <MotionDetectionLayout>
    <!--ro, opt, object, layout configuration-->
    <sensitivityLevel>
      <!--ro, req, int, sensitivity, range:[0,100]-->0
    </sensitivityLevel>
    <layout>
      <!--ro, opt, object, layout information-->
      <gridMap>
        <!--ro, opt, string, grid map, desc:the image is divided in to N × N grids, and motion detection can be enabled (value: 1) or disabled (value: 0) for each grid. Each four horizontal grids consist of a hexadecimal number, e.g., the corresponding hexadecimal number of "0011" is 3-->test
      </gridMap>
      <RegionList>
        <!--ro, opt, array, this node is available when the regionType is set to "region", subType:object-->
        <Region>
          <!--ro, opt, object, list-->
          <id>
            <!--ro, req, string, region ID, range:[1,4]-->test
          </id>
          <RegionCoordinatesList>
            <!--ro, opt, array, rectangle, subType:object, range:[0,4], desc:rectangle-->
            <RegionCoordinates>
              <!--ro, opt, object, list, desc:the origin is the lower-left corner of the screen-->
              <positionX>
                <!--ro, req, int, X-coordinate, range:[0,1000]-->0
              </positionX>
              <positionY>
                <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
              </positionY>
            </RegionCoordinates>
          </RegionCoordinatesList>
        </Region>
      </RegionList>
    </layout>
    <targetType>
      <!--ro, opt, enum, target classification, subType:string, desc:"human", "vehicle"-->human
    </targetType>
  </MotionDetectionLayout>
</MotionDetection>

```

### 16.2.10.12 Get the motion detection capability of a specific channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<MotionDetection xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, motion detection parameters, attr:version{req, string, protocolVersion}-->
  <enabled opt="true,false">
    <!--ro, req, bool, whether to enable motion detection, attr:opt{req, string}-->true
  </enabled>
  <enableHighlight opt="true,false">
    <!--ro, opt, bool, whether to enable dynamic analysis, attr:opt{req, string}-->true
  </enableHighlight>
  <samplingInterval>
    <!--ro, opt, int, frame interval-->0
  </samplingInterval>
  <startTriggerTime>
    <!--ro, opt, int, start time, unit:ms-->0
  </startTriggerTime>
  <endTriggerTime>
    <!--ro, opt, int, end time, unit:ms-->0
  </endTriggerTime>
  <regionType opt="grid,roi,none,region">
    <!--ro, req, enum, region type, subType:string, attr:opt{req, string}, desc:"grid", "roi" (region of interest), "none", "region" (multiple regions)-->
  >grid
  </regionType>
  <Grid>
    <!--ro, opt, object, grid-->
    <rowGranularity>
      <!--ro, req, int, total number of rows-->0
    </rowGranularity>
    <columnGranularity>
      <!--ro, req, int, total number of columns-->0
    </columnGranularity>
  </Grid>
  <MotionDetectionLayout>
    <!--ro, req, object, layout configuration capability-->
    <sensitivityLevel min="0" max="100" step="20">
      <!--ro, req, int, sensitivity, range:[0,100], attr:min{req, int},max{req, int},step{opt, int}-->0
    </sensitivityLevel>
    <layout>
      <!--ro, opt, object, layout information-->
      <gridMap>
        <!--ro, opt, string, grid map, desc:the image is divided in to N × N grids, and motion detection can be enabled (value: 1) or disabled (value: 0)
        for each grid. Each four horizontal grids consist of a hexadecimal number, e.g., the corresponding hexadecimal number of "0011" is 3-->test
        </gridMap>
        <RegionList size="4">
          <!--ro, opt, array, this node is available when the regionType is set to "region", subType:object, attr:size{req, int}-->
          <Region>
            <!--ro, opt, object, list-->
            <id>
              <!--ro, req, string, region ID-->test
            </id>
            <RegionCoordinatesList>
              <!--ro, opt, array, rectangle, subType:object, range:[0,4], desc:rectangle-->
              <RegionCoordinates>
                <!--ro, opt, object, list, desc:the origin is the lower-left corner of the screen-->
                <positionX>
                  <!--ro, req, int, X-coordinate, range:[0,1000]-->0
                </positionX>
                <positionY>
                  <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
                </positionY>
              </RegionCoordinates>
            </RegionCoordinatesList>
          </Region>
        </RegionList>
      </layout>
      <targetType opt="human,vehicle">
        <!--ro, opt, enum, target classification, subType:string, attr:opt{req, string}, desc:"human", "vehicle"-->human
      </targetType>
      <regionEdgesNum min="4" max="10">
        <!--ro, opt, int, number of detection area sides, range:[4,10], attr:min{req, int},max{req, int}, desc:if this node is not returned, the default
        detection area is quadrilateral-->0
      </regionEdgesNum>
    </MotionDetectionLayout>
  </MotionDetection>

```

### 16.2.10.13 Set grid layout parameters of motion detection of a specific channel

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection/layout/gridLayout?regionType=<regionType>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionType	enum	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<MotionDetectionGridLayout xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, grid Layout parameters of motion detection, attr:version{req, string, protocolVersion}-->
  <sensitivityLevel>
    <!--req, int, sensitivity, range:[0,100], desc:sensitivity-->
  </sensitivityLevel>
</MotionDetectionGridLayout>
```

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->0
  </subStatusCode>
</ResponseStatus>
```

### 16.2.10.14 Get grid layout parameters of motion detection of a specific channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection/layout/gridLayout?regionType=<regionType>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionType	enum	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<MotionDetectionGridLayout xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <sensitivityLevel>
    <!--ro, req, int, sensitivity, range:[0,100], desc:sensitivity-->1
  </sensitivityLevel>
</MotionDetectionGridLayout>
```

### 16.2.10.15 Get the region parameters of motion detection of a specific channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection/layout?regionType=<regionType>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionType	enum	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<MotionDetectionLayout xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <sensitivityLevel>
    <!--req, int, sensitivity-->1
  </sensitivityLevel>
  <layout>
    <!--req, opt, object-->
    <gridMap>
      <!--req, string, grid map, desc:the image is divided in to N × N grids, and motion detection can be enabled (value: 1) or disabled (value 0) for each grid. Each four horizontal grids consist of a hexadecimal number-->0011
    </gridMap>
  </layout>
</MotionDetectionLayout>
```

## 16.2.10.16 Set the region parameters of motion detection of a specific channel

### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/motionDetection/layout?regionType=<regionType>

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionType	enum	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<MotionDetectionLayout xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <sensitivityLevel>
    <!--req, int, sensitivity-->1
  </sensitivityLevel>
  <layout>
    <!--opt, object-->
    <gridMap>
      <!--req, string, grid map, desc:the image is divided in to N × N grids, and motion detection can be enabled (value: 1) or disabled (value 0) for each grid. Each four horizontal grids consist of a hexadecimal number-->0011
    </gridMap>
  </layout>
</MotionDetectionLayout>
```

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.10.17 Get motion detection parameters of a specified video input channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/motionDetectionExt

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<MotionDetectionExt xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, motion detection parameters of a specified video input channel, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool, whether to enable the function-->true
  </enabled>
  <minObjectSize>
    <!--ro, opt, int, minimum target resolution-->0
  </minObjectSize>
  <maxObjectSize>
    <!--ro, opt, int, maximum target resolution-->0
  </maxObjectSize>
  <ROI>
    <!--ro, opt, object, region of interest (ROI)-->
  </ROI>
  <enableHighlight>
    <!--ro, opt, bool, whether to enable highlight-->true
  </enableHighlight>
  <MotionDetectionSwitch>
    <!--ro, opt, object, day/night settings switch-->
    <type>
      <!--ro, opt, enum, type, subType:string, desc:"off", "auto", "schedule" (scheduled)-->off
    </type>
    <Schedule>
      <!--ro, opt, object, schedule-->
      <scheduleType>
        <!--ro, req, enum, schedule type, subType:string, desc:"day" (daytime), "night" (night)-->day
      </scheduleType>
      <TimeRange>
        <!--ro, req, object, time period-->
        <beginTime>
          <!--ro, req, time, start time-->00:00:00+08:00
        </beginTime>
        <endTime>
          <!--ro, req, time, end time-->00:00:00+08:00
        </endTime>
      </TimeRange>
    </Schedule>
  </MotionDetectionSwitch>
  <activeMode>
    <!--ro, opt, enum, effective mode, subType:string, desc:"normal", "expert"-->normal
  </activeMode>
  <MotionDetectionRegionList size="10">
    <!--ro, opt, array, detection area List, subType:object, attr:size{req, int}-->
    <MotionDetectionRegion>
      <!--ro, opt, object, detection area-->
      <id>
        <!--ro, req, string, ID-->test
      </id>
      <enabled>
        <!--ro, req, bool, whether to enable the function-->true
      </enabled>
      <sensitivityLevel>
        <!--ro, opt, int, sensitivity, range:[0,100]-->0
      </sensitivityLevel>
      <daySensitivityLevel>
        <!--ro, opt, int, sensitivity (during day time), range:[0,100]-->0
      </daySensitivityLevel>
      <nightSensitivityLevel>
        <!--ro, opt, int, sensitivity (during night), range:[0,100]-->0
      </nightSensitivityLevel>
      <RegionCoordinatesList>
        <!--ro, req, array, rule region List, subType:object, range:[0,4], desc:rectangle-->
        <RegionCoordinates>
          <!--ro, opt, object, area, desc:the origin is the Lower-Left corner of the screen-->
          <positionX>
            <!--ro, req, int, x-coordinate, range:[0,1000]-->0
          </positionX>
          <positionY>
            <!--ro, req, int, y-coordinate, range:[0,1000]-->0
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </MotionDetectionRegion>
  </MotionDetectionRegionList>
</MotionDetectionExt>

```

### 16.2.10.18 Set motion detection parameters of a specified video input channel

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/motionDetectionExt

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<MotionDetectionExt xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, motion detection parameters of a specified video input channel, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool, whether to enable the function-->true
  </enabled>
  <minObjectSize>
    <!--opt, int, minimum target resolution-->0
  </minObjectSize>
  <maxObjectSize>
    <!--opt, int, maximum target resolution-->0
  </maxObjectSize>
  <ROI>
    <!--opt, object, region of interest (ROI)-->
  </ROI>
  <enableHighlight>
    <!--opt, bool, whether to enable highlight-->true
  </enableHighlight>
  <MotionDetectionSwitch>
    <!--opt, object, day/night settings switch-->
    <type>
      <!--opt, enum, type, subType:string, desc:"off", "auto", "schedule" (scheduled)-->off
    </type>
    <Schedule>
      <!--opt, object, schedule-->
      <scheduleType>
        <!--req, enum, schedule type, subType:string, desc:"day" (daytime), "night" (night)-->day
      </scheduleType>
      <TimeRange>
        <!--req, object, time period-->
        <beginTime>
          <!--req, time, start time-->00:00:00+08:00
        </beginTime>
        <endTime>
          <!--req, time, end time-->00:00:00+08:00
        </endTime>
      </TimeRange>
    </Schedule>
  </MotionDetectionSwitch>
  <activeMode>
    <!--opt, enum, effective mode, subType:string, desc:"normal", "expert"-->normal
  </activeMode>
  <MotionDetectionRegionList size="10">
    <!--opt, array, detection area list, subType:object, attr:size{req, int}-->
    <MotionDetectionRegion>
      <!--opt, object, detection area-->
      <id>
        <!--req, string, ID-->test
      </id>
      <enabled>
        <!--req, bool, whether to enable the function-->true
      </enabled>
      <sensitivityLevel>
        <!--opt, int, sensitivity, range:[0,100]-->0
      </sensitivityLevel>
      <daySensitivityLevel>
        <!--opt, int, sensitivity (during day time), range:[0,100]-->0
      </daySensitivityLevel>
      <nightSensitivityLevel>
        <!--opt, int, sensitivity (during night), range:[0,100]-->0
      </nightSensitivityLevel>
      <RegionCoordinatesList>
        <!--req, array, rule region list, subType:object, range:[0,4], desc:rectangle-->
        <RegionCoordinates>
          <!--opt, object, area, desc:the origin is the lower-left corner of the screen-->
          <positionX>
            <!--req, int, x-coordinate, range:[0,1000]-->0
          </positionX>
          <positionY>
            <!--req, int, y-coordinate, range:[0,1000]-->0
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </MotionDetectionRegion>
  </MotionDetectionRegionList>
</MotionDetectionExt>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL, range:[0,1024]-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, desc:sub status code-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, desc:sub status code-->OK
    </subStatusCode>
    <description>
      <!--ro, opt, string, custom error information description, range:[0,1024], desc:the detailed information of custom error returned by device applications, which is used for fast debugging-->badXmlFormat
    </description>
  </statusCode>
</ResponseStatus>

```

### 16.2.10.19 Set the motion detection parameters of a single region of a specific channel

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/motionDetectionExt/regions/<regionID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<MotionDetectionRegion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, detection area, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string, ID-->test
  </id>
  <enabled>
    <!--req, bool-->true
  </enabled>
  <sensitivityLevel>
    <!--opt, int, sensitivity, range:[0,100]-->0
  </sensitivityLevel>
  <daySensitivityLevel>
    <!--opt, int, sensitivity (during day time), range:[0,100]-->0
  </daySensitivityLevel>
  <nightSensitivityLevel>
    <!--opt, int, sensitivity (during night), range:[0,100]-->0
  </nightSensitivityLevel>
  <RegionCoordinatesList>
    <!--req, array, rectangle, subType:object, range:[0,4], desc:rectangle-->
    <RegionCoordinates>
      <!--opt, object, the origin is the lower-left corner of the screen, desc:the origin is the lower-left corner of the screen-->
      <positionX>
        <!--req, int, X-coordinate, range:[0,1000]-->0
      </positionX>
      <positionY>
        <!--req, int, Y-coordinate, range:[0,1000]-->0
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</MotionDetectionRegion>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

## 16.2.10.20 Motion detection

### EventType:VMD

```

<?xml version="1.0" encoding="UTF-8"?>

<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, EventNotificationAlert, attr:version{req, string, protocolVersion}-->
  <ipAddress>
    <!--ro, opt, string, IPv4 address-->172.6.64.7
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPv6 Address-->1080:0:0:0:8:800:200C:417A
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, device's communication port No.-->1
  </portNo>
  <protocol>
    <!--ro, opt, enum, transmission communication protocol, subType:string, desc:"HTTP", "HTTPS", "EHome"-->HTTP
  </protocol>
  <macAddress>
    <!--ro, opt, string, MAC address-->test
  </macAddress>
  <dynChannelID>
    <!--ro, opt, string, digital channel No.-->1
  </dynChannelID>
  <channelID>
    <!--ro, opt, string, channel No.-->1
  </channelID>
  <relatedChannelList>
    <!--ro, opt, string, list of alarm related channels, which are of the same camera with channelID, desc:this parameter is used for live view or playback on the platform; multiple channel No.s are separated by commas-->1,2,3
  </relatedChannelList>
  <dateTime>
    <!--ro, req, datetime, event occurred time-->2017-04-22T15:39:01+08:00
  </dateTime>
  <activePostCount>
    <!--ro, req, int, event triggering frequency-->1
  </activePostCount>
  <eventType>
    <!--ro, req, enum, event type, subType:string, desc:"VMD" (motion detection)-->VMD
  </eventType>
  <eventState>
    <!--ro, req, enum, event status (for persistent event), subType:string, desc:for durative event: active (valid), inactive (invalid)-->active
  </eventState>
  <eventDescription>
    <!--ro, req, string, event description-->test
  </eventDescription>
  <DetectionRegionList>
    <!--ro, opt, array, detection area parameter List (for expert mode only), subType:object-->
    <DetectionRegionEntry>
      <!--ro, opt, object, parameters of a detection area (for expert mode only), desc:parameters of a detection area (for expert mode only)-->
      <regionID>
        <!--ro, req, string, detection area ID-->test
      </regionID>
      <sensitivityLevel>
        <!--ro, req, int, sensitivity, range:[0,100]-->1
      </sensitivityLevel>
      <RegionCoordinatesList>
        <!--ro, req, array, detection area coordinates, subType:object, range:[0,4], desc:rectangle-->
        <RegionCoordinates>
          <!--ro, opt, object, detection area origin coordinates, desc:the origin is the upper-left corner of the screen-->
          <positionX>
            <!--ro, req, int, x-coordinate, range:[0,1000]-->1
          </positionX>
          <positionY>
            <!--ro, req, int, y-coordinate, range:[0,1000]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </DetectionRegionEntry>
  </DetectionRegionList>
</EventNotificationAlert>

```

```

</RegionCoordinates>
</RegionCoordinatesList>
</DetectionRegionEntry>
</DetectionRegionList>
<channelName>
  <!--ro, opt, string, channel name-->test
</channelName>
<deviceID>
  <!--ro, opt, string, device ID-->1
</deviceID>
<Extensions>
  <!--ro, opt, object, extended information-->
<serialNumber>
  <!--ro, req, string, serial No.-->test
</serialNumber>
<eventPush>
  <!--ro, req, string, event notification contents-->test
</eventPush>
</Extensions>
<targetType>
  <!--ro, opt, string, target classification, desc:"human", "vehicle", "human,vehicle" (human and vehicle). If detection target classification is not supported, this field is not returned-->human
</targetType>
<bkgUrl>
  <!--ro, opt, string, background picture URL-->test
</bkgUrl>
<URLCertificationType>
  <!--ro, opt, enum, picture URL authentication method, subType:string, desc:"no" (it is for the cloud storage protocol), "digest" (digest authentication, it is for the device local storage)-->no
</URLCertificationType>
</EventNotificationAlert>

```

## 16.2.11 OSD Settings

### 16.2.11.1 Get the OSD parameters of a specified channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/overlays

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<VideoOverlay xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, text overlay, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, opt, bool, whether to enable, desc:true (OSD is enabled), false (OSD is disabled). The default value is true-->true
  </enabled>
  <normalizedScreenSize>
    <!--ro, req, object, screen size-->
    <normalizedScreenWidth>
      <!--ro, req, int, normalized width-->1
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--ro, req, int, normalized height-->1
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <attribute>
    <!--ro, opt, object, attributes-->
    <transparent>
      <!--ro, req, bool, whether it is transparent-->true
    </transparent>
    <flashing>
      <!--ro, req, bool, whether it is flickering-->true
    </flashing>
    <backgroundColor>
      <!--ro, opt, object, background color-->
      <enabled>
        <!--ro, req, bool, whether to enable the background color-->true
      </enabled>
      <color>
        <!--ro, opt, string, background color RGB, desc:hexadecimal digit-->000000
      </color>
      <transparency>
        <!--ro, opt, int, background color transparency, range:[0,100]-->50
      </transparency>
    
```

```

        </transparency>
        </backgroundColor>
</attribute>
<TextOverlayList>
    <!--ro, opt, array, text content list, subType:object-->
    <TextOverlay>
        <!--ro, opt, object, text content-->
        <id>
            <!--ro, req, string, text No.-->1
        </id>
        <enabled>
            <!--ro, req, bool, whether to enable-->true
        </enabled>
        <positionX>
            <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
        </positionX>
        <positionY>
            <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
        </positionY>
        <displayText>
            <!--ro, req, string, overlay text-->test
        </displayText>
        <isPersistentText>
            <!--ro, opt, bool, whether the overlaid text remains the previous settings after device rebooting, desc:if the value is true or the node is not transmitted, it indicates to write OSD to flash, and it is not suitable for frequent or fast callings. If the value is false, it indicates not to writer OSD to flash and the OSD settings will not restore-->true
        </isPersistentText>
        <OverlayRegion>
            <!--ro, opt, object, OSD overlay frame of a single text will be returned when text overlay succeeded (the origin is the upper-left corner of the screen)-->
            <width>
                <!--ro, req, int, width, range:[0,1000]-->0
            </width>
            <height>
                <!--ro, req, int, height, range:[0,1000]-->0
            </height>
        </OverlayRegion>
    </TextOverlay>
</TextOverlayList>
<DateTimeOverlay>
    <!--ro, opt, object, channel time information overlay-->
    <enabled>
        <!--ro, req, bool, whether to enable-->true
    </enabled>
    <positionX>
        <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>
    <positionY>
        <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionY>
    <dateStyle>
        <!--ro, opt, enum, date format, subType:string, desc:"YYYY-MM-DD", "MM-DD-YYYY", "DD-MM-YYYY", "CHR-YYYY-MM-DD", "CHR-MM-DD-YYYY", "CHR-DD-MM-YYYY", "CHR-YYYY/MM/DD", "CHR-MM/DD/YYYY", "CHR-DD/MM/YYYY"-->YYYY-MM-DD
    </dateStyle>
    <timeStyle>
        <!--ro, opt, enum, time format, subType:string, desc:"12hour", "24hour"-->12hour
    </timeStyle>
    <displayWeek>
        <!--ro, opt, bool, whether to display week information-->true
    </displayWeek>
    <displayDate>
        <!--ro, opt, bool, whether to display the date (year-month-day)-->true
    </displayDate>
    <displayTime>
        <!--ro, opt, bool, whether to display the time (hour-minute-second)-->true
    </displayTime>
    <OverlayRegion>
        <!--ro, opt, object, OSD overlay frame of a single text will be returned when text overlay succeeded (the origin is the upper-left corner of the screen)-->
        <width>
            <!--ro, req, int, width, range:[0,1000]-->0
        </width>
        <height>
            <!--ro, req, int, height, range:[0,1000]-->0
        </height>
    </OverlayRegion>
</DateTimeOverlay>
<channelNameOverlay>
    <!--ro, opt, object, channel name overlay-->
    <enabled>
        <!--ro, req, bool, whether to enable channel name overlay-->true
    </enabled>
    <positionX>
        <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>
    <positionY>
        <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionY>

```

```

</positionY>
<OverlayRegion>
    <!--ro, opt, object, OSD overlay frame of a single text will be returned when text overlay succeeded (the origin is the upper-left corner of the screen)-->
    <width>
        <!--ro, req, int, width, range:[0,1000]-->0
    </width>
    <height>
        <!--ro, req, int, height, range:[0,1000]-->0
    </height>
</OverlayRegion>
<channelNameOverlay>
<HumitureOverlay>
    <!--ro, opt, object, temperature and humidity-->
    <enabled>
        <!--ro, req, bool, whether to display temperature and humidity-->true
    </enabled>
    <updateInterval>
        <!--ro, opt, int, update interval-->1
    </updateInterval>
    <positionX>
        <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>
    <positionY>
        <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionY>
</HumitureOverlay>
<fontSize>
    <!--ro, opt, enum, font size, subType:string, desc:"adaptive", "16*16", "32*32", "48*48", "64*64", "96*96", "128*128"-->96*96
</fontSize>
<frontColorMode>
    <!--ro, opt, enum, font color mode, subType:string, desc:"auto", "customize", "outline"-->auto
</frontColorMode>
<frontColor>
    <!--ro, opt, string, font color, desc:hexadecimal digit-->000000
</frontColor>
<BatteryPowerOverlay>
    <!--ro, opt, object, battery power information overlay-->
    <enabled>
        <!--ro, req, bool, whether to enable the function-->true
    </enabled>
    <positionX>
        <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>
    <positionY>
        <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionY>
</BatteryPowerOverlay>
<alignment>
    <!--ro, opt, enum, alignment mode, subType:string, desc:"customize" (custom), "alignRight" (right align), "alignLeft" (left align), "GB", "adsorption" (adsorption mode)-->alignRight
</alignment>
<publicSecurity>
    <!--ro, opt, bool-->true
</publicSecurity>
<DeviceStatusDisplay>
    <!--ro, opt, object, whether to display the device status-->
    <batteryCapacityDisplay>
        <!--ro, opt, bool, whether to display the battery capacity-->true
    </batteryCapacityDisplay>
    <chargingStatusDisplay>
        <!--ro, opt, bool, whether to display the charging status-->true
    </chargingStatusDisplay>
    <bluetoothStatusDisplay>
        <!--ro, opt, bool, whether to display the bluetooth status-->true
    </bluetoothStatusDisplay>
    <dialStatusDisplay>
        <!--ro, opt, bool, whether to display the dial-up status-->true
    </dialStatusDisplay>
</DeviceStatusDisplay>
<boundary>
    <!--ro, opt, int, the minimum boundary, range:[0,2], dep:and,{$.VideoOverlay.alignment,eq,GB}, desc:unit: number of characters-->1
</boundary>
<FloorNumOverlay>
    <!--ro, opt, object, elevator floor No.-->
    <enabled>
        <!--ro, req, bool, whether to display the elevator floor No.-->true
    </enabled>
    <positionX>
        <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>
    <positionY>
        <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionY>
</FloorNumOverlay>
<upDownBoundary>
    <!--ro, opt, int, range:[0,2]-->1
</upDownBoundary>

```

```

<!-->
<leftRightBoundary>
  <!--ro, opt, int, range:[0,2]-->1
</leftRightBoundary>
<Angle>
  <!--ro, opt, object, device angle-->
<TiltAngle>
  <!--ro, opt, object, pitch angle-->
<enabled>
  <!--ro, req, bool, whether to enable-->true
</enabled>
<positionX>
  <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</TiltAngle>
<MultiAngleOverlay>
  <!--ro, opt, object, multiple angles information overlay, desc:pitch angle, range, and rotation angle-->
<enabled>
  <!--ro, req, bool, whether to enable-->true
</enabled>
<positionX>
  <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</MultiAngleOverlay>
<Angle>
<GPSOverlay>
  <!--ro, opt, object, GPS information overlay-->
<enabled>
  <!--ro, req, bool, whether to enable-->true
</enabled>
<positionX>
  <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</GPSOverlay>
<PackingSpaceRecognitionOverlay>
  <!--ro, opt, object, display of parking space detection result-->
<isParkedEnabled>
  <!--ro, opt, bool, whether to display the parking space status (occupied or free)-->true
</isParkedEnabled>
</PackingSpaceRecognitionOverlay>
<WirelessDialOverlay>
  <!--ro, opt, object, wireless dial-up status overlay-->
<enabled>
  <!--ro, req, bool, whether to enable. If it is enabled, the information such as wireless network type (5G/4G), upstream speed, and downstream speed
will be overlaid-->true
</enabled>
<positionX>
  <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</WirelessDialOverlay>
<MethaneDetectionOverlay>
  <!--ro, opt, object, methane (CH4) detection information overlay-->
<enabled>
  <!--ro, req, bool, whether to enable. If it is enabled, the information such as methane density and Light intensity will be overlaid-->true
</enabled>
<positionX>
  <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</MethaneDetectionOverlay>
<WaterLevelDetectionOverlay>
  <!--ro, opt, object, water level monitoring-->
<enabled>
  <!--ro, req, bool, whether to display the water level height-->true
</enabled>
<positionX>
  <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>

```

```

<!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</WaterLevelDetectionOverlay>
<visibilityDisplay>
<!--ro, opt, object, visibility information overlay-->
<enabled>
<!--ro, req, bool, whether to enable. If it is enabled, the visibility information will be overlaid-->true
</enabled>
<positionX>
<!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
<!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</VisibilityDisplay>
<RoadSurfaceStatusDisplay>
<!--ro, opt, object, road status overlay-->
<enabled>
<!--ro, req, bool, whether to enable. If it is enabled, the road status information will be overlaid-->true
</enabled>
<positionX>
<!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
<!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</RoadSurfaceStatusDisplay>
<FiveElementWeatherDisplay>
<!--ro, opt, object, 5-element weather overlay-->
<enabled>
<!--ro, req, bool, whether to enable. If it is enabled, the 5-element weather information will be overlaid-->true
</enabled>
<positionX>
<!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
<!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</FiveElementWeatherDisplay>
<MEFStatusDisplay>
<!--ro, opt, object-->
<enabled>
<!--ro, req, bool-->true
</enabled>
</MEFStatusDisplay>
<RainfallOverlay>
<!--ro, opt, object, rainfall information overlay-->
<enabled>
<!--ro, req, bool, whether to enable. If it is enabled, the rainfall information will be overlaid-->true
</enabled>
<positionX>
<!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
<!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</RainfallOverlay>
<WaterLevelThresholdOverlay>
<!--ro, opt, object-->
<enabled>
<!--ro, req, bool-->true
</enabled>
<positionX>
<!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
<!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</WaterLevelThresholdOverlay>
<SolarPanelOverlay>
<!--ro, opt, object-->
<enabled>
<!--ro, req, bool-->true
</enabled>
<positionX>
<!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
<!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</SolarPanelOverlay>

```

```

</SolarPanelOverlay>
<BatteryInfoOverlay>
    <!--ro, opt, object, battery information overlay, desc:display the output voltage and output current of battery-->
    <enabled>
        <!--ro, req, bool, whether to enable battery information overlay-->true
    </enabled>
    <positionX>
        <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>
    <positionY>
        <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionY>
</BatteryInfoOverlay>
<WirelessDataOverlay>
    <!--ro, opt, object-->
    <enabled>
        <!--ro, req, bool-->true
    </enabled>
    <positionX>
        <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>
    <positionY>
        <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionY>
</WirelessDataOverlay>
<AccelerationOverlay>
    <!--ro, opt, object, display the acceleration, desc:pitch acceleration and rotation acceleration-->
    <enabled>
        <!--ro, req, bool, whether to enable acceleration overlay-->true
    </enabled>
    <positionX>
        <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>
    <positionY>
        <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionY>
</AccelerationOverlay>
<FirmwareVersionOverlay>
    <!--ro, opt, object, device firmware version information-->
    <enabled>
        <!--ro, req, bool, whether to enable overlaying of device firmware version information-->true
    </enabled>
    <positionX>
        <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>
    <positionY>
        <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionY>
</FirmwareVersionOverlay>
<WifiOverlay>
    <!--ro, opt, object, Wi-Fi status information, desc:enable/disable-->
    <enabled>
        <!--ro, req, bool, whether to display the Wi-Fi status information-->true
    </enabled>
    <positionX>
        <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>
    <positionY>
        <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionY>
</WifiOverlay>
<StorageStatusOverlay>
    <!--ro, opt, object, storage status information overlay, desc:whether there is SD card, the remaining storage space-->
    <enabled>
        <!--ro, req, bool, whether to display the storage status information-->true
    </enabled>
    <positionX>
        <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>
    <positionY>
        <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionY>
</StorageStatusOverlay>
<RecordStatusOverlay>
    <!--ro, opt, object, recording status overlay, desc:enable/disable-->
    <enabled>
        <!--ro, req, bool, whether to enable the recording status overlay-->true
    </enabled>
    <positionX>
        <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>

```

```

<positionY>
  <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->
</positionY>
</RecordStatusOverlay>
<laserRangingOverlay>
  <!--ro, opt, object, Laser distance measurement-->
<enabled>
  <!--ro, req, bool, whether to enable-->true
</enabled>
<positionX>
  <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</laserRangingOverlay>
<sensorDataOverlay>
  <!--ro, opt, object, sensor data overlay-->
<positionX>
  <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</sensorDataOverlay>
<frameCentreOverlayEnabled>
  <!--ro, req, bool-->true
</frameCentreOverlayEnabled>
<elevatorRunningOverlay>
  <!--ro, opt, object-->
<enabled>
  <!--ro, req, bool-->true
</enabled>
<positionX>
  <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</elevatorRunningOverlay>
<elevatorPeopleNumOverlay>
  <!--ro, opt, object, overlay the number of passengers in an elevator-->
<enabled>
  <!--ro, req, bool, whether to enable overlaying the number of passengers in an elevator-->false
</enabled>
<positionX>
  <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</elevatorPeopleNumOverlay>
</VideoOverlay>

```

### 16.2.11.2 Set OSD parameters of a specific video input channel

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/overlays

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>

<VideoOverlay xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, text overlay, attr:version{req, string, protocolVersion}-->
<enabled>
  <!--opt, bool, whether to enable, desc:true (OSD is enabled), false (OSD is disabled). The default value is true-->true
</enabled>
<normalizedScreenSize>
  <!--req, object, screen size-->
<normalizedScreenWidth>

```

```

<!--req, int, normalized width-->1
</normalizedScreenWidth>
<normalizedScreenHeight>
  <!--req, int, normalized height-->1
</normalizedScreenHeight>
</normalizedScreenSize>
<attribute>
  <!--opt, object, attributes-->
  <transparent>
    <!--req, bool, whether it is transparent-->true
  </transparent>
  <flashing>
    <!--req, bool, whether it is flickering-->true
  </flashing>
  <backgroundColor>
    <!--opt, object, background color-->
    <enabled>
      <!--req, bool-->true
    </enabled>
    <color>
      <!--opt, string, background color RGB, desc:hexadecimal digit-->000000
    </color>
    <transparency>
      <!--opt, int, background color transparency, range:[0,100]-->50
    </transparency>
  </backgroundColor>
</attribute>
<TextOverlayList>
  <!--opt, array, text content list, subType:object-->
  <TextOverlay>
    <!--opt, object, text content-->
    <id>
      <!--req, string, text No.-->1
    </id>
    <enabled>
      <!--req, bool, whether to enable-->true
    </enabled>
    <positionX>
      <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionX>
    <positionY>
      <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
    </positionY>
    <displayText>
      <!--req, string, overlay text-->text
    </displayText>
    <isPersistentText>
      <!--opt, bool, whether the overlaid text remains the previous settings after device rebooting, desc:if the value is true or the node is not transmitted, it indicates to write OSD to flash, and it is not suitable for frequent or fast callings. If the value is false, it indicates not to writer OSD to flash and the OSD settings will not restore-->true
    </isPersistentText>
    <OverlayRegion>
      <!--opt, object, OSD overlay frame of a single text will be returned when text overlay succeeded (the origin is the upper-left corner of the screen)-->
      <width>
        <!--req, int, width, range:[0,1000]-->0
      </width>
      <height>
        <!--req, int, height, range:[0,1000]-->0
      </height>
    </OverlayRegion>
  </TextOverlay>
</TextOverlayList>
<DateTimeOverlay>
  <!--opt, object, channel time information overlay-->
  <enabled>
    <!--req, bool, whether to enable-->true
  </enabled>
  <positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
  <dateStyle>
    <!--opt, enum, date, subType:string, desc:"YYYY-MM-DD", "MM-DD-YYYY", "DD-MM-YYYY", "CHR-YYYY-MM-DD", "CHR-MM-DD-YYYY", "CHR-DD-MM-YYYY", "CHR-YYYY/MM/DD", "CHR-MM/DD/YYYY", "CHR-DD/MM/YYYY"-->YYYY-MM-DD
  </dateStyle>
  <timeStyle>
    <!--opt, enum, time format, subType:string, desc:time format-->12hour
  </timeStyle>
  <displayWeek>
    <!--opt, bool, whether to display week information-->true
  </displayWeek>
  <displayDate>
    <!--opt, bool, whether to display the date (year-month-day)-->true
  </displayDate>
  <displayTime>
    <!--opt, bool, whether to display the time (hour-minute-second)-->true
  </displayTime>
</DateTimeOverlay>

```

```

</displayFrame>
<OverlayRegion>
  <!--opt, object, OSD overlay frame of a single text will be returned when text overlay succeeded (the origin is the upper-left corner of the screen)-->
<>
  <width>
    <!--req, int, width, range:[0,1000]-->0
  </width>
  <height>
    <!--req, int, height, range:[0,1000]-->0
  </height>
</OverlayRegion>
</DateTimeOverlay>
<channelNameOverlay>
  <!--opt, object, channel name overlay-->
<enabled>
  <!--req, bool, whether to enable channel name overlay-->true
</enabled>
<positionX>
  <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
<OverlayRegion>
  <!--opt, object, OSD overlay frame of a single text will be returned when text overlay succeeded (the origin is the upper-left corner of the screen)-->
<>
  <width>
    <!--req, int, width, range:[0,1000]-->0
  </width>
  <height>
    <!--req, int, height, range:[0,1000]-->0
  </height>
</OverlayRegion>
</channelNameOverlay>
<HumitureOverlay>
  <!--opt, object, temperature and humidity-->
<enabled>
  <!--req, bool, whether to display temperature and humidity-->true
</enabled>
<updateInterval>
  <!--opt, int, update interval-->1
</updateInterval>
<positionX>
  <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</HumitureOverlay>
<fontSize>
  <!--opt, enum, font size, subType:string, desc:"adaptive", "16*16", "32*32", "48*48", "64*64", "96*96", "128*128"-->96*96
</fontSize>
<frontColorMode>
  <!--opt, enum, font color mode, subType:string, desc:"auto", "customize", "outline"-->auto
</frontColorMode>
<frontColor>
  <!--opt, string, font color, desc:hexadecimal digit-->000000
</frontColor>
<BatteryPowerOverlay>
  <!--opt, object, battery power information overlay-->
<enabled>
  <!--req, bool, whether to enable the function-->true
</enabled>
<positionX>
  <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</BatteryPowerOverlay>
<alignment>
  <!--opt, enum, alignment mode, subType:string, desc:"customize" (custom), "alignRight" (right align), "alignLeft" (left align), "GB", "adsorption" (adsorption mode)-->alignRight
</alignment>
<publicSecurity>
  <!--opt, bool-->true
</publicSecurity>
<DeviceStatusDisplay>
  <!--opt, object, whether to display the device status-->
<batteryCapacityDisplay>
  <!--opt, bool, whether to display the battery capacity-->true
</batteryCapacityDisplay>
<chargingStatusDisplay>
  <!--opt, bool, whether to display the charging status-->true
</chargingStatusDisplay>
<bluetoothStatusDisplay>
  <!--opt, bool, whether to display the bluetooth status-->true
</bluetoothStatusDisplay>

```

```

<dialStatusDisplay>
    <!--opt, bool, whether to display the dial-up status-->true
</dialStatusDisplay>
</DeviceStatusDisplay>
<boundary>
    <!--opt, int, range:[0,2], dep:and, ${.VideoOverlay.alignment},eq,GB}, desc:unit: number of characters-->000000
</boundary>
<FloorNumOverlay>
    <!--opt, object, elevator floor No.-->
<enabled>
    <!--req, bool, whether to display the elevator floor No.-->true
</enabled>
<positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</FloorNumOverlay>
<upDownBoundary>
    <!--opt, int, range:[0,2]-->1
</upDownBoundary>
<leftRightBoundary>
    <!--opt, int, range:[0,2]-->1
</leftRightBoundary>
<Angle>
    <!--opt, object, device angle-->
<TiltAngle>
    <!--opt, object, pitch angle-->
<enabled>
    <!--req, bool, whether to enable-->true
</enabled>
<positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
<MultiAngleOverlay>
    <!--opt, object, multiple angles information overlay, desc:pitch angle, range, and rotation angle-->
<enabled>
    <!--req, bool, whether to enable-->true
</enabled>
<positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
<MultiAngleOverlay>
<Angle>
<GPSOverlay>
    <!--opt, object, GPS information overlay-->
<enabled>
    <!--req, bool, whether to enable-->true
</enabled>
<positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</GPSOverlay>
<ParkingSpaceRecognitionOverlay>
    <!--opt, object, display of parking space detection result-->
<isParkedEnabled>
    <!--opt, bool, whether to display the parking space status (occupied or free)-->true
</isParkedEnabled>
<ParkingSpaceRecognitionOverlay>
<WirelessDialOverlay>
    <!--opt, object, wireless dial-up status overlay-->
<enabled>
    <!--req, bool, whether to enable. If it is enabled, the information such as wireless network type (5G/4G), upstream speed, and downstream speed will be overlaid-->true
</enabled>
<positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</WirelessDialOverlay>

```

```

<MethaneDetectionOverlay>
  <!--opt, object, methane (CH4) detection information overlay-->
  <enabled>
    <!--req, bool, whether to enable. If it is enabled, the information such as methane density and light intensity will be overlaid-->true
  </enabled>
  <positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
</MethaneDetectionOverlay>
<WaterLevelDetectionOverlay>
  <!--opt, object, water level monitoring-->
  <enabled>
    <!--req, bool, whether to display the water level height-->true
  </enabled>
  <positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
</WaterLevelDetectionOverlay>
<VisibilityDisplay>
  <!--opt, object, visibility information overlay-->
  <enabled>
    <!--req, bool, whether to enable. If it is enabled, the visibility information will be overlaid-->true
  </enabled>
  <positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
</VisibilityDisplay>
<RoadSurfaceStatusDisplay>
  <!--opt, object, road status overlay-->
  <enabled>
    <!--req, bool, whether to enable. If it is enabled, the road status information will be overlaid-->true
  </enabled>
  <positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
</RoadSurfaceStatusDisplay>
<FiveElementWeatherDisplay>
  <!--opt, object, 5-element weather overlay-->
  <enabled>
    <!--req, bool, whether to enable. If it is enabled, the 5-element weather information will be overlaid-->true
  </enabled>
  <positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
</FiveElementWeatherDisplay>
<MEFStatusDisplay>
  <!--opt, object-->
  <enabled>
    <!--req, bool-->true
  </enabled>
</MEFStatusDisplay>
<RainfallOverlay>
  <!--opt, object, rainfall information overlay-->
  <enabled>
    <!--req, bool, whether to enable. If it is enabled, the rainfall information will be overlaid-->true
  </enabled>
  <positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
</RainfallOverlay>
<WaterLevelThresholdOverlay>
  <!--opt, object-->
  <enabled>
    <!--req, bool-->true
  </enabled>

```

```

</enabled>
<positionX>
  <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</WaterLevelThresholdOverlay>
<SolarPanelOverlay>
<!--opt, object-->
<enabled>
  <!--req, bool-->true
</enabled>
<positionX>
  <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</SolarPanelOverlay>
<BatteryInfoOverlay>
<!--opt, object, desc:display the output voltage and output current of battery-->
<enabled>
  <!--req, bool, whether to enable battery information overlay-->true
</enabled>
<positionX>
  <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</BatteryInfoOverlay>
<WirelessDataOverlay>
<!--opt, object-->
<enabled>
  <!--req, bool-->true
</enabled>
<positionX>
  <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</WirelessDataOverlay>
<AccelerationOverlay>
<!--opt, object, display the acceleration-->
<enabled>
  <!--req, bool, whether to enable acceleration overlay-->true
</enabled>
<positionX>
  <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</AccelerationOverlay>
<FirmwareVersionOverlay>
<!--opt, object, device firmware version information-->
<enabled>
  <!--req, bool, whether to enable overlaying of device firmware version information-->true
</enabled>
<positionX>
  <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>
</FirmwareVersionOverlay>
<WifiOverlay>
<!--opt, object, Wi-Fi status information, desc:enable/disable-->
<enabled>
  <!--req, bool, whether to display the Wi-Fi status information-->true
</enabled>
<positionX>
  <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionX>
<positionY>
  <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of
normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
</positionY>

```

```

</WifiOverlay>
<StorageStatusOverlay>
  <!--opt, object, storage status information overlay, desc:whether there is SD card, the remaining storage space-->
  <enabled>
    <!--req, bool, whether to display the storage status information-->true
  </enabled>
  <positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
</StorageStatusOverlay>
<RecordStatusOverlay>
  <!--opt, object, recording status overlay, desc:enable/disable-->
  <enabled>
    <!--req, bool, whether to enable the recording status overlay-->true
  </enabled>
  <positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
</RecordStatusOverlay>
<laserRangingOverlay>
  <!--opt, object, Laser distance measurement-->
  <enabled>
    <!--req, bool, whether to enable-->true
  </enabled>
  <positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
</laserRangingOverlay>
<sensorDataOverlay>
  <!--opt, object, sensor data overlay-->
  <positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
</sensorDataOverlay>
<frameCentreOverlayEnabled>
  <!--opt, bool-->true
</frameCentreOverlayEnabled>
<elevatorRunningOverlay>
  <!--ro, opt, object-->
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
  <positionX>
    <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
</elevatorRunningOverlay>
<elevatorPeopleNumOverlay>
  <!--ro, opt, object, overlay the number of passengers in an elevator-->
  <enabled>
    <!--ro, req, bool, whether to enable overlaying the number of passengers in an elevator-->false
  </enabled>
  <positionX>
    <!--ro, req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen. The range is the same with that of normalizedScreenSize. Default value: x[0,704] y[0,576]-->1
  </positionY>
</elevatorPeopleNumOverlay>
</VideoOverlay>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL, range:[0,1024]-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, desc:sub status code-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, desc:sub status code-->OK
    </subStatusCode>
    <description>
      <!--ro, opt, string, custom error information description, range:[0,1024], desc:the detailed information of custom error returned by device applications, it is used for fast debugging-->badXmlFormat
    </description>
  </statusCode>
</ResponseStatus>

```

### 16.2.11.3 Get the OSD capability of a specific video input channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/overlays/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<VideoOverlay xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, see details in the message of XML_BatteryPowerOverlay, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, opt, bool-->true
  </enabled>
  <normalizedScreenSize>
    <!--ro, req, object-->
    <normalizedScreenWidth>
      <!--ro, req, int, normalized width-->1
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--ro, req, int, normalized height-->1
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <attribute>
    <!--ro, opt, object, attributes-->
    <transparent opt="true,false">
      <!--ro, req, bool, whether it is transparent, attr:opt{req, string}-->true
    </transparent>
    <flashing opt="true,false">
      <!--ro, req, bool, attr:opt{req, string}-->true
    </flashing>
  </attribute>
  <TextOverlayList size="8">
    <!--ro, opt, array, subType:object, attr:size{req, int}-->
    <TextOverlay>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string, listening host ID-->1
      </id>
      <enabled>
        <!--ro, req, bool, whether to enable-->true
      </enabled>
      <positionX>
        <!--ro, req, int, range:[0,704]-->1
      </positionX>
      <positionY>
        <!--ro, req, int, range:[0,576]-->1
      </positionY>
      <displayText>
        <!--ro, req, string-->test
      </displayText>
    </TextOverlay>
  </TextOverlayList>
</VideoOverlay>

```

```
</displayText>
<isPersistentText>
  <!--ro, opt, bool-->true
</isPersistentText>
<isEditText>
  <!--ro, opt, bool-->true
</isEditText>
<isSupportOverlayRegion>
  <!--ro, opt, bool-->true
</isSupportOverlayRegion>
</TextOverlay>
</TextOverlayList>
<DateTimeOverlay>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, whether to enable, attr:opt{req, string}-->true
  </enabled>
  <positionX>
    <!--ro, req, int, range:[0,704]-->1
  </positionX>
  <positionY>
    <!--ro, req, int, range:[0,576]-->1
  </positionY>
  <dateStyle opt="YYYY-MM-DD,MM-DD-YYYY,DD-MM-YYYY,CHR-YYYY-MM-DD,CHR-MM-DD-YYYY,CHR-DD-MM-YYYY,CHR-YYYY/MM/DD,CHR-MM/DD/YYYY,CHR-DD/MM/YYYY">
    <!--ro, opt, string, attr:opt{req, string}-->YYYY-MM-DD
  </dateStyle>
  <timeStyle opt="12hour,24hour">
    <!--ro, opt, string, time format, attr:opt{req, string}-->12hour
  </timeStyle>
  <displayWeek opt="true,false">
    <!--ro, opt, bool, attr:opt{req, string}-->true
  </displayWeek>
  <displayDate>
    <!--ro, opt, bool-->true
  </displayDate>
  <displayTime>
    <!--ro, opt, bool-->true
  </displayTime>
  <isSupportOverlayRegion>
    <!--ro, opt, bool-->true
  </isSupportOverlayRegion>
</DateTimeOverlay>
<channelNameOverlay>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <positionX>
    <!--ro, req, int, range:[0,704]-->1
  </positionX>
  <positionY>
    <!--ro, req, int, range:[0,576]-->1
  </positionY>
  <isSupportOverlayRegion>
    <!--ro, opt, bool-->true
  </isSupportOverlayRegion>
</channelNameOverlay>
<HumitureOverlay>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <updateInterval min="1" max="10">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
  </updateInterval>
  <positionX min="0" max="704">
    <!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
  </positionX>
  <positionY min="0" max="576">
    <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
  </positionY>
</HumitureOverlay>
<fontSize opt="16*16,32*32,48*48,64*64,96*96,adaptive">
  <!--ro, opt, string, font size, attr:opt{req, string}-->adaptive
</fontSize>
<frontColorMode opt="auto,customize,outline" def="auto">
  <!--ro, opt, string, font color mode, attr:opt{req, string},def{req, string}-->outline
</frontColorMode>
<frontColor>
  <!--ro, opt, string, font color-->#000000
</frontColor>
<BatteryPowerOverlay>
  <!--ro, opt, object-->
  <enabled>
    <!--ro, req, bool, whether to enable the function-->true
  </enabled>
  <positionX>
    <!--ro, req, int, range:[0,704]-->1
  </positionX>
  <positionY>
    <!--ro, req, int, range:[0,576]-->1
  </positionY>
</BatteryPowerOverlay>
<DeviceStatusDisplay>
```

```

<!--ro, opt, object-->
<batteryCapacityDisplay>
  <!--ro, opt, bool-->true
</batteryCapacityDisplay>
<chargingStatusDisplay>
  <!--ro, opt, bool-->true
</chargingStatusDisplay>
<bluetoothStatusDisplay>
  <!--ro, opt, bool-->true
</bluetoothStatusDisplay>
<dialStatusDisplay>
  <!--ro, opt, bool-->true
</dialStatusDisplay>
</DeviceStatusDisplay>
<alignment opt="customize,alignRight,alignLeft,GB,allRight,allLeft,adsorption">
  <!--ro, opt, string, alignment mode, attr:opt{req, string}-->customize
</alignment>
<lowRightTextNumber min="1" max="6">
  <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
</lowRightTextNumber>
<lowLeftTextNumber min="7" max="8">
  <!--ro, opt, int, attr:min{req, int},max{req, int}-->7
</lowLeftTextNumber>
<boundary min="0" max="2">
  <!--ro, opt, int, dep:and,{$.VideoOverlay.alignment,eq,GB}, attr:min{req, int},max{req, int}-->1
</boundary>
<TextOverlaySize min="0" max="0">
  <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
</TextOverlaySize>
<FloorNumOverlay>
  <!--ro, opt, object-->
<enabled>
  <!--ro, req, bool-->true
</enabled>
<positionX>
  <!--ro, req, int, range:[0,704]-->1
</positionX>
<positionY>
  <!--ro, req, int, range:[0,576]-->1
</positionY>
</FloorNumOverlay>
<upDownboundary>
  <!--ro, opt, int-->1
</upDownboundary>
<leftRightboundary>
  <!--ro, opt, int-->1
</leftRightboundary>
<Angle>
  <!--ro, opt, object-->
<TiltAngle>
  <!--ro, opt, object-->
<enabled opt="true,false">
  <!--ro, req, bool, whether to enable, attr:opt{req, string}-->true
</enabled>
<positionX min="0" max="704">
  <!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
</positionX>
<positionY min="0" max="576">
  <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
</positionY>
</TiltAngle>
<MultiAngleOverlay>
  <!--ro, opt, object-->
<enabled opt="true,false">
  <!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<positionX min="0" max="704">
  <!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
</positionX>
<positionY min="0" max="576">
  <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
</positionY>
</MultiAngleOverlay>
</Angle>
<isSupportTextOverlayRegions>
  <!--ro, opt, bool-->true
</isSupportTextOverlayRegions>
<GPSOverlay>
  <!--ro, opt, object-->
<enabled opt="true,false">
  <!--ro, req, bool, whether to enable, attr:opt{req, string}-->true
</enabled>
<positionX min="0" max="704">
  <!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
</positionX>
<positionY min="0" max="576">
  <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
</positionY>
</GPSOverlay>
<PackingSpaceRecognitionOverlay>
  <!--ro, opt, object-->
<isParkedEnabled opt="true,false">
  <!--ro, opt, bool, attr:opt{req, string}-->true

```

```

</isParkedEnabled>
<PackingSpaceRecognitionOverlay>
<WirelessDialOverlay>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <positionX min="0" max="704">
    <!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
  </positionX>
  <positionY min="0" max="576">
    <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
  </positionY>
</WirelessDialOverlay>
<MethaneDetectionOverlay>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <positionX min="0" max="704">
    <!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
  </positionX>
  <positionY min="0" max="576">
    <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
  </positionY>
</MethaneDetectionOverlay>
<WaterLevelDetectionOverlay>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <positionX min="0" max="704">
    <!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
  </positionX>
  <positionY min="0" max="576">
    <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
  </positionY>
</WaterLevelDetectionOverlay>
<objectAccumulationDetectionOverlay opt="true,false">
  <!--ro, opt, bool, attr:opt{req, string}-->true
</objectAccumulationDetectionOverlay>
<visibilityDisplay>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <positionX min="0" max="704">
    <!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
  </positionX>
  <positionY min="0" max="576">
    <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
  </positionY>
</visibilityDisplay>
<RoadSurfaceStatusDisplay>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <positionX min="0" max="704">
    <!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
  </positionX>
  <positionY min="0" max="576">
    <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
  </positionY>
</RoadSurfaceStatusDisplay>
<FiveElementWeatherDisplay>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <positionX min="0" max="704">
    <!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
  </positionX>
  <positionY min="0" max="576">
    <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
  </positionY>
</FiveElementWeatherDisplay>
<MEFStatusDisplay>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
</MEFStatusDisplay>
<RainfallOverlay>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <positionX min="0" max="704">
    <!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
  </positionX>
  <positionY min="0" max="576">
    <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
  </positionY>

```

```

</positionX>
</RainfallOverlay>
<WaterLevelThresholdOverlay>
<!--ro, opt, object-->
<enabled opt="true,false">
<!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<positionX min="0" max="704">
<!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
</positionX>
<positionY min="0" max="576">
<!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
</positionY>
</WaterLevelThresholdOverlay>
<SolarPanelOverlay>
<!--ro, opt, object-->
<enabled opt="true,false">
<!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<positionX min="0" max="704">
<!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
</positionX>
<positionY min="0" max="576">
<!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
</positionY>
</SolarPanelOverlay>
<BatteryInfoOverlay>
<!--ro, opt, object-->
<enabled opt="true,false">
<!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<positionX min="0" max="704">
<!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
</positionX>
<positionY min="0" max="576">
<!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
</positionY>
</BatteryInfoOverlay>
<WirelessDataOverlay>
<!--ro, opt, object-->
<enabled opt="true,false">
<!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<positionX min="0" max="704">
<!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
</positionX>
<positionY min="0" max="576">
<!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
</positionY>
</WirelessDataOverlay>
<AccelerationOverlay>
<!--ro, opt, object-->
<enabled opt="true,false">
<!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<positionX min="0" max="704">
<!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
</positionX>
<positionY min="0" max="576">
<!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
</positionY>
</AccelerationOverlay>
<FirmwareVersionOverlay>
<!--ro, opt, object-->
<enabled opt="true,false">
<!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<positionX min="0" max="704">
<!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
</positionX>
<positionY min="0" max="576">
<!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
</positionY>
</FirmwareVersionOverlay>
<WifiOverlay>
<!--ro, opt, object-->
<enabled opt="true,false">
<!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<positionX min="0" max="704">
<!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
</positionX>
<positionY min="0" max="576">
<!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
</positionY>
</WifiOverlay>
<StorageStatusOverlay>
<!--ro, opt, object-->
<enabled opt="true,false">
<!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<positionX min="0" max="704">
<!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
</positionX>

```

```

<positionY min="0" max="576">
  <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
</positionY>
</StorageStatusOverlay>
<RecordStatusOverlay>
  <!--ro, opt, object-->
<enabled opt="true,false">
  <!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<positionX min="0" max="704">
  <!--ro, req, int, range:[0,704], attr:min{req, int},max{req, int}-->1
</positionX>
<positionY min="0" max="576">
  <!--ro, req, int, range:[0,576], attr:min{req, int},max{req, int}-->1
</positionY>
</RecordStatusOverlay>
</VideoOverlay>

```

#### 16.2.11.4 Get the channel name overlay parameters of a specified video input channel

##### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/overlays/channelNameOverlay

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<channelNameOverlay xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, channel name overlay parameters, attr:version{req, string, protocolVersion}-->
<enabled>
  <!--ro, req, bool, whether to enable channel name overlay-->true
</enabled>
<positionX>
  <!--ro, req, int, X-coordinate-->1
</positionX>
<positionY>
  <!--ro, req, int, Y-coordinate-->1
</positionY>
</channelNameOverlay>

```

#### 16.2.11.5 Set the channel name overlay parameters for a specified video input channel

##### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/overlays/channelNameOverlay

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<channelNameOverlay xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
<enabled>
  <!--req, bool, whether to enable channel name overlay-->true
</enabled>
<positionX>
  <!--req, int, X-coordinate-->1
</positionX>
<positionY>
  <!--req, int, Y-coordinate-->1
</positionY>
</channelNameOverlay>

```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

## 16.2.11.6 Set the parameters of date and time information overlay for a specified video input channel

### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/overlays/dateTimeOverlay

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<DateTimeOverlay xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, configurations of date and time information overlay of video input channels, attr:version{req, string, protocolVersion}-->
</DateTimeOverlay>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

## 16.2.11.7 Get the parameters of date and time information overlay of a specific channel

### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/overlays/dateTimeOverlay

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<DateTimeOverlay xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
</DateTimeOverlay>
```

### 16.2.11.8 Delete the text overlay parameters of a specific video input channel

#### Request URL

DELETE /ISAPI/System/Video/inputs/channels/<channelID>/overlays/text

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code description-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.11.9 Set the text overlay parameters of a specific video input channel

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/overlays/text

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>

<TextOverlayList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, see details in the message of XML_TextOverlay, subType:object, attr:version{opt, string, protocolVersion}-->
  <TextOverlay>
    <!--opt, object-->
    <id>
      <!--req, string, text No.-->test
    </id>
    <enabled>
      <!--req, bool-->true
    </enabled>
    <positionX>
      <!--req, int, X-coordinate-->1
    </positionX>
    <positionY>
      <!--req, int, Y-coordinate-->1
    </positionY>
    <displayText>
      <!--req, string, overlay character-->test
    </displayText>
    <OverlayRegion>
      <!--opt, object-->
      <width>
        <!--req, int, width, range:[0,1000]-->0
      </width>
      <height>
        <!--req, int, height, range:[0,1000]-->0
      </height>
    </OverlayRegion>
  </TextOverlay>
</TextOverlayList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.11.10 Get parameters of text information overlay of a specified video input channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/overlays/text

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<TextOverlayList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, text information overlay, subType:object, attr:version{req, string, protocolVersion}-->
  <TextOverlay>
    <!--ro, opt, object, text overlay configuration-->
    <id>
      <!--ro, req, string, text No.-->test
    </id>
    <enabled>
      <!--ro, req, bool, whether to enable-->true
    </enabled>
    <positionX>
      <!--ro, req, int, X-coordinate of the upper-left corner of the overlaid position, range:[0,704], desc:the origin is the lower-left corner of the screen-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y-coordinate of the upper-left corner of the overlaid position, range:[0,576], desc:the origin is the lower-left corner of the screen-->1
    </positionY>
    <displayText>
      <!--ro, req, string, overlaid text-->test
    </displayText>
    <OverlayRegion>
      <!--ro, opt, object-->
      <width>
        <!--ro, req, int, width, range:[0,1000]-->0
      </width>
      <height>
        <!--ro, req, int, height, range:[0,1000]-->0
      </height>
    </OverlayRegion>
  </TextOverlay>
</TextOverlayList>

```

### 16.2.11.11 Set the parameters of a specific piece of text overlay information of a specific video input channel

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/overlays/text/<textID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
textID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<TextOverlay xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{opt, string, protocolVersion}-->
  <id>
    <!--req, string, text No.-->test
  </id>
  <enabled>
    <!--req, bool-->true
  </enabled>
  <positionX>
    <!--req, int, x-coordinate, range:[0,704], desc:the origin is the lower-left corner of the screen-->1
  </positionX>
  <positionY>
    <!--req, int, y-coordinate, range:[0,576], desc:the origin is the lower-left corner of the screen-->1
  </positionY>
  <displayText>
    <!--req, string, overlay character-->test
  </displayText>
</TextOverlay>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
      <statusString>
        <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
        <subStatusCode>
          <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
        </subStatusCode>
      </statusString>
    </statusCode>
  </statusCode>
</ResponseStatus>

```

### 16.2.11.12 Get the text overlay parameters of a specified video input channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/overlays/text/<textID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
textID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<TextOverlay xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, text overlay configuration, attr:version{opt, string, protocolVersion}-->
  <id>
    <!--ro, req, string, text No.-->test
  </id>
  <enabled>
    <!--ro, req, bool, whether to enable-->true
  </enabled>
  <positionX>
    <!--ro, req, int, X-coordinate of the upper-left corner of the overlaid position, range:[0,704], desc:the origin is the Lower-Left corner of the screen-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y-coordinate of the upper-left corner of the overlaid position, range:[0,576], desc:the origin is the Lower-Left corner of the screen-->1
  </positionY>
  <displayText>
    <!--ro, req, string, overlaid text-->test
  </displayText>
</TextOverlay>

```

### 16.2.11.13 Delete a specified piece of text overlay of a video input channel

#### Request URL

DELETE /ISAPI/System/Video/inputs/channels/<channelID>/overlays/text/<textID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
textID	string	--

#### Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

## 16.2.11.14 Get the OSD language

### Request URL

GET /ISAPI/System/Video/inputs/OSDLanguage

### Query Parameter

None

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Language xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, OSD Language, attr:version{req, string, protocolVersion}-->
  <type>
    <!--ro, req, enum, Language type, subType:string, desc:"GBK", "EUC-KR"-->GBK
  </type>
</Language>
```

## 16.2.11.15 Set OSD language parameters

### Request URL

PUT /ISAPI/System/Video/inputs/OSDLanguage

### Query Parameter

None

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Language xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <type>
    <!--req, enum, "GBK,EUC-KR",def="GBK", subType:string, desc:"GBK", "EUC-KR"-->GBK
  </type>
</Language>
```

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

## 16.2.12 Privacy Mask

### 16.2.12.1 Get the privacy mask configuration capability of a specified channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/privacyMask/privacyMaskCap

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PrivacyMaskCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, the privacy mask configuration capability, attr:version{req, string, protocolVersion}-->
  <maskTypeDescriptor>
    <!--ro, opt, enum, mask type, subType:string, desc:"gray", "red", "yellow", "blue", "orange", "green", "transparent", "half-transparent", "mosaic", "black"-->gray
  </maskTypeDescriptor>
  <videoPrivacyType opt="privacyMask,privacyCover">
    <!--ro, opt, enum, privacy type, subType:string, attr:opt{req, string}, desc:"privacyMask"-privacy mask, "privacyCover"-video tampering-->privacyMask
  </videoPrivacyType>
</PrivacyMaskCap>

```

### 16.2.12.2 Get the parameters of all privacy mask areas of a specified channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/privacyMask/regions

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PrivacyMaskRegionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, privacy mask area configurations, subType:object, attr:version{req, string, protocolVersion}-->
</PrivacyMaskRegionList>

```

### 16.2.12.3 Set the parameters of all privacy mask areas of a specified channel

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/privacyMask/regions

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<PrivacyMaskRegionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, array, privacy mask area configurations, subType:object, attr:version{req, string, protocolVersion}-->
</PrivacyMaskRegionList>
```

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
    <requestURL>
        <!--ro, req, string, request URL-->null
    </requestURL>
    <statusCode>
        <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    </statusCode>
    <statusString>
        <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusString>
    <subStatusCode>
        <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
</ResponseStatus>
```

### 16.2.12.4 Delete parameters of all privacy mask areas of the specified channel

#### Request URL

DELETE /ISAPI/System/Video/inputs/channels/<channelID>/privacyMask/regions

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
    <requestURL>
        <!--ro, req, string, request URL-->null
    </requestURL>
    <statusCode>
        <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    </statusCode>
    <statusString>
        <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusString>
    <subStatusCode>
        <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
</ResponseStatus>
```

### 16.2.12.5 Get parameters of a privacy mask area of a specified channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/privacyMask/regions/<regionID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<PrivacyMaskRegion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, configurations of a privacy mask area, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, int, region ID-->1
  </id>
  <enabled>
    <!--ro, req, bool, whether to enable the area-->true
  </enabled>
  <RegionCoordinatesList>
    <!--ro, req, array, area coordinate List, subType:object, range:[0,4], desc:rectangle-->
    <RegionCoordinates>
      <!--ro, opt, object, the origin is the lower-left corner of the screen, desc:the origin is the lower-left corner of the screen-->
      <positionX>
        <!--ro, req, int, X-coordinate, range:[0,704]-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y-coordinate, range:[0,576]-->1
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</PrivacyMaskRegion>
```

### 16.2.12.6 Delete parameters of a privacy mask area of the specified channel

#### Request URL

DELETE /ISAPI/System/Video/inputs/channels/<channelID>/privacyMask/regions/<regionID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
  <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.12.7 Set parameters of a privacy mask area of a specified channel

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/privacyMask/regions/<regionID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PrivacyMaskRegion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, configurations of a privacy mask area, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, int, area ID-->1
  </id>
  <enabled>
    <!--req, bool, whether to enable the area-->true
  </enabled>
  <RegionCoordinatesList>
    <!--req, array, area coordinates List, subType:object, range:[0..4], desc:rectangle-->
    <RegionCoordinates>
      <!--opt, object, region coordinates, desc:the origin is the lower-left corner of the screen-->
      <position>
        <!--req, int, X-coordinate, range:[0..704]-->1
      </positionX>
      <positionY>
        <!--req, int, Y-coordinate, range:[0..576]-->1
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</PrivacyMaskRegion>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
  <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.12.8 Get the privacy mask parameters of a specific channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/privacyMask?switch=<switchType>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
switchType	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<PrivacyMask xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, privacy mask configurations, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool, whether to enable the privacy mask-->true
  </enabled>
  <normalizedScreenSize>
    <!--ro, opt, object, normalized size-->
    <normalizedScreenWidth>
      <!--ro, req, int, normalized width-->1
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--ro, req, int, normalized height-->1
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <PrivacyMaskRegionList size="8">
    <!--ro, opt, array, list of video tampering region(s), subType:object, attr:size{req, int}-->
    <PrivacyMaskRegion>
      <!--ro, opt, object, configurations of a privacy mask region-->
      <id>
        <!--ro, req, int, region ID-->1
      </id>
      <enabled>
        <!--ro, req, bool, whether to enable the region-->true
      </enabled>
      <RegionCoordinatesList>
        <!--ro, req, array, region coordinates list, subType:object, range:[0,4], desc:rectangle-->
        <RegionCoordinates>
          <!--ro, opt, object, region coordinates, desc:the origin is the lower-left corner of the screen-->
          <positionX>
            <!--ro, req, int, X-coordinate, range:[0,704]-->1
          </positionX>
          <positionY>
            <!--ro, req, int, Y-coordinate, range:[0,576]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </PrivacyMaskRegion>
  </PrivacyMaskRegionList>
</PrivacyMask>
```

### 16.2.12.9 Set the privacy mask parameters of a specified channel

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/privacyMask?switch=<switchType>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
switchType	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PrivacyMask xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, privacy mask configurations, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool, whether to enable the privacy mask-->true
  </enabled>
  <normalizedScreenSize>
    <!--opt, object, normalized size-->
    <normalizedScreenWidth>
      <!--req, int, normalized width-->1
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--req, int, normalized height-->1
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <PrivacyMaskRegionList size="8">
    <!--opt, array, list of video privacy mask area(s), subType:object, attr:size{req, int}-->
    <PrivacyMaskRegion>
      <!--opt, object, configurations of a privacy mask area-->
      <id>
        <!--req, int, area ID-->1
      </id>
      <enabled>
        <!--req, bool, whether to enable the area-->true
      </enabled>
      <RegionCoordinatesList>
        <!--req, array, area coordinates list, subType:object, range:[0,4], desc:rectangle-->
        <RegionCoordinates>
          <!--opt, object, area coordinates, desc:the origin is the lower-left corner of the screen-->
          <positionX>
            <!--req, int, X-coordinate, range:[0,704]-->1
          </positionX>
          <positionY>
            <!--req, int, Y-coordinate, range:[0,576]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </PrivacyMaskRegion>
  </PrivacyMaskRegionList>
</PrivacyMask>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

## 16.2.13 Real-Time Streaming

### 16.2.13.1 Get the Real-Time Transport Control Protocol (RTCP) parameters of a specified channel

#### Request URL

GET /ISAPI/Streaming/channels/<trackStreamID>/RTCP

#### Query Parameter

Parameter Name	Parameter Type	Description
trackStreamID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RTCP xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
</RTCP>

```

### 16.2.13.2 Set the Real-Time Transport Control Protocol (RTCP) parameters of a specified channel

#### Request URL

PUT /ISAPI/Streaming/channels/<trackStreamID>/RTCP

#### Query Parameter

Parameter Name	Parameter Type	Description
trackStreamID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RTCP xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool-->true
  </enabled>
</RTCP>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.14 Scene Change Detection

#### 16.2.14.1 Set the arming schedule of scene change detection for all channels

#### Request URL

PUT /ISAPI/Event/schedules/sceneChangeDetections

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<SceneChangeDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, subType:object, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--opt, object-->
    <id>
      <!--req, string-->sceneChangeDetection_video1
    </id>
    <eventType>
      <!--opt, enum, subType:string-->sceneChangeDetection
    </eventType>
    <videoInputChannelID>
      <!--opt, string-->1
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--req, array, subType:object, attr:size{opt, int}-->
      <TimeBlock>
        <!--opt, object-->
        <dayOfWeek>
          <!--opt, enum, subType:int-->1
        </dayOfWeek>
        <TimeRange>
          <!--req, object-->
          <beginTime>
            <!--req, time-->10:00:00
          </beginTime>
          <endTime>
            <!--req, time-->10:00:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>
</SceneChangeDetectionScheduleList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.14.2 Get the arming schedule of scene change detection for all channels

#### Request URL

GET /ISAPI/Event/schedules/sceneChangeDetections

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<SceneChangeDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, subType:object, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--ro, opt, object-->
    <id>
      <!--ro, req, string-->sceneChangeDetection_video1
    </id>
    <eventType>
      <!--ro, opt, enum, subType:string-->sceneChangeDetection
    </eventType>
    <videoInputChannelID>
      <!--ro, opt, string-->1
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--ro, req, array, subType:object, attr:size{opt, int}-->
      <TimeBlock>
        <!--ro, opt, object-->
        <dayOfWeek>
          <!--ro, opt, enum, subType:int-->1
        </dayOfWeek>
        <TimeRange>
          <!--ro, req, object-->
          <beginTime>
            <!--ro, req, time-->10:00:00
          </beginTime>
          <endTime>
            <!--ro, req, time-->10:00:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>
</SceneChangeDetectionScheduleList>

```

### 16.2.14.3 Get the arming schedule of scene change detection for a specified channel

#### Request URL

GET /ISAPI/Event/schedules/sceneChangeDetections/sceneChangeDetection\_video<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->sceneChangeDetection_video1
  </id>
  <eventType>
    <!--ro, opt, enum, subType:string-->sceneChangeDetection
  </eventType>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--ro, req, array, subType:object, attr:size{opt, int}-->
    <TimeBlock>
      <!--ro, opt, object-->
      <dayOfWeek>
        <!--ro, opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--ro, req, object-->
        <beginTime>
          <!--ro, req, time-->10:00:00
        </beginTime>
        <endTime>
          <!--ro, req, time-->10:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>

```

#### 16.2.14.4 Set the arming schedule of scene change detection for a specified channel

##### Request URL

PUT /ISAPI/Event/schedules/sceneChangeDetections/sceneChangeDetection\_video<channelID>

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->sceneChangeDetection_video1
  </id>
  <eventType>
    <!--opt, enum, subType:string-->sceneChangeDetection
  </eventType>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--req, array, subType:object, attr:size{opt, int}-->
    <TimeBlock>
      <!--opt, object-->
      <dayOfWeek>
        <!--opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--req, object-->
        <beginTime>
          <!--req, time-->10:00:00
        </beginTime>
        <endTime>
          <!--req, time-->10:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

## 16.2.15 Video Tampering

### 16.2.15.1 Set the arming schedule of video tampering detection for all channels

#### Request URL

PUT /ISAPI/Event/schedules/tamperDetections

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<TamperDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, subType:object, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--req, object-->
    <id>
      <!--req, string-->tamperedetection_video1
    </id>
    <eventType>
      <!--opt, string-->tamperDetection
    </eventType>
    <videoInputChannelID>
      <!--opt, string-->1
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--req, array, subType:object, attr:size{opt, int}-->
      <TimeBlock>
        <!--opt, object-->
        <dayOfWeek>
          <!--opt, enum, subType:int-->1
        </dayOfWeek>
        <TimeRange>
          <!--req, object-->
          <beginTime>
            <!--req, time-->10:00:00
          </beginTime>
          <endTime>
            <!--req, time-->10:00:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>
</TamperDetectionScheduleList>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.15.2 Get the arming schedule of video tampering detection for all channels

#### Request URL

GET /ISAPI/Event/schedules/tamperDetections

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<TamperDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, subType:object, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--ro, req, object-->
    <id>
      <!--ro, req, string-->tamperedetection_video1
    </id>
    <eventType>
      <!--ro, opt, string-->tamperDetection
    </eventType>
    <videoInputChannelID>
      <!--ro, opt, string-->1
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--ro, req, array, subType:object, attr:size{opt, int}-->
      <TimeBlock>
        <!--ro, opt, object-->
        <dayOfWeek>
          <!--ro, opt, enum, subType:int-->1
        </dayOfWeek>
        <TimeRange>
          <!--ro, req, object-->
          <beginTime>
            <!--ro, req, time-->10:00:00
          </beginTime>
          <endTime>
            <!--ro, req, time-->10:00:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>
</TamperDetectionScheduleList>

```

### 16.2.15.3 Get the arming schedule of video tampering detection for a specified channel

#### Request URL

GET /ISAPI/Event/schedules/tamperDetections/tamperedetection\_video<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->tamperdetection_video1
  </id>
  <eventType>
    <!--ro, opt, enum, subType:string-->tamperDetection
  </eventType>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--ro, req, array, subType:object, attr:size{opt, int}-->
    <TimeBlock>
      <!--ro, opt, object-->
      <dayOfWeek>
        <!--ro, opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--ro, req, object-->
        <beginTime>
          <!--ro, req, time-->10:00:00
        </beginTime>
        <endTime>
          <!--ro, req, time-->10:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>
```

## 16.2.15.4 Set the arming schedule of video tampering detection for a specified channel

### Request URL

PUT /ISAPI/Event/schedules/tamperDetections/tamperdetection\_video<channelID>

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->tamperdetection_video1
  </id>
  <eventType>
    <!--opt, string-->tamperDetection
  </eventType>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--req, array, subType:object, attr:size{opt, int}-->
    <TimeBlock>
      <!--opt, object-->
      <dayOfWeek>
        <!--opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--req, object-->
        <beginTime>
          <!--req, time-->10:00:00
        </beginTime>
        <endTime>
          <!--req, time-->10:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.15.5 Get the linkage parameters of video tampering detection for a specified channel

#### Request URL

GET /ISAPI/Event/triggers/tamper-<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->tamper-1
  </id>
  <eventType>
    <!--ro, req, string-->tamper
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subtype:object-->
    <EventTriggerNotification>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->test
      </id>
      <notificationMethod>
        <!--ro, req, enum, subType:string-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--ro, opt, enum, subType:string-->beginning
      </notificationRecurrence>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>
```

### 16.2.15.6 Delete the linkage of video tampering detection for a specified channel

#### Request URL

DELETE /ISAPI/Event/triggers/tamper-<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

## 16.2.15.7 Set the linkage parameters of video tampering detection for a specified channel

### Request URL

PUT /ISAPI/Event/triggers/tamper-<channelID>

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->tamper-1
  </id>
  <eventType>
    <!--req, string-->tamper
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
    <EventTriggerNotification>
      <!--opt, object-->
      <id>
        <!--req, string-->test
      </id>
      <notificationMethod>
        <!--req, enum, subType:string-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--opt, enum, subType:string-->beginning
      </notificationRecurrence>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>
```

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.15.8 Set the linkage triggering parameters of video tampering detection for a specified channel

#### Request URL

PUT /ISAPI/Event/triggers/tamper-<channelID>/notifications

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotification xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
</EventTriggerNotification>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.15.9 Get the linkage triggering parameters of video tampering detection for a specified channel

#### Request URL

GET /ISAPI/Event/triggers/tamper-<channelID>/notifications

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotification xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
</EventTriggerNotification>

```

### 16.2.15.10 Set the video tampering parameters of a single detection area for a specified channel

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/tamperDetection/regions/<regionID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<TamperDetectionRegion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, video tampering detection areas, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string, area ID-->test
  </id>
  <enabled>
    <!--opt, bool, whether to enable area detection-->true
  </enabled>
  <sensitivityLevel>
    <!--req, int, area detection sensitivity, range:[0,100]-->1
  </sensitivityLevel>
  <RegionCoordinatesList>
    <!--req, array, list of area coordinates, subType:object, range:[0,4], desc:rectangle-->
    <RegionCoordinates>
      <!--req, object, area coordinates, desc:the origin is the Lower-Left corner of the screen-->
      <positionX>
        <!--req, int, X-coordinate, range:[0,704]-->1
      </positionX>
      <positionY>
        <!--req, int, Y-coordinate, range:[0,576]-->1
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</TamperDetectionRegion>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.2.15.11 Video tampering alarm

#### EventType:shelteralarm

```

<?xml version="1.0" encoding="UTF-8"?>

<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, event details, attr:version{req, string, protocolVersion}-->
  <ipAddress>
    <!--ro, opt, string, IPv4 Address-->10.10.10.10
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPv6 Address-->ABCD:EF01:2345:6789:ABCD:EF01:2345:6789
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, device's communication port No.-->1
  </portNo>
  <protocol>
    <!--ro, opt, enum, transmission communication protocol type, subType:string, desc:"HTTP, HTTPS, EHome". The value should be "HTTP" when ISAPI protocol is transmitted via EZ protocol. The value should be "EHome" when ISAPI protocol is transmitted via ISUP.-->HTTP
  </protocol>
  <macAddress>
    <!--ro, opt, string, MAC address-->test
  </macAddress>
  <dynChannelID>
    <!--ro, opt, string, digital channel No.-->test
  </dynChannelID>
  <channelID>
    <!--ro, opt, string, channel No.-->1
  </channelID>
  <relatedChannelList>
    <!--ro, opt, string, List of alarm related channels, which are of the same camera with channelID, desc:this parameter is used for Live view or playback on the platform; multiple channel No.s are separated by commas-->1,2,3
  </relatedChannelList>
  <dateTime>
    <!--ro, req, datetime, event occurred time-->1970-01-01T00:00:00+08:00
  </dateTime>
  <activePostCount>
    <!--ro, req, int, event triggering frequency-->1
  </activePostCount>
  <eventType>
    <!--ro, req, enum, event type: shelteralarm, subType:string, desc:event type: shelteralarm-->shelteralarm
  </eventType>
  <eventState>
    <!--ro, req, enum, event status (for persistent event), subType:string, desc:for durative event: active (valid), inactive (invalid)-->active
  </eventState>
  <eventDescription>
    <!--ro, req, string, event description-->test
  </eventDescription>
  <channelName>
    <!--ro, opt, string, channel name-->test
  </channelName>
  <deviceID>
    <!--ro, opt, string, device ID-->test
  </deviceID>
  <Extensions>
    <!--ro, opt, object, extended information-->
    <serialNumber>
      <!--ro, req, string, serial No.-->test
    </serialNumber>
    <eventPush>
      <!--ro, req, string, event notification contents-->test
    </eventPush>
  </Extensions>
  <Image>
    <!--ro, opt, object, event related pictures-->
    <resourcesContentType>
      <!--ro, opt, enum, resource transmission type, subType:string, desc:"url" (transmitted via URL), "binary" (transmitted in binary format). When the returned alarm message contains additional resource (e.g., pictures), this field should be returned.-->binary
    </resourcesContentType>
    <resourcesContent>
      <!--ro, opt, string, resource ID, range:[1,128], desc:when the returned alarm message contains additional resources (e.g., pictures), this field should be returned. When the value of resourcesContentType is "binary", it must be the same as Content-ID of the resource. When the value of resourcesContentType is "url", its value is the resource URL-->image
    </resourcesContent>
  </Image>
</EventNotificationAlert>

```

Parameter Name	Parameter Value	Parameter Type(Content-Type)	Content-ID	File Name	Description
EventNotificationAlert	[Message content]	application/xml	--	--	--
image	[Binary picture data]	image/jpeg	image	image.jpg	--

**Note:** The protocol is transmitted in form format. See Chapter 4.5.1.4 for form framework description, as shown in the instance below.

```
--<frontier>
Content-Disposition: form-data; name=Parameter Name;filename=File Name
Content-Type: Parameter Type
Content-Length: ****
Content-ID: Content ID
Parameter Value
```

- Parameter Name: the name property of Content-Disposition in the header of form unit; it refers to the form unit name.
- Parameter Type (Content-Type): the Content-Type property in the header of form unit.
- File Name (filename): the filename property of Content-Disposition of form unit Headers. It exists only when the transmitted data of form unit is file, and it refers to the file name of form unit body.
- Parameter Value: the body content of form unit.

## 16.2.16 White Balance

### 16.2.16.1 Set the white balance parameters of s specific channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/whiteBalance

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<WhiteBalance xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, white balance, attr:version{req, string, protocolVersion}-->
  <WhiteBalanceStyle>
    <!--opt, enum, white balance type, subType:string, desc:"auto", "auto1", "auto2" "manual", "indoor", "outdoor", "autotrace", "once", "sodiumlight",
"mercuryLight"-->auto
  <WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--req, int, this node depends on <WhiteBalanceStyle>, desc:this node depends on <WhiteBalanceStyle>-->1
  <WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--req, int, this node depends on <WhiteBalanceStyle>, desc:this node depends on <WhiteBalanceStyle>-->1
  <WhiteBalanceBlue>
</WhiteBalance>
```

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0-OK, 1-OK, 2-Device Busy, 3-Device Error, 4-Invalid Operation, 5-Invalid XML Format, 6-Invalid XML
Content, 7-Reboot Required-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format",
"Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.2.16.2 Get the white balance (in day mode) parameters of a specified channel

#### Request URL

GET /ISAPI/Image/channels/<channelID>/whiteBalance/day

## Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<WhiteBalance xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <WhiteBalanceStyle>
    <!--ro, opt, enum, subType:string-->auto
  </WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--ro, req, int-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--ro, req, int-->1
  </WhiteBalanceBlue>
</WhiteBalance>
```

## 16.2.16.3 Set the white balance (in day mode) parameters of a specified channel

### Request URL

PUT /ISAPI/Image/channels/<channelID>/whiteBalance/day

## Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<WhiteBalance xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <WhiteBalanceStyle>
    <!--opt, enum, subType:string-->auto
  </WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--req, int-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--req, int-->1
  </WhiteBalanceBlue>
</WhiteBalance>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

## 16.2.16.4 Get the white balance (in night mode) parameters of a specified channel

### Request URL

GET /ISAPI/Image/channels/<channelID>/whiteBalance/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<WhiteBalance xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <WhiteBalanceStyle>
    <!--ro, opt, enum, subType:string-->auto
  </WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--ro, req, int-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--ro, req, int-->1
  </WhiteBalanceBlue>
</WhiteBalance>
```

### 16.2.16.5 Set the white balance (in night mode) parameters of a specified channel

#### Request URL

PUT /ISAPI/Image/channels/<channelID>/whiteBalance/night

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<WhiteBalance xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <WhiteBalanceStyle>
    <!--opt, enum, subType:string-->auto
  </WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--req, int-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--req, int-->1
  </WhiteBalanceBlue>
</WhiteBalance>
```

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->0K
  </subStatusCode>
</ResponseStatus>
```

### 16.2.16.6 Get the white balance parameters of s specified channel

## Request URL

GET /ISAPI/Image/channels/<channelID>/whiteBalance?parameterType=<parameterType>

## Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	--

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<WhiteBalance xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, white balance, attr:version{req, string, protocolVersion}-->
  <WhiteBalanceStyle>
    <!--ro, opt, enum, "auto,manual,indoor,outdoor,autotrace,onece,sodiumlight,mercurylight,auto0,auto1,fluorescent,natural,warm,incandescent",
    subType:string, desc:"auto", "auto1", "auto2", "manual", "indoor", "outdoor", "autotrace", "onece", "sodiumlight", "mercuryLight"-->auto
    <WhiteBalanceStyle>
    <WhiteBalanceRed>
      <!--ro, req, int, white balance, desc:white balance-->1
    </WhiteBalanceRed>
    <WhiteBalanceBlue>
      <!--ro, req, int, white balance, desc:white balance-->1
    </WhiteBalanceBlue>
  </WhiteBalance>
```

## 16.3 Video Management

### 16.3.1 Manual Recording

#### 16.3.1.1 Start the manual recording

## Request URL

PUT /ISAPI/ContentMgmt/record/control/manual/start/tracks/<trackStreamID>

## Query Parameter

Parameter Name	Parameter Type	Description
trackStreamID	string	--

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

#### 16.3.1.2 Stop the manual recording

## Request URL

PUT /ISAPI/ContentMgmt/record/control/manual/stop/tracks/<trackStreamID>

### Query Parameter

Parameter Name	Parameter Type	Description
trackStreamID	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

## 16.3.2 Scheduled Recording

### 16.3.2.1 Get all parameters of a recording schedule

#### Request URL

GET /ISAPI/ContentMgmt/record/tracks

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<TrackList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, get all parameters of a recording schedule, subType:object, attr:version{req, string, protocolVersion}-->
  <Track>
    <!--ro, opt, object, recording schedule configuration-->
    <id>
      <!--ro, req, int, channel number corresponding to stream type, desc:the values of <id> and <Channel> are the same. For example, 201 refers to the main stream of channel 2, 202 refers to the sub-stream of channel 2, and so on. If the field SrcUrl of video source description is valid, the channel stream type should be subject to that of SrcUrl-->201
    </id>
    <Channel>
      <!--ro, req, int, channel No.-->1
    </Channel>
    <Enable>
      <!--ro, req, bool, whether to enable, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
    </Enable>
    <Description>
      <!--ro, req, string, recording description-->trackType=standard,sourceTag=AXIS210a,contentType=video,codecType=MPEG4-SP,resolution=640x480,frameRate=20 fps,bitrate=6000 kbps
    </Description>
    <TrackGUID>
      <!--ro, req, string, GUID, desc:generated by the client-->{A01AAAAA-BBBB-CCCC-DDDD-033595353625}
    </TrackGUID>
    <Duration min="0" max="750">
      <!--ro, opt, string, video expiry time, attr:min{req, int},max{req, int}, desc:e.g., P10DT15H indicates the video will be expired in 10 days and 15 hours; 0 indicates the video is valid until it is recurrently overwritten-->P10DT15H
    </Duration>
    <DefaultRecordingMode>
      <!--ro, req, enum, default record type, subType:string, desc:"CMR" (capture by schedule), "MOTION" (capture motion detection pictures), "ALARM" (capture alarm pictures), "EDR" (capture alarm pictures or motion detection pictures), "ALARMANDMOTION" (capture alarm pictures and motion detection pictures), "ALLEvent" (all events), "POS" (POS recording), "PECCANCY" (forensic marking recording), "SMART" (smart recording), "temperatureIntervalMeasurement" (temperature range detection), "methaneConcentrationException" (dangerous gas detection: methane concentration exception by default), "thermalVehicleDetection" (thermal imaging vehicle detection), "fishingShipDetection" (fishing ship detection), "consultation", "panicAlarm" (panic alarm), "zoneAlarm" (zone alarm), "parallelParking" (vehicle entering or exiting during parallel parking, "parallelParkingContinuousRecord" (from
    </DefaultRecordingMode>
  </Track>
</TrackList>
```

vehicle entering to vehicle exiting for parallel parking), "linedetection" (line crossing detection), "toiletTarry" (in-toilet overtime detection), "getup" (get-off-bed detection), "retention" (overstay detection), "failDown" (getting up or falling down detection)-->CMR  
 </DefaultRecordingMode>  
 <LoopEnable>  
     <!--ro, opt, bool, whether to support recurrently overwriting-->true  
 </LoopEnable>  
 <SrcDescriptor>  
     <!--ro, opt, object, video source description-->  
 <SrcGUID>  
     <!--ro, req, string, GUID generated by the client-->{E800A543-9D53-4520-8BB8-9509062C692D}  
 </SrcGUID>  
 <SrcChannel>  
     <!--ro, req, int, source channel-->1  
 </SrcChannel>  
 <StreamHint>  
     <!--ro, req, string, stream description-->video,mp4,640x480,20 fps,6000 kbps  
 </StreamHint>  
 <SrcDriver>  
     <!--ro, req, string, stream executable driver name-->RTP/RTSP  
 </SrcDriver>  
 <SrcType>  
     <!--ro, opt, string, source type-->mp4 video  
 </SrcType>  
 <SrcUrl>  
     <!--ro, req, string, source URL, desc:the format of source URL is: rtsp://localhost/ISAPI/Streaming/channels/<trackStreamID>. The <trackStreamID> refers to the channel stream ID = channel number \* 100 + stream type (1 indicates main stream, 2 indicates sub stream). For example, 201 indicates the main stream of channel 2-->rtsp://localhost/ISAPI/Streaming/channels/101  
 </SrcUrl>  
 <SrcLogin>  
     <!--ro, req, string, login source password-->admin:admin  
 </SrcLogin>  
</SrcDescriptor>  
<TrackSchedule>  
 <!--ro, opt, object, recording schedule-->  
 <ScheduleBlockList>  
     <!--ro, opt, array, recording schedule list, subType:object-->  
 <ScheduleBlock>  
     <!--ro, opt, object, recording schedule module-->  
 <ScheduleBlockGUID>  
     <!--ro, opt, string, schedule GUID-->{ABC12345-CDEF-4520-8BB8-71357890C8790}  
 </ScheduleBlockGUID>  
 <ScheduleBlockType>  
     <!--ro, req, string, schedule type-->/ISAPI/recording/schedule/default  
 </ScheduleBlockType>  
 <ScheduleAction>  
     <!--ro, opt, array, schedule, subType:object, desc:N/A-->  
 <id>  
     <!--ro, req, int, ID-->1  
 </id>  
 <ScheduleActionStartTime>  
     <!--ro, opt, object, scheduled start time-->  
 <DayOfWeek>  
     <!--ro, req, enum, day of a week, subType:string, desc:"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"-->Monday  
 </DayOfWeek>  
 <TimeOfDay>  
     <!--ro, opt, time, time-->08:00:00  
 </TimeOfDay>  
 <ScheduleActionEndTime>  
     <!--ro, opt, object, scheduled end time-->  
 <DayOfWeek>  
     <!--ro, req, enum, day of a week, subType:string, desc:"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"-->Monday  
 </DayOfWeek>  
 <TimeOfDay>  
     <!--ro, opt, time, time-->08:00:00  
 </TimeOfDay>  
 </ScheduleActionEndTime>  
 <ScheduledDSTEnable>  
     <!--ro, req, bool, whether to enable Daylight Saving Time (DST)-->true  
 </ScheduledDSTEnable>  
 <Description>  
     <!--ro, req, string, description-->PreMorning (Midnight to 8am,local time)  
 </Description>  
 <Actions>  
     <!--ro, opt, object, trigger action, desc:alarm or motion detection that triggers a recording-->  
 <Record>  
     <!--ro, opt, bool, record, desc:this node is retained, and returned by some devices, but has no practical meaning-->true  
 </Record>  
 <ActionRecordingMode>  
     <!--ro, req, enum, recording schedule type, subType:string, desc:"CMR" (capture by schedule), "MOTION" (capture motion detection pictures), "ALARM" (capture alarm pictures), "EDR" (capture alarm pictures or motion detection pictures), "ALARMANDMOTION" (capture alarm pictures and motion detection pictures), "COMMAND" (recording by commands), "SMART" (smart recording), "thermalVehicleDetection" (thermal imaging vehicle detection), "fishingShipDetection" (fishing ship detection), "methaneConcentrationException" (dangerous gas detection: methane concentration exception by default), "AllEvent" (all events), "consultation", "panicAlarm" (panic alarm), "zoneAlarm" (zone alarm), "temperatureIntervalMeasurement" (temperature range detection), "linedetection" (line crossing detection), "toiletTarry" (in-toilet overtime detection), "getUp" (get-off-bed detection), "retention" (overstay detection), "faildown" (getting up or falling down detection)-->CMR  
 </ActionRecordingMode>  
     <Actions>  
     </Actions>  
 </ScheduleAction>  
 <ScheduleBlockList>  
</TrackSchedule>  
<CustomExtensionList>  
 <!--ro, opt, array, custom extension list, subType:object-->

```

<CustomExtension>
  <!--ro, opt, object, custom extension-->
<CustomExtensionName>
  <!--ro, opt, string, custom extension name-->www.xxxx.com/RaCM/trackExt/ver10
</CustomExtensionName>
<enableSchedule>
  <!--ro, opt, bool, whether to enable recording schedule configuration-->true
</enableSchedule>
<PreRecordTimeSeconds>
  <!--ro, opt, int, pre-record, unit:s-->1
</PreRecordTimeSeconds>
<PostRecordTimeSeconds>
  <!--ro, opt, int, post-record, unit:s-->1
</PostRecordTimeSeconds>
</CustomExtension>
</CustomExtensionList>
<durationEnabled>
  <!--ro, opt, bool, whether to enable video expiry time, desc:this node is not returned when it is not supported; you can configure video expiry time via the node "Duration" when it is supported-->true
</durationEnabled>
<Track>
</TrackList>

```

### 16.3.2.2 Set all recording schedules

#### Request URL

PUT /ISAPI/ContentMgmt/record/tracks

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<TrackList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, all recording schedules parameters, subType:object, attr:version{req, string, protocolVersion}-->
  <Track>
    <!--opt, object, recording schedule configuration-->
    <id>
      <!--req, int, channel corresponding to the stream type, desc:the values of <id> and <Channel> are the same. For example, 201 refers to the main stream of channel 2, 202 refers to the sub-stream of channel 2, and so on. If the field SrcUrl of video source description is valid, the channel stream type should be subject to that of SrcUrl-->201
    </id>
    <Channel>
      <!--req, int, channel No.-->1
    </Channel>
    <Enable>
      <!--req, bool, whether to enable, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
    </Enable>
    <Description>
      <!--req, string, recording description-->trackType=standard,sourceTag=AXIS210a,contentType=video,codecType=MPEG4-SP,resolution=640x480,frameRate=20
      fps,bitrate=6000 kbps
    </Description>
    <TrackGUID>
      <!--req, string, GUID, desc:generated by the client-->{A01AAAAA-BBBB-CCCC-DDDD-033595353625}
    </TrackGUID>
    <Duration min="0" max="750">
      <!--req, string, video expiry time, attr:min{req, int},max{req, int}, desc:e.g., P10DT15H indicates the video will be expired in 10 days and 15 hours; 0 indicates the video is valid until it is recurrently overwritten-->P10DT15H
    </Duration>
    <DefaultRecordingMode>
      <!--req, enum, default recording type, subType:string, desc:"OMR" (capture by schedule), "MOTION" (capture motion detection pictures), "ALARM" (capture alarm pictures), "EDR" (capture alarm pictures or motion detection pictures), "ALARMANDMOTION" (capture alarm pictures and motion detection pictures), "ALLEvent" (all events), "POS" (POS recording), "PECCANCY" (forensic marking recording), "SMART" (smart recording), "temperatureIntervalMeasurement" (temperature range detection), "methaneConcentrationException" (dangerous gas detection: methane concentration exception by default), "thermalVehicleDetection" (thermal imaging vehicle detection), "fishingShipDetection" (fishing ship detection), "consultation", "panicAlarm" (panic alarm), "zoneAlarm" (zone alarm), "parallelParking" (vehicle entering or exiting during parallel parking), "parallelParkingContinuousRecord" (from vehicle entering to vehicle exiting for parallel parking), "linedetection" (line crossing detection), "toiletTarry" (in-toilet overtime detection), "getUp" (get-off-bed detection), "retention" (overstay detection), "failDown" (getting up or falling down detection)-->CMR
    </DefaultRecordingMode>
    <LoopEnable>
      <!--opt, bool, whether to support recurrently overwriting-->true
    </LoopEnable>
    <SrcDescriptor>
      <!--opt, object, video source description-->
      <SrcGUID>
        <!--req, string, GUID generated by the client-->{E800A543-9D53-4520-8BB8-9509062C692D}
      </SrcGUID>
      <SrcChannel>
        <!--req, int, source channel-->1
      </SrcChannel>
      <StreamHint>
        <!--req, string, stream description-->video,mp4,640x480,20 fps,6000 kbps
      </StreamHint>
      <SrcDriver>
        <!--req, string, stream executable driver name-->RTP/RTSP
      </SrcDriver>
    </SrcDescriptor>
  </Track>
</TrackList>

```

```

<SrcType>
    <!--opt, string, source type-->mp4 video
</SrcType>
<SrcUrl>
    <!--req, string, source URL is: rtsp://localhost/ISAPI/Streaming/channels/<trackStreamID>. The <trackStreamID> refers
to the channel stream ID = channel number * 100 + stream type (1 indicates main stream, 2 indicates sub stream). For example, 201 indicates the main stream
of channel 2-->rtsp://localhost/ISAPI/Streaming/channels/101
</SrcUrl>
<SrcLogin>
    <!--req, string, login source password-->admin:admin
</SrcLogin>
</SrcDescriptor>
<TrackSchedule>
    <!--opt, object, recording schedule-->
<ScheduleBlockList>
    <!--opt, array, recording schedule list, subType:object-->
<ScheduleBlock>
    <!--opt, object, recording schedule module-->
<ScheduleBlockGUID>
    <!--opt, string, schedule GUID-->{ABC12345-CDEF-4520-8BB8-7135789C8790}
</ScheduleBlockGUID>
<ScheduleBlockType>
    <!--req, string, schedule type-->/ISAPI/recording/schedule/default
</ScheduleBlockType>
<ScheduleAction>
    <!--opt, array, schedule, subType:object, desc:N/A-->
<id>
    <!--req, int, ID-->1
</id>
<ScheduleActionStartTime>
    <!--opt, object, scheduled start time-->
<DayOfWeek>
    <!--req, enum, day of a week, subType:string, desc:"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"-->Monday
</DayOfWeek>
<TimeOfDay>
    <!--opt, time, time-->08:00:00
</TimeOfDay>
</ScheduleActionStartTime>
<ScheduleActionEndTime>
    <!--opt, object, scheduled end time-->
<DayOfWeek>
    <!--req, enum, day of a week, subType:string, desc:"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"-->Monday
</DayOfWeek>
<TimeOfDay>
    <!--opt, time, time-->08:00:00
</TimeOfDay>
</ScheduleActionEndTime>
<ScheduledDSTEnable>
    <!--req, bool, whether to enable Daylight Saving Time (DST)-->true
</ScheduledDSTEnable>
<Description>
    <!--req, string, description-->PreMorning (Midnight to 8am, local time)
</Description>
<Actions>
    <!--opt, object, trigger action, desc:alarm or motion detection that triggers a recording-->
<Record>
    <!--opt, bool, record, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
</Record>
<ActionRecordingMode>
    <!--req, enum, recording schedule type, subType:string, desc:"CMR" (capture by schedule), "MOTION" (capture motion detection pictures),
"ALARM" (capture alarm pictures), "EDR" (capture alarm pictures or motion detection pictures), "ALARMANDMOTION" (capture alarm pictures and motion detection
pictures), "COMMAND" (recording by commands), "SMART" (smart recording), "thermalVehicleDetection" (thermal imaging vehicle detection),
"fisheringShipDetection" (fishing ship detection), "methaneConcentrationException" (dangerous gas detection: methane concentration exception by default),
"ALLEvent" (all events), "consultation", "panicAlarm" (panic alarm), "zoneAlarm" (zone alarm), "temperatureIntervalMeasurement" (temperature range
detection), "linedetection" (line crossing detection), "toiletTarry" (in-toilet overtime detection), "getUp" (get-off-bed detection), "retention" (overstay
detection), "fallDown" (getting up or falling down detection)-->CMR
</ActionRecordingMode>
</Actions>
</ScheduleAction>
</ScheduleBlock>
</ScheduleBlockList>
</TrackSchedule>
<CustomExtensionList>
    <!--opt, array, custom extension list, subType:object-->
<CustomExtension>
    <!--opt, object, custom extension-->
<CustomExtensionName>
    <!--opt, string, custom extension name-->www.xxx.com/RaCM/trackExt/ver10
</CustomExtensionName>
<enableSchedule>
    <!--opt, bool, whether to enable recording schedule configuration-->true
</enableSchedule>
<PreRecordTimeSeconds>
    <!--opt, int, pre-record, unit:s-->1
</PreRecordTimeSeconds>
<PostRecordTimeSeconds>
    <!--opt, int, post-record, unit:s-->1
</PostRecordTimeSeconds>
</CustomExtension>
</CustomExtensionList>
<durationEnabled>
    <!--opt, bool, whether to enable video expiry time, desc:this node is not returned when it is not supported; you can configure video expiry time via
the node Duration when it is supported-->true

```

```

</durationEnabled>
</Track>
</TrackList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
<!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
<requestURL>
<!--ro, req, string, request URL, range:[0,1024]-->null
</requestURL>
<statusCode>
<!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
</statusCode>
<statusString>
<!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
</statusString>
<subStatusCode>
<!--ro, req, string, sub status code, desc:sub status code-->OK
</subStatusCode>
<description>
<!--ro, opt, string, custom error information description, range:[0,1024], desc:detailed information of custom error returned by device applications, used for fast debugging-->badXmlFormat
</description>
</ResponseStatus>

```

### 16.3.2.3 Get the configuration capability of the recording schedule

#### Request URL

GET /ISAPI/ContentMgmt/record/tracks/<trackStreamID>/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
trackStreamID	string	Channel number corresponding to the stream type, for example, 101 indicates main stream of channel 1.
security	string	When "security" does not exist, it indicates that the data is not encrypted; when "security" is 1, it indicates that the node SrcLogin in the message are encrypted in AES128 CBC mode; when "security" is 2, it indicates that the node SrcLogin in the message are encrypted in AES256 CBC mode.
iv	string	The initialization vector. The maximum length is 32 and is required when security is 1 or 2.

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<Track xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
<!--ro, req, object, configuration capability of the recording schedule, attr:version{req, string, protocolVersion}-->
<id>
<!--ro, opt, int, channel corresponding to the stream type, desc:the values of <id> and <Channel> are the same. For example, 201 refers to the main stream of channel 1, 202 refers to the sub-stream of channel 2, and so on-->201
</id>
<Channel>
<!--ro, req, int, channel No.-->1
</Channel>
<Enable>
<!--ro, req, bool, whether to enable, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
</Enable>
<Description>
<!--ro, req, string, recording description-->trackType=standard,sourceTag=AXIS210a,contentType=video,codecType=MPEG4-SP,resolution=640x480,frameRate=20
fps,bitrate=6000 kbps
</Description>
<TrackGUID>
<!--ro, req, string, GUID, desc:generated by the client-->{A01AAAAA-BBBB-CCCC-DDDD-033595353625}
</TrackGUID>

```

```

<!-->
<Size>
    <!--ro, opt, int, recording size-->1
</Size>
<Duration min="0" max="750">
    <!--ro, opt, string, video expiry time, attr:min{req, int},max{req, int}, desc:e.g., P10DT15H indicates the video will be expired in 10 days and 15 hours-->P10DT15H
</Duration>
<DefaultRecordingMode>
opt="CMR,MOTION,ALARM,EDR,ALARMANDMOTION,AllEvent,POS,PECCANCY,SMART,temperatureIntervalMeasurement,methaneConcentrationException,thermalVehicleDetection,fishingShipDetection,consultation,panicAlarm,zoneAlarm,parallelParking,parallelParkingContinuousRecord,linedetection,toiletTarry,getUp,retention,failDown"
    <!--ro, req, enum, default recording type, subType:string, attr:opt{req, string}, desc:"CMR" (capture by schedule), "MOTION" (capture motion detection pictures), "ALARM" (capture alarm pictures), "EDR" (capture alarm pictures or motion detection pictures), "ALARMANDMOTION" (capture alarm pictures and motion detection pictures), "AllEvent" (all events), "POS" (POS recording), "PECCANCY" (forensic marking recording), "SMART" (smart recording), "temperatureIntervalMeasurement" (temperature range detection), "methaneConcentrationException" (dangerous gas detection: methane concentration exception by default), "thermalVehicleDetection" (thermal imaging vehicle detection), "fishingShipDetection" (fishing ship detection), "consultation", "panicAlarm" (panic alarm), "zoneAlarm" (zone alarm), "parallelParking" (vehicle entering or exiting during parallel parking, "parallelParkingContinuousRecord" (from vehicle entering to vehicle exiting for parallel parking), "linedetection" (line crossing detection), "toiletTarry" (in-toilet overtime detection), "getUp" (get-off-bed detection), "retention" (overstay detection), "failDown" (getting up or falling down detection)-->CMR
</DefaultRecordingMode>
<LoopEnable>
    <!--ro, opt, bool, whether to support recurrent overwriting-->true
</LoopEnable>
<SrcDescriptor>
    <!--ro, opt, object, video source description-->
<SrcGUID>
    <!--ro, req, string, GUID generated by the client-->{E800A543-9D53-4520-8BB8-9509062C692D}
</SrcGUID>
<SrcChannel>
    <!--ro, req, int, source channel-->1
</SrcChannel>
<StreamHint>
    <!--ro, req, string, stream description-->video,mp4,640x480,20 fps,6000 kbps
</StreamHint>
<SrcDriver>
    <!--ro, req, string, stream executable driver name-->RTP/RTSP
</SrcDriver>
<SrcType>
    <!--ro, opt, string, source type-->mp4 video
</SrcType>
<SrcUrl>
    <!--ro, req, string, source URL, attr:opt{req, string}-->rtsp://10.3.2.26/mpeg4/media.amp
</SrcUrl>
<SrcUrlMethods>
    <!--ro, req, string, methods supported by the source, desc:"DESCRIBE" (subscribe), "SETUP", "PLAY", "TEARDOWN" (stop)-->DESCRIBE
</SrcUrlMethods>
<SrcLogin>
    <!--ro, req, string, login source password, desc:user name for source channel RTSP authentication returned by device; encryption required for sensitive information-->admin
</SrcLogin>
</SrcDescriptor>
<TrackSchedule>
    <!--ro, opt, object, recording schedule-->
<ScheduleBlockList>
    <!--ro, opt, array, recording schedule list, subType:object-->
<ScheduleBlock>
    <!--ro, opt, object, recording schedule module-->
<ScheduleBlockGUID>
    <!--ro, opt, string, schedule GUID-->{ABC12345-CDEF-4520-8BB8-7135789C8790}
</ScheduleBlockGUID>
<ScheduleBlockType>
    <!--ro, req, string, schedule type-->/ISAPI/recording/schedule/default
</ScheduleBlockType>
<ScheduleAction>
    <!--ro, opt, array, schedule, subType:object, desc:N/A-->
<id>
    <!--ro, req, int, ID-->1
</id>
<ScheduleActionStartTime>
    <!--ro, opt, object, scheduled start time-->
<DayOfWeek>
    <!--ro, req, enum, day of a week, subType:string, desc:"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"-->Monday
</DayOfWeek>
<TimeOfDay>
    <!--ro, opt, time, time-->08:00:00
</TimeOfDay>
</ScheduleActionStartTime>
<ScheduleActionEndTime>
    <!--ro, opt, object, scheduled end time-->
<DayOfWeek>
    <!--ro, req, enum, day of a week, subType:string, desc:"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"-->Monday
</DayOfWeek>
<TimeOfDay>
    <!--ro, opt, time, time-->08:00:00
</TimeOfDay>
</ScheduleActionEndTime>
<ScheduleDSTEnable>
    <!--ro, req, bool, whether to enable Daylight Saving Time (DST)-->true
</ScheduleDSTEnable>
<Description>
    <!--ro, req, string, description-->PreMorning (Midnight to 8am, local time)
</Description>

```

```

<Actions>
  <!--ro, opt, object, trigger action, desc:alarm or motion detection that triggers a recording-->
<Record>
  <!--ro, opt, bool, record, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
</Record>
<Log>
  <!--ro, opt, bool, log, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
</Log>
<SaveImg>
  <!--ro, opt, bool, save image, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
</SaveImg>
<ActionRecordingMode>
  <!--ro, req, enum, recording schedule type, subType:string, desc:"CMR" (capture by schedule), "MOTION" (capture motion detection pictures), "ALARM" (capture alarm pictures), "EDR" (capture alarm pictures or motion detection pictures), "ALARMANDMOTION" (capture alarm pictures and motion detection pictures), "Command" (recording by commands), "SMART" (smart recording), "ALLEvent" (all events), "temperatureIntervalMeasurement" (temperature range detection), "thermalVehicleDetection" (thermal imaging vehicle detection), "fishingShipDetection" (fishing ship detection), "methaneConcentrationException" (dangerous gas detection: methane concentration exception by default), "consultation", "panicAlarm" (panic alarm), "zoneAlarm" (zone alarm), "linedetection" (line crossing detection), "toiletTarry" (in-toilet overtime detection), "getUp" (get-off-bed detection), "retention" (overstay detection), "failDown" (getting up or falling down detection)-->CMR
    </ActionRecordingMode>
    <PreRecordTimeSeconds min="0" max="10" def="5">
      <!--ro, opt, int, pre-record duration, unit:s, attr:min{req, int},max{req, int},def{req, int}, desc:from 0 to 10 seconds; default: 5 seconds-->5
    </PreRecordTimeSeconds>
    <PostRecordTimeSeconds min="30" max="600" def="30">
      <!--ro, opt, int, post-record duration, unit:s, attr:min{req, int},max{req, int},def{req, int}, desc:from 30 to 600 seconds; default: 30 seconds-->30
    </PostRecordTimeSeconds>
  </Actions>
  </ScheduleAction>
</ScheduleBlock>
</ScheduleBlockList>
</TrackSchedule>
<CustomExtensionList>
  <!--ro, opt, array, custom extension list, subType:object-->
<CustomExtension>
  <!--ro, opt, object, custom extension-->
  <CustomExtensionName>
    <!--ro, opt, string, custom extension name-->www.xxx.com/RaCM/trackExt/ver10
  </CustomExtensionName>
  <enableSchedule>
    <!--ro, opt, bool, whether to enable recording schedule configuration-->true
  </enableSchedule>
  <SaveAudio>
    <!--ro, opt, bool, whether to enable audio recording-->true
  </SaveAudio>
  <PreRecordTimeSeconds>
    <!--ro, opt, int, pre-record, unit:s-->1
  </PreRecordTimeSeconds>
  <PostRecordTimeSeconds>
    <!--ro, opt, int, post-record, unit:s-->1
  </PostRecordTimeSeconds>
</CustomExtensionList>
<HolidaySchedule>
  <!--ro, opt, object, holiday schedule-->
  <ScheduleBlock>
    <!--ro, opt, object, schedule module-->
    <ScheduleBlockGUID>
      <!--ro, req, string, schedule module GUID-->{00000000-0000-0000-0000-000000000000}
    </ScheduleBlockGUID>
    <ScheduleBlockType>
      <!--ro, opt, string, schedule module type-->www.xxx.com/racm/schedule/ver10
    </ScheduleBlockType>
  </ScheduleBlock>
</HolidaySchedule>
</CustomExtension>
</CustomExtensionList>
<IntelligentRecord>
  <!--ro, opt, enum, whether to enable VCA recording, subType:string, desc:0-no, 1-yes.-->0
</IntelligentRecord>
<delayTime opt="0,3,4,5,10,30,60,120,300">
  <!--ro, opt, int, capture delay time, unit:s, attr:opt{req, string}-->1
</delayTime>
<durationEnabled opt="true,false">
  <!--ro, opt, bool, whether to enable video expiry date, attr:opt{req, string}, desc:1. front-end device: this node is not returned when this function is not supported; the video expiry date will be set by the node <Duration> when this function is supported and <Duration> is not 0
  2. back-end device: the video expiry date will be set by the node <Duration> when this node is not returned; this node is "false" when <Duration> is 0-->true
</durationEnabled>
<redundancyRec opt="true,false">
  <!--ro, opt, bool, whether it is redundant recording, attr:opt{req, string}-->false
</redundancyRec>
<passbackRecord opt="true,false">
  <!--ro, opt, bool, whether to copy back recording, attr:opt{req, string}-->false
</passbackRecord>
<lockDuration min="0" max="65535">
  <!--ro, opt, int, recording Locking duration, unit:h, attr:min{req, int},max{req, int}, desc:0-unlocked or this node does not exist, 0xffff-Locked permanently; if the total recording is longer than the locked part, it will not be locked-->0
</lockDuration>
<recordBackup opt="true,false">
  <!--ro, opt, bool, whether to enable video archiving, attr:opt{req, string}-->false
</recordBackup>
<SVCLevel opt="0,1,2,3">
  <!--ro, opt, enum, SVC frame extracting type, subType:int, attr:opt{req, string}, desc:0-not extracted, 1 half extracted, 2 three quarters extracted, 3 full extracted-->0

```

```

<!--ro, opt, enum, SVC frame extracting type, subtype:uri, attr:opt{req, string}, desc:0-not extracted, 1-half extracted, 2-three quarters extracted-->
</SVCLevel>
<recordManage opt="true,false">
    <!--ro, opt, bool, whether to enable video scheduling, attr:opt{req, string}, desc:enabled-scheduled recording; disabled-no scheduled recording but the recording schedule still in use for, e.g., motion detection and copy-back-->false
</recordManage>
<extraSaveAudio opt="true,false">
    <!--ro, opt, bool, whether to enable separate audio storage, attr:opt{req, string}-->false
</extraSaveAudio>
</Track>

```

#### 16.3.2.4 Get the recording schedule parameters of a specified channel

##### Request URL

GET /ISAPI/ContentMgmt/record/tracks/<trackStreamID>

##### Query Parameter

Parameter Name	Parameter Type	Description
trackStreamID	string	Channel number corresponding to the stream type, for example, 101 indicates main stream of channel 1.
security	string	When "security" does not exist, it indicates that the data is not encrypted; when "security" is 1, it indicates that the node SrcLogin in the message are encrypted in AES128 CBC mode; when "security" is 2, it indicates that the node SrcLogin in the message are encrypted in AES256 CBC mode.
iv	string	The initialization vector. The maximum length is 32 and is required when security is 1 or 2.

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Track xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, parameters of a recording schedule, attr:version{req, string, protocolVersion}-->
    <id>
        <!--ro, opt, int, channel corresponding to the stream type, desc:the values of <id> and <Channel> are the same. For example, 201 refers to the main stream of channel 2, 202 refers to the sub-stream of channel 2, and so on. If the field SrcUrl of video source description is valid, the channel stream type should be subject to that of SrcUrl-->201
    </id>
    <Channel>
        <!--ro, req, int, channel No.-->1
    </Channel>
    <Enable>
        <!--ro, req, bool, whether to enable, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
    </Enable>
    <Description>
        <!--ro, req, string, recording description-->trackType=standard,sourceTag=AXIS210a,contentType=video,codecType=MPEG4-SP,resolution=640x480,frameRate=20
        fps,bitrate=6000 kbps
    </Description>
    <TrackGUID>
        <!--ro, req, string, GUID, desc:generated by the client-->{A01AAAAA-BBBB-CCCC-DDDD-033595353625}
    </TrackGUID>
    <Duration min="0" max="750">
        <!--ro, opt, string, video expiry time, attr:min{req, int},max{req, int}, desc:e.g., P10DT15H indicates the video will be expired in 10 days and 15 hours-->P10DT15H
    </Duration>
    <DefaultRecordingMode>
        <!--ro, req, enum, default recording type, subtype:string, desc:"CMR" (capture by schedule), "MOTION" (capture motion detection pictures), "ALARM" (capture alarm pictures), "EDR" (capture alarm pictures or motion detection pictures), "ALARMANDMOTION" (capture alarm pictures and motion detection pictures), "ALLEvent" (all events), "POS" (POS recording), "PECCANCY" (forensic marking recording), "SMART" (smart recording), "temperatureIntervalMeasurement" (temperature range detection), "methaneConcentrationException" (dangerous gas detection: methane concentration exception by default), "thermalVehicleDetection" (thermal imaging vehicle detection), "fishingShipDetection" (fishing ship detection), "consultation", "panicAlarm" (panic alarm), "zoneAlarm" (zone alarm), "parallelParking" (vehicle entering or exiting during parallel parking, "parallelParkingContinuousRecord" (from vehicle entering to vehicle exiting for parallel parking), "Linedetection" (line crossing detection), "toiletTarry" (in-toilet overtime detection), "getUp" (get-off-bed detection), "retention" (overstay detection), "failDown" (getting up or falling down detection)-->CMR
    </DefaultRecordingMode>
    <LoopEnable>
        <!--ro, opt, bool, whether to support recurrent overwriting-->true
    </LoopEnable>
    <SrcDescriptor>
        <!--ro, opt, object, video source description-->
        <SrcGUID>
            <!--ro, req, string, GUID generated by the client-->{E800A543-9D53-4520-8BB8-9509062C692D}
        </SrcGUID>
    </SrcDescriptor>

```

```

<!-->
<SrcChannel>
    <!--ro, req, int, source channel-->1
</SrcChannel>
<StreamHint>
    <!--ro, req, string, stream description-->video,mp4,640x480,20 fps,6000 kbps
</StreamHint>
<SrcDriver>
    <!--ro, req, string, stream executable driver name-->RTP/RTSP
</SrcDriver>
<SrcType>
    <!--ro, opt, string, source type-->mp4 video
</SrcType>
<SrcUrl>
    <!--ro, req, string, source URL is: rtsp://localhost/ISAPI/Streaming/channels/<trackStreamID>. The <trackStreamID> refers to the channel stream ID = channel number * 100 + stream type (1 indicates main stream, 2 indicates sub stream). For example, 201 indicates the main stream of channel 2-->rtsp://localhost/ISAPI/Streaming/channels/101
</SrcUrl>
<SrcLogin>
    <!--ro, req, string, login source password, desc:user name for source channel RTSP authentication returned by device; encryption required for sensitive information-->admin
</SrcLogin>
<SrcDescriptor>
<TrackSchedule>
    <!--ro, opt, object, recording schedule-->
    <ScheduleBlockList>
        <!--ro, opt, array, recording schedule list, subType:object-->
        <ScheduleBlock>
            <!--ro, opt, object, recording schedule module-->
            <ScheduleBlockGUID>
                <!--ro, opt, string, schedule GUID-->{ABC12345-CDEF-4520-8BBB-7135789C8790}
            </ScheduleBlockGUID>
            <ScheduleBlockType>
                <!--ro, req, string, schedule type-->/ISAPI/recording/schedule/default
            </ScheduleBlockType>
            <ScheduleAction>
                <!--ro, opt, array, schedule, subType:object, desc:N/A-->
                <id>
                    <!--ro, req, int, ID-->1
                </id>
                <ScheduleActionStartTime>
                    <!--ro, opt, object, scheduled start time-->
                    <DayOfWeek>
                        <!--ro, req, enum, day of a week, subType:string, desc:"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"-->Monday
                    </DayOfWeek>
                    <TimeOfDay>
                        <!--ro, opt, time, time-->08:00:00
                    </TimeOfDay>
                </ScheduleActionStartTime>
                <ScheduleActionEndTime>
                    <!--ro, opt, object, scheduled end time-->
                    <DayOfWeek>
                        <!--ro, req, enum, day of a week, subType:string, desc:"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"-->Monday
                    </DayOfWeek>
                    <TimeOfDay>
                        <!--ro, opt, time, time-->08:00:00
                    </TimeOfDay>
                </ScheduleActionEndTime>
                <ScheduleDSTEnable>
                    <!--ro, req, bool, whether to enable Daylight Saving Time (DST)-->true
                </ScheduleDSTEnable>
                <Description>
                    <!--ro, req, string, description-->PreMorning (Midnight to 8am, local time)
                </Description>
                <Actions>
                    <!--ro, opt, object, trigger action, desc:alarm or motion detection that triggers a recording-->
                    <Record>
                        <!--ro, opt, bool, record, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
                    </Record>
                    <ActionRecordingMode>
                        <!--ro, req, enum, recording schedule type, subType:string, desc:"CMR" (capture by schedule), "MOTION" (capture motion detection pictures), "ALARM" (capture alarm pictures), "EDR" (capture alarm pictures or motion detection pictures), "ALARMANDMOTION" (capture alarm pictures and motion detection pictures), "COMMAND" (recording by commands), "SMART" (smart recording), "thermalVehicleDetection" (thermal imaging vehicle detection), "fishingShipDetection" (fishing ship detection), "methaneConcentrationException" (dangerous gas detection: methane concentration exception by default), "ALLEvent" (all events), "consultation", "panicAlarm" (panic alarm), "zoneAlarm" (zone alarm), "temperatureIntervalMeasurement" (temperature range detection), "linedetection" (line crossing detection), "toiletTarry" (in-toilet overtime detection), "getUp" (get-off-bed detection), "retention" (overstay detection), "fallDown" (getting up or falling down detection)-->CMR
                    </ActionRecordingMode>
                <Actions>
                </Actions>
            </ScheduleAction>
        </ScheduleBlock>
    </ScheduleBlockList>
</TrackSchedule>
<CustomExtensionList>
    <!--ro, opt, array, custom extension list, subType:object-->
    <CustomExtension>
        <!--ro, opt, object, custom extension-->
        <CustomExtensionName>
            <!--ro, opt, string, custom extension name-->www.xxx.com/RaCM/trackExt/ver10
        </CustomExtensionName>
        <enableSchedule>
            <!--ro, opt, bool, whether to enable recording schedule configuration-->true
        </enableSchedule>
        <PreRecordTimeSeconds>

```

```

<!--ro, opt, int, pre-record, unit:s-->1
</PreRecordTimeSeconds>
<PostRecordTimeSeconds>
    <!--ro, opt, int, post-record, unit:s-->1
    </PostRecordTimeSeconds>
</CustomExtension>
</CustomExtensionList>
<durationEnabled>
    <!--ro, opt, bool, whether to enable video expiry time, desc:this node is not returned when it is not supported; you can configure video expiry time via the node Duration when it is supported-->true
</durationEnabled>
</Track>

```

### 16.3.2.5 Set the recording schedule for a specified channel

#### Request URL

PUT /ISAPI/ContentMgmt/record/tracks/<trackStreamID>

#### Query Parameter

Parameter Name	Parameter Type	Description
trackStreamID	string	Channel number corresponding to the stream type, for example, 101 indicates main stream of channel 1.
security	string	When "security" does not exist, it indicates that the data is not encrypted; when "security" is 1, it indicates that the node SrcLogin in the message are encrypted in AES128 CBC mode; when "security" is 2, it indicates that the node SrcLogin in the message are encrypted in AES256 CBC mode.
iv	string	The initialization vector. The maximum length is 32 and is required when security is 1 or 2.

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Track xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--req, object, get the parameters of a recording schedule, attr:version{req, string, protocolVersion}-->
    <id>
        <!--opt, int, channel corresponding to the stream type, desc:the values of <id> and <Channel> are the same. For example, 201 refers to the main stream of channel 2, 202 refers to the sub-stream of channel 2, and so on. If the field SrcUrl of video source description is valid, the channel stream type should be subject to that of SrcUrl-->201
    </id>
    <Channel>
        <!--req, int, channel No.-->1
    </Channel>
    <Enable>
        <!--req, bool, whether to enable, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
    </Enable>
    <Description>
        <!--req, string, recording description-->trackType=standard,sourceTag=AXIS210a,contentType=video,codecType=MPEG4-SP,resolution=640x480,frameRate=20
        fps,bitrate=6000 kbps
    </Description>
    <TrackGUID>
        <!--req, string, GUID, desc:generated by the client-->{A01AAAAA-BBBB-CCCC-033595353625}
    </TrackGUID>
    <Duration min="0" max="750">
        <!--opt, string, video expiry time, attr:min{req, int},max{req, int}, desc:e.g., P10DT15H indicates the video will be expired in 10 days and 15 hours-->P10DT15H
    </Duration>
    <DefaultRecordingMode>
        <!--req, enum, default record type, subType:string, desc:"CMR" (capture by schedule), "MOTION" (capture motion detection pictures), "ALARM" (capture alarm pictures), "EDR" (capture alarm pictures or motion detection pictures), "ALARMANDMOTION" (capture alarm pictures and motion detection pictures), "ALLEvent" (all events), "POS" (POS recording), "PECCANCY" (forensic marking recording), "SMART" (smart recording), "temperatureIntervalMeasurement" (temperature range detection), "methaneConcentrationException" (dangerous gas detection: methane concentration exception by default), "thermalVehicleDetection" (thermal imaging vehicle detection), "fishingShipDetection" (fishing ship detection), "consultation", "panicAlarm" (panic alarm), "zoneAlarm" (zone alarm), "parallelParking" (vehicle entering or exiting during parallel parking, "parallelParkingContinuousRecord" (from vehicle entering to vehicle exiting for parallel parking), "linedetection" (line crossing detection), "toiletTarry" (in-toilet overtime detection), "getUp" (get-off-bed detection), "retention" (overstay detection), "failDown" (getting up or falling down detection)-->CMR
    </DefaultRecordingMode>
    <LoopEnable>
        <!--opt, bool, whether to support recurrently overwriting-->true
    </LoopEnable>
    <SrcDescriptor>
        <!--opt, object, video source description-->
        <SrcGUID>
            <!--req, string, GUID generated by the client-->{E800A543-9D53-4520-8BB8-9509062C692D}
        </SrcGUID>
        <SrcChannel>
            <!--req, int, source channel-->1
        </SrcChannel>
    </SrcDescriptor>

```

```

    ... req, opt, source channel ...
</SrcChannel>
<StreamHint>
    <!--req, string, stream description-->video,mp4,640x480,20 fps,6000 kbps
</StreamHint>
<SrcDriver>
    <!--req, string, stream executable driver name-->RTP/RTSP
</SrcDriver>
<SrcType>
    <!--opt, string, source type-->mp4 video
</SrcType>
<SrcUrl>
    <!--req, string, source URL, desc:the format of source URL is: rtsp://localhost/ISAPI/Streaming/channels/<trackStreamID>. The <trackStreamID> refers to the channel stream ID = channel number * 100 + stream type (1 indicates main stream, 2 indicates sub stream). For example, 201 indicates the main stream of channel 2-->rtsp://localhost/ISAPI/Streaming/channels/101
</SrcUrl>
<SrcLogin>
    <!--req, string, Login source password, desc:user name for source channel RTSP authentication; encryption required for sensitive information-->
admin:admin
</SrcLogin>
</SrcDescriptor>
<TrackSchedule>
    <!--opt, object, recording schedule-->
<ScheduleBlockList>
    <!--opt, array, recording schedule List, subType:object-->
<ScheduleBlock>
    <!--opt, object, recording schedule module-->
<ScheduleBlockGUID>
    <!--opt, string, schedule GUID-->{ABC12345-CDEF-4520-8BBB-7135789C8790}
</ScheduleBlockGUID>
<ScheduleBlockType>
    <!--req, string, schedule type-->/ISAPI/recording/schedule/default
</ScheduleBlockType>
<ScheduleAction>
    <!--opt, array, schedule, subType:object, desc:N/A-->
<id>
    <!--req, int, ID-->1
</id>
<ScheduleActionStartTime>
    <!--opt, object, scheduled start time-->
<DayOfWeek>
    <!--req, enum, day of a week, subType:string, desc:"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"-->Monday
</DayOfWeek>
<TimeOfDay>
    <!--opt, time, time-->08:00:00
</TimeOfDay>
</ScheduleActionStartTime>
<ScheduleActionEndTime>
    <!--opt, object, scheduled end time-->
<DayOfWeek>
    <!--req, enum, day of a week, subType:string, desc:"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"-->Monday
</DayOfWeek>
<TimeOfDay>
    <!--opt, time, time-->08:00:00
</TimeOfDay>
</ScheduleActionEndTime>
<ScheduleDSTEnable>
    <!--req, bool, whether to enable Daylight Saving Time (DST)-->true
</ScheduleDSTEnable>
<Description>
    <!--req, string, Description-->PreMorning (Midnight to 8am,local time)
</Description>
<Actions>
    <!--opt, object, trigger action, desc:alarm or motion detection that triggers a recording-->
<Record>
    <!--opt, bool, record, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
</Record>
<ActionRecordingMode>
    <!--req, enum, recording schedule type, subType:string, desc:"CMR" (capture by schedule), "MOTION" (capture motion detection pictures), "ALARM" (capture alarm pictures), "EDR" (capture alarm pictures or motion detection pictures), "ALARMANDMOTION" (capture alarm pictures and motion detection pictures), "COMMAND" (recording by commands), "SMART" (smart recording), "thermalVehicleDetection" (thermal imaging vehicle detection), "fishingShipDetection" (fishing ship detection), "methaneConcentrationException" (dangerous gas detection: methane concentration exception by default), "AllEvent" (all events), "consultation", "panicAlarm" (panic alarm), "zoneAlarm" (zone alarm), "temperatureIntervalMeasurement" (temperature range detection), "linedetection" (line crossing detection), "toiletTarry" (in-toilet overtime detection), "getUp" (get-off-bed detection), "retention" (overstay detection), "fallDown" (getting up or falling down detection)-->CMR
</ActionRecordingMode>
<Actions>
    <!--opt, object, ScheduleAction-->
</Actions>
</ScheduleAction>
</ScheduleBlock>
</ScheduleBlockList>
</TrackSchedule>
<CustomExtensionList>
    <!--opt, array, custom extension List, subType:object-->
<CustomExtension>
    <!--opt, object, custom extension-->
<CustomExtensionName>
    <!--opt, string, custom extension name-->www.xxx.com/RaCM/trackExt/ver10
</CustomExtensionName>
<enableSchedule>
    <!--opt, bool, whether to enable recording schedule configuration-->true
</enableSchedule>
<PreRecordTimeSeconds>
    <!--opt, int, pre-record, unit:s-->1
</PreRecordTimeSeconds>

```

```

<PostRecordTimeSeconds>
  <!--opt, int, post-record, unit:s-->1
</PostRecordTimeSeconds>
</CustomExtension>
</CustomExtensionList>
<durationEnabled>
  <!--opt, bool, whether to enable video expiry time, desc:this node is not returned when it is not supported; you can configure video expiry time via the node Duration when it is supported-->true
</durationEnabled>
</Track>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL, range:[0,1024]-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code-->OK
  </subStatusCode>
  <description>
    <!--ro, opt, string, custom error information description, range:[0,1024], desc:detailed information of custom error returned by device applications, used for fast debugging-->badXmlFormat
  </description>
</ResponseStatus>

```

## 16.4 PTZ

### 16.4.1 PTZ Maintenance

#### 16.4.1.1 Get the PTZ capability supported by the device

##### Request URL

GET /ISAPI/PTZCtrl/capabilities

##### Query Parameter

None

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PTZCtrlCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, PTZ control capability, attr:version{req, string, protocolVersion}-->
  <maxPTZChannleNum min="1" max="10">
    <!--ro, opt, int, number of PTZ channels, attr:min{req, int},max{req, int}-->1
  </maxPTZChannleNum>
</PTZCtrlCap>

```

#### 16.4.1.2 Set control parameter of all PTZ.

##### Request URL

PUT /ISAPI/PTZCtrl/channels

##### Query Parameter

None

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PTZChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, PTZ channels' configuration list, subType:object, attr:version{req, string, protocolVersion}-->
</PTZChannelList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

## 16.5 People Counting

### 16.5.1 Real-Time People Counting

#### 16.5.1.1 Get the arming schedule of people counting for all channels

##### Request URL

GET /ISAPI/Event/schedules/countings

##### Query Parameter

None

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<CountingScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, subType:object, attr:version{opt, string, protocolVersion}-->
</CountingScheduleList>

```

#### 16.5.1.2 Set arming schedules of people counting of all channels

##### Request URL

PUT /ISAPI/Event/schedules/countings

##### Query Parameter

None

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<CountingScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, subType:object, attr:version{opt, string, protocolVersion}-->
</CountingScheduleList>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, desc:sub status code, which describes the error in details-->OK
    </statusString>
  </statusCode>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.5.1.3 Set parameters for overlaying people counting POS information on the stream of a video input channel

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/counting/posInfoOverlay

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PosInfoOverlay xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, parameters for overlaying people counting POS information on the stream of a video input channel, attr:version{opt, string, protocolVersion}-->
  <enabled>
    <!--opt, bool, whether to enable-->true
  </enabled>
</PosInfoOverlay>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, desc:sub status code, which describes the error in details-->OK
    </statusString>
  </statusCode>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.5.1.4 Get parameters for overlaying people counting POS information on the stream of a video input channel

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/counting/posInfoOverlay

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<PosInfoOverlay xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, parameters for overlaying people counting POS information on the stream of a video input channel, attr:version{opt, string, protocolVersion}-->
  <enabled>
    <!--ro, opt, bool, whether to enable-->true
  </enabled>
</PosInfoOverlay>
```

## 16.5.1.5 Get the capability of overlaying people counting POS information on the stream of a video input channel

### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/counting/posInfoOverlay/capabilities

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<PosInfoOverlay xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, the capability of overlaying people counting POS information on the stream of a video input channel, attr:version{opt, string, protocolVersion}-->
  <enabled opt="true,false">
    <!--ro, req, bool, whether to enable, attr:opt{opt, string}-->true
  </enabled>
</PosInfoOverlay>
```

## 16.5.1.6 Get people counting status by channel

### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/counting/status

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<CountingStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <status>
    <!--ro, req, enum, "counting,stopped,pause", subType:string, desc:"counting", "stopped", "paused"-->counting
  </status>
  <time>
    <!--ro, req, time, operation time (Local time + time zone)-->22:00:00+08:00
  </time>
</CountingStatus>

```

### 16.5.1.7 People counting event

#### EventType:PeopleCounting

```

<?xml version="1.0" encoding="UTF-8"?>

<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, people counting event details, attr:version{req, string, protocolVersion}-->
  <ipAddress>
    <!--ro, opt, string, IPv4 address-->172.6.64.7
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPv6 Address-->1080:0:0:0:8:800:200C:417A
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, device's communication port No.-->1
  </portNo>
  <protocol>
    <!--ro, opt, enum, transmission communication protocol, subType:string, desc:"HTTP, HTTPS, EHome". The value should be "EHome" when ISAPI protocol is transmitted via ISUP-->HTTP
  </protocol>
  <macAddress>
    <!--ro, opt, string, MAC address-->test
  </macAddress>
  <dynChannelID>
    <!--ro, opt, string, digital channel No.-->test
  </dynChannelID>
  <channelID>
    <!--ro, opt, string, channel No.-->1
  </channelID>
  <relatedChannelList>
    <!--ro, opt, string, List of alarm related channels, which are of the same camera with channelID, desc:this parameter is used for Live view or playback on the platform; multiple channel No.s are separated by commas-->1,2,3
  </relatedChannelList>
  <dateTime>
    <!--ro, req, datetime, event occurred time-->1970-01-01T00:00:00+08:00
  </dateTime>
  <activePostCount>
    <!--ro, req, int, event triggering frequency-->1
  </activePostCount>
  <eventType>
    <!--ro, req, enum, event type, subType:string, desc:"PeopleCounting" (people counting)-->PeopleCounting
  </eventType>
  <eventState>
    <!--ro, req, enum, event status, subType:string, desc:"active" (valid), "inactive" (invalid)-->active
  </eventState>
  <eventDescription>
    <!--ro, req, string, event description-->test
  </eventDescription>
  <channelName>
    <!--ro, opt, string, channel name-->test
  </channelName>
  <deviceID>
    <!--ro, opt, string, PUID, range:[1,64], desc:required in Ehome alarm, corresponding to deviceID in /ISAPI/System/Network/Ehome-->test
  </deviceID>
  <peopleCounting>
    <!--ro, opt, object, people counting-->
    <statisticalMethods>
      <!--ro, opt, enum, people counting method, subType:string, desc:"realTime" (real-time people counting), "timeRange" (scheduled people counting), "signalTrigger" (alarm input triggered people counting)-->realTime
    </statisticalMethods>
    <RealTime>
      <!--ro, opt, object, real-time, desc:it is required when statisticalMethods is "realTime"-->
      <time>
        <!--ro, req, datetime, people counting time-->1970-01-01T00:00:00+08:00
      </time>
    </Realtime>
    <TimeRange>
      <!--ro, opt, object, time period for real-time people counting, desc:it is required when statisticalMethods is "timeRange" or "signalTrigger".-->
      <startTime>
        <!--ro, req, datetime, start time of people counting-->1970-01-01T00:00:00+08:00
      </startTime>
      <endTime>
        <!--ro, req, datetime, end time of people counting-->1970-01-01T00:00:00+08:00
      </endTime>
    </TimeRange>
    <enter>

```

```

<!--ro, opt, int, number of entered people-->1
</enter>
<exit>
<!--ro, opt, int, number of exited people-->1
</exit>
<pass>
<!--ro, opt, int, passers-by number, desc:passers-by number-->1
</pass>
<duplicatePeople>
<!--ro, opt, int, number of duplicated people, desc:number of duplicated people-->1
</duplicatePeople>
<FlowStatisticsList>
<!--ro, opt, array, flow direction counting List, subType:object-->
<FlowStatistics>
<!--ro, opt, object, flow direction counting, desc:for deduplicated people counting statistics, the flow direction analysis is only valid in the
following circumstances: the rule area is a quadrilateral, the counting broken Line is formed with four points, and the counting Line has only two points
that overlap with the rule area Line-->
<countingLineID>
<!--ro, opt, int, counting Line ID-->1
</countingLineID>
<pass>
<!--ro, opt, int, passers-by number, desc:people who enter the counting area but do not cross the counting Line-->1
</pass>
<movingDirectionList>
<!--ro, opt, array, people moving direction list, subType:object, desc:count the moving direction of people who enter the counting area and cross
the counting Line-->
<MovingDirection>
<!--ro, opt, object, people moving direction-->
<countingLineSegmentID>
<!--ro, opt, int, area ID, range:[1,3], desc:the four points of the broken line are marked as 1, 2, 3, and 4 respectively in counterclockwise
direction from the counting direction. Draw two lines parallel to the counting direction respectively starting from point 2 and point 3 to intersect the
counting rule area. The formed three areas are marked as 1, 2, and 3 in counterclockwise direction from the counting direction-->1
</countingLineSegmentID>
<count>
<!--ro, opt, int, the number of people-->0
</count>
</MovingDirection>
</movingDirectionList>
</FlowStatistics>
</FlowStatisticsList>
<countingSceneMode>
<!--ro, opt, enum, people counting scene mode, subType:string, desc:"entrance", "exit". The default mode is entrance-->entrance
</countingSceneMode>
<deDuplicateEnter>
<!--ro, opt, int, deduplicated number of entered people-->0
</deDuplicateEnter>
<deDuplicateExit>
<!--ro, opt, int, deduplicated number of exited people-->0
</deDuplicateExit>
<faceLibDuplicatePeopleList>
<!--ro, opt, array, List of duplicated people in the face picture library, subType:object, desc:count the number of duplicated people in every face
picture library-->
<faceLibDuplicatePeople>
<!--ro, opt, object, number of duplicated people in the face picture library-->
<FDID>
<!--ro, req, string, face picture Library ID, desc:check details of the Library with /ISAPI/Intelligent/FDLib/search?format=json-->1
</FDID>
<name>
<!--ro, req, string, face picture Library name-->11
</name>
<duplicate>
<!--ro, opt, int, number of duplicated people in the library-->1
</duplicate>
</faceLibDuplicatePeople>
</faceLibDuplicatePeopleList>
</peopleCounting>
<childCounting>
<!--ro, opt, object, children counting-->
<enter>
<!--ro, opt, int, number of entered people-->1
</enter>
<exit>
<!--ro, opt, int, number of exited people-->1
</exit>
</childCounting>
<FaceExpressionList>
<!--ro, opt, array, facial expression statistics, subType:object-->
<FaceExpression>
<!--ro, opt, object, facial expression-->
<value>
<!--ro, req, enum, facial attributes, subType:string, desc:"unknown", "poker-faced" (no facial expression), "happy", "surprised", "disgusted",
"sad", "angry", "contemptuous", "panic", "all"-->unknown
</value>
<count>
<!--ro, req, int, number of people corresponding to the face expression-->1
</count>
</FaceExpression>
</FaceExpressionList>
<GenderList>
<!--ro, opt, array, gender statistics, subType:object-->
<Gender>
<!--ro, opt, object, gender-->
<value>
<!--ro, req, enum, options, subType:string, desc:"unknown", "female", "male", "all"-->unknown
</value>

```

```

<!--ro, req, enum, options, subType:string, desc: unknown, female, male, all -->unknown
</value>
<count>
  <!--ro, req, int, number of people corresponding to the gender-->1
</count>
</Gender>
</GenderList>
<MaskList>
  <!--ro, opt, array, mask statistics, subType:object-->
  <Mask>
    <!--ro, opt, object, mask wearing status-->
    <value>
      <!--ro, req, enum, options, subType:string, desc:"unknown", "yes" (wearing a mask), "no" (not wearing a mask)-->unknown
    </value>
    <count>
      <!--ro, req, int, number of people corresponding to the masks wearing status-->1
    </count>
    </Mask>
  </MaskList>
<AgeGroupList>
  <!--ro, opt, array, age group statistics, subType:object-->
  <AgeGroup>
    <!--ro, opt, object, age group-->
    <value>
      <!--ro, req, enum, options, subType:string, desc:"unknown", "child", "young", "middle", "old", "infant", "kid", "teenager", "prime", "middleAged" (the middle age), "all"-->unknown
    </value>
    <count>
      <!--ro, req, int, number of people corresponding to the age group-->1
    </count>
  </AgeGroup>
</AgeGroupList>
<GlassList>
  <!--ro, opt, array, glasses statistics, subType:object-->
  <Glass>
    <!--ro, opt, object, glasses wearing status-->
    <value>
      <!--ro, req, enum, options, subType:string, desc:"unknown", "yes" (wearing glasses), "no" (not wearing glasses), "sunglasses" (wearing sunglasses), "all"-->unknown
    </value>
    <count>
      <!--ro, req, int, number of people corresponding to the glass wearing attribute-->1
    </count>
  </Glass>
</GlassList>
<RegionList>
  <!--ro, opt, array, people counting information in the area, subType:object-->
  <Region>
    <!--ro, opt, object, area-->
    <id>
      <!--ro, req, int, area ID-->1
    </id>
    <name>
      <!--ro, opt, string, area name-->test
    </name>
    <enter>
      <!--ro, opt, int, number of entered people-->1
    </enter>
    <exit>
      <!--ro, opt, int, number of exited people-->1
    </exit>
    <totalEnter>
      <!--ro, opt, int, total number of entered people, desc:before resetting on the current day, the number will accumulate the number will be initialized only after manual resetting on the current day or auto resetting on the next day (if the device supports duplicated people counting, the number will be the deduplicated number of people of the current day)-->1
    </totalEnter>
    <totalExit>
      <!--ro, opt, int, total number of exited people, desc:before resetting on the current day, the number will accumulate the number will be initialized only after manual resetting on the current day or auto resetting on the next day (if the device supports duplicated people counting, the number will be the deduplicated number of people of the current day)-->1
    </totalExit>
    <totalPass>
      <!--ro, opt, int-->1
    </totalPass>
  </Region>
</RegionList>
<targetAttrs>
  <!--ro, opt, string, target attributes, range:[1,1024], desc:transparent transmission field, which corresponds to targetAttrs in applying task, format:>
  <![CDATA[{
    /*opt, transparent transmission field, the following three fields are recommended*/
    "deviceId": "1",
    /*opt, device ID, string, maximum length 64*/
    "deviceChannel": 1,
    /*opt, device channel, integer32*/
    "deviceName": "shebet"
    /*opt, device name, string, maximum length 128*/
  }]], example: <![CDATA[{"deviceId": "1", "deviceChannel": 1, "deviceName": "shebet"}]]-->test
</targetAttrs>
<targetInfo>
  <!--ro, opt, object, target information, desc:corresponding to targetInfo in heat analysis task /ISAPI/SDT/Management/HeatAnalysis/video/AddTask? format=json. Either targetInfo or targetAttrs is required-->
  <cameraInfo>
    <!--ro, opt, string, camera information-->test

```

```

<cameraIndexCode>
    <!--ro, req, string, camera ID-->test
</cameraIndexCode>
<cameraName>
    <!--ro, opt, string, camera name-->test
</cameraName>
</cameraInfo>
<mapfileID>
    <!--ro, opt, string, map file ID-->test
</mapfileID>
<storeID>
    <!--ro, opt, string, store ID-->test
</storeID>
</targetInfo>
<customInfo>
    <!--ro, opt, string, custom information-->test
</customInfo>
<TaskInfo>
    <!--ro, opt, object, task information submitted by data center products-->
<taskID>
    <!--ro, req, string, task ID-->test
</taskID>
<algorithmType>
    <!--ro, opt, enum, algorithm type, subType:int, desc:1-"hightQualityVehicle" (high-quality vehicle), 2-"hightPerformanceVehicle" (high-performance vehicle), 3-"personStructModel" (person structural modeling), 4-"vehicleAndPersonStructModel" (person and vehicle structural modeling), 5-"videoFaceDetect" (video face detection), 6-"videoFaceDetectModel" (video face detection and modeling), 7-"smart" (face detection + human body detection + behavior detection)-->1
        </algorithmType>
<taskName>
    <!--ro, opt, string, task name-->test
</taskName>
<algorithmID>
    <!--ro, opt, string, algorithm package ID-->test
</algorithmID>
<monitorPointID>
    <!--ro, opt, string, camera No.-->test
</monitorPointID>
<monitorPointName>
    <!--ro, opt, string, camera name-->test
</monitorPointName>
<streamType>
    <!--ro, opt, enum, stream type, subType:string, desc:"realtime", "historyvideo", "localvideo"-->realtime
</streamType>
<ruleID>
    <!--ro, opt, string, rule ID, desc:it is used to distinguish different tasks with the same name, and the ruleID in the event uploading message indicates that the event is related to this task-->test
</ruleID>
<ruleCustomName>
    <!--ro, req, string, user-defined event name, desc:it is defined by the caller, and it is included in the message when the alarm is uploaded. Up to 128 characters are allowed-->test
</ruleCustomName>
<algorithmInfo>
    <!--ro, opt, object, algorithm version information, desc:this field is required for secondary analysis-->
<algorithmId>
    <!--ro, opt, string, algorithm package ID-->test
</algorithmId>
<algoVendorId>
    <!--ro, opt, string, manufacturer ID-->test
</algoVendorId>
<algoVersionId>
    <!--ro, opt, string, algorithm version No.-->test
</algoVersionId>
<algoChiptype>
    <!--ro, opt, string, chip type-->test
</algoChiptype>
<algoModelVersion>
    <!--ro, opt, string, model version-->test
</algoModelVersion>
</algorithmInfo>
</TaskInfo>
<bkgUrl>
    <!--ro, opt, string, background picture URL, desc:background picture URL-->test
</bkgUrl>
<timeStamp>
    <!--ro, opt, string, timestamp of the frame, which triggers the alarm, desc:ISO 8601 format, e.g., 2019-12-30T15:00:00.23+08:00-->test
</timeStamp>
<DetectionRegionList>
    <!--ro, opt, array, detection area, subType:object-->
<DetectionRegionEntry>
    <!--ro, opt, object, detection area-->
<regionID>
    <!--ro, req, string, area ID-->test
</regionID>
<sensitivityLevel>
    <!--ro, opt, int, sensitivity Level, range:[0,100]-->1
</sensitivityLevel>
<RegionCoordinatesList>
    <!--ro, opt, array, target area, subType:object, range:[3,10]-->
<RegionCoordinates>
    <!--ro, opt, object, area coordinates, desc:the origin is the upper-left corner of the screen-->
<positionX>
    <!--ro, req, int, X-coordinate, range:[0,1000]-->1
</positionX>
<positionY>
    <!--ro, req, int, Y-coordinate, range:[0,1000]-->1
</positionY>

```

```

<positionY>
  <!--ro, req, int, Y-coordinate, range:[0,1000]-->1
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</DetectionRegionEntry>
</DetectionRegionList>
<HatList>
  <!--ro, opt, array, wearing hat statistics, subType:object-->
<Hat>
  <!--ro, opt, object, whether the person wears a hat-->
<value>
  <!--ro, req, enum, options, subType:string, desc:"unknown", "yes" (wearing a hat), "no" (not wearing a hat), "helmet" (wearing a helmet)-->unknown
</value>
<count>
  <!--ro, req, int, number of people wearing hats-->1
</count>
</Hat>
</HatList>
<ThingsList>
  <!--ro, opt, array, carrying objects statistics, subType:object-->
<Things>
  <!--ro, opt, object, carrying objects-->
<value>
  <!--ro, req, enum, options, subType:string, desc:"unknown", "yes", "no"-->unknown
</value>
<count>
  <!--ro, req, int, number of people carrying objects-->1
</count>
</Things>
</ThingsList>
<isDataRetransmission>
  <!--ro, opt, bool, whether the data is replenished, desc:if the real-time detection data upload fails due to the network condition, the data will be
uploaded again after the device recovers-->true
</isDataRetransmission>
</EventNotificationAlert>

```

## 16.6 Vehicle Recognition

### 16.6.1 Vehicle Detection

#### 16.6.1.1 Get the arming schedule of vehicle detection

##### Request URL

GET /ISAPI/Event/schedules/vehicledetects

##### Query Parameter

None

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<VehicledetectScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, subType:object, attr:version{req, string, protocolVersion}-->
</VehicledetectScheduleList>

```

#### 16.6.1.2 Set ANPR arming schedules of all channels

##### Request URL

PUT /ISAPI/Event/schedules/vehicledetects

##### Query Parameter

None

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<VehicledetectScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, arming time list, subType:object, attr:version{req, string, protocolVersion}-->
</VehicledetectScheduleList>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.6.1.3 Restore the traffic violation dictionary to default settings

#### Request URL

PUT /ISAPI/ITC/illegalDictionary/reset

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL, range:[0,1024]-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status description, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, error code description, desc:error code description-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.6.1.4 License plate recognition

#### EventType:ANPR

```

<?xml version="1.0" encoding="UTF-8"?>

<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, license plate recognition result, attr:version{opt, string, protocolVersion}-->
  <ipAddress>
    <!--ro, req, string, IPv4 address of the device that triggers the alarm-->172.6.64.7
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPv6 address of the device that triggers the alarm-->1080:0:0:0:8:800:200C:417A
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, communication port No. of the device that triggers the alarm-->80
  </portNo>
  <protocol>
    <!--ro, req, enum, protocol type, subType:string, desc:transmission communication protocol type: "HTTP", "HTTPS", "EHome". The value should be "HTTP" when ISAPI protocol is transmitted via EZ protocol. The value should be "EHome" when ISAPI protocol is transmitted via ISUP-->HTTP
  </protocol>
  <macAddress>
    <!--ro, opt, string, MAC address-->01:17:24:45:D9:F4
  </macAddress>
  <dynChannelID>
    <!--ro, opt, string, digital channel No.-->test
  </dynChannelID>
  <channelID>
    <!--ro, opt, string, channel ID-->1
  </channelID>
</EventNotificationAlert>

```

```

<!--ro, opt, int, channel no. of the device that triggers the alarm, desc:video channel no. that triggers the alarm-->1
</channelID>
<relatedChannelList>
    <!--ro, opt, string, list of alarm related channels, which are of the same camera with channelID, desc:this parameter is used for live view or playback on the platform; multiple channel No.s are separated by commas-->1,2,3
</relatedChannelList>
<dateTime>
    <!--ro, req, datetime, alarm trigger time-->1970-01-01T00:00:00+08:00
</dateTime>
<activePostCount>
    <!--ro, opt, int, times that the same alarm has been uploaded, desc:event triggering frequency-->1
</activePostCount>
<eventType>
    <!--ro, req, enum, event type, subType:string, desc:"ANPR" (license plate recognition)-->ANPR
</eventType>
<eventState>
    <!--ro, req, enum, continuous event status, subType:string, desc:for durative event: active (valid), inactive (invalid)-->active
</eventState>
<eventDescription>
    <!--ro, req, enum, event description, subType:string, desc:"ANPR" (license plate recognition)-->ANPR
</eventDescription>
<channelName>
    <!--ro, opt, string, channel name, range:[1,128]-->test
</channelName>
<deviceID>
    <!--ro, opt, string, device ID, desc:it should be returned for ISUP alarms, e.g., test0123 (Ehome2.0, Ehome4.0, and ISUP5.0)-->12345
</deviceID>
<ANPR>
    <!--ro, opt, object, information about license plate recognition alarm-->
<region>
    <!--ro, opt, enum, region, subType:string, desc:"ER" (Russian-speaking region), "EU" (Europe), "EUandCIS" (Europe and Russia), "ME" (Middle East), "other" (other regions), "APAC" (Asia-Pacific region), "AFandAM" (Africa and America), "THAandLA" (Thailand and Laos), "HKandMO" (Hong Kong China and Macao China), "ALL" (all regions),-->ER
        </region>
<country>
    <!--ro, opt, enum, country or region, subType:int, desc:253-invalid, 70-Philippines, 17-United Kingdom (previously Great Britain), 23-Macedonia (North Macedonia since 2018), 188-Dominican, 227-Australia, 60-Bahrain, 95-Burma/Myanmar, 228-New Zealand, 28-Azerbaijan, 131-Gambia, 132-Mali, 159-Saint Helena, 133-Burkina Faso, 134-Guinea, 135-Guinea-Bissau, 136-Cape Verde, 137-Sierra Leone, 138-Liberia, 139-Ivory Coast, 140-Ghana, 141-Togo, 142-Benin, 143-Niger, 144-Zambia, 145-Angola, 146-Zimbabwe, 147-Malawi, 148-Mozambique, 149-Botswana, 150-Namibia, 151-South Africa, 152-Swaziland, 153-Lesotho, 154-Madagascar, 155-Comoros, 156-Mauritius, 157-Nigeria, 158-South Sudan, 160-Mayotte, 161-Reunion, 162-Canary Islands, 163-AZORES, 164-Madeira, 165-reserved, 166-reserved, 167-reserved, 168-reserved, 169-Canada, 170-Greenland Nuuk, 171-Pierre and Miquelon, 172-United States, 173-Bermuda, 174-Mexico, 175-Guatemala, 176-Belize, 177-El Salvador, 178-Honduras, 179-Nicaragua, 180-Costa Rica, 181-Panama, 182-Bahamas, 183-Turks and Caicos Islands, 184-Cuba, 185-Jamaica, 186-Cayman Islands, 187-Haiti, 189-Puerto Rico, 190-United States Virgin Islands, 191-British Virgin Islands, 192-Anguilla, 193-Antigua and Barbuda, 194-Collectivite de Saint-Martin, 195-Autonomous country, 196-Saint Barthlemy, 197-Saint Kitts and Nevis, 198-Montserrat, 199-Guadeloupe, 200-Dominica, 201-Martinique, 202-St. Lucia, 203-Saint Vincent and the Grenadines, 204-Grenada, 205-Barbados, 206-Trinidad and Tobago, 207-Curaçao, 0-Unsupported, 1-Czech Republic, 2-France, 3-Germany, 4-Spain, 5-Italy, 6-Netherlands, 7-Poland, 8-Slovakia, 9-Belarus, 10-Moldova, 11-Russia, 12-Ukraine, 13-Belgium, 14-Bulgaria, 15-Denmark, 16-Finland, 18-Greece, 19-Croatia, 20-Hungary, 21-Israel(), 22-Luxembourg, 24-Norway, 25-Portugal, 26-Romania, 27-Serbia, 29-Georgia(), 30-Kazakhstan(), 31-Lithuania, 32-Turkmenistan(), 33-Uzbekistan(), 34-Latvia, 35-Estonia, 36-Albania, 37-Austria, 38-Bosnia and Herzegovina, 39-Ireland, 40-Iceland, 41-Vatican, 42-Malta, 43-Sweden, 44-Switzerland, 45-Cyprus, 46-Turkey, 47-Slovenia, 48-Montenegro, 49-Kosovo, 50-Andorra, 51-Armenia(), 52-Monaco, 53-Liechtenstein, 54-San Marino, 55-reserved, 56-reserved, 57-reserved, 58-reserved, 59-China, 91-Vietnam, 61-South Korea, 62-Lebanon, 63-Nepal, 64-Thailand, 65-Pakistan, 66-United Arab Emirates, 67-Bhutan, 68-Oman, 69-North Korea, 71-Cambodia, 72-Qatar, 73-Kyrgyzstan, 74-Maldives, 75-Malaysia, 76-Mongolia, 77-Saudi Arabia, 78-Brunei, 79-Laos, 80-Japan, 81-Turkey, 82-Palestinian, 83-Tajikistan, 84-Kuwait, 85-Syria, 86-India, 87-Indonesia, 88-Afghanistan, 89-Sri Lanka, 92-Iran, 93-Yemen, 94-Jordan, 96-Sikkim, 97-Bangladesh, 98-Singapore, 99-Democratic Republic of Timor-Leste, 100-reserved, 101-reserved, 102-reserved, 103-reserved, 104-Egypt, 105-Libya, 106-Sudan, 107-Tunisia, 108-Algeria, 109-Morocco, 110-Ethiopia, 111-Eritrea, 112-Somalia Democratic, 113-Djibouti, 114-Kenya, 115-Tanzania, 116-Uganda, 117-Rwanda, 118-Burundi, 119-Seychelles, 120-Chad, 121-Central African, 122-Cameroun, 123-Equatorial Guinea, 124-Gabon, 125-Congo, 126-Democratic Republic of the Congo, 127-Sao Tome and Principe, 128-Mauritania, 129-Western Sahara, 130-Senegal, 208-Aruba, 209-Netherlands Antilles, 210-Colombia, 211-Venezuela, 212-Guyana, 213-Suriname, 214-Guyane Francaise, 215-Ecuador, 216-Peru, 217-Bolivia, 218-Paraguay, 219-Chile, 220-Brazil, 221-Uruguay, 222-Argentina, 223-reserved, 224-reserved, 225-reserved, 226-reserved, 229-Papua New Guinea, 230-Salomonen, 231-Vanuatu, 232-New Caledonia, 233-Palau, 234-Federated States of Micronesia, 235-Marshall Island, 236-Northern Mariana Islands, 237-Guam, 238-Nauru, 239-Kiribati, 240-Fidschi, 241-Tonga, 242-Tuvalu, 243-Wallis et Futuna, 244-Samoa, 245-Eastern Samoa, 246-Tokelau, 247-Niue, 248-Cook Islands, 249-French Polynesia, 250-Pitcairn Islands, 251-Hawaii State, 252-reserved, 254-unrecognized, 255-all-->253
</country>
<area>
    <!--ro, opt, enum, United Arab Emirates, subType:string, desc:"FJR" (Fujairah), "AD" (Abu Dhabi), "unknown", "UMN" (Umm al-Qaiwain), "other", "AM" (Ajman), "RAK" (Ras al-Kaimah), "DB" (Dubai), "SJ" (Sharjah)-->FJR
</area>
<licensePlate>
    <!--ro, req, string, License plate number, range:[1,32], desc:noPlate (vehicle without license plate), unknown (no license plate recognized), XXXXXXXX (specific license plate number; color information required by Chinese license plate; for motor vehicles, it is a 16-byte string; for non-motor vehicles, it is a 48-byte string)-->A283KY77
</licensePlate>
<cameraNo>
    <!--ro, opt, string, device No., range:[1,48], desc:corresponds to <cameraNum> from /ISAPI/System/Video/inputs/channels/<ID>/cameraInfo (recommended for new devices) and /ISAPI/Traffic/channels/<channelID>/cameraInfo applied by some traffic monitoring device-->test
</cameraNo>
<line>
    <!--ro, req, int, recognized Lane No.-->1
</line>
<direction>
    <!--ro, opt, enum, license plate recognition direction, subType:string, desc:"reverse", "forward", "unknown"-->reverse
</direction>
<confidenceLevel>
    <!--ro, req, int, confidence Level, range:[0,100]-->50
</confidenceLevel>
<plateType>
    <!--ro, opt, enum, license plate type, subType:string, desc:"unknown", "arm" (police vehicle), "92TypeArm" (type 92 armed police vehicle), "02TypePersonalized" (type 02 custom vehicle), "yellowTwoLine" (yellow two-line rear license plate), "embassy" (embassy vehicle), "oneLineArm"(new armed police vehicle, one-line), "twoLineArm" (new armed police vehicle, two-line), "yellow1225FarmVehicle" (agricultural vehicle, yellow 1225), "green1325FarmVehicle" (agricultural vehicle, green 1325), "yellow1325FarmVehicle" (agricultural vehicle, yellow 1325), "motorola" (motorcycle), "coach" (driver-training vehicle), "tempTravL" (vehicle with temporary license plate), "trailer" (trailer), "consulate" (consular vehicle), "hongKongMacao" (Hong Kong China/Macao China entrance vehicle), "tempEntry" (temporary vehicle), "civilAviation" (civil aviation license plate), "newEnergy" (new energy vehicle), "92FarmVehicle" (two-line license plate civil vehicle), "emergency" (emergency license plate), "oneLineArmHeadquarters" (armed police headquarter license plate, one-line), "twoLineArmHeadquarters" (armed police headquarter license plate, two-line), "twoWheelVehicle" (two wheeler)-->unknown
</plateType>

```

```

<plateColor>
  <!--ro, opt, enum, license plate color, subType:string, desc:"black", "blue", "golden", "orange", "red", "yellow", "white", "unknown", "other", "newEnergyYellowGreen"-->new energy yellow-green,"civilAviationBlack", "civilAviationGreen" (civil aviation green), "green", "mixedColor", "newEnergyGreen" (new energy green), "brown"-->black
</plateColor>
<licenseBright>
  <!--ro, opt, int, license plate brightness, which ranges from 0 to 255, range:[0,255]-->50
</licenseBright>
<Rect>
  <!--ro, opt, object, coordinates of the license plate thumbnail in the matched picture, desc:used by back-end DeepinMind; the origin is the upper-left corner of the screen-->
    <height>
      <!--ro, req, float, height, range:[0.000,1.000]-->1.000
    </height>
    <width>
      <!--ro, req, float, width, range:[0.000,1.000]-->1.000
    </width>
    <x>
      <!--ro, req, float, the reference origin is the upper left corner of image, range:[0.000,1.000]-->1.000
    </x>
    <y>
      <!--ro, req, float, the reference origin is the upper left corner of image, range:[0.000,1.000]-->1.000
    </y>
  </Rect>
<pilotsafebelt>
  <!--ro, opt, enum, whether the driver is wearing a safety belt, subType:string, desc:"unknown", "yes", "no"-->unknown
</pilotsafebelt>
<vicepilotsafebelt>
  <!--ro, opt, enum, whether the co-driver is wearing a safety belt, subType:string, desc:"unknown", "yes", "no"-->unknown
</vicepilotsafebelt>
<pilotsunvisor>
  <!--ro, opt, enum, whether the driver room's sun visor is open, subType:string, desc:"unknown", "yes", "no"-->unknown
</pilotsunvisor>
<vicepilotunvisor>
  <!--ro, opt, enum, whether the co-driver room's sun visor is open, subType:string, desc:"unknown", "yes", "no"-->unknown
</vicepilotunvisor>
<envprosign>
  <!--ro, opt, enum, whether it is a yellow-label vehicle, subType:string, desc:"unknown", "green" (green-label), "yellow" (yellow-label), "yes", "no"-->unknown
</envprosign>
<dangmark>
  <!--ro, opt, enum, whether it is a dangerous goods vehicle, subType:string, desc:"unknown", "yes", "no"-->unknown
</dangmark>
<uphone>
  <!--ro, opt, enum, whether the driver is making a phone call, subType:string, desc:"unknown", "yes", "no"-->unknown
</uphone>
<pendant>
  <!--ro, opt, enum, whether there are window hangings detected, subType:string, desc:"unknown", "yes", "no"-->unknown
</pendant>
<tissueBox>
  <!--ro, opt, enum, whether there is a tissue box detected, subType:string, desc:"unknown", "yes", "no"-->unknown
</tissueBox>
<frontChild>
  <!--ro, opt, enum, whether the co-driver is with a baby in arm, subType:string, desc:"unknown", "yes", "no"-->unknown
</frontChild>
<label>
  <!--ro, opt, enum, whether there are stickers detected, subType:string, desc:"unknown", "yes", "no"-->unknown
</label>
<decoration>
  <!--ro, opt, enum, whether there are decorations detected, subType:string, desc:"unknown", "yes", "no"-->unknown
</decoration>
<smoking>
  <!--ro, opt, enum, whether anyone is smoking, subType:string, desc:"unknown", "yes", "no"-->yes
</smoking>
<perfumeBox>
  <!--ro, opt, enum, whether there is perfume box detected, subType:string, desc:"unknown", "yes", "no"-->unknown
</perfumeBox>
<pdvs>
  <!--ro, opt, enum, whether there is a person sticking out of sunroof, subType:string, desc:"unknown", "yes", "no"-->unknown
</pdvs>
<helmet>
  <!--ro, opt, enum, whether there is helmet detected, subType:string, desc:"unknown", "yes", "no"-->no
</helmet>
<twoWheelVehicle>
  <!--ro, opt, enum, whether there is a two-wheel vehicle detected, subType:string, desc:"unknown", "yes", "no"-->unknown
</twoWheelVehicle>
<threeWheelVehicle>
  <!--ro, opt, enum, whether there is a three-wheel vehicle detected, subType:string, desc:"unknown", "yes", "no"-->unknown
</threeWheelVehicle>
<blackness>
  <!--ro, opt, int, Ringelmann emittance, desc:it is used for black smoke detection-->2
</blackness>
<plateCharBelieve>
  <!--ro, opt, string, confidence of each character in the recognized License plate, desc:if "A12345" is detected as a license plate number and the confidence is 10, 20, 30, 40, 50, 60, it indicates that the accuracy of "A" is 20%-->test
</plateCharBelieve>
<speedLimit>
  <!--ro, opt, int, maximum speed limit, desc:it is valid only when overspeeding occurred-->50
</speedLimit>
<illegalInfo>
  <!--ro, opt, object, vehicle traffic violation information, desc:illegal action information-->
  <illegalCode>
    <!--ro, req, string, illegal action code, range:[0,64]-->1301
  </illegalCode>

```

```

</illegalCode>
<illegalName>
  <!--ro, req, string, illegal action name, range:[0,128]-->逆行
</illegalName>
<illegalDescription>
  <!--ro, opt, string, illegal action description, range:[0,256]-->车辆逆行
</illegalDescription>
</illegalInfo>
<vehicleType>
  <!--ro, req, enum, vehicle type, subType:string, desc:vehicle type:>
"unknown",LargeBus,truck,vehicle,van,buggy,pedestrian,twoWheelVehicle,threeWheelVehicle,SUMMPV,mediumBus,motorVehicle,nonmotorVehicle,smallCar,miniCar,pickup
Truck"-->nonmotorVehicle
</vehicleType>
<postPicFileName>
  <!--ro, opt, string, name of the picture selected as the checkpoint picture when illegal action occurs, desc:"none" (no picture selected)-->test
</postPicFileName>
<featurePicFileName>
  <!--ro, opt, string, name of the picture selected as the close-up picture when running the red Light in the intersection violation system is detected, desc:"none" (no picture selected)-->test
</featurePicFileName>
<detectDir>
  <!--ro, opt, enum, detection direction, subType:int, desc:1 (upward), 2 (downward), 3 (bidirectional), 4 (westward), 5 (northward), 6 (eastward), 7 (southward), 8 (other)-->1
</detectDir>
<detectType>
  <!--ro, opt, enum, detection type, subType:int, desc:0 (vehicle detection), 1 (inductive loop trigger), 2 (video trigger), 3 (multiple-frame recognition), 4 (radar trigger), 5 (mixed-traffic detection)-->1
</detectType>
<barrierGateCtrlType>
  <!--ro, opt, enum, whether to enable elapsed time, subType:int, desc:0 (enabled), 1 (disabled)-->0
</barrierGateCtrlType>
<alarmDataType>
  <!--ro, opt, enum, alarm data type (real-time data or history data), subType:int, desc:0 (real-time data), 1 (history data)-->0
</alarmDataType>
<dwIllegalTime>
  <!--ro, opt, int, illegal action duration, unit:ms, desc:it is the difference between the capture time of the last picture and the capture time of the first picture-->100
</dwIllegalTime>
<vehicleInfo>
  <!--ro, opt, object, vehicle information-->
<index>
  <!--ro, req, int, vehicle No.-->1
</index>
<vehicleType>
  <!--ro, opt, enum, vehicle type, subType:int, desc:vehicle type:>
"unknown",LargeBus,truck,vehicle,van,buggy,pedestrian,twoWheelVehicle,threeWheelVehicle,SUMMPV,mediumBus,motorVehicle,nonmotorVehicle,smallCar,miniCar,pickup
Truck"-->0
</vehicleType>
<colorDepth>
  <!--ro, req, enum, shade of the vehicle color, subType:int, desc:0 (deep color), 1 (light color), 2 (unknown)-->0
</colorDepth>
<color>
  <!--ro, req, enum, vehicle color, subType:string, desc:"unknown", "green", "brown", "pink", "purple", "deepGray" (dark gray), "cyan", "orange", "white", "silver" (silvery), "gray", "black", "red", "deepBlue" (dark blue), "blue", "yellow"-->green
</color>
<speed>
  <!--ro, opt, int, vehicle speed, unit:km/h-->1
</speed>
<length>
  <!--ro, req, int, Length of the former vehicle, unit:dm-->10
</length>
<vehicleLogoRecog>
  <!--ro, req, int, vehicle main brand-->1
</vehicleLogoRecog>
<vehicleSubLogoRecog>
  <!--ro, opt, int, vehicle sub-brand-->1
</vehicleSubLogoRecog>
<vehicleModel>
  <!--ro, opt, int, model year of vehicle sub-brand-->1
</vehicleModel>
<vehicleTypeByWeight>
  <!--ro, opt, enum, vehicle type according to the vehicle weight, subType:int, desc:1 (class 1: buses with seven or less seats, trucks with capacity of 2 tons or less), 2 (class 2: buses with 8 to 19 seats, trucks with capacity of 2 to 5 (included) tons), 3 (class 3: buses with 20 to 39 seats, trucks with capacity of 5 to 10 (included) tons), 4 (class 4: buses with 40 or more seats, trucks with capacity of 10 to 15 (included) tons), 5 (class 5: trucks with capacity of more than 15 tons), 6 (class 6: special vehicle such as ambulance, fire truck, and garbage truck)-->4
</vehicleTypeByWeight>
<tollRoadVehicleSeries>
  <!--ro, opt, enum, vehicles on tollway, subType:int, desc:N/A-->2
</tollRoadVehicleSeries>
<tollRoadVehicleType>
  <!--ro, opt, enum, vehicle types on tollway, subType:int, desc:N/A-->4
</tollRoadVehicleType>
<carWindowFeature>
  <!--ro, opt, object, vehicle window feature, desc:configured via <CarWindowFeature> from /ISAPI/ITC/carFeatureParam-->
<tempPlate>
  <!--ro, opt, enum, whether there is temporary license plate, subType:string, desc:detect whether there is temporary license plate on the window:>
"unknown", "yes", "no"-->unknown
</tempPlate>
<passCard>
  <!--ro, opt, enum, whether there is entry & exit pass, subType:string, desc:detect whether there is entry & exit pass on the window:>
"unknown", "yes", "no"-->unknown
</passCard>
<carCard>
  <!--ro, opt, enum, whether there are cards detected, subType:string, desc:whether there is any card (business cards, flyers, etc.) attached to the

```

```

    ... opt, enum, whether there are cards detected, subType:string, desc:whether there is any card (business card, id card, etc) attached to the
window: "unknown", "yes", "no"-->unknown
</carCard>
</CarWindowFeature>
<CarBodyFeature>
    <!--ro, opt, object, bodywork feature, desc:configured via <CarBodyFeature> from /ISAPI/ITC/carFeatureParam-->
</spareTire>
    <!--ro, opt, enum, whether it is with a spare tire, subType:string, desc:"unknown", "yes", "no"-->unknown
</spareTire>
</rack>
    <!--ro, opt, enum, whether it is with a Luggage rack, subType:string, desc:"unknown", "yes", "no"-->unknown
</rack>
</sunRoof>
    <!--ro, opt, enum, whether there is a sunroof, subType:string, desc:"unknown", "yes", "no"-->unknown
</sunRoof>
</words>
    <!--ro, opt, enum, whether the vehicle is painted, subType:string, desc:"unknown", "yes", "no"-->unknown
</words>
</slatTruckCoverPlate>
    <!--ro, opt, enum, whether the back cover of the dump truck is covered, subType:string, desc:"unknown", "yes", "no"-->unknown
</slatTruckCoverPlate>
</reflectiveTape>
    <!--ro, opt, enum, subType:string-->unknown
</reflectiveTape>
</closureWindow>
    <!--ro, opt, enum, subType:string-->unknown
</closureWindow>
</CarBodyFeature>
</vehicleUseType>
    <!--ro, opt, enum, vehicle purpose type, subType:string, desc:configured via vehicleUseEnable from /ISAPI/ITC/carFeatureParam: "taxi", "ambulance",
"bus", "schoolbus", "coach", "unknown"-->taxi
</vehicleUseType>
</carWindowInfo>
    <!--ro, opt, object, vehicle window information, desc:configured via /ISAPI/ITC/TriggerMode/HVLaneDetection?format=json-->
</carWindowNum>
    <!--ro, opt, int, number of vehicle windows-->4
</carWindowNum>
</carWindowStatusList>
    <!--ro, opt, object, vehicle window status list-->
</carWindowStatus>
    <!--ro, opt, object, vehicle window status-->
</passengers>
    <!--ro, opt, int, number of persons behind the window, range:[0,10]-->1
</passengers>
</windowStatus>
    <!--ro, opt, enum, whether the vehicle windows are transparent, subType:string, desc:"unknown", "yes", "no"-->yes
</windowStatus>
</carWindowStatus>
</carWindowStatusList>
</carWindowInfo>
</actualPassengers>
    <!--ro, opt, int, number of passengers in the vehicle, range:[0,100]-->1
</actualPassengers>
</vehiclePosition>
    <!--ro, opt, enum, detected vehicle body part, subType:string, desc:"LeftSide", "rightSide", "headStock", "tailStock", "roof"-->leftSide
</vehiclePosition>
</containerInfo>
    <!--ro, opt, object, container information-->
</containerNum>
    <!--ro, opt, int, number of transferred containers, range:[1,10]-->2
</containerNum>
</containerList>
    <!--ro, opt, array, list of container information, subType:object, range:[1,10]-->
</container>
    <!--ro, opt, object, container information-->
</containerMainNum>
    <!--ro, opt, string, major No. of captured container, range:[1,32]-->abcd1234
</containerMainNum>
</containerSubNum>
    <!--ro, opt, string, minor No. of captured container, range:[1,32]-->abcd1234
</containerSubNum>
</containerISONum>
    <!--ro, opt, string, ISO No. of captured container, range:[1,32]-->abcd1234
</containerISONum>
</containerNumConfidence>
    <!--ro, opt, int, confidence of captured container No., range:[0,100]-->0
</containerNumConfidence>
</container>
</containerList>
</containerInfo>
</RFID>
    <!--ro, opt, string, Link RFID, range:[1,64], dep:and, ${$.EventNotificationAlert.ANPR.vehicleType,eq,nonmotorVehicle}-->12345
</RFID>
</vehicleCarryFeature>
    <!--ro, opt, object-->
</isCarry>
    <!--ro, opt, enum, subType:string-->unknown
</isCarry>
</isTarpaulin>
    <!--ro, opt, enum, subType:string-->unknown
</isTarpaulin>
</isCoverplate>
    <!--ro, opt, enum, subType:string-->unknown
</isCoverplate>

```

```

</vehicleCarryFeature>
</vehicleInfo>
<EntranceInfo>
    <!--ro, opt, object, entrance and exit information-->
    <parkingID>
        <!--ro, opt, string, parking space No.-->test
    </parkingID>
    <gateID>
        <!--ro, opt, string, entrance and exit No.-->test
    </gateID>
    <direction>
        <!--ro, opt, enum, target direction, subType:string, desc:"enter", "Leave"-->enter
    </direction>
    <cardNo>
        <!--ro, opt, string, card No.-->test
    </cardNo>
    <parkType>
        <!--ro, opt, enum, parking type, subType:string, desc:"temporary", "permanent"-->temporary
    </parkType>
</EntranceInfo>
<pictureInfoList>
    <!--ro, req, array, picture List, subType:object, range:[0,8]-->
    <pictureInfo>
        <!--ro, req, object, picture information-->
        <fileName>
            <!--ro, req, enum, picture name, subType:string, desc:"detectionPicture.jpg" (background picture), "licensePlatePicture.jpg" (license plate picture), "pilotPicture.jpg" (driver's picture matting), "copilotPicture.jpg" (co-driver's picture matting), "compositePicture.jpg" (composite picture), "plateBinaryPicture.jpg" (license plate binary picture), "nonMotorPicture.jpg" (non-motor vehicle picture matting), "pedestrianDetectionPicture.jpg" (pedestrian picture), "pedestrianPicture.jpg" (pedestrian's picture matting), "vehiclePicture.jpg" (vehicle picture). Picture name, which must correspond to the picture name transmitted with the alarm message-->detectionPicture.jpg
        </fileName>
        <type>
            <!--ro, req, enum, picture type, subType:string, desc:"detectionPicture" (background picture), "licensePlatePicture" (license plate picture), "pilotPicture" (driver's picture matting), "copilotPicture" (co-driver's picture matting), "compositePicture" (composite picture), "plateBinaryPicture" (license plate binary picture), "nonMotorPicture" (non-motor vehicle picture matting), "pedestrianDetectionPicture" (pedestrian picture), "pedestrianPicture" (pedestrian's picture matting), "vehiclePicture" (vehicle picture)-->vehiclePicture
        </type>
        <dataType>
            <!--ro, req, enum, data type, subType:int, desc:0 (binary), 1 (URL)-->0
        </dataType>
        <picRecogMode>
            <!--ro, opt, enum, license plate recognition mode, subType:int, desc:0 (front license plate recognition), 1 (rear license plate recognition)-->0
        </picRecogMode>
        <redLightTime>
            <!--ro, opt, int, red light time elapsed, unit:s-->60
        </redLightTime>
        <vehicleHead>
            <!--ro, opt, enum, license plate recognition direction, subType:string, desc:"unknown", "forward" (front license plate recognition), "back" (rear license plate recognition)-->unknown
        </vehicleHead>
        <absTime>
            <!--ro, opt, string, absolute time, desc:format: yyyyMMddHHmmssxxx, e.g.: 20090810235959999, the last three digits are time in millisecond-->20090810235959999
        </absTime>
        <plateRect>
            <!--ro, opt, object, license plate area coordinates, desc:this field is valid when the value of type is "detectionPicture". The normalized value is the current image size in percentage multiplying 1000 and it is accurate to three decimal places. The origin is the upper-left corner of screen-->
            <X>
                <!--ro, req, int, X-coordinate of the upper-left corner of the boundary frame, range:[0,1000]-->1000
            </X>
            <Y>
                <!--ro, req, int, Y-coordinate of the upper-left corner of the boundary frame, range:[0,1000]-->1000
            </Y>
            <width>
                <!--ro, req, int, width of the boundary frame, range:[0,1000]-->1000
            </width>
            <height>
                <!--ro, req, int, height of the boundary frame, range:[0,1000]-->1000
            </height>
        </plateRect>
        <vehicleRect>
            <!--ro, opt, object, the vehicle area coordinates, desc:this field is valid when the value of type is "detectionPicture". The normalized value is the current image size in percentage multiplying 1000. The origin is the upper-left corner of screen-->
            <X>
                <!--ro, req, int, X-coordinate of the upper-left corner of the boundary frame, range:[0,1000]-->1000
            </X>
            <Y>
                <!--ro, req, int, Y-coordinate of the upper-left corner of the boundary frame, range:[0,1000]-->1000
            </Y>
            <width>
                <!--ro, req, int, width of the boundary frame, range:[0,1000]-->1000
            </width>
            <height>
                <!--ro, req, int, height of the boundary frame, range:[0,1000]-->1000
            </height>
        </vehicleRect>
        <pictureURL>
            <!--ro, opt, string, picture URL, desc:it is valid when the value of dataType is "URL"-->test
        </pictureURL>
        <pId>
            <!--ro, opt, string, strlen.max=32, range:[1,32], desc:strlen.max=32-->test
        </pId>
        <province>
            <!--ro, opt, enum, province index, subType:int, desc:N/A-->A
        </province>

```

```

<!--ro, opt, enum, province index, subType:int, desc:IV/A-->
</province>
<PilotRect>
<!--ro, opt, object-->
<x>
<!--ro, req, int, range:[0,1000]-->1000
</x>
<y>
<!--ro, req, int, range:[0,1000]-->1000
</y>
<width>
<!--ro, req, int, range:[0,1000]-->1000
</width>
<height>
<!--ro, req, int, range:[0,1000]-->1000
</height>
</PilotRect>
<VicepilotRect>
<!--ro, opt, object-->
<x>
<!--ro, req, int, range:[0,1000]-->1000
</x>
<y>
<!--ro, req, int, range:[0,1000]-->1000
</y>
<width>
<!--ro, req, int, range:[0,1000]-->1000
</width>
<height>
<!--ro, req, int, range:[0,1000]-->1000
</height>
</VicepilotRect>
<VehicleWindowRect>
<!--ro, opt, object-->
<x>
<!--ro, req, int, range:[0,1000]-->1000
</x>
<y>
<!--ro, req, int, range:[0,1000]-->1000
</y>
<width>
<!--ro, req, int, range:[0,1000]-->1000
</width>
<height>
<!--ro, req, int, range:[0,1000]-->1000
</height>
</VehicleWindowRect>
<extraPlateRect>
<!--ro, opt, object-->
<x>
<!--ro, req, int, range:[0,1000]-->1000
</x>
<y>
<!--ro, req, int, range:[0,1000]-->1000
</y>
<width>
<!--ro, req, int, range:[0,1000]-->1000
</width>
<height>
<!--ro, req, int, range:[0,1000]-->1000
</height>
</extraPlateRect>
</pictureInfo>
</pictureInfolist>
<hasMoreData>
<!--ro, opt, bool, whether there is more data, desc:this field is used to report the license plate information first, and then report XML message with picture data; the XM message with picture data and license plate information are linked by UUID-->true
</hasMoreData>
<listType>
<!--ro, opt, enum, List Property, subType:string, desc:"white" (allowlist), "black" (blocklist), "temporary" (temporary list)-->white
</listType>
<originalLicensePlate>
<!--ro, opt, string, original license plate number, desc:when the original license plate number contains non-English characters, the original license plate number can be returned-->test
</originalLicensePlate>
<CRIndex>
<!--ro, opt, enum, country or region index, subType:int, desc:country or region index-->258
</CRIndex>
<VehicleGPSInfo>
<!--ro, opt, object, GPS information of the vehicle-->
<longitudeType>
<!--ro, req, enum, Longitude, subType:string, desc:"E" (Eastern Hemisphere), W" (West Hemisphere)-->E
</longitudeType>
<latitudeType>
<!--ro, req, enum, Latitude, subType:string, desc:"S" (Southern Hemisphere), "N" (North Hemisphere)-->S
</latitudeType>
<Longitude>
<!--ro, req, object, longitude information-->
<degree>
<!--ro, req, int, degree(s)-->60
</degree>
<minute>
<!--ro, req, int, minute(s), range:[0,59]-->59
</minute>

```

```

<sec>
  <!--ro, req, float, second(s), range:[0,59.99999]-->59.000000
</sec>
</Longitude>
<Latitude>
  <!--ro, req, object, latitude information-->
<degree>
  <!--ro, req, int, degree(s)-->60
</degree>
<minute>
  <!--ro, req, int, minute(s), range:[0,59]-->59
</minute>
<sec>
  <!--ro, req, float, second(s), range:[0,59.99999], desc:the value is accurate to six decimal places-->59.000000
</sec>
</Latitude>
</VehicleGPSInfo>
<vehiclePositionControl>
  <!--ro, opt, enum, vehicle arming type, subType:string, desc:"vehicleMonitor" (intelligent arming of vehicle, related URI: PUT /ISAPI/Traffic/channels/<ID>/vehicleMonitor/<taskID>/startTask), "manualVehicleMonitor" (manual arming of vehicle, related URI: PUT /ISAPI/Traffic/channels/<ID>/manualVehicleMonitor), "dailyVehicleMonitor" (daily arming of vehicle)-->dailyVehicleMonitor
</vehiclePositionControl>
<vehicleMonitorTaskID>
  <!--ro, opt, string, task ID of intelligent arming of vehicle, range:[1,64], desc:the task ID is sent to the device by the upper layer, which should guarantee the uniqueness of ID. It's valid when the value of vehiclePositionControl is "vehicleMonitor"-->test
</vehicleMonitorTaskID>
<vehicleListName>
  <!--ro, opt, string, name of the list that the vehicle belongs to, range:[0,128], desc:name of the list that the vehicle belongs to, the maximum size is 128 bytes-->test
</vehicleListName>
<vehicleThermometryEnabled>
  <!--ro, opt, bool, whether to enable vehicle temperature measurement-->true
</vehicleThermometryEnabled>
<currTemperature>
  <!--ro, opt, float, temperature-->36.5
</currTemperature>
<thermometryUnit>
  <!--ro, opt, enum, temperature unit, subType:string, desc:"celsius", "fahrenheit", "kelvin"-->celsius
</thermometryUnit>
<plateCategory>
  <!--ro, opt, string, license plate additional information, range:[0,8], desc:license plate additional information-->test
</plateCategory>
<plateSize>
  <!--ro, opt, enum, license plate size, subType:int, desc:license plate size-->1
</plateSize>
<isNeedVerification>
  <!--ro, opt, bool, whether it requires verification, desc:false-do not verify-->true
</isNeedVerification>
<ISO-CR>
  <!--ro, opt, string, ISO 3166-1 standard country/region code, desc:refer to protocol dictionary-->CN
</ISO-CR>
<vehicleRectName>
  <!--ro, opt, string, country/region where the vehicle triggering events belongs, range:[0,64]-->11
</vehicleRectName>
<roadType>
  <!--ro, opt, enum, road type, subType:string, desc:"entrance" (entrance and exit), "city", "custom", "alarmInput" (alarm input triggered by coil), "post" (checkpoint), "publicSecurity" (general video security mode, in which the scene with mixed vehicles and without lane lines can be detected)-->entrance
</roadType>
<Custom>
  <!--ro, opt, object, custom parameters, dep:and,{$.EventNotificationAlert.ANPR.roadType,eq,custom}-->
</direction>
  <!--ro, opt, enum, target direction, subType:string, desc:"enter", "parkind", "Leave"-->enter
</direction>
</Custom>
<accelerationNoise>
  <!--ro, opt, object, vehicle revving behavior that causes huge noise-->
<startTime>
  <!--ro, req, datetime, start time of the behavior-->1970-01-01T00:00:00+08:00
</startTime>
<endTime>
  <!--ro, req, datetime, end time of the behavior-->1970-01-01T00:00:00+08:00
</endTime>
<noiseDecibel>
  <!--ro, req, float, noise decibel-->36.5
</noiseDecibel>
</accelerationNoise>
<ADRNo>
  <!--ro, opt, string, hazardous material ID, range:[1,8]-->231965
</ADRNo>
<pilotmask>
  <!--ro, opt, enum, subType:string-->unknown
</pilotmask>
<vicepilotMask>
  <!--ro, opt, enum, subType:string-->unknown
</vicepilotMask>
<nonMotorMask>
  <!--ro, opt, enum, subType:string-->unknown
</nonMotorMask>
<pedestrianMask>
  <!--ro, opt, enum, subType:string-->unknown
</pedestrianMask>
<vehicleTemperature>
  <!--ro, opt, float, temperature-->36.5
</vehicleTemperature>

```

```

<!--ro, opt, object-->
<vehicleHeadTemperature>
  <!--ro, opt, float, range:[-20.0,550.0]-->50.2
</vehicleHeadTemperature>
<vehicleBodyTemperature>
  <!--ro, opt, float, range:[-20.0,550.0]-->35.2
</vehicleBodyTemperature>
<leftWheelHubList>
  <!--ro, opt, array, subType:object, range:[0,6]-->
  <leftWheelHub>
    <!--ro, opt, object-->
    <ID>
      <!--ro, opt, int, range:[1,6]-->1
    </ID>
    <wheelHubTemperature>
      <!--ro, opt, float, range:[-20.0,550.0]-->38.2
    </wheelHubTemperature>
  </leftWheelHub>
</leftWheelHubList>
<rightWheelHubList>
  <!--ro, opt, array, subType:object, range:[0,6]-->
  <rightWheelHub>
    <!--ro, opt, object-->
    <ID>
      <!--ro, opt, int, range:[1,6]-->1
    </ID>
    <wheelHubTemperature>
      <!--ro, opt, float, range:[-20.0,550.0]-->38.2
    </wheelHubTemperature>
  </rightWheelHub>
</rightWheelHubList>
</vehicleTemperature>
<deliveryVan>
  <!--ro, opt, enum, subType:string-->unknown
</deliveryVan>
<nonMotorShedUmbrella>
  <!--ro, opt, enum, subType:string-->unknown
</nonMotorShedUmbrella>
<nonMotorManned>
  <!--ro, opt, enum, subType:string-->unknown
</nonMotorManned>
</ANPR>
<UUID>
  <!--ro, opt, string, unique common ID, which is used to link the same capture across multiple servers, desc:if the vehicle is captured many times, the UUID is the same-->test
</UUID>
<picNum>
  <!--ro, opt, int, number of pictures-->2
</picNum>
<monitoringSiteID>
  <!--ro, opt, string, camera No., desc:corresponds to <positionNum> from /ISAPI/System/Video/inputs/channels/<channelID>/cameraInfo (recommended for new devices) and /ISAPI/Traffic/channels/<channelID>/cameraInfo applied by some traffic monitoring device-->test
</monitoringSiteID>
<ePlateUUID>
  <!--ro, opt, string, electronic license plate ID, desc:if this field has a value, it indicates that an electronic license plate is linked-->test
</ePlateUUID>
<isDataRetransmission>
  <!--ro, opt, bool, whether the data is replenished-->true
</isDataRetransmission>
<SceneInfo>
  <!--ro, opt, object, scene information-->
  <scenesID>
    <!--ro, opt, string, scene ID, which is between 1 and 16-->test
  </scenesID>
  <sceneName>
    <!--ro, opt, string, scene name, range:[0,128]-->test
  </sceneName>
<PTZPos>
  <!--ro, opt, object, PTZ information-->
  <elevation>
    <!--ro, opt, int, tilting angle, range:[-900,2700]-->0
  </elevation>
  <azimuth>
    <!--ro, opt, int, panning angle, range:[0,3600]-->0
  </azimuth>
  <absoluteZoom>
    <!--ro, opt, int, zooming ratio, range:[0,1000]-->1
  </absoluteZoom>
</PTZPos>
</SceneInfo>
<monitorDescription>
  <!--ro, opt, string, camera information, range:[0,256], desc:corresponds to <positionInfo> from /ISAPI/System/Video/inputs/channels/<ID>/cameraInfo (recommended for new devices) and /ISAPI/Traffic/channels/<channelID>/cameraInfo applied by some traffic monitoring device-->test
</monitorDescription>
<DeviceGPSInfo>
  <!--ro, opt, object, device GPS information-->
  <longitudeType>
    <!--ro, req, enum, Longitude, subType:string, desc:"E" (Eastern Hemisphere), W" (West Hemisphere)-->E
  </longitudeType>
  <latitudeType>
    <!--ro, req, enum, Latitude, subType:string, desc:"S" (Southern Hemisphere), "N" (North Hemisphere)-->S
  </latitudeType>
  <Longitude>
    <!--ro, req, object, longitude information-->

```

```

<degree>
  <!--ro, req, int, degree(s)-->60
</degree>
<minute>
  <!--ro, req, int, minute(s), range:[0,59]-->59
</minute>
<sec>
  <!--ro, req, float, second(s), range:[0,59.999999]-->59.000000
</sec>
</Longitude>
<Latitude>
  <!--ro, req, object, latitude information-->
<degree>
  <!--ro, req, int, degree(s)-->60
</degree>
<minute>
  <!--ro, req, int, minute(s), range:[0,59]-->59
</minute>
<sec>
  <!--ro, req, float, second(s), range:[0,59.999999], desc:the value is accurate to six decimal places-->59.000000
</sec>
</Latitude>
</DeviceGPSInfo>
<pilotStandardSafebelt>
  <!--ro, opt, enum, whether the driver wears the seat belt properly, subType:string, desc:"unknown", "yes", "no"-->yes
</pilotStandardSafebelt>
<vicepilotStandardSafebelt>
  <!--ro, opt, enum, whether the front passenger wears the seat belt properly, subType:string, desc:"unknown", "yes", "no"-->yes
</vicepilotStandardSafebelt>
<trafficLightSnap>
  <!--ro, opt, enum, whether it is captured at the traffic lights, subType:string, desc:"yes", "no"-->yes
</trafficLightSnap>
<sequence>
  <!--ro, opt, int, No. of vehicle capture from network-triggered burst, range:[1,4294967295], desc:command applied by network-triggered burst control;
related API: /ISAPI/Traffic/startRecognition; SDK interface: NET_DVR_ContinuousShoot-->0
</sequence>
<relLaneDirectionType>
  <!--ro, opt, enum, Linked Lane direction, subType:int, desc:a camera can be linked with multiple Lane directions, which is independent of the camera
setup angle <detectDir>; 0 (others),1 (from east to west), 2 (from west to east), 3 (from south to north), 4 (from north to south), 5 (from southeast to
northwest), 6 (from northwest to southeast), 7 (from northeast to southwest), 8 (from southwest to northeast)-->1
</relLaneDirectionType>
<carDirectionType>
  <!--ro, opt, enum, vehicle running direction on the Lane, subType:int, desc:independent of the camera setup angle <detectDir>; 0-from up to down, 1-from
down to up-->1
</carDirectionType>
<targetID>
  <!--ro, opt, string, target ID, range:[1,64], desc:UUID, of which the uniqueness is guaranteed by device, and corresponds to <dMatchNo> in the private
protocol-->test
</targetID>
<isSecondCamera>
  <!--ro, opt, bool, whether to enable capture by the second camera, desc:the second camera: e.g., the perspective camera from distant/close-view capture
cameras, the back camera from front/back capture camera, etc.-->false
</isSecondCamera>
<dataAnalysisType>
  <!--ro, opt, enum, data analysis type, subType:int, desc:0-data not analyzed, 1-data analyzed-->1
</dataAnalysisType>
<RecordInfo>
  <!--ro, opt, object, vehicle video information-->
<fileName>
  <!--ro, req, enum, video name, subType:string, desc:the video name should match <Content-ID> for binary transmission: "record.mp4" (vehicle video)-->record.mp4
</fileName>
<dataType>
  <!--ro, req, enum, data type, subType:int, desc:0 (binary), 1 (URL)-->0
</dataType>
<URL>
  <!--ro, opt, string, URL, desc:it is valid when the value of dataType is "URL"-->test
</URL>
</RecordInfo>
<VehicleWeightInfo>
  <!--ro, opt, object, vehicle weighing information, desc:this node is currently only supported by weighing data management server-->
<isOverWeight>
  <!--ro, opt, bool, whether it is overweight, desc:set overweight reporting or overrun reporting as priority via <illegalPriorityType> from
/ISAPI/Traffic/channels/<channelID>/vehicleWeight/capabilities?format=json-->false
</isOverWeight>
<axleNum>
  <!--ro, opt, int, number of axles, range:[1,10]-->4
</axleNum>
<axleModel>
  <!--ro, opt, enum, vehicle type by axle, subType:int, desc:0 (unknown), 512 (2-axle truck), 756 (3-axle centre-axle trailer combination), 766 (3-axle
articulated vehicle), 767 (3-axle truck), 768 (3-axle truck), 1024 (type-125 4-axle centre-axle trailer combination), 1025 (type-152 4-axle centre-axle
trailer combination), 1026 (4-axle articulated vehicle), 1027 (4-axle full trailer combination), 1028 (4-axle truck), 1280 (type-155 5-axle centre-axle
trailer combination), 1281 (type-1125 5-axle centre-axle trailer combination), 1282 (type-155 5-axle articulated vehicle), 1283 (type-1125 5-axle
articulated vehicle), 1284 (type-129 5-axle articulated vehicle), 1285 (type-1522 5-axle full trailer combination), 1286 (type-11222 5-axle full trailer
combination), 1536 (type-159 6-axle centre-axle trailer combination), 1537 (type-159-2 6-axle centre-axle trailer combination), 1538 (type-1155-1 6-axle
centre-axle trailer combination), 1539 (type-1155-2 6-axle centre-axle trailer combination), 1540 (type-159-3 6-axle articulated vehicle), 1541 (type-159-4
6-axle articulated vehicle), 1542 (type-1129 6-axle articulated vehicle), 1543 (type-11522-1 6-axle full trailer combination), 1544 (type-11522-2 6-axle
full trailer combination)-->0
</axleModel>
<overWeight>
  <!--ro, opt, float, over-limit weight (unit: ton), range:[0.000,100.000], unit:吨, desc:accurate to three decimal places-->4.502
</overWeight>
<weight>

```

```

<!--ro, opt, float, vehicle weight (unit: ton), range:[0.000,100.000], unit:kg, desc:accurate to three decimal places-->4.502
</weight>
</limitWeight>
<!--ro, opt, float, weight limit (unit: ton), range:[0.000,100.000], unit:kg, desc:accurate to three decimal places-->4.502
</limitWeight>
<axleLen>
<!--ro, opt, float, distance between tyres, range:[0.00,100.00], unit:mm, desc:accurate to two decimal places-->4.502
</axleLen>
<devDescInfo>
<!--ro, opt, string, device description, range:[0,64]-->test
</devDescInfo>
<AxeInfoList>
<!--ro, opt, object, axle information list, desc:<axleNum>, number of axles-->
<AxeInfo>
<!--ro, opt, object, information of a single axle-->
<axleWeight>
<!--ro, opt, float, axle weight, range:[0.00,10000.00], unit:kg-->100.00
</axleWeight>
<axleDistance>
<!--ro, opt, int, distance between axles, range:[0,100000], unit:mm, desc:the distance between the current axle and the next axle-->30000
</axleDistance>
<wheelNum>
<!--ro, opt, int, range:[0,8]-->2
</wheelNum>
</AxeInfo>
</AxeInfoList>
<length>
<!--ro, opt, int, vehicle length, range:[1,1000000], unit:cm-->4000
</length>
<width>
<!--ro, opt, int, vehicle width, range:[1,1000000], unit:cm-->4000
</width>
<height>
<!--ro, opt, int, vehicle height, range:[1,1000000], unit:cm-->4000
</height>
<tollwayVehicleType>
<!--ro, opt, enum, vehicle types on tollway, subType:int, desc:0 (unknown), 1 (type-1 minibus), 2 (type-1 small-sized bus), 3 (type-2 medium-sized bus), 4 (type-2 bus and passenger trailer combination), 5 (type-3 large-sized bus), 6 (type-4 large-sized bus), 7 (type-1 2-axle truck), 8 (type-2 2-axle truck), 9 (type-3 3-axle truck), 10 (type-4 4-axle truck), 11 (type-5 5-axle truck), 12 (type-6 6-axle truck), 13 (type-1 2-axle operation vehicle), 14 (type-2 2-axle operation vehicle), 15 (type-3 3-axle operation vehicle), 16 (type-4 4-axle operation vehicle), 17 (type-5 5-axle operation vehicle), 18 (type-6 operation vehicle with 6 or more axles)-->1
</tollwayVehicleType>
<tiresNum>
<!--ro, opt, int, number of tyres, range:[1,20]-->4
</tiresNum>
<approvedPassengers>
<!--ro, opt, int, authorized passenger capacity, range:[1,100]-->7
</approvedPassengers>
<limitLength>
<!--ro, opt, int, vehicle length limit, range:[1,1000000], unit:cm, desc:configured via /ISAPI/Traffic/channels/<channelID>/vehicleWeight/capabilities?format=json-->4000
</limitLength>
<limitWidth>
<!--ro, opt, int, vehicle width limit, range:[1,1000000], unit:cm, desc:configured via /ISAPI/Traffic/channels/<channelID>/vehicleWeight/capabilities?format=json-->4000
</limitWidth>
<limitHeight>
<!--ro, opt, int, vehicle height limit, range:[1,1000000], unit:cm, desc:configured via /ISAPI/Traffic/channels/<channelID>/vehicleWeight/capabilities?format=json-->4000
</limitHeight>
<overLength>
<!--ro, opt, int, over-limit length, range:[1,1000000], unit:cm, desc:the over-limit part equals <length> minus <limitLength>-->4000
</overLength>
<overWidth>
<!--ro, opt, int, over-limit width, range:[1,1000000], unit:cm, desc:the over-limit part equals <width> minus <limitWidth>-->4000
</overWidth>
<overHeight>
<!--ro, opt, int, over-limit height, range:[1,1000000], unit:cm, desc:the over-limit part equals <height> minus <limitHeight>-->4000
</overHeight>
<isOverLimit>
<!--ro, opt, bool, whether it is over-limit, desc:set overweight reporting or overrun reporting as priority via <illegalPriorityType> from /ISAPI/Traffic/channels/<channelID>/vehicleWeight/capabilities?format=json-->false
</isOverLimit>
<axleNumConfidence>
<!--ro, opt, int, confidence of captured number of axles, range:[0,100]-->0
</axleNumConfidence>
<standardAxleTypeInfo>
<!--ro, opt, object-->
<axleType>
<!--ro, req, enum, subType:int-->12
</axleType>
<axleDescribeInfo>
<!--ro, req, string, range:[0,64]-->2轴载货汽车
</axleDescribeInfo>
</standardAxleTypeInfo>
<VehicleWeightInfo>
<isNotSlowZebraCrossing>
<!--ro, opt, bool, whether the vehicle did not slow down at zebra crossing-->false
</isNotSlowZebraCrossing>
<isTurnRightStop>
<!--ro, opt, bool, whether the vehicle did not stop at a right turn-->false
</isTurnRightStop>
<PlateInfoList>
<!--ro, opt, object, this node is returned when license plates from Mainland, Honakona, and Maokao are all supported-->

```

```

.. ., opt, object, desc field is returned when license plates , e.g. handwritten, hangzhou, and license are not supported .
<PlateInfo>
  <!--ro, opt, object, information of a single license plate-->
  <plateRect>
    <!--ro, opt, object, license plate area coordinates, desc:this field is valid when the value of type is "detectionPicture". The normalized value is
the current image size in percentage multiplying 1000 and it is accurate to three decimal places. The origin is the upper-left corner of screen-->
    <x>
      <!--ro, req, int, X-coordinate of the upper-left corner of the boundary frame, range:[0,1000]-->1000
    </x>
    <y>
      <!--ro, req, int, Y-coordinate of the upper-left corner of the boundary frame, range:[0,1000]-->1000
    </y>
    <width>
      <!--ro, req, int, width of the boundary frame, range:[0,1000]-->1000
    </width>
    <height>
      <!--ro, req, int, height of the boundary frame, range:[0,1000]-->1000
    </height>
  </plateRect>
  <plateColor>
    <!--ro, req, enum, license plate color, subType:string, desc:License plate color: "white", "yellow", "blue", "black", "green", "newEnergyGreen"-new
energy green, "newEnergyYellowGreen"-new energy flavogreen, "other"-other color-->black
  </plateColor>
  <licensePlate>
    <!--ro, req, string, license plate number, range:[1,32], desc:license plate number,e.g., "123456"-->A283KY77
  </licensePlate>
  <confidenceLevel>
    <!--ro, req, int, confidence level, range:[0,100]-->50
  </confidenceLevel>
  <province>
    <!--ro, opt, enum, province index, subType:int, desc:N/A-->0
  </province>
  <isExtraPlate>
    <!--ro, opt, bool-->true
  </isExtraPlate>
</PlateInfo>
</PlateInfoList>
<deviceUUID>
  <!--ro, opt, string, device No., range:[1,32], desc:the device number is the device UUID which is the device's serial number and can be edited by the
node <deviceID> in the message of /ISAPI/System/deviceInfo-->12345
</deviceUUID>
<VehicleGATInfo>
  <!--ro, opt, object, N/A, desc:N/A-->
  <palteTypeByGAT>
    <!--ro, opt, enum, N/A, subType:int, desc:N/A-->1
  </palteTypeByGAT>
  <plateColorByGAT>
    <!--ro, opt, enum, N/A, subType:int, desc:0-white, 1-yellow, 2-blue, 3-black, 4-green, 5-not recognized, 9, other colors-->0
  </plateColorByGAT>
  <vehicleTypeByGAT>
    <!--ro, opt, enum, N/A, subType:string, desc:GA/T 16.4 complied: "K10" (large-sized bus), "K20" (medium-sized bus), "K30" (small-sized bus), "K33"
(medium-sized car), "H10" (heavy truck), "H20" (medium-sized truck), "H30" (light truck), "M10" (three-wheel motorcycle), "M20" (two-wheel motorcycle), "X99"
(others)-->K10
  </vehicleTypeByGAT>
  <colorByGAT>
    <!--ro, opt, enum, N/A, subType:string, desc:GA/T 16.8 complied: A-white, B-gray, C-yellow, D-pink, E-red, F-purple, G-green, H-blue, I-brown, J-
black, K-not recognized, Z-other colors-->A
  </colorByGAT>
</VehicleGATInfo>
<EPlateInfo>
  <!--ro, opt, object, RFID detection result-->
  <EPlateChipNo>
    <!--ro, opt, string, chip No. of electronic license plate (detected by RFID), range:[0,16]-->E8550000011F9AAAS
  </EPlateChipNo>
  <EPlateCardNo>
    <!--ro, opt, string, electronic license plate No. (detected by RFID), range:[0,12], desc:electronic license plate No. (12 digits; the first two
represent the province code-->E8550000011F9AAAS
  </EPlateCardNo>
  <EPlateNo>
    <!--ro, opt, string, license plate No. (detected by RFID), range:[0,32], desc:"noPlate" (vehicle without license plate detected), "unknown" (license
plate not detected; temporary license plate detected)-->浙A12345
  </EPlateNo>
  <EPlateManufacturingDate>
    <!--ro, opt, string, production date (detected by RFID), range:[0,7], desc:format: yyyy-MM-->2014-01
  </EPlateManufacturingDate>
  <EPlateInspectionValidity>
    <!--ro, opt, string, inspection validity (detected by RFID), range:[0,7], desc:accurate to month; format: yyyy-MM-->2014-01
  </EPlateInspectionValidity>
  <EPlateMandaScrapPeriod>
    <!--ro, opt, int, compulsory scrapping period (detected by RFID), range:[0,100], unit:年, desc:from the start of inspection validity to the compulsory
scrapping time (unit: year)-->10
  </EPlateMandaScrapPeriod>
  <EPlatePassengersCapacity>
    <!--ro, opt, int, authorized passenger capacity (detected by RFID), range:[0,100], desc:e.g. 1-->1
  </EPlatePassengersCapacity>
  <EPlateCarryingCapacity>
    <!--ro, opt, float, tractive tonnage, range:[0,100], unit:t/t, desc:tractive tonnage (unit: ton)-->2.0
  </EPlateCarryingCapacity>
  <EPlateVehicleColor>
    <!--ro, opt, enum, vehicle color (detected by RFID), subType:string, desc:"unknown", "white", "silver", "gray", "black", "red", "deepBlue", "blue",
"yellow", "green", "brown", "pink", "purple", "deepGray", "cyan", "orange"-->black
  </EPlateVehicleColor>
  <EPlateNonMotorType>
    <!--ro, opt, enum, non-motor vehicle type (detected by RFID), subType:string, desc:"unknown", "threeWheelVehicle" (tricycle), "twoWheelVehicle" (two
wheel vehicle)-->black
  </EPlateNonMotorType>

```

```

wheeler)-->threeWheelVehicle
</EPlateNonMotorType>
<EPlateNonMotorPlateType>
  <!--ro, opt, enum, non-motor vehicle license plate type (detected by RFID), subType:string, desc:"unknown", "temporary", "normal"-->unknown
</EPlateNonMotorPlateType>
<EPlateColor>
  <!--ro, opt, enum, vehicle color (detected by RFID), subType:string, desc:"black", "blue", "golden", "orange", "red", "yellow", "white", "unknown", "other", "newEnergyYellowGreen"-->new energy yellow-green, "civilAviationBlack", "civilAviationGreen" (civil aviation green), "green", "mixedColor", "newEnergyGreen" (new energy green), "brown"-->black
</EPlateColor>
<EPlateUseType>
  <!--ro, opt, enum, non-motor vehicle purpose (detected by RFID), subType:string, desc:"unknown", "nonOperating" (non-operating), "takeOut" (take-out delivery), "express" (express delivery)-->nonOperating
</EPlateUseType>
<EPlateTriggerTime>
  <!--ro, req, datetime, device detection trigger time of electronic license plate (detected by RFID)-->1970-01-01T00:00:00+08:00
</EPlateTriggerTime>
<EPlateListenAuthCode>
  <!--ro, opt, string, licensing authority code of electronic license plate (detected by RFID)-->粤B
</EPlateListenAuthCode>
</EPlateInfo>
<accelerationNoise>
  <!--ro, opt, object, vehicle revving behavior that causes huge noise-->
<isExistsAccelerationNoise>
  <!--ro, req, bool, whether there is vehicle revving behavior that causes huge noise-->true
</isExistsAccelerationNoise>
<startTime>
  <!--ro, req, datetime, start time of the behavior, desc:accurate to millisecond-->1970-01-01T00:00:00.000+08:00
</startTime>
<endTime>
  <!--ro, req, datetime, end time of the behavior, desc:accurate to millisecond-->1970-01-01T00:00:00.000+08:00
</endTime>
</accelerationNoise>
<noiseDecibel>
  <!--ro, opt, float, noise decibel of vehicle running, range:[0,300], unit:分贝, desc:accurate to one decimal place-->65.5
</noiseDecibel>
<exhaustPipeNum>
  <!--ro, opt, int, number of exhaust pipes, range:[0,4)-->2
</exhaustPipeNum>
<exhaustPipePositionType>
  <!--ro, opt, enum, exhaust pipe installation location, subType:string, desc:"unilateral", "bilateral", "middle"-->unilateral
</exhaustPipePositionType>
<openGateType>
  <!--ro, opt, enum, reasons for opening the barrier gate, subType:string, dep:and,{$.EventNotificationAlert.ANPR.barrierGateCtrlType,eq,0}, desc:"white" (auto-open barrier gate via device according to the allowlist), "manual" (manually open barrier gate via platform), "abnormal" (auto-open barrier gate via device for exceptional circumstances)-->white
</openGateType>
<plateStandardStatus>
  <!--ro, opt, enum, whether the license plate is valid, subType:string, desc:"yes", "no"-->yes
</plateStandardStatus>
<limitSurfaceSpeed>
  <!--ro, opt, int, cross-sectional device-detected speed limit (required for overspeed and violation), range:[0,255], unit:km/h-->1
</limitSurfaceSpeed>
<startCaptureSpeed>
  <!--ro, opt, int, device-detected capture triggered speed (required for overspeed and violation), range:[0,255], unit:km/h-->1
</startCaptureSpeed>
<redlightIllegalCode>
  <!--ro, opt, string, traffic lights and lane matching code for red light running, range:[0,4], desc:@all lanes, 1-turn left, 2-go straight, 3-turn right, 4-go straight & turn left, 5-go straight & turn right-->1
</redlightIllegalCode>
<redlightSignalStartTime>
  <!--ro, opt, string, start time of the red light (required for red light running), desc:ISO 8601 format yyyy-MM-ddTHH:mm:ss+current zone, e.g., 2017-08-01T17:30:08+08:00-->2004-05-03T17:30:08+08:00
</redlightSignalStartTime>
<redlightSignalEndTime>
  <!--ro, opt, string, end time of the red light (required for red light running), desc:ISO 8601 format yyyy-MM-ddTHH:mm:ss+current zone, e.g., 2017-08-01T17:30:08+08:00-->2004-05-03T17:30:08+08:00
</redlightSignalEndTime>
<trafficControlTimeSpanList>
  <!--ro, opt, array, time period for traffic restriction, subType:object-->
<TimeRange>
  <!--ro, req, object, time period-->
<beginTime>
  <!--ro, req, time, start time-->10:00:00
</beginTime>
<endTime>
  <!--ro, req, time, end time-->10:00:00
</endTime>
</TimeRange>
</trafficControlTimeSpanList>
<detectionBackgroundImageResolution>
  <!--ro, opt, object, -ro, desc:corresponds to the picture represented by <pictureURL> and <pId> when <fileName> is "detectionPicture.jpg" and <type> is "detectionPicture" from <pictureInfoList> and <pictureInfo> in ANPR when the normalized coordinates of <Rect>, <pictureInfoList>, <pictureInfo>, <plateRect>, <vehicleRect>, <PlateInfoList>, <PlateInfo>, and <plateRect> are converted to the resolution coordinates, this node is required-->
<height>
  <!--ro, req, int, resolution (height), range:[1,65535], unit:px-->1
</height>
<width>
  <!--ro, req, int, resolution (width), range:[1,65535], unit:px-->1
</width>
</detectionBackgroundImageResolution>
<trafficSurveyVehicleType>
  <!--ro, opt, enum, subType:int-->1
</trafficSurveyVehicleType>

```

```
</array>
<!-->
```

<doorTextInfoList>

```
<!--ro, opt, array, subType:object, range:[0,2]-->
<doorTextInfo>
<!--ro, opt, object-->
<doorID>
<!--ro, opt, int, range:[0,2]-->1
</doorID>
<doorTextList>
<!--ro, opt, array, subType:object, range:[0,4]-->
<doorText>
<!--ro, opt, string, range:[0,128]-->载荷50人
</doorText>
</doorTextList>
<approvedPassengers>
<!--ro, opt, int, range:[1,100]-->7
</approvedPassengers>
<doorTextConfidence>
<!--ro, opt, int, range:[0,100]-->0
</doorTextConfidence>
</doorTextInfo>
</doorTextInfoList>
<splicingDetectInfo>
<!--ro, opt, object-->
<startSplicingTime>
<!--ro, req, datetime-->1970-01-01T00:00:00 000+08:00
</startSplicingTime>
<endSplicingTime>
<!--ro, req, datetime-->1970-01-01T00:00:00 000+08:00
</endSplicingTime>
</splicingDetectInfo>
</EventNotificationAlert>
```

Parameter Name	Parameter Value	Parameter Type(Content-Type)	Content-ID	File Name	Description
anpr.xml	[Message content]	application/xml	--	--	--
.value	[Binary picture data]	image/jpeg	detectionPicture	detectionPicture.jpg	--
.value	[Binary picture data]	image/jpeg	licensePlatePicture	licensePlatePicture.jpg	--
.value	[Binary picture data]	image/jpeg	pilotPicture	pilotPicture.jpg	--
.value	[Binary picture data]	image/jpeg	copilotPicture	copilotPicture.jpg	--
.value	[Binary picture data]	image/jpeg	compositePicture	compositePicture.jpg	--
.value	[Binary picture data]	image/jpeg	plateBinaryPicture	plateBinaryPicture.jpg	--
.value	[Binary picture data]	image/jpeg	nonMotorPicture	nonMotorPicture.jpg	--
.value	[Binary picture data]	image/jpeg	pedestrianDetectionPicture	pedestrianDetectionPicture.jpg	--
.value	[Binary picture data]	image/jpeg	vehiclePicture	vehiclePicture.jpg	--
.value	[Binary picture data]	image/jpeg	featurePicture	featurePicture.jpg	--
.value	[Binary picture data]	image/jpeg	panoramaPicture	,panoramaPicture.jpg	--
.value	[Binary picture data]	image/jpeg	extraLicensePlatePicture	extraLicensePlatePicture.jpg	--

**Note:** The protocol is transmitted in form format. See Chapter 4.5.1.4 for form framework description, as shown in the instance below.

```
--<frontier>
Content-Disposition: form-data; name=Parameter Name;filename=File Name
Content-Type: Parameter Type
Content-Length: ****
Content-ID: Content ID
Parameter Value
```

- Parameter Name: the name property of Content-Disposition in the header of form unit; it refers to the form unit name.
- Parameter Type (Content-Type): the Content-Type property in the header of form unit.
- File Name (filename): the filename property of Content-Disposition of form unit Headers. It exists only when the transmitted data of form unit is file, and it refers to the file name of form unit body.
- Parameter Value: the body content of form unit.

## 16.7 Abnormal Event Detection

### 16.7.1 Fast Moving Detection

#### 16.7.1.1 Get the arming schedule of fast moving detection for all channels

##### Request URL

GET /ISAPI/Event/schedules/rapidMoves

##### Query Parameter

None

##### Request Message

None

##### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<RapidMoveScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
</RapidMoveScheduleList>
```

#### 16.7.1.2 Set the arming schedule of fast moving detection for all channels

##### Request URL

PUT /ISAPI/Event/schedules/rapidMoves

##### Query Parameter

None

##### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<RapidMoveScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
</RapidMoveScheduleList>
```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

## 16.7.2 Loitering Detection

### 16.7.2.1 Set arming schedules of loitering detection of all channels

#### Request URL

PUT /ISAPI/Event/schedules/loiterings

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<LoiteringScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, arming schedule list, subType:object, attr:version{req, string, protocolVersion}-->
</LoiteringScheduleList>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code-->0
  </subStatusCode>
</ResponseStatus>

```

### 16.7.2.2 Get arming schedules of the loitering detection of all channels

#### Request URL

GET /ISAPI/Event/schedules/loiterings

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<LoiteringScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, arming schedule list, subType:object, attr:version{req, string, protocolVersion}-->
</LoiteringScheduleList>

```

### 16.7.2.3 Set loitering detection parameters of all channels

#### Request URL

PUT /ISAPI/Smart/loitering

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<LoiteringList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, subType:object, attr:version{req, string, protocolVersion}-->
  <Loitering>
    <!--opt, object, Loitering detection parameters configuration of a channel-->
    <id>
      <!--req, string, channel No.-->test
    </id>
    <enabled>
      <!--req, bool-->true
    </enabled>
    <normalizedScreenSize>
      <!--opt, object, normalized size-->
      <normalizedScreenWidth>
        <!--req, int, normalized width-->1
      </normalizedScreenWidth>
      <normalizedScreenHeight>
        <!--req, int, normalized height-->1
      </normalizedScreenHeight>
    </normalizedScreenSize>
    <LoiteringRegionList>
      <!--opt, array, information list of loitering detection areas, subType:object-->
      <LoiteringRegion>
        <!--opt, object, Loitering detection area-->
        <id>
          <!--req, string-->1
        </id>
        <sensitivityLevel>
          <!--opt, int, sensitivity, range:[0,100], desc:sensitivity-->1
        </sensitivityLevel>
        <timeThreshold>
          <!--opt, int, time threshold-->1
        </timeThreshold>
        <RegionCoordinatesList>
          <!--opt, array, area coordinate List, subType:object, range:[0,4], desc:rectangle-->
        </RegionCoordinatesList>
      </LoiteringRegion>
    </LoiteringRegionList>
    <isSupportMultiScene>
      <!--opt, bool, whether supports multiple scene detection-->true
    </isSupportMultiScene>
  </Loitering>
</LoiteringList>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code description-->OK
  </subStatusCode>
</ResponseStatus>

```

#### 16.7.2.4 Get loitering detection parameters of all channels

##### Request URL

GET /ISAPI/Smart/loitering

##### Query Parameter

None

##### Request Message

None

##### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<LoiteringList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, array, subType:object, attr:version{req, string, protocolVersion}-->
    <Loitering>
        <!--ro, opt, object, Loitering detection parameters configuration of a channel-->
        <id>
            <!--ro, req, string, channel No.-->test
        </id>
        <enabled>
            <!--ro, req, bool, whether to enable the function-->true
        </enabled>
        <normalizedScreenSize>
            <!--ro, opt, object, normalized size-->
            <normalizedScreenWidth>
                <!--ro, req, int, normalized width-->1
            </normalizedScreenWidth>
            <normalizedScreenHeight>
                <!--ro, req, int, normalized height-->1
            </normalizedScreenHeight>
        </normalizedScreenSize>
        <LoiteringRegionList>
            <!--ro, opt, array, information list of Loitering detection areas, subType:object-->
            <LoiteringRegion>
                <!--ro, opt, object, Loitering detection area-->
                <id>
                    <!--ro, req, string-->1
                </id>
                <sensitivityLevel>
                    <!--ro, opt, int, sensitivity, range:[0,100], desc:sensitivity-->1
                </sensitivityLevel>
                <timeThreshold>
                    <!--ro, opt, int, time threshold-->1
                </timeThreshold>
                <RegionCoordinatesList>
                    <!--ro, opt, array, area coordinate list, subType:object, range:[0,4], desc:rectangle-->
                </RegionCoordinatesList>
            </LoiteringRegion>
        </LoiteringRegionList>
        <isSupportMultiScene>
            <!--ro, opt, bool, whether supports multiple scene detection-->true
        </isSupportMultiScene>
    </Loitering>
</LoiteringList>
```

#### 16.7.3 Object Removal Detection

##### 16.7.3.1 Get the arming schedule of object removal detection for all channels

##### Request URL

GET /ISAPI/Event/schedules/attendedBaggages

##### Query Parameter

None

##### Request Message

None

##### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<AttendBaggageScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, array, subType:object, attr:version{req, string, protocolVersion}-->
</AttendBaggageScheduleList>
```

### 16.7.3.2 Set the arming schedule of object removal detection for all channels

#### Request URL

PUT /ISAPI/Event/schedules/attendedBaggages

#### Query Parameter

None

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<AttendBaggageScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--req, array, subType:object, attr:version{req, string, protocolVersion}-->
</AttendBaggageScheduleList>
```

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
    <requestURL>
        <!--ro, req, string-->null
    </requestURL>
    <statusCode>
        <!--ro, req, enum, subType:int-->0
    </statusCode>
    <statusString>
        <!--ro, req, enum, subType:string-->OK
    </statusString>
    <subStatusCode>
        <!--ro, req, string-->OK
    </subStatusCode>
</ResponseStatus>
```

### 16.7.4 People Gathering Detection

#### 16.7.4.1 Get the arming time parameters of people gathering for all channels

#### Request URL

GET /ISAPI/Event/schedules/groups

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<GroupScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, people gathering arming time configuration list, attr:version{req, string, protocolVersion}-->
</GroupScheduleList>
```

#### 16.7.4.2 Set the arming time parameters of people gathering for all channels

#### Request URL

PUT /ISAPI/Event/schedules/groups

#### Query Parameter

None

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<GroupScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, object, people gathering arming time configuration list, attr:version{req, string, protocolVersion}-->
</GroupScheduleList>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

## 16.7.5 Student Standing Up

### 16.7.5.1 Get the arming schedule of student standing detection for all channels

#### Request URL

GET /ISAPI/Event/schedules/studentsStoodUp

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<StudentsStoodUpScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, subType:object, attr:version{req, string, protocolVersion}-->
</StudentsStoodUpScheduleList>
```

### 16.7.5.2 Set the arming schedule of student standing detection for all channels

#### Request URL

PUT /ISAPI/Event/schedules/studentsStoodUp

#### Query Parameter

None

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<StudentsStoodUpScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, subType:object, attr:version{req, string, protocolVersion}-->
</StudentsStoodUpScheduleList>
```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

## 16.7.6 Unattended Baggage Detection

### 16.7.6.1 Set the arming schedule of unattended baggage detection for all channels

#### Request URL

PUT /ISAPI/Event/schedules/unattendedBaggages

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<UnattendBaggageScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, subType:object, attr:version{req, string, protocolVersion}-->
</UnattendBaggageScheduleList>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.7.6.2 Get the arming schedule of unattended baggage detection for all channels

#### Request URL

GET /ISAPI/Event/schedules/unattendedBaggages

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<UnattendBaggageScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, subType:object, attr:version{req, string, protocolVersion}-->
</UnattendBaggageScheduleList>

```

# 16.8 Face Capture

## 16.8.1 Face Capture Settings

### 16.8.1.1 Get the arming schedule of face capture

#### Request URL

GET /ISAPI/Event/schedules/faceSnap

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<FaceSnapScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, array, arming schedule of face capture for all channels, subType:object, attr:version{req, string, protocolVersion}-->
    <Schedule>
        <!--ro, opt, object, arming schedule of face capture for a specified channel-->
        <id>
            <!--ro, req, string, channel No.-->test
        </id>
        <eventTypes>
            <!--ro, req, enum, event type, subType:string, desc:"ADAS", "ADASAlarm", "AID", "ANPR", "AccessControllerEvent", "CDSStatus", "DBD", "GPSUpload", "HFP", "IO", "IOTD", "LES", "LFPD", "PALMismatch", "PIR", "PeopleCounting", "PeopleNumChange", "Standup", "TMA", "TMPAM", "VMD", "abnormalAcceleration", "abnormalDriving", "advReachHeight", "alarmResult", "attendance", "attendedBaggage", "audioAbnormal", "audioexception", "behaviorResult", "blindSpotDetection", "cardMatch", "changedStatus", "collision", "containerDetection", "crowdSituationAnalysis", "databaseException", "defocus", "diskInformat", "diskerror", "diskfull", "driverConditionMonitor", "emergencyAlarm", "faceCapture", "facesnapModeling", "facedetection", "failDown", "faultAlarm", "fieldDetection", "fireDetection", "fireEscapeDetection", "flowOverrun", "framesPeopleCounting", "getUp", "group", "hdBadBlock", "hdImpact", "heatmap", "highHDTemperature", "highTempAlarm", "hotSpare", "illaccess", "ipctTransferAbnormal", "ipConflict", "keyPersonGetUp", "LeavePosition", "linedetection", "listSyncException", "loitering", "LowHDTemperature", "mixedTargetDetection", "modelError", "nicbroken", "nodeOffline", "nonPoliceIntrusio", "overSpeed", "overtimeTarry", "parking", "peopleNumChange", "peopleNumCounting", "personAbnormalAlarm", "personDensityDetection", "personQueueCounting", "personQueueDetection", "personQueueRealTime", "personQueueTime", "playCellPhone", "pocException", "poe", "policeAbsent", "radarAlarm", "radarFieldDetection", "radarLineDetection", "radarPerimeterRule", "radarTargetDetection", "radarVideoDetection", "raidException", "rapidMove", "reachHeight", "recordCycleAbnormal", "recordException", "regionEntrance", "regionExiting", "retention", "rollover", "running", "safetyHelmetDetection", "scenecangedetection", "sensorAlarm", "severeHDFailure", "shelteralarm", "shipsDetection", "sitQuietly", "smokeAndFireDetection", "smokeDetection", "softIO", "spacingChange", "sysStorFull", "takingElevatorDetection", "targetCapture", "temperature", "thermometry", "thirdPartyException", "toiletTarry", "tolCodeInfo", "tossing", "unattendedBaggage", "vehicleMatchResult", "vehicleRcogResult", "versionAbnormal", "videoException", "videoloss", "violationAlarm", "violentMotion", "yardTarry", "faceSnap"-->ADAS
        </eventTypes>
        <videoInputChannelID>
            <!--ro, opt, string, video input channel No.-->test
        </videoInputChannelID>
        <TimeBlockList size="8">
            <!--ro, req, array, arming time list, subType:object, attr:size{req, int}-->
            <TimeBlock>
                <!--ro, opt, object, arming time-->
                <dayOfWeek>
                    <!--ro, opt, enum, day of a week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->1
                </dayOfWeek>
                <TimeRange>
                    <!--ro, req, object, time range-->
                    <beginTime>
                        <!--ro, req, time, start time-->00:00:00+08:00
                    </beginTime>
                    <endTime>
                        <!--ro, req, time, end time-->00:00:00+08:00
                    </endTime>
                </TimeRange>
            </TimeBlock>
        </TimeBlockList>
    </Schedule>
</FaceSnapScheduleList>
```

### 16.8.1.2 Set the arming schedule of face capture

#### Request URL

PUT /ISAPI/Event/schedules/faceSnap

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FaceSnapScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--wo, req, array, arming schedule of face capture for all channels, subType:object, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--opt, object, arming schedule of face capture for a specified channel-->
    <id>
      <!--wo, req, string, channel No.-->test
    </id>
    <eventType>
      <!--wo, req, enum, event type, subType:string, desc:"ADAS", "ADASAlarm", "AID", "ANPR", "AccessControllerEvent", "CDsStatus", "DBD", "GPSUpload", "HFP", "IO", "IOTD", "LES", "LFPDI", "PALMiMatch", "PIR", "PeopleCounting", "PeopleNumChange", "Standup", "TMA", "TMAM", "MP", "abnormalAcceleration", "abnormalDriving", "advReachHeight", "alarmResult", "attendance", "attendedBaggage", "audioAbnormal", "audioexception", "behaviorResult", "blindSpotDetection", "cardMatch", "changedStatus", "collision", "containerDetection", "crowdSituationAnalysis", "databaseException", "defocus", "diskUnformat", "diskerror", "diskfull", "driverConditionMonitor", "emergencyAlarm", "faceCapture", "faceSnapModeling", "facedetection", "failDown", "faultAlarm", "fielddetection", "fireDetection", "fireEscapeDetection", "flowOverrun", "framesPeopleCounting", "getUp", "group", "hdBadBlock", "hdImpact", "heatmap", "highHDTemperature", "highTempAlarm", "hotSpare", "illAccess", "ipcTransferAbnormal", "ipConflict", "reyPersonGetUp", "LeavePosition", "linedetection", "listSyncException", "loitering", "lowHDTemperature", "mixedTargetDetection", "modelError", "nicbroken", "nodeOffline", "nonPoliceIntrusio", "overSpeed", "overtimeTarry", "parking", "peopleNumChange", "peopleNumCounting", "personAbnormalAlarm", "personDensityDetection", "personQueueCounting", "personQueueDetection", "personQueueRealTime", "personQueueTime", "playCellPhone", "pocException", "poe", "policeAbsent", "radarAlarm", "radarFieldDetection", "radarLineDetection", "radarPerimeterRule", "radarTargetDetection", "radarVideoDetection", "raidException", "rapidMove", "reachHeight", "recordCycleAbnormal", "recordException", "regionEntrance", "regionExiting", "retention", "rollOver", "running", "safetyHelmetDetection", "scenecangedetection", "sensorAlarm", "severeHDFailure", "shelterAlarm", "shipsDetection", "sitQuietly", "smokeAndFireDetection", "smokeDetection", "softIO", "spacingChange", "sysStorFull", "takingElevatorDetection", "targetCapture", "temperature", "thermometry", "thirdPartyException", "toiletTarry", "tollCodeInfo", "tossing", "unattendedBaggage", "vehicleMatchResult", "vehicleRocgResult", "versionAbnormal", "videoException", "videoloss", "violationAlam", "violentMotion", "yardTarry", "faceSnap"-->faceSnap
    </eventType>
    <videoInputChannelID>
      <!--wo, opt, string, video input channel No.-->test
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--wo, req, array, arming time list, subType:object, attr:size{req, int}-->
      <TimeBlock>
        <!--wo, opt, object, arming slots-->
        <dayOfWeek>
          <!--opt, enum, days of the week, subType:int, desc:days of the week-->1
        </dayOfWeek>
        <TimeRange>
          <!--wo, req, object, time range-->
          <beginTime>
            <!--wo, req, time, start time-->00:00:00+08:00
          </beginTime>
          <endTime>
            <!--wo, req, time, end time-->00:00:00+08:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>
</FaceSnapScheduleList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.8.1.3 Get the arming schedule of face capture for a specified channel

#### Request URL

GET /ISAPI/Event/schedules/faceSnap/faceSnap-<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, arming schedule of face capture for a specified channel, attr:version{req, string, protocolVersion}-->
    <id>
        <!--ro, req, string, channel No.-->test
    </id>
    <eventType>
        <!--ro, req, enum, event type, subType:string, desc:"ADAS", "ADASAlarm", "AID", "ANPR", "AccessControllerEvent", "CDsStatus", "DBD", "GPSUpload", "HFP", "IO", "IOTD", "LES", "LFPD", "PALMiMatch", "PIR", "PeopleCounting", "PeopleNumChange", "Standup", "TMA", "TMPAM", "VMD", "abnormalAcceleration", "abnormalDriving", "advReachHeight", "alarmResult", "attendance", "attendedBaggage", "audioAbnormal", "audioexception", "behaviorResult", "blindSpotDetection", "cardMatch", "changedStatus", "collision", "containerDetection", "crowdSituationAnalysis", "databaseException", "defocus", "diskUnformat", "diskrror", "diskfull", "driverConditionMonitor", "emergencyAlarm", "faceCapture", "faceSnapModeling", "facedetection", "failDown", "faultAlarm", "fielddetection", "fireDetection", "fireEscapeDetection", "flowOverrun", "framesPeopleCounting", "getUp", "group", "hdBadBlock", "hdImpact", "heatmap", "highHDTemperature", "highTempAlarm", "hotSpare", "illAccess", "ipcTransferAbnormo", "ipConflict", "reyPersonGetUp", "LeavePosition", "linedetection", "listSyncException", "loitering", "LowHDTemperature", "mixedTargetDetection", "ModelError", "nicbroken", "nodeOffline", "nonPoliceIntrusio", "overSpeed", "overtimeTarry", "parking", "peopleNumChange", "peopleNumCounting", "personAbnormalAlarm", "personDensityDetection", "personQueueCounting", "personQueueDetection", "personQueueRealTime", "personQueueTime", "playCellPhone", "pocException", "poe", "policeAbsent", "radarAlarm", "radarFieldDetection", "radarLineDetection", "radarPerimeterRule", "radarTargetDetection", "radarVideoDetection", "raidException", "rapidMove", "reachHeight", "recordCycleAbnormal", "recordException", "regionEntrance", "regionExiting", "retention", "rollOver", "running", "safetyHelmetDetection", "scenedchangedetection", "sensorAlarm", "severeHDFailure", "shelteralarm", "shipsDetection", "sitquietly", "smokeAndFireDetection", "smokeDetection", "softIO", "spacingChange", "sysStorFull", "takingElevatorDetection", "targetCapture", "temperature", "thermometry", "thirdPartyException", "toiletTarry", "tollCodeInfo", "tossing", "unattendedBaggage", "vehicleMatchResult", "vehicleRocgResult", "versionAbnormal", "videoException", "videoloss", "violationAlarm", "violentMotion", "yardTarry", "faceSnap"-->faceSnap
    </eventType>
    <videoInputChannelID>
        <!--ro, opt, string, video input channel No.-->test
    </videoInputChannelID>
    <TimeBlockList size="8">
        <!--ro, req, array, arming time list, subType:object, attr:size{req, int}-->
        <TimeBlock>
            <!--ro, opt, object, arming time-->
            <dayOfWeek>
                <!--ro, opt, enum, days of the week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->
                >1
                    </dayOfWeek>
                    <TimeRange>
                        <!--ro, req, object, time range-->
                        <beginTime>
                            <!--ro, req, time, start time-->00:00:00+08:00
                        </beginTime>
                        <endTime>
                            <!--ro, req, time, end time-->00:00:00+08:00
                        </endTime>
                    </TimeRange>
                </TimeBlock>
            </TimeBlockList>
        </Schedule>
    <!-->
```

### 16.8.1.4 Set the arming schedule of face capture for a specified channel

#### Request URL

PUT /ISAPI/Event/schedules/faceSnap/faceSnap-<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--wo, opt, object, arming schedule of face capture for a specified channel, attr:version{req, string, protocolVersion}-->
    <id>
        <!--wo, req, string, channel No.-->test
    </id>
    <eventType>
        <!--wo, req, enum, event type, subType:string, desc:"ADAS", "ADASAlarm", AID, ANPR, AccessControllerEvent, CDsStatus, DBD, GPSUpload, HFP, IO, IOTD, LES, LFPD, PALMismatch, PIR, PeopleCounting, PeopleNumChange, Standup, TMA, TMPAM, VMD, abnormalAcceleration, abnormalDriving, advReachheight, alarmResult, attendance, attendedBaggage, audioAbnormal, audioexception, behaviorResult, blindSpotDetection, cardMatch, changedStatus, collision, containerDetection, crowdSituationAnalysis, databaseException, defocus, diskUnformat, diskerror, diskfull, driverConditionMonitor, emergencyAlarm, faceCapture, faceSnapModeling, facedetection, fallDown, faultAlarm, fielddetection, fireDetection, fireEscapeDetection, flowOverrun, framesPeopleCounting, getUp, group, hdBadBlock, hdImpact, heatmap, highHDTemperature, highTempAlarm, hotSpace, illAccess, ipcTransferAbnorma, ipConflict, keyPersonGetUp, leavePosition, linedetection, listSyncException, loitering, lowHDTemperature, mixedTargetDetection, modelError, nicbroken, nodeOffline, nonPoliceIntrust, overSpeed, overtimeTarry, parking, peopleNumChange, peopleNumCounting, personAbnormalAlarm:, personDensityDetection, personQueueCounting, personQueueDetection, personQueueRealTime, personQueueTime, playCellPhone, pocException, poe, policeAbsent, radarAlarm, radarFieldDetection, radarLineDetection, radarPerimeterRule, radarTargetDetection, radarVideoDetection, raidException, rapidMove, reachHeight, recordCycleAbnormal, recordException, regionEntrance, regionExiting, retention, rollover, running, safetyHelmetDetection, scenechangedetection, sensorAlarm, severeHDFailure, shelteralarm, shipsDetection, sitQuietly, smokeAndFireDetection, smokeDetection, softIO, spacingChange, sysStorFull, takingElevatorDetection, targetCapture, temperature, thermometry, thirdPartyException, toiletTarry, tollCodeInfo, tossing, unattendedBaggage, vehicleMatchResult, vehicleRcogResult, versionAbnormal, videoException, videoLoss, violationAlarm, violentMotion, yardTarry, faceSnap-->faceSnap
    </eventType>
    <videoInputChannelID>
        <!--wo, opt, string, video input channel No.-->test
    </videoInputChannelID>
    <TimeBlockList size="8">
        <!--wo, req, array, arming time List, subType:object, attr:size{req, int}-->
        <TimeBlock>
            <!--wo, opt, object, arming time-->
            <dayOfWeek>
                <!--wo, opt, enum, days of the week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->
                >1
                </dayOfWeek>
                <TimeRange>
                    <!--wo, req, object, time range-->
                    <beginTime>
                        <!--wo, req, time, start time-->00:00:00+08:00
                    </beginTime>
                    <endTime>
                        <!--wo, req, time, end time-->00:00:00+08:00
                    </endTime>
                </TimeRange>
            </TimeBlock>
        </TimeBlockList>
    </Schedule>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
    <requestURL>
        <!--ro, req, string, request URL-->null
    </requestURL>
    <statusCode>
        <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    </statusCode>
    <statusString>
        <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusString>
    <subStatusCode>
        <!--ro, req, string, sub status code, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
</ResponseStatus>

```

### 16.8.1.5 Set the arming schedule of face modeling for all channels

#### Request URL

PUT /ISAPI/Event/schedules/faceSnapModeling

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FaceSnapModelingScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, subType:object, attr:version{req, string, protocolVersion}-->
</FaceSnapModelingScheduleList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.8.1.6 Get the arming schedule of face modeling for all channels

#### Request URL

GET /ISAPI/Event/schedules/faceSnapModeling

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FaceSnapModelingScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, subType:object, attr:version{req, string, protocolVersion}-->
</FaceSnapModelingScheduleList>

```

### 16.8.1.7 Delete the linkage of face capture

#### Request URL

DELETE /ISAPI/Event/triggers/faceSnap-<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.8.1.8 Set the linkage parameters of face capture

#### Request URL

PUT /ISAPI/Event/triggers/faceSnap-<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--wo, req, object, Configure Linkage actions:, attr:version{req, string, protocolVersion}-->
  <id>
    <!--wo, req, string, channel No.-->test
  </id>
  <eventType>
    <!--wo, req, enum, event type, subType:string, desc:"ADAS", "ADASAlarm", "AID", "ANPR", "AccessControllerEvent", "CDsStatus", "DBD", "GPSUpload", "HFP", "IO", "IOTD", "LES", "LFPD", "PALMismatch", "PIR", "PeopleCounting", "PeopleNumChange", "Standup", "TMA", "TMPAM", "VMD", "abnormalAcceleration", "abnormalDriving", "advReachHeight", "alarmResult", "attendance", "attendedBaggage", "audioAbnormal", "audioexception", "behaviorResult", "blindsightDetection", "cardMatch", "changedstatus", "collision", "containerDetection", "crowdsituationAnalysis", "databaseException", "defocus", "diskInformat", "diskerror", "diskfull", "driverConditionMonitor", "emergencyAlarm", "faceCapture", "facesnapModeling", "facedetection", "failDown", "FaultAlarm", "fielddetection", "fireDetection", "fireEscapeDetection", "flowOverrun", "framesPeopleCounting", "getUp", "group", "ndBadBlock", "ndImpact", "heatmap", "highHDTemperature", "highTempAlarm", "hotSpare", "illAccess", "ipcTransferAbnormal", "ipConflict", "keyPersonGetUp", "LeavePosition", "linedetection", "listSyncException", "loitering", "LowHDTemperature", "mixedTargetDetection", "modelError", "nicbroken", "nodeOffline", "nonPoliceIntrusio", "overSpeed", "overtimeTarry", "parking", "peopleNumChange", "peopleNumCounting", "personAbnormalAlarm", "personDensityDetection", "personQueueCounting", "personQueueDetection", "personQueueRealTime", "personQueueTime", "playCellphone", "pocException", "poe", "policeAbsent", "radarAlarm", "radarFieldDetection", "radarLineDetection", "radarPerimeterRule", "radarTargetDetection", "radarVideoDetection", "raidException", "rapidMove", "reachHeight", "recordCycleAbnormal", "recordException", "regionEntrance", "regionExiting", "retention", "rollover", "running", "safetyHelmetDetection", "scenedechangedetection", "sensorAlarm", "severeHDFailure", "shelteralarm", "shipsDetection", "sitQuietly", "smokeAndFireDetection", "smokedetection", "softIO", "spacingChange", "sysStorFull", "takingElevatorDetection", "targetCapture", "temperature", "thermometry", "thirdPartyException", "toiletTarry", "tollCodeInfo", "tossing", "unattendedBaggage", "vehicleMatchResult", "vehicleRcogResult", "versionAbnormal", "videoException", "videoloss", "violationAlarm", "violentMotion", "yardTarry", "facesnap"-->faceSnap
  </eventType>
  <eventDescription>
    <!--wo, opt, string, event description-->test
  </eventDescription>
  <videoInputChannelID>
    <!--wo, opt, string, video input channel ID, desc:only valid when event types are VMD, videoloss, shelteralarm, regionEntrance, regionExiting, loitering, group, rapidMove, parking, unattendedBaggage, attendedBaggage, thermal, zoneTriggeredAlarm, criticalAlerts, businessCounselingLinkage-->test
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--wo, opt, string, dynamic video input channel ID-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--wo, opt, array, configuration list of event linkage, subType:object-->
    <EventTriggerNotification>
      <!--wo, opt, object, list-->
      <id>
        <!--wo, req, string, Linkage configuration ID-->test
      </id>
      <notificationMethod>
        <!--wo, req, enum, Linkage action, subType:string, desc:"FTP" (uploading via FTP), "HTTP" (uploading via HTTP), "IM" (instant messaging), "IO" (I/O output), "LightAudioAlarm" (light and audio alarm), "audio" (audio), "beep" (buzzer), "center" (uploading to the center), "cloud" (cloud storage), "email", "focus" (focus), "monitorAlarm" (alarm on monitor), "ptz" (PTZ), "record" (recording), "syslog" (system log), "trace" (tracking), "whiteLight" (white light)-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--wo, opt, enum, Linkage cycle, subType:string, desc:"beginning", "beginningandend", "recurring"-->beginning
      </notificationRecurrence>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.8.1.9 Get the linkage parameters of face capture

#### Request URL

GET /ISAPI/Event/triggers/faceSnap-<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, Linkage configuration, attr:version{req, string, protocolVersion}-->
    <id>
        <!--ro, req, string, channel No.-->test
    </id>
    <eventType>
        <!--ro, req, enum, event type, subType:string, desc:"ADAS", "ADASAlarm", "AID", "ANPR", "AccessControlLerEvent", "CDsStatus", "DBD", "GPSUpload", "HFP", "IO", "IOTD", "LES", "LFPD", "PALMismatch", "PIR", "PeopleCounting", "PeopleNumChange", "Standup", "TMA", "TPAM", "VMD", "abnormalAcceleration", "abnormalDriving", "advReachHeight", "alarmResult", "attendance", "attendedBaggage", "audioAbnormal", "audioexception", "behaviorResult", "blindsightDetection", "cardMatch", "changedStatus", "Collision", "containerDetection", "crowdsituationAnalysis", "databaseException", "defocus", "diskUnformat", "diskrror", "diskfull", "driverConditionMonitor", "emergencyAlarm", "faceCapture", "faceSnapModeling", "facedetection", "failDown", "faultAlarm", "fielddetection", "fireDetection", "fireEscapeDetection", "flowOverrun", "framesPeopleCounting", "getUp", "group", "hdBadBlock", "hdImpact", "heatmap", "highHDTemperature", "highTempAlarm", "hotSpare", "illAccess", "ipcTransferAbnorm", "ipconflict", "keyPersonGetUp", "LeavePosition", "linedetection", "listSyncException", "loitering", "LowHDTemperature", "mixedTargetDetection", "modelError", "nicbroken", "nodeOffline", "nonPoliceIntrusio", "overspeed", "overtimeTarry", "parking", "peopleNumChange", "peopleNumCounting", "personAbnormalAlarm", "personDensityDetection", "personQueueCounting", "personQueueDetection", "personQueueRealTime", "personQueueTime", "playCellphone", "pocException", "poe", "policeAbsent", "radarAlarm", "radarFieldDetection", "radarLineDetection", "radarPerimeterRule", "radarTargetDetection", "radarVideoDetection", "raidException", "rapidMove", "reachHeight", "recordCycleAbnormal", "recordException", "regionEntrance", "regionExiting", "retention", "rollOver", "running", "safetyHelmetDetection", "scenedechangedetection", "sensorAlarm", "severeHDFailure", "shelteralarm", "shipsDetection", "sitQuietly", "smokeAndFireDetection", "smokedetection", "softIO", "spacingChange", "sysStorFull", "takingElevatorDetection", "targetCapture", "temperature", "thermometry", "thirdPartyException", "toiletTarry", "tollCodeInfo", "tossing", "unattendedBaggage", "attendedBaggage", "thermal", "zoneTriggeredAlarm", "criticalAlerts", "businessCounselingLinkage"-->test
    </videoInputChannelID>
    <dynVideoInputChannelID>
        <!--ro, opt, string, dynamic video input channel ID-->1
    </dynVideoInputChannelID>
    <EventTriggerNotificationList>
        <!--ro, opt, array, configuration List of event Linkage, subType:object-->
        <EventTriggerNotification>
            <!--ro, opt, object, List-->
            <id>
                <!--ro, req, string, Linkage configuration ID-->test
            </id>
            <notificationMethod>
                <!--ro, req, enum, Linkage action, subType:string, desc:"FTP" (uploading via FTP), "HTTP" (uploading via HTTP), "IM" (instant messaging), "IO" (I/O output), "LightAudioAlarm" (light and audio alarm), "audio" (audio), "beep" (buzzer), "center" (uploading to the center), "cloud" (cloud storage), "email", "focus" (focus), "monitorAlarm" (alarm on monitor), "ptz" (PTZ), "record" (recording), "syslog" (system log), "trace" (tracking), "whiteLight" (white light)-->FTP
            </notificationMethod>
            <notificationRecurrence>
                <!--ro, opt, enum, Linkage cycle, subType:string, desc:"beginning", "beginningandend", "recurring"-->beginning
            </notificationRecurrence>
        </EventTriggerNotification>
    </EventTriggerNotificationList>
</EventTrigger>

```

### 16.8.1.10 Set the picture information overlay configurations of a device

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/picInfoOverlap

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PicInfoOverlap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--wo, opt, object, attr:version{req, string, protocolVersion}-->
  <Overlap>
    <!--wo, opt, object-->
    <enabled>
      <!--wo, opt, bool-->true
    </enabled>
    <OverlapItemList>
      <!--wo, opt, array, List of overlay information, subType:object-->
      <OverlapItem>
        <!--wo, opt, object, overlay information-->
        <id>
          <!--wo, req, int-->1
        </id>
        <item>
          <!--wo, req, enum, subType:string-->positionInfo
        </item>
      </OverlapItem>
    </OverlapItemList>
  </Overlap>
</PicInfoOverlap>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
  </ResponseStatus>

```

### 16.8.1.11 Face capture and recognition event

#### EventType:faceCapture

```

{
  "ipAddress": "172.6.64.7",
  /*ro, req, string, IPv4 address of the device that triggers the alarm*/
  "ipv6Address": "1080:0:0:0:8:800:200C:417A",
  /*ro, opt, string, IPv6 address of the device that triggers the alarm*/
  "portNo": 80,
  /*ro, opt, int, communication port No. of the device that triggers the alarm*/
  "protocol": "HTTP",
  /*ro, opt, enum, protocol type, subType:string, desc:"HTTP", "HTTPS", "EHome"*/
  "macAddress": "01:17:24:45:D9:F4",
  /*ro, opt, string, MAC address*/
  "channelID": 1,
  /*ro, opt, int, channel No. of the device that triggers the alarm*/
  "relatedChannelList": [1, 2, 3],
  /*ro, opt, array, list of alarm related channels, which are of the same camera with channelID, subType:int, desc:this parameter is used for live view or playback on the platform*/
  "dateTime": "2004-05-03T17:30:08+08:00",
  /*ro, req, datetime, alarm trigger time*/
  "activePostCount": 1,
  /*ro, req, int, times that the same alarm has been uploaded*/
  "eventType": "faceCapture",
  /*ro, req, string, event type*/
  "eventState": "active",
  /*ro, req, enum, event status, subType:string, desc:for durative event: active (valid), inactive (invalid)*/
  "eventDescription": "faceCapture",
  /*ro, req, string, event description, desc:face capture data upload*/
  "channelName": "ABC",
  /*ro, req, string, channel name (camera name)*/
  "deviceID": "test0123",
  /*ro, opt, string, device ID (PUID)*/
  "isDataRetransmission": true,
  /*ro, opt, bool, whether the data is replenished*/
  "faceCapture": [
    /*ro, opt, array, face capture and recognition results, one result can contain multiple pictures and one picture can contains multiple faces, subType:object*/
  ]
}

```

```

"errorCode": 1,
/*ro, opt, int, error code*/
"errorMsg": "ok",
/*ro, opt, string, error code description*/
"taskId": "temp_dfasdffsafdfsadada",
/*ro, opt, string, task ID*/
"faceTime": "2004-05-03T17:30:08+08:00",
/*ro, opt, datetime, capture time*/
"traceUuid": "sdddddddf",
/*ro, opt, string, target ID, the same person in multiple capture results corresponds to same target ID*/
"traceIdx": 1,
/*ro, opt, int, No. of picture with highest comparison similarity*/
"targetAttrs": {
/*ro, opt, object, face attributes*/
"deviceId": "ee3c4000-a22d-11b4-8334-5803fb77894c",
/*ro, opt, string, device UUID*/
"deviceChannel": 1,
/*ro, opt, int, device channel No.*/
"deviceName": "shebei",
/*ro, opt, string, device name*/
"faceTime": "2004-05-03T17:30:08+08:00",
/*ro, opt, datetime, capture time*/
"bkgUrl": "test",
/*ro, opt, string, background picture URL*/
"thermalURL": "test",
/*ro, opt, string, thermal picture URL*/
"contentID": "backgroundImage",
/*ro, opt, string, background picture (binary)*/
"ID": "F4F665D6A18E41308CE9934DCDD01111"
/*ro, opt, string*/
},
"faces": [
/*ro, opt, array, face picture information, which contains the recognition results, face attributes, modeling, matched list, and so on,
subType:object*/
{
"faceId": 1,
/*ro, opt, int, face ID*/
"faceRect": {
/*ro, opt, object, face thumbnail frame, desc:the origin is the upper-left corner of the screen*/
"height": 1.000,
/*ro, opt, float, height, range:[0.000,1.000]*/
"width": 1.000,
/*ro, opt, float, width, range:[0.000,1.000]*/
"x": 0.000,
/*ro, opt, float, X-coordinate, range:[0.000,1.000]*/
"y": 0.000
/*ro, opt, float, Y-coordinate, range:[0.000,1.000]*/
},
"age": {
/*ro, opt, object, age*/
"range": 5,
/*ro, opt, int, age deviation*/
"value": 23,
/*ro, opt, int, age*/
"ageGroup": "old"
/*ro, opt, string, age group*/
},
"gender": {
/*ro, opt, object, gender*/
"confidence": 99.999,
/*ro, opt, float, confidence*/
"value": "male"
/*ro, opt, enum, gender, subType:string, desc:"male", "female", "unknown"*/
},
"glass": {
/*ro, opt, object, whether the person wears glasses*/
"confidence": 99.945,
/*ro, opt, float, confidence*/
"value": "yes"
/*ro, opt, enum, whether the person wears glasses, subType:string, desc:"no", "yes"*/
},
"mask": {
/*ro, opt, object, whether the person wears a mask*/
"confidence": 99.945,
/*ro, opt, float, confidence*/
"value": "yes"
/*ro, opt, enum, whether the person wears a mask, subType:string, desc:"no", "yes"*/
},
"smile": {
/*ro, opt, object, whether the person is smiling*/
"confidence": 99.945,
/*ro, opt, float, confidence*/
"value": "no"
/*ro, opt, enum, whether the person is smiling, subType:string, desc:"no", "yes"*/
},
"faceExpression": {
/*ro, opt, object, facial expression*/
"value": "unknown"
/*ro, opt, enum, facial expression, subType:string, desc:"unknown", "surprised", "panic", "disgusted", "happy", "sad", "angry", "poker-faced"*/
},
"hat": {
/*ro, opt, object, whether the person wears a hat*/
"value": "no"
}
]
}

```

```

    /*ro, opt, enum, whether the person wears a hat, subType:string, desc:"no", "yes", "unknown"*/
},
"URL": "xxxxxx",
/*ro, opt, string, face picture thumbnail URL*/
"contentID": "faceImage",
/*ro, opt, string, face picture thumbnail (binary)*/
"stayDuration": 20,
/*ro, opt, int, dwell time duration in the image, unit:ms*/
"faceScore": 30,
/*ro, req, int, face score (face picture quality), range:[0,100]*/
"captureEndMark": true,
/*ro, req, bool, end flag of capture, desc:true (ended), false (not ended)*/
"targetID": "xxxxxx",
/*ro, opt, string, face and human body linkage ID*/
"FacePictureRect": {
/*ro, opt, object, rectangle frame for human face, desc:the origin is the upper-left corner of the screen*/
    "height": 1.000,
    /*ro, req, float, height, range:[0.000,1.000]*/
    "width": 1.000,
    /*ro, req, float, width, range:[0.000,1.000]*/
    "x": 0.000,
    /*ro, req, float, X-coordinate, range:[0.000,1.000]*/
    "y": 0.000,
    /*ro, req, float, Y-coordinate, range:[0.000,1.000]*/
},
"blockingState": "noBlocking",
/*ro, opt, enum, target face shielding status, subType:string, desc:"noBlocking" (unshielded), "instantaneousInsignificantBlocking" (lightly shielded), "continuousInsignificantBlocking" (keep shielding), "significantBlocking" (seriously shielded), "OSD", "unknown"*/
"swingAngle": -90,
/*ro, opt, int, panning angle, range:[-90,90]*/
"tiltAngle": -90,
/*ro, opt, int, tilting angle, range:[-90,90]*/
"pupilDistance": 10,
/*ro, opt, int, pupil distance of target face, desc:the maximum value is 10 bytes, and the maximum value is (resolution width/1.6), unit: pixel*/
"livenessDetectionStatus": "liveFace",
/*ro, opt, enum, face anti-spoofing detection status, subType:string, desc:"liveFace" (live face), "notLiveFace" (the face is not real), "detectionFailed" (detection failed), "notEnabled" (disabled)*/
"pid": "F4F665D6A18E41308CE9934DCDD222",
/*ro, opt, string*/
"enterTime": "2004-05-03T17:30:08+08:00",
/*ro, opt, datetime*/
"exitTime": "2004-05-03T17:30:08+08:00",
/*ro, opt, datetime*/
"perClassify": 0,
/*ro, opt, int, face data presorting*/
"faceSnapThermometryEnabled": true,
/*ro, opt, bool, whether to enable uploading captured face picture with temperature information*/
"currTemperature": 36.8,
/*ro, opt, float, face temperature*/
"isAbnormalTemperature": true,
/*ro, opt, bool, whether the temperature is abnormal or not: true, false*/
"thermometryUnit": "celsius",
/*ro, opt, enum, temperature unit, subType:string, desc:"celsius", "fahrenheit", "kelvin"*/
"alarmTemperature": 36.7,
/*ro, opt, float, temperature threshold to trigger alarm*/
"interiorPosition": "driver"
/*ro, opt, enum, subType:string*/
}
],
"uid": "2019052010392202600Kb4s3iQPzLignclsRUbFT0hr039Y0YzuLt0VLf199yZi",
/*ro, opt, string, the unique ID of event uploading*/
"OperatorInfo": {
/*ro, opt, object, operator information, desc:operator information*/
    "personUUID": "test",
    /*ro, opt, string, range:[1,32]*/
    "personName": "test"
    /*ro, opt, string, range:[1,64]*/
},
"MEFData": [
/*ro, opt, array, subType:object*/
{
    "time": "2004-05-03T17:30:08.000+08:00",
    /*ro, req, datetime, it should be accurate to the millisecond, desc:it should be accurate to the millisecond*/
    "IMSI": "460001357924680",
    /*ro, req, string, range:[1,15]*/
    "bandIdx": 11
    /*ro, opt, enum, subType:int*/
}
],
"isNeedVerification": true
/*ro, opt, bool*/
}
],
"GPS": {
/*ro, opt, object*/
    "divisionEW": "E",
    /*ro, req, enum, "E" (Eastern Hemisphere), "W" (West Hemisphere), subType:string, desc:"E" (Eastern Hemisphere), "W" (West Hemisphere)*/
    "longitude": 100,
    /*ro, req, int, longitude, desc:longitude*/
    "divisionNS": "N",
    /*ro, req, enum, "S" (Southern Hemisphere), "N" (North Hemisphere), subType:string, desc:"S" (Southern Hemisphere), "N" (North Hemisphere)*/
    "latitude": 100
}

```

```

    /*ro, req, int, latitude, desc:latitude*/
},
"URLCertificationType": "digest"
/*ro, opt, enum, picture URL authentication method, subType:string, desc:"no" (no authentication, it is used for the cloud protocol), "digest" (digest authentication, it is used for local picture URL returned by NVR or DVR)*/
}

```

Parameter Name	Parameter Value	Parameter Type(Content-Type)	Content-ID	File Name	Description
faceCapture	[Message content]	application/json	--	--	--

**Note:** The protocol is transmitted in form format. See Chapter 4.5.1.4 for form framework description, as shown in the instance below.

```

--<frontier>
Content-Disposition: form-data; name=Parameter Name;filename=File Name
Content-Type: Parameter Type
Content-Length: ****
Content-ID: Content ID
Parameter Value

```

- Parameter Name: the name property of Content-Disposition in the header of form unit; it refers to the form unit name.
- Parameter Type (Content-Type): the Content-Type property in the header of form unit.
- File Name (filename): the filename property of Content-Disposition of form unit Headers. It exists only when the transmitted data of form unit is file, and it refers to the file name of form unit body.
- Parameter Value: the body content of form unit.

## 16.8.2 Face Picture Comparison

### 16.8.2.1 Get the arming schedule of all face comparison channels

#### Request URL

GET /ISAPI/Event/schedules/faceContrast

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FaceContrastScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, array, arming schedule List, subType:object, attr:version{req, string, protocolVersion}-->
</FaceContrastScheduleList>

```

### 16.8.2.2 Set the arming schedule of multiple face picture comparison alarms

#### Request URL

PUT /ISAPI/Event/schedules/faceContrast

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FaceContrastScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, array, schedule information List, subType:object, attr:version{req, string, protocolVersion}-->
</FaceContrastScheduleList>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

## 16.9 Road Traffic

### 16.9.1 Enforcement

#### 16.9.1.1 Search for the status of manual traffic law enforcement

##### Request URL

PUT /ISAPI/Traffic/channels/<channelID>/manualItsCapStatus

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ManualItsCapStatus xmlns="http://www.isapi.org/ver20/XMLSchema">
  <!--ro, opt, object-->
</ManualItsCapStatus>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ManualItsCapStatusResult xmlns="http://www.isapi.org/ver20/XMLSchema">
  <!--ro, opt, object-->
</ManualItsCapStatusResult>

```

### 16.9.2 Traffic Event Detection

#### 16.9.2.1 Set a single scene auto-switch schedule by channel

##### Request URL

PUT /ISAPI/Traffic/channels/<channelID>/sceneCruiseSchedule/<SID>

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<CruiseScheduleBlock xmlns="http://www.isapi.org/ver20/XMLSchema">
    <!--opt, object, patrol schedule-->
    <dayOfweek>
        <!--req, enum, days of the week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->1
    </dayOfweek>
    <CruiseScheduleRange>
        <!--opt, object, schedule time-->
        <timeID>
            <!--req, string, arming time, range:[1,16]-->1
        </timeID>
        <beginTime>
            <!--req, time, start time-->00:00:00+08:00
        </beginTime>
        <endTime>
            <!--req, time, end time-->00:00:00+08:00
        </endTime>
        <SceneList>
            <!--opt, array, scene list, subType:object-->
            <Scene>
                <!--opt, object, scene-->
                <id>
                    <!--req, string, Linked scene ID, range:[1,16], desc:the default value is 1-->1
                </id>
                <duration>
                    <!--req, int, duration-->0
                </duration>
            </Scene>
        </SceneList>
    </CruiseScheduleRange>
</CruiseScheduleBlock>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
    <requestURL>
        <!--ro, req, string, request URL-->null
    </requestURL>
    <statusCode>
        <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    </statusCode>
    <statusString>
        <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusString>
    <subStatusCode>
        <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
</ResponseStatus>

```

### 16.9.2.2 Get a single scene auto-switch schedule by channel

#### Request URL

GET /ISAPI/Traffic/channels/<channelID>/sceneCruiseSchedule/<SID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<CruiseScheduleBlock xmlns="http://www.isapi.org/ver20/XMLSchema">
  <!--ro, opt, object-->
  <dayOfWeek>
    <!--ro, req, enum, day of the week based on ISO8601, "1"=Monday,..., subType:int, desc:day of the week based on ISO8601, "1"=Monday,...-->1
  </dayOfWeek>
  <CruiseScheduleRange>
    <!--ro, opt, object-->
    <timeID>
      <!--ro, req, string, arming schedule ID, range:[1,16]-->1
    </timeID>
    <beginTime>
      <!--ro, req, time, start time-->00:00:00+08:00
    </beginTime>
    <endTime>
      <!--ro, req, time, end time-->00:00:00+08:00
    </endTime>
    <SceneList>
      <!--ro, opt, array, subType:object-->
      <Scene>
        <!--ro, opt, object-->
        <id>
          <!--ro, req, string, Linked scene ID, range:[1,16], desc:the default value is 1-->1
        </id>
        <duration>
          <!--ro, req, int, duration-->0
        </duration>
      </Scene>
    </SceneList>
  </CruiseScheduleRange>
</CruiseScheduleBlock>

```

## 16.10 Smart Event (Perimeter Intrusion Detection)

### 16.10.1 Intrusion Detection

#### 16.10.1.1 Set intrusion detection arming schedules of all channels

##### Request URL

PUT /ISAPI/Event/schedules/fieldDetections

##### Query Parameter

None

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FieldDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, refer to the message XML_Schedule for details, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--opt, object-->
    <id>
      <!--req, string-->test
    </id>
    <eventType>
      <!--opt, enum, event type, subType:string-->fieldDetection
    </eventType>
    <videoInputChannelID>
      <!--opt, string, video input channel ID-->test
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--req, object, arming time list, attr:size{req, int}-->
      <TimeBlock>
        <!--opt, object, arming time-->
        <dayOfWeek>
          <!--req, enum, day of the week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->1
        </dayOfWeek>
        <TimeRange>
          <!--req, object, time range-->
          <beginTime>
            <!--req, time, start time-->00:00:00
          </beginTime>
          <endTime>
            <!--req, time, end time-->00:00:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>
</FieldDetectionScheduleList>

```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code description-->OK
  </subStatusCode>
  <subStatusString>
    <!--ro, req, string, sub status code, desc:sub status code description-->OK
  </subStatusString>
</ResponseStatus>
```

### 16.10.1.2 Get arming schedules of all channels for intrusion detection

#### Request URL

GET /ISAPI/Event/schedules/fieldDetections

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<FieldDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, refer to the message XML_Schedule for details, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--ro, opt, object, configuration of intrusion detection arming schedules-->
    <id>
      <!--ro, req, string, ID, desc:format: event type-channel No.-->test
    </id>
    <eventType>
      <!--ro, opt, enum, event type, subType:string-->fieldDetection
    </eventType>
    <videoInputChannelID>
      <!--ro, opt, string, video input channel ID-->test
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--ro, req, object, list of arming schedules, attr:size{req, int}-->
      <TimeBlock>
        <!--ro, opt, object, arming schedules-->
        <dayOfWeek>
          <!--ro, opt, enum, day of a week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->1
        </dayOfWeek>
        <TimeRange>
          <!--ro, req, object, time range-->
          <beginTime>
            <!--ro, req, time, start time-->00:00:00
          </beginTime>
          <endTime>
            <!--ro, req, time, end time-->00:00:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>
</FieldDetectionScheduleList>
```

### 16.10.1.3 Set the arming schedule of intrusion detection for a specified channel

#### Request URL

PUT /ISAPI/Event/schedules/fieldDetections/fieldDetection-<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->格式: 事件类型-通道号
  </id>
  <eventType>
    <!--opt, string-->fieldDetection
  </eventType>
  <videoInputChannelID>
    <!--opt, string-->test
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--req, object, attr:size{req, int}-->
    <TimeBlock>
      <!--opt, object-->
      <dayOfWeek>
        <!--opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--req, object-->
        <beginTime>
          <!--req, time-->00:00:00
        </beginTime>
        <endTime>
          <!--req, time-->00:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>
```

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.10.1.4 Get the arming schedule of intrusion detection for a specified channel

#### Request URL

GET /ISAPI/Event/schedules/fieldDetections/fieldDetection-<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->test
  </id>
  <eventType>
    <!--ro, opt, string-->fieldDetection
  </eventType>
  <videoInputChannelID>
    <!--ro, opt, string-->test
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--ro, req, object, attr:size{req, int}-->
    <TimeBlock>
      <!--ro, opt, object-->
      <dayOfWeek>
        <!--ro, opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--ro, req, object-->
        <beginTime>
          <!--ro, req, time-->00:00:00
        </beginTime>
        <endTime>
          <!--ro, req, time-->00:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>

```

### 16.10.1.5 Get arming schedules of all radar channels for intrusion detection

#### Request URL

GET /ISAPI/Event/schedules/radarFieldDetections

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RadarFieldDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, List of arming schedules, subType:object, attr:version{req, string, protocolVersion}-->
</RadarFieldDetectionScheduleList>

```

### 16.10.1.6 Set arming schedules of all radar channels for intrusion detection

#### Request URL

PUT /ISAPI/Event/schedules/radarFieldDetections

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RadarFieldDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, refer to the message XML_Schedule for details, subType:object, attr:version{req, string, protocolVersion}-->
</RadarFieldDetectionScheduleList>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, desc:sub status code description-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, desc:sub status code description-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

### 16.10.1.7 Set the arming schedule of radar intrusion detection for a specified channel

#### Request URL

PUT /ISAPI/Event/schedules/radarFieldDetections/radarFieldDetection-<radarChannelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
radarChannelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->radarFieldDetection-1
  </id>
  <eventType>
    <!--opt, enum, subType:string-->radarFieldDetection
  </eventType>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--req, array, subType:object, attr:size{opt, int}-->
    <TimeBlock>
      <!--opt, object-->
      <dayOfWeek>
        <!--opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--req, object-->
        <beginTime>
          <!--req, time-->10:00:00
        </beginTime>
        <endTime>
          <!--req, time-->10:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.1.8 Get the arming schedule of radar intrusion detection for a specified channel

#### Request URL

GET /ISAPI/Event/schedules/radarFieldDetections/radarFieldDetection-<radarChannelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
radarChannelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->radarFieldDetection-1
  </id>
  <eventType>
    <!--ro, opt, enum, subType:string-->radarFieldDetection
  </eventType>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--ro, req, array, subType:object, attr:size{opt, int}-->
    <TimeBlock>
      <!--ro, opt, object-->
      <dayOfWeek>
        <!--ro, opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--ro, req, object-->
        <beginTime>
          <!--ro, req, time-->10:00:00
        </beginTime>
        <endTime>
          <!--ro, req, time-->10:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>

```

### 16.10.1.9 Set the linkage triggering parameters of intrusion detection event for a specified channel

#### Request URL

PUT /ISAPI/Event/triggers/FieldDetection-<channelID>/notifications

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, subType:object, attr:version{req, string, protocolVersion}-->
  <EventTriggerNotification>
    <!--opt, object-->
    <id>
      <!--req, string-->test
    </id>
    <notificationMethod>
      <!--req, enum, subType:string-->email
    </notificationMethod>
    <notificationRecurrence>
      <!--opt, enum, subType:string-->beginning
    </notificationRecurrence>
    <videoInputID>
      <!--opt, string-->1
    </videoInputID>
  </EventTriggerNotification>
</EventTriggerNotificationList>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.10.1.10 Get the linkage triggering parameters of intrusion detection event for a specified channel

#### Request URL

GET /ISAPI/Event/triggers/FieldDetection-<channelID>/notifications

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, subType:object, attr:version{req, string, protocolVersion}-->
  <EventTriggerNotification>
    <!--ro, opt, object-->
    <id>
      <!--ro, req, string-->test
    </id>
    <notificationMethod>
      <!--ro, req, enum, subType:string-->email
    </notificationMethod>
    <notificationRecurrence>
      <!--ro, opt, enum, subType:string-->beginning
    </notificationRecurrence>
    <videoInputID>
      <!--ro, opt, string-->1
    </videoInputID>
  </EventTriggerNotification>
</EventTriggerNotificationList>
```

### 16.10.1.11 Set the linkage parameters of intrusion detection event for a specified channel

#### Request URL

PUT /ISAPI/Event/triggers/FieldDetection-<channelID>?triggersEnabledStatus=<triggersEnabledStatus>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
triggersEnabledStatus	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->1
  </id>
  <eventType>
    <!--req, enum, subType:string-->fielddetection
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
    <EventTriggerNotification>
      <!--opt, object-->
      <id>
        <!--req, string-->test
      </id>
      <notificationMethod>
        <!--req, enum, subType:string-->email
      </notificationMethod>
      <notificationRecurrence>
        <!--opt, enum, subType:string-->beginning
      </notificationRecurrence>
      <videoInputID>
        <!--opt, string-->1
      </videoInputID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>
```

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.10.1.12 Get the linkage parameters of intrusion detection event for a specified channel

#### Request URL

GET /ISAPI/Event/triggers/FieldDetection-<channelID>?triggersEnabledStatus=<triggersEnabledStatus>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
triggersEnabledStatus	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->1
  </id>
  <eventType>
    <!--ro, req, enum, subType:string-->FieldDetection
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subType:object-->
    <EventTriggerNotification>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->test
      </id>
      <notificationMethod>
        <!--ro, req, enum, subType:string-->email
      </notificationMethod>
      <notificationRecurrence>
        <!--ro, opt, enum, subType:string-->beginning
      </notificationRecurrence>
      <videoInputID>
        <!--ro, opt, string-->1
      </videoInputID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>
```

### 16.10.1.13 Set the linkage parameters of radar intrusion detection event for a specified channel

#### Request URL

PUT /ISAPI/Event/triggers/radarFieldDetection-<radarChannelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
radarChannelID	string	--

### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->radarFieldDetection-1
  </id>
  <eventType>
    <!--req, enum, subType:string-->radarFieldDetection
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
    <EventTriggerNotification>
      <!--opt, object-->
      <id>
        <!--req, string-->test
      </id>
      <notificationMethod>
        <!--req, enum, subType:string-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--opt, enum, subType:string-->beginning
      </notificationRecurrence>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.1.14 Get the linkage parameters of radar intrusion detection event for a specified channel

#### Request URL

GET /ISAPI/Event/triggers/radarFieldDetection-<radarChannelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
radarChannelID	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->radarFieldDetection-1
  </id>
  <eventType>
    <!--ro, req, enum, subType:string-->radarFieldDetection
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subType:object-->
    <EventTriggerNotification>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->test
      </id>
      <notificationMethod>
        <!--ro, req, enum, subType:string-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--ro, opt, enum, subType:string-->beginning
      </notificationRecurrence>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>
```

#### 16.10.1.15 Set intrusion detection parameters of all channels

##### Request URL

PUT /ISAPI/Smart/FieldDetection

##### Query Parameter

None

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FieldDetectionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, subType:object, attr:version{req, string, protocolVersion}-->
  <FieldDetection>
    <!--opt, object-->
    <id>
      <!--req, string, channel No.-->test
    </id>
    <enabled>
      <!--req, bool-->true
    </enabled>
    <startTriggerTime>
      <!--req, int, unit:ms-->1
    </startTriggerTime>
    <endTriggerTime>
      <!--req, int, unit:ms-->1
    </endTriggerTime>
    <normalizedScreenSize>
      <!--opt, object, normalized size-->
      <normalizedScreenWidth>
        <!--req, int, normalized width-->1
      </normalizedScreenWidth>
      <normalizedScreenHeight>
        <!--req, int, normalized height-->1
      </normalizedScreenHeight>
    </normalizedScreenSize>
    <FieldDetectionRegionList size="4">
      <!--opt, array, subType:object, attr:size{req, int}-->
      <FieldDetectionRegion>
        <!--opt, object, intrusion region-->
        <id>
          <!--req, string-->1
        </id>
        <enabled>
          <!--req, bool, whether to enable the function-->true
        </enabled>
        <sensitivityLevel>
          <!--req, int, sensitivity, range:[0,100]-->1
        </sensitivityLevel>
        <timeThreshold>
          <!--req, int-->1
        </timeThreshold>
        <detectionTarget>
          <!--opt, enum, detection target, subType:string, desc:detection target-->all
        </detectionTarget>
        <RegionCoordinatesList>
          <!--opt, array, rectangle, subType:object, range:[0,4], desc:rectangle-->
          <RegionCoordinates>
            <!--opt, object, area coordinates, desc:the origin is the lower-left corner of the screen-->
            <positionX>
              <!--req, int, x-coordinate, range:[0,1000]-->1
            </positionX>
            <positionY>
              <!--req, int, y-coordinate, range:[0,1000]-->1
            </positionY>
          </RegionCoordinates>
        </RegionCoordinatesList>
        <alarmConfidence>
          <!--opt, enum, subType:string-->low
        </alarmConfidence>
      </FieldDetectionRegion>
    </FieldDetectionRegionList>
  </FieldDetection>
</FieldDetectionList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code description-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.1.16 Get intrusion detection parameters of all channels

#### Request URL

GET /ISAPI/Smart/FieldDetection

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<FieldDetectionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, array, intrusion detection parameter List, subType:object, attr:version{req, string, protocolVersion}-->
    <FieldDetection>
        <!--ro, opt, object, intrusion detection parameters-->
        <id>
            <!--ro, req, string, channel No.-->test
        </id>
        <enabled>
            <!--ro, req, bool-->true
        </enabled>
        <startTriggerTime>
            <!--ro, req, int, unit:ms-->1
        </startTriggerTime>
        <endTriggerTime>
            <!--ro, req, int, unit:ms-->1
        </endTriggerTime>
        <normalizedScreenSize>
            <!--ro, opt, object, normalized size-->
            <normalizedScreenWidth>
                <!--ro, req, int, normalized width-->1
            </normalizedScreenWidth>
            <normalizedScreenHeight>
                <!--ro, req, int, normalized height-->1
            </normalizedScreenHeight>
        </normalizedScreenSize>
        <FieldDetectionRegionList size="4">
            <!--ro, opt, array, subType:object, attr:size{req, int}-->
            <FieldDetectionRegion>
                <!--ro, opt, object-->
                <id>
                    <!--ro, req, string-->1
                </id>
                <enabled>
                    <!--ro, req, bool, whether to enable the function-->true
                </enabled>
                <sensitivityLevel>
                    <!--ro, req, int, sensitivity-->1
                </sensitivityLevel>
                <timeThreshold>
                    <!--ro, req, int, time threshold-->1
                </timeThreshold>
                <detectionTarget>
                    <!--ro, opt, enum, detection target, subType:string, desc:detection target-->all
                </detectionTarget>
                <RegionCoordinatesList>
                    <!--ro, opt, array, rectangle, subType:object, range:[0,4], desc:rectangle-->
                    <RegionCoordinates>
                        <!--ro, opt, object, area coordinates, desc:the origin is the lower-left corner of the screen-->
                        <positionX>
                            <!--ro, req, int, X-coordinate, range:[0,1000]-->1
                        </positionX>
                        <positionY>
                            <!--ro, req, int, Y-coordinate, range:[0,1000]-->1
                        </positionY>
                    </RegionCoordinates>
                </RegionCoordinatesList>
                <alarmConfidence>
                    <!--ro, opt, enum, subType:string-->low
                </alarmConfidence>
            </FieldDetectionRegion>
        </FieldDetectionRegionList>
    </FieldDetection>
</FieldDetectionList>
```

### 16.10.1.17 Get intrusion detection parameters by video input channel

#### Request URL

GET /ISAPI/Smart/FieldDetection/<channelID>

## Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<FieldDetection xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, intrusion detection, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string, channel No.-->test
  </id>
  <enabled>
    <!--ro, req, bool, whether to enable the function-->true
  </enabled>
  <startTriggerTime>
    <!--ro, req, int, unit: milliseconds, unit:ms-->1
  </startTriggerTime>
  <endTriggerTime>
    <!--ro, req, int, unit: milliseconds, unit:ms-->1
  </endTriggerTime>
  <normalizedScreenSize>
    <!--ro, opt, object, normalized size-->
    <normalizedScreenWidth>
      <!--ro, req, int, normalized width-->1
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--ro, req, int, normalized height-->1
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <FieldDetectionRegionList size="4">
    <!--ro, opt, array, subtype:object, attr:size{req, int}-->
    <FieldDetectionRegion>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string, ID-->1
      </id>
      <enabled>
        <!--ro, req, bool, whether to enable the function-->true
      </enabled>
      <sensitivityLevel>
        <!--ro, req, int, sensitivity-->1
      </sensitivityLevel>
      <timeThreshold>
        <!--ro, req, int, time threshold-->1
      </timeThreshold>
      <detectionTarget>
        <!--ro, opt, enum, detection target, subType:string-->all
      </detectionTarget>
      <RegionCoordinatesList>
        <!--ro, opt, array, area coordinate list, subType:object, range:[0,4], desc:rectangle-->
        <RegionCoordinates>
          <!--ro, opt, object, area coordinates, desc:the origin is the lower-left corner of the screen-->
          <positionX>
            <!--ro, req, int, X-coordinate, range:[0,1000]-->1
          </positionX>
          <positionY>
            <!--ro, req, int, Y-coordinate, range:[0,1000]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
      <alarmConfidence>
        <!--ro, opt, enum, subType:string-->low
      </alarmConfidence>
    </FieldDetectionRegion>
  </FieldDetectionRegionList>
</FieldDetection>
```

## 16.10.1.18 Set intrusion detection parameters by channel

### Request URL

PUT /ISAPI/Smart/FieldDetection/<channelID>

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FieldDetection xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, intrusion detection parameters, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string, channel No.-->test
  </id>
  <enabled>
    <!--req, bool, whether to enable the function-->true
  </enabled>
  <startTriggerTime>
    <!--req, int, trigger start time, unit:ms-->1
  </startTriggerTime>
  <endTriggerTime>
    <!--req, int, trigger end time, unit:ms-->1
  </endTriggerTime>
  <normalizedScreenSize>
    <!--opt, object, normalized size-->
    <normalizedScreenWidth>
      <!--req, int, normalized width-->1
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--req, int, normalized height-->1
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <FieldDetectionRegionList size="4">
    <!--opt, array, intrusion detection region list, subType:object, attr:size{req, int}-->
    <FieldDetectionRegion>
      <!--opt, object, intrusion detection region-->
      <id>
        <!--req, string, region ID-->1
      </id>
      <enabled>
        <!--req, bool, whether to enable the function-->true
      </enabled>
      <sensitivityLevel>
        <!--req, int, sensitivity-->1
      </sensitivityLevel>
      <timeThreshold>
        <!--req, int, time threshold-->1
      </timeThreshold>
      <detectionTarget>
        <!--opt, enum, detection target, subType:string-->test
      </detectionTarget>
      <RegionCoordinatesList>
        <!--opt, array, region coordinate list, subType:object, range:[0,4], desc:rectangle-->
        <RegionCoordinates>
          <!--opt, object, region coordinates, desc:the origin is the Lower-left corner of the screen-->
          <positionX>
            <!--req, int, X-coordinate, range:[0,1000]-->1
          </positionX>
          <positionY>
            <!--req, int, Y-coordinate, range:[0,1000]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
      <alarmConfidence>
        <!--opt, enum, alarm confidence, subType:string, desc:"Low" (default value), "mediumLow", "mediumHigh", "high"-->low
      </alarmConfidence>
    </FieldDetectionRegion>
  </FieldDetectionRegionList>
</FieldDetection>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.1.19 Get configuration capability of intrusion detection by channel

#### Request URL

GET /ISAPI/Smart/FieldDetection/<channelID>/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FieldDetection xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string, channel No.-->test
  </id>
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <enableDualVca opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enableDualVca>
  <normalizedScreenSize>
    <!--ro, opt, object-->
    <normalizedScreenWidth>
      <!--ro, req, int-->1
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--ro, req, int-->1
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <FieldDetectionRegionList size="4">
    <!--ro, opt, array, subType:object, attr:size{req, int}-->
    <FieldDetectionRegion>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string, region ID-->1
      </id>
      <minRegionCoordinatesNum>
        <!--ro, opt, int-->1
      </minRegionCoordinatesNum>
      <maxRegionCoordinatesNum>
        <!--ro, opt, int-->1
      </maxRegionCoordinatesNum>
      <sensitivityLevel>
        <!--ro, req, int, sensitivity-->1
      </sensitivityLevel>
      <timeThreshold>
        <!--ro, req, int, time threshold-->1
      </timeThreshold>
      <detectionTarget>
        <!--ro, opt, enum, detection target, subType:string-->all
      </detectionTarget>
      <alarmConfidence opt="low,mediumLow,mediumHigh,high" def="low">
        <!--ro, opt, enum, subType:string, attr:opt{req, string},def{req, string}-->low
      </alarmConfidence>
    </FieldDetectionRegion>
  </FieldDetectionRegionList>
  <isSupportMultiScene opt="true,false">
    <!--ro, req, bool, whether it supports multiple scene detection, attr:opt{req, string}-->true
  </isSupportMultiScene>
  <isSupportHumanMisinfoFilter>
    <!--ro, opt, bool, whether it supports false human body alarm prevention, desc:if this function is supported, this node will be returned and its value is true; otherwise, this node will not be returned-->true
  </isSupportHumanMisinfoFilter>
  <isSupportVehicleMisinfoFilter>
    <!--ro, opt, bool, whether it supports false vehicle alarm prevention, desc:if this function is supported, this node will be returned and its value is true; otherwise, this node will not be returned-->true
  </isSupportVehicleMisinfoFilter>
  <isSupportTargetMultiSelect>
    <!--ro, opt, bool, whether it supports selecting multiple detection targets, desc:if this function is supported, this node will be returned and its value is true; otherwise, this node will not be returned-->true
  </isSupportTargetMultiSelect>
  <isSupportAllDayUpload>
    <!--ro, opt, bool, whether it supports uploading all-day event-->true
  </isSupportAllDayUpload>
</FieldDetection>

```

### 16.10.1.20 Get parameters of all intrusion detection regions of a specific channel

#### Request URL

GET /ISAPI/Smart/FieldDetection/<channelID>/regions

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FieldDetectionRegionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, subType:object, attr:version{req, string, protocolVersion}-->
  <FieldDetectionRegion>
    <!--ro, opt, object, intrusion detection-->
    <id>
      <!--ro, req, string, ID-->1
    </id>
    <enabled>
      <!--ro, req, bool-->true
    </enabled>
    <sensitivityLevel>
      <!--ro, req, int, sensitivity, range:[0,100]-->1
    </sensitivityLevel>
    <timeThreshold>
      <!--ro, req, int, time threshold-->1
    </timeThreshold>
    <detectionTarget>
      <!--ro, opt, enum, detection target, subType:string, desc:"all", "human", "vehicle", "others"-->all
    </detectionTarget>
    <RegionCoordinatesList>
      <!--ro, opt, array, List of area coordinates, subType:object, range:[0,4], desc:rectangle-->
      <RegionCoordinates>
        <!--ro, opt, object, area coordinates, desc:the origin is the lower-left corner of the screen-->
        <positionX>
          <!--ro, req, int, X-coordinate, range:[0,1000]-->1
        </positionX>
        <positionY>
          <!--ro, req, int, Y-coordinate, range:[0,1000]-->1
        </positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
    <alarmConfidence>
      <!--ro, opt, enum, confidence of alarm notification, subType:string-->low
    </alarmConfidence>
  </FieldDetectionRegion>
</FieldDetectionRegionList>

```

#### 16.10.1.21 Set parameters of all intrusion detection regions of a specified channel

##### Request URL

PUT /ISAPI/Smart/FieldDetection/<channelID>/regions

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FieldDetectionRegionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!-- opt, array, parameters of all intrusion detection regions, subType:object, attr:version{req, string, protocolVersion}-->
  <FieldDetectionRegion>
    <!-- opt, object, intrusion detection regions-->
    <id>
      <!--req, string, No.-->1
    </id>
    <enabled>
      <!--req, bool, whether to enable the function-->true
    </enabled>
    <sensitivityLevel>
      <!--req, int, sensitivity-->1
    </sensitivityLevel>
    <timeThreshold>
      <!--req, int, time threshold-->1
    </timeThreshold>
    <detectionTarget>
      <!--opt, enum, detection target, subType:string, desc:"all", "human", "vehicle", "others" (targets except human bodies and vehicles). The value "all" is mutually exclusive with "others"-->all
    </detectionTarget>
    <RegionCoordinatesList>
      <!-- opt, array, region coordinate List, subType:object, range:[0,4], desc:rectangle-->
      <RegionCoordinates>
        <!--opt, object, region coordinates, desc:the origin is the lower-left corner of the screen-->
        <position>
          <!--req, int, X-coordinate, range:[0,1000]-->1
        </positionX>
        <positionY>
          <!--req, int, Y-coordinate, range:[0,1000]-->1
        </positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
    <alarmConfidence>
      <!--opt, enum, alarm confidence, subType:string, desc:"Low" (default value), "mediumLow", "mediumHigh", "high"-->low
    </alarmConfidence>
  </FieldDetectionRegion>
</FieldDetectionRegionList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.1.22 Get parameters of an intrusion detection region of a specific channel

#### Request URL

GET /ISAPI/Smart/FieldDetection/<channelID>/regions/<regionID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FieldDetectionRegion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string, ID-->1
  </id>
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
  <sensitivityLevel>
    <!--ro, req, int, sensitivity, range:[0,100]-->1
  </sensitivityLevel>
  <timeThreshold>
    <!--ro, req, int, time threshold-->1
  </timeThreshold>
  <detectionTarget>
    <!--ro, opt, enum, detection target, subType:string, desc:"all", "human", "vehicle", "others"-->all
  </detectionTarget>
  <RegionCoordinatesList>
    <!--ro, opt, object, list of area coordinates-->
    <RegionCoordinates>
      <!--ro, opt, object, area coordinates-->
      <positionX>
        <!--ro, req, int, X-coordinate-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y-coordinate-->1
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
  <alarmConfidence>
    <!--ro, opt, enum, confidence of alarm notification, subType:string, desc:confidence of alarm notification: "Low", "mediumLow", "mediumHigh", "high", "Low"-->low
  </alarmConfidence>
</FieldDetectionRegion>

```

### 16.10.1.23 Set parameters of an intrusion detection region of a specified channel

#### Request URL

PUT /ISAPI/Smart/FieldDetection/<channelID>/regions/<regionID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<FieldDetectionRegion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, parameters of an intrusion detection region of a specified channel, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string, No.-->1
  </id>
  <enabled>
    <!--req, bool, whether to enable the function-->true
  </enabled>
  <sensitivityLevel>
    <!--req, int, sensitivity-->1
  </sensitivityLevel>
  <timeThreshold>
    <!--req, int, time threshold-->1
  </timeThreshold>
  <detectionTarget>
    <!--opt, enum, detection target, subType:string, desc:"all", "human", "vehicle", "others" (targets except human bodies and vehicles). The value "all" is mutually exclusive with "others"-->all
  </detectionTarget>
  <RegionCoordinatesList>
    <!--opt, object, region coordinate list-->
    <RegionCoordinates>
      <!--opt, object, region coordinates-->
      <positionX>
        <!--req, int, X-coordinate-->1
      </positionX>
      <positionY>
        <!--req, int, Y-coordinate-->1
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
  <alarmConfidence>
    <!--opt, enum, alarm confidence, subType:string, desc:"Low" (default value), "mediumLow", "mediumHigh", "high"-->low
  </alarmConfidence>
</FieldDetectionRegion>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.1.24 Get configuration capability of intrusion detection by channel

#### Request URL

GET /ISAPI/Smart/FieldDetection/<channelID>/regions/<regionID>/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<FieldDetectionRegion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string, detection area ID-->1
  </id>
  <enabled>
    <!--ro, req, bool, whether to enable the function-->true
  </enabled>
  <sensitivityLevel min="1" max="100">
    <!--ro, req, int, sensitivity, range:[1,100], attr:min{req, int},max{req, int}-->1
  </sensitivityLevel>
  <timeThreshold min="1" max="100">
    <!--ro, req, int, range:[1,100], attr:min{req, int},max{req, int}-->1
  </timeThreshold>
  <objectOccupation min="1" max="100">
    <!--ro, req, int, range:[1,100], attr:min{req, int},max{req, int}-->1
  </objectOccupation>
  <maxRegionCoordinatesNum>
    <!--ro, req, int, the maximum number of coordinates-->1
  </maxRegionCoordinatesNum>
  <minRegionCoordinatesNum>
    <!--ro, req, int, the minimum number of coordinates-->1
  </minRegionCoordinatesNum>
  <detectionTarget opt="all,human,vehicle,others">
    <!--ro, opt, enum, detection target, subType:string, attr:opt{req, string}-->all
  </detectionTarget>
  <RegionCoordinatesList>
    <!--ro, opt, array, area coordinates list, subType:object, range:[0,4], desc:rectangle-->
    <RegionCoordinates>
      <!--ro, opt, object, area coordinates, desc:the origin is the lower-left corner of the screen-->
      <positionX>
        <!--ro, req, int, X-coordinate, range:[0,1000]-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y-coordinate, range:[0,1000]-->1
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
  <humanMisinfoFilterEnabled opt="true,false">
    <!--ro, opt, bool, false alarm reduction of human body, attr:opt{req, string}-->true
  </humanMisinfoFilterEnabled>
  <vehicleMisinfoFilterEnabled opt="true,false">
    <!--ro, opt, bool, false alarm reduction of vehicle, attr:opt{req, string}-->true
  </vehicleMisinfoFilterEnabled>
  <priorities opt="low,middle,high">
    <!--ro, opt, enum, priority, subType:string, attr:opt{req, string}, desc:priority: "Low,middle,high"-->low
  </priorities>
</FieldDetectionRegion>

```

### 16.10.1.25 Intrusion detection

#### EventType:fielddetection

```

<?xml version="1.0" encoding="UTF-8"?>

<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, event alarm, attr:version{opt, string, protocolVersion}-->
  <ipAddress>
    <!--ro, req, string, IPv4 address of the device that triggers the alarm-->172.6.64.7
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPv6 address of the device that triggers the alarm-->1080:0:0:0:8:800:200C:417A
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, communication port No. of the device that triggers the alarm-->80
  </portNo>
  <protocol>
    <!--ro, opt, enum, transmission communication protocol type, subType:string, desc:when ISAPI protocol is transmitted via HCNetSDK, the channel No. is the video channel No. of private protocol. When ISAPI protocol is transmitted via EZ protocol, the channel No. is the video channel No. of EZ protocol. When ISAPI protocol is transmitted via ISUP, the channel No. is the video channel No. of ISUP-->HTTP
  </protocol>
  <macAddress>
    <!--ro, opt, string, MAC address-->01:17:24:45:D9:F4
  </macAddress>
  <dynChannelID>
    <!--ro, opt, string, digital channel No.-->test
  </dynChannelID>
  <channelID>
    <!--ro, opt, int, channel No. of the device that triggers the alarm, desc:video channel No. that triggers the alarm-->1
  </channelID>
  <relatedChannelList>
    <!--ro, opt, string, list of alarm related channels, which are of the same camera with channelID, desc:this parameter is used for live view or playback on the platform; multiple channel No.s are separated by commas-->1,2,3
  </relatedChannelList>
  <dateTime>
    <!--ro, req, datetime, alarm trigger time-->2004-05-03T17:30:08+08:00
  </dateTime>
</EventNotificationAlert>

```

```

</dateTime>
<activePostCount>
  <!--ro, opt, int, times that the same alarm has been uploaded, desc:event triggering frequency-->1
</activePostCount>
<eventType>
  <!--ro, req, enum, event type, subType:string, desc:"fielddetection" (intrusion detection)-->fielddetection
</eventType>
<eventState>
  <!--ro, req, enum, event status, subType:string, desc:for durative event: "active" (valid), "inactive" (invalid)-->active
</eventState>
<eventDescription>
  <!--ro, req, string, event description-->test
</eventDescription>
<deviceID>
  <!--ro, opt, string, device ID, desc:it should be returned for ISUP alarms, e.g., test0123 (Ehome2.0, Ehome4.0, and ISUP5.0)-->12345
</deviceID>
<DetectionRegionList>
  <!--ro, opt, array, detection area List, subType:object-->
<DetectionRegionEntry>
  <!--ro, opt, object, detection area-->
<regionID>
  <!--ro, req, string, area ID-->test
</regionID>
<sensitivityLevel>
  <!--ro, opt, int, sensitivity, range:[0,100]-->1
</sensitivityLevel>
<RegionCoordinatesList>
  <!--ro, opt, array, area coordinates, subType:object, range:[0,4], desc:rectangle-->
<RegionCoordinates>
  <!--ro, opt, object, area point coordinates, desc:the origin is the upper-left corner of the screen-->
<positionX>
  <!--ro, req, int, X-coordinate, range:[0,1000]-->50
</positionX>
<positionY>
  <!--ro, req, int, Y-coordinate, range:[0,1000]-->50
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
<detectionTarget>
  <!--ro, opt, enum, target type, subType:string, desc:"human", "vehicle", "others"-->human
</detectionTarget>
<TargetRect>
  <!--ro, opt, object, target area, desc:the origin is the upper-left corner of the screen-->
<X>
  <!--ro, req, float, target X-coordinate, range:[0.000,1.000]-->0.000
</X>
<Y>
  <!--ro, req, float, target Y-coordinate, range:[0.000,1.000]-->0.000
</Y>
<width>
  <!--ro, req, float, target width, range:[0.000,1.000]-->1.000
</width>
<height>
  <!--ro, req, float, target height, range:[0.000,1.000]-->1.000
</height>
</TargetRect>
<sceneID>
  <!--ro, opt, int, scene ID-->1
</sceneID>
</DetectionRegionEntry>
</DetectionRegionList>
<channelName>
  <!--ro, opt, string, channel name, range:[1,64]-->test
</channelName>
<Extensions>
  <!--ro, opt, object, extended information-->
<serialNumber>
  <!--ro, opt, string, serial No.-->test
</serialNumber>
<eventPush>
  <!--ro, opt, string, event notification-->test
</eventPush>
</Extensions>
<detectionPictureTransType>
  <!--ro, opt, enum, picture transmission mode, subType:string, desc:"url", "binary", "base64"-->url
</detectionPictureTransType>
<detectionPicturesNumber>
  <!--ro, opt, int, number of pictures contained in the alarm received in arming mode, desc:this field will not be returned when there is no picture contained in the alarm-->1
</detectionPicturesNumber>
<bkgUrl>
  <!--ro, opt, string, background picture URL, desc:background picture URL-->test
</bkgUrl>
<URLCertificationType>
  <!--ro, opt, enum, picture URL authentication method, subType:string, desc:authentication mode of picture URL: "no"-no authentication, "digest"-digest authentication (it is used for local picture URL returned by NVR or DVR)-->no
</URLCertificationType>
<pId>
  <!--ro, opt, string, picture ID, desc:the field value is the picture Content-ID when the value of detectionPictureTransType is "binary"-->test
</pId>
<visibleLightURL>
  <!--ro, opt, string, visible light picture URL-->test
</visibleLightURL>
<+normal1IDI>

```

```

<!--ro, opt, string, thermal imaging picture URL-->test
</thermalURL>
<targetAttrs>
  <!--ro, opt, object, transmitted parameters-->{ /*可选,透传字段,推荐使用以下三个字段*/ "deviceId": "1", /*可选,设备ID, string类型, 最大长度为64*/
"deviceChannel": 1, /*可选,设备通道号, integer32类型*/ "deviceName": "shebei", /*可选,设备名称, string类型, 最大长度为128*/ }
</targetAttrs>
<TaskInfo>
  <!--ro, opt, object, task information submitted by data center products-->
<taskID>
  <!--ro, req, string, task ID, range:[0,64]-->test
</taskID>
<algorithmType>
  <!--ro, opt, enum, algorithm type, subType:int, desc:1-"hightQualityVehicle" (high-quality vehicle),2-"hightPerformanceVehicle" (high-performance vehicle),3-"personStructModel" (person structural modeling),4-"vehicleAndPersonStructModel" (person and vehicle structural modeling),5-"videoFaceDetect" (video face detection),6-"videoFaceDetectModel" (video face detection and modeling),7-"smart" (face detection (output frame without recognizing properties)+human body detection (output properties and frame)+behavior detection)-->1
</algorithmType>
<taskName>
  <!--ro, opt, string, task name-->test
</taskName>
<algorithmID>
  <!--ro, opt, string, algorithm package ID-->test
</algorithmID>
<monitorPointID>
  <!--ro, opt, string, camera No.-->test
</monitorPointID>
<monitorPointName>
  <!--ro, opt, string, camera name-->test
</monitorPointName>
<streamType>
  <!--ro, opt, enum, stream type, subType:string, desc:"realtime" (real-time stream), "historyvideo" (history stream), "localvideo" (local stream)-->realtime
</streamType>
<ruleID>
  <!--ro, opt, string, rule ID, desc:it is used to distinguish different tasks with the same name, and the ruleID in the event uploading message indicates that the event is related to this task-->test
</ruleID>
<ruleCustomName>
  <!--ro, req, string, user-defined event name, range:[0,128], desc:it is defined by the caller, and it is included in the message when the alarm is uploaded. Up to 128 characters are allowed-->test
</ruleCustomName>
<algorithmInfo>
  <!--ro, opt, object, algorithm version information, desc:this field is required for secondary analysis-->
<algorithmId>
  <!--ro, opt, string, algorithm package ID-->test
</algorithmId>
<algoVendorId>
  <!--ro, opt, string, manufacturer ID-->test
</algoVendorId>
<algoVersionId>
  <!--ro, opt, string, algorithm version No.-->test
</algoVersionId>
<algoChipType>
  <!--ro, opt, string, chip type-->test
</algoChipType>
<algoModelVersion>
  <!--ro, opt, string, model version-->test
</algoModelVersion>
</algorithmInfo>
</TaskInfo>
<timeStamp>
  <!--ro, opt, string, timestamp of the frame, which triggers the alarm-->test
</timeStamp>
<targetSpeed>
  <!--ro, opt, int-->0
</targetSpeed>
<targetDistance>
  <!--ro, opt, int, unit:m-->2
</targetDistance>
<triggerType>
  <!--ro, opt, enum, subType:string-->video
</triggerType>
<uid>
  <!--ro, opt, string, the unique ID of event uploading, range:[0,64], desc:it can be composed of time (accurate to the millisecond) and random digits-->test
</uid>
<isDataRetransmission>
  <!--ro, opt, bool, whether it is a resending data (if the real-time detection data upload fails due to the network condition, the data will be uploaded again after the device recovers.)-->true
</isDataRetransmission>
<ruleID>
  <!--ro, opt, int, rule No., range:[0,7], desc:rule No.-->0
</ruleID>
<ruleName>
  <!--ro, opt, string, rule name, range:[0,32], desc:rule name-->test
</ruleName>
<nonTargetRate>
  <!--ro, opt, int, range:[1,100], unit:%-->1
</nonTargetRate>
<visibleLightAbsoluteHigh>
  <!--ro, opt, object-->
<elevation>
  <!--ro, opt, float, range:[-90.00,270.00], step:0.001-->0.00

```

```

</elevation>
<azimuth>
  <!--ro, opt, float, range:[0,360.00], step:0.001-->0.00
</azimuth>
<absoluteZoom>
  <!--ro, opt, float, zooming ratio, range:[0,10000.00]-->0.00
</absoluteZoom>
<focus>
  <!--ro, opt, int, range of the focus, range:[1,100000], step:1-->1
</focus>
</visibleLightAbsoluteHigh>
<thermalAbsoluteHigh>
<!--ro, opt, object-->
<elevation>
  <!--ro, opt, float, range:[-90.00,270.00], step:0.001-->0.00
</elevation>
<azimuth>
  <!--ro, opt, float, range:[0,360.00], step:0.001-->0.00
</azimuth>
<absoluteZoom>
  <!--ro, opt, float, zooming ratio, range:[0,10000.00]-->0.00
</absoluteZoom>
<focus>
  <!--ro, opt, int, range of the focus, range:[1,100000], step:1-->1
</focus>
</thermalAbsoluteHigh>
<deviceLocation>
<longitudeType>
  <!--ro, req, enum, subType:string-->W
</longitudeType>
<latitudeType>
  <!--ro, req, enum, subType:string, desc:"S"-south, "N"-north-->S
</latitudeType>
<longitude>
  <!--ro, req, object-->
<degree>
  <!--ro, req, int, degree(s), range:[0,180]-->1
</degree>
<minute>
  <!--ro, req, float, minute(s), range:[0,60]-->0.0
</minute>
<sec>
  <!--ro, req, float, range:[0,60]-->0.0
</sec>
</longitude>
<latitude>
  <!--ro, req, object-->
<degree>
  <!--ro, req, int, range:[0,90]-->1
</degree>
<minute>
  <!--ro, req, float, minute(s), range:[0,60]-->0.0
</minute>
<sec>
  <!--ro, req, float, range:[0,60]-->0.0
</sec>
</latitude>
</deviceLocation>
<laserRanging>
  <!--ro, opt, int, range:[1,1000], step:1, unit:m-->1
</laserRanging>
</EventNotificationAlert>

```

Parameter Name	Parameter Value	Parameter Type(Content-Type)	Content-ID	File Name	Description
fielddetection	[Message content]	application/xml	--	--	--
visibleLightImage	[Binary picture data]	image/jpeg	visibleLightImage	visibleLightImage.jpg	--
thermallImage	[Binary picture data]	image/jpeg	thermallImage	thermallImage.jpg	--
targetImage	[Binary picture data]	image/jpeg	targetImage	targetImage.jpg	--

**Note:** The protocol is transmitted in form format. See Chapter 4.5.1.4 for form framework description, as shown in the instance below.

```
--<frontier>
Content-Disposition: form-data; name=Parameter Name;filename=File Name
Content-Type: Parameter Type
Content-Length: ****
Content-ID: Content ID
Parameter Value
```

- Parameter Name: the name property of Content-Disposition in the header of form unit; it refers to the form unit name.
- Parameter Type (Content-Type): the Content-Type property in the header of form unit.
- File Name (filename): the filename property of Content-Disposition of form unit Headers. It exists only when the transmitted data of form unit is file, and it refers to the file name of form unit body.
- Parameter Value: the body content of form unit.

## 16.10.2 Line Crossing Detection

### 16.10.2.1 Get the arming schedules of line crossing detection of all channels

#### Request URL

GET /ISAPI/Event/schedules/lineDetections

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<LineDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, the arming schedules of line crossing detection of all channels, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--ro, opt, object, arming schedule information-->
    <id>
      <!--ro, req, string, ID-->test
    </id>
    <eventType>
      <!--ro, opt, enum, event type, subType:string-->lineDetection
    </eventType>
    <videoInputChannelID>
      <!--ro, opt, string, video input channel ID-->1
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--ro, req, object, arming time list, attr:size{req, int}-->
      <TimeBlock>
        <!--ro, opt, object, arming time-->
        <dayOfWeek>
          <!--ro, opt, enum, day of a week, subType:int (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->1
        </dayOfWeek>
        <TimeRange>
          <!--ro, req, object, time range-->
          <beginTime>
            <!--ro, req, time, start time-->00:00:00
          </beginTime>
          <endTime>
            <!--ro, req, time, end time-->00:00:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>
</LineDetectionScheduleList>
```

### 16.10.2.2 Set line crossing detection arming schedules of all channels

#### Request URL

PUT /ISAPI/Event/schedules/lineDetections

#### Query Parameter

None

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<LineDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, object, refer to the message XML_Schedule for details, attr:version{req, string, protocolVersion}-->
    <Schedule>
        <!--opt, object, arming information-->
        <id>
            <!--req, string-->test
        </id>
        <eventType>
            <!--opt, enum, event type, subType:string-->lineDetection
        </eventType>
        <videoInputChannelID>
            <!--opt, string, video input channel ID-->1
        </videoInputChannelID>
        <TimeBlockList size="8">
            <!--req, object, attr:size{req, int}-->
            <TimeBlock>
                <!--opt, object-->
                <dayOfWeek>
                    <!--opt, enum, day of the week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->1
                </dayOfWeek>
                <TimeRange>
                    <!--req, object, time range-->
                    <beginTime>
                        <!--req, time, start time-->00:00:00
                    </beginTime>
                    <endTime>
                        <!--req, time, end time-->00:00:00
                    </endTime>
                </TimeRange>
            </TimeBlock>
        </TimeBlockList>
    </Schedule>
</LineDetectionScheduleList>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
    <requestURL>
        <!--ro, req, string, request URL-->null
    </requestURL>
    <statusCode>
        <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    </statusCode>
    <statusString>
        <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusString>
    <subStatusCode>
        <!--ro, req, string, sub status code, desc:sub status code description-->OK
    </subStatusCode>
</ResponseStatus>
```

### 16.10.2.3 Get the arming schedule of line crossing detection for a specified channel

#### Request URL

GET /ISAPI/Event/schedules/lineDetections/lineDetections\_video<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

None

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->test
  </id>
  <eventType>
    <!--ro, opt, enum, subType:string-->lineDetection
  </eventType>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--ro, req, object, attr:size{req, int}-->
    <TimeBlock>
      <!--ro, opt, object-->
      <dayOfWeek>
        <!--ro, opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--ro, req, object-->
        <beginTime>
          <!--ro, req, time-->00:00:00
        </beginTime>
        <endTime>
          <!--ro, req, time-->00:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>

```

#### 16.10.2.4 Set the arming schedule of line crossing detection for a specified channel

##### Request URL

PUT /ISAPI/Event/schedules/lineDetections/lineDetections\_video<channelID>

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->test
  </id>
  <eventType>
    <!--opt, enum, subType:string-->lineDetection
  </eventType>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--req, object, attr:size{req, int}-->
    <TimeBlock>
      <!--opt, object-->
      <dayOfWeek>
        <!--opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--req, object-->
        <beginTime>
          <!--req, time-->00:00:00
        </beginTime>
        <endTime>
          <!--req, time-->00:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.2.5 Set the arming schedule of radar intrusion detection for all channels

#### Request URL

PUT /ISAPI/Event/schedules/radarLineDetections

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RadarLineDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, subType:object, attr:version{req, string, protocolVersion}-->
</RadarLineDetectionScheduleList>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.2.6 Get arming schedules of all radar channels for line crossing detection

#### Request URL

GET /ISAPI/Event/schedules/radarLineDetections

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RadarLineDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, list of arming schedules, subType:object, attr:version{req, string, protocolVersion}-->
</RadarLineDetectionScheduleList>

```

### 16.10.2.7 Get the arming schedule of radar line crossing detection for a specified channel

## Request URL

GET /ISAPI/Event/schedules/radarLineDetections/radarLineDetection-<radarChannelID>

## Query Parameter

Parameter Name	Parameter Type	Description
radarChannelID	string	--

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->radarLineDetection-1
  </id>
  <eventType>
    <!--ro, opt, enum, subType:string-->radarLineDetection
  </eventType>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--ro, req, array, subType:object, attr:size{opt, int}-->
    <TimeBlock>
      <!--ro, opt, object-->
      <dayOfWeek>
        <!--ro, opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--ro, req, object-->
        <beginTime>
          <!--ro, req, time-->10:00:00
        </beginTime>
        <endTime>
          <!--ro, req, time-->10:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>
```

## 16.10.2.8 Set the arming schedule of radar line crossing detection for a specified channel

### Request URL

PUT /ISAPI/Event/schedules/radarLineDetections/radarLineDetection-<radarChannelID>

### Query Parameter

Parameter Name	Parameter Type	Description
radarChannelID	string	--

### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->radarLineDetection-1
  </id>
  <eventType>
    <!--opt, enum, subType:string-->radarLineDetection
  </eventType>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--req, array, subType:object, attr:size{opt, int}-->
    <TimeBlock>
      <!--opt, object-->
      <dayOfWeek>
        <!--opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--req, object-->
        <beginTime>
          <!--req, time-->10:00:00
        </beginTime>
        <endTime>
          <!--req, time-->10:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.2.9 Set alarm linkage parameters of line crossing detection by channel

#### Request URL

PUT /ISAPI/Event/triggers/LineDetection-<channelID>/notifications

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, Linkage action List, subType:object, attr:version{req, string, protocolVersion}-->
  <EventTriggerNotification>
    <!--opt, object, Linkage action-->
    <id>
      <!--req, string, No.-->test
    </id>
    <notificationMethod>
      <!--req, enum, Linkage action, subType:string, desc:"email", "IM" (instant messaging), "IO" (I/O output), "syslog" (system Log), "HTTP" (uploading via HTTP), "FTP" (uploading via FTP), "beep" (buzzer), "ptz" (PTZ), "record" (recording), "monitorAlarm" (alarm on monitor), "center" (uploading to the center), "LightAudioAlarm" (light and audio alarm), "focus" (focus), "trace" (tracking), "whiteLight" (white Light), "audio" (audio), "supplementLight" (supplement Light)-->email
    </notificationMethod>
    <notificationRecurrence>
      <!--opt, enum, subType:string-->beginning
    </notificationRecurrence>
    <videoInputID>
      <!--opt, string, video channel ID-->1
    </videoInputID>
  </EventTriggerNotification>
</EventTriggerNotificationList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->0
  </subStatusCode>
</ResponseStatus>

```

### 16.10.2.10 Get alarm linkage parameters of a line crossing detection channel

#### Request URL

GET /ISAPI/Event/triggers/LineDetection-<channelID>/notifications

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, subType:object, attr:version{req, string, protocolVersion}-->
  <EventTriggerNotification>
    <!--ro, opt, object, linkage type-->
    <id>
      <!--ro, req, string-->test
    </id>
    <notificationMethod>
      <!--ro, req, enum, linkage type, subType:string-->email
    </notificationMethod>
    <notificationRecurrence>
      <!--ro, opt, enum, subType:string-->beginning
    </notificationRecurrence>
    <videoInputID>
      <!--ro, opt, string-->1
    </videoInputID>
  </EventTriggerNotification>
</EventTriggerNotificationList>

```

### 16.10.2.11 Get the linkage parameters of line crossing detection event for a specified channel

#### Request URL

GET /ISAPI/Event/triggers/LineDetection-<channelID> ?triggersEnabledStatus=<triggersEnabledStatus>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
triggersEnabledStatus	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->1
  </id>
  <eventType>
    <!--ro, req, enum, subType:string-->LineDetection
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subtype:object-->
    <EventTriggerNotification>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->test
      </id>
      <notificationMethod>
        <!--ro, req, enum, subType:string-->email
      </notificationMethod>
      <notificationRecurrence>
        <!--ro, opt, enum, subType:string-->beginning
      </notificationRecurrence>
      <videoInputID>
        <!--ro, opt, string-->1
      </videoInputID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

### 16.10.2.12 Set the linkage parameters of line crossing detection event for a specified channel

#### Request URL

PUT /ISAPI/Event/triggers/LineDetection-<channelID>?triggersEnabledStatus=<triggersEnabledStatus>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
triggersEnabledStatus	string	--

#### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->1
  </id>
  <eventType>
    <!--req, enum, subType:string-->LineDetection
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
    <EventTriggerNotification>
      <!--opt, object-->
      <id>
        <!--req, string-->test
      </id>
      <notificationMethod>
        <!--req, enum, subType:string-->email
      </notificationMethod>
      <notificationRecurrence>
        <!--opt, enum, subType:string-->beginning
      </notificationRecurrence>
      <videoInputID>
        <!--opt, string-->1
      </videoInputID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>
```

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.10.2.13 Set the linkage parameters of radar line crossing detection event for a specified channel

#### Request URL

PUT /ISAPI/Event/triggers/radarLineDetection-<radarChannelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
radarChannelID	string	--

### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->radarLineDetection-1
  </id>
  <eventType>
    <!--req, enum, subType:string-->radarLineDetection
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
    <EventTriggerNotification>
      <!--opt, object-->
      <id>
        <!--req, string-->test
      </id>
      <notificationMethod>
        <!--req, enum, subType:string-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--opt, enum, subType:string-->beginning
      </notificationRecurrence>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

## 16.10.2.14 Get the linkage parameters of radar line crossing detection event for a specified channel

### Request URL

GET /ISAPI/Event/triggers/radarLineDetection-<radarChannelID>

### Query Parameter

Parameter Name	Parameter Type	Description
radarChannelID	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->radarLineDetection-1
  </id>
  <eventType>
    <!--ro, req, enum, subType:string-->radarLineDetection
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subType:object-->
    <EventTriggerNotification>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->test
      </id>
      <notificationMethod>
        <!--ro, req, enum, subType:string-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--ro, opt, enum, subType:string-->beginning
      </notificationRecurrence>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>
```

#### 16.10.2.15 Set line crossing detection parameters of all channels

##### Request URL

PUT /ISAPI/Smart/LineDetection

##### Query Parameter

None

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<LineDetectionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, subType:object, attr:version{req, string, protocolVersion}-->
  <LineDetection>
    <!--opt, object-->
    <id>
      <!--req, string-->test
    </id>
    <enabled>
      <!--req, bool-->true
    </enabled>
    <normalizedScreenSize>
      <!--opt, object-->
      <normalizedScreenWidth>
        <!--req, int-->1
      </normalizedScreenWidth>
      <normalizedScreenHeight>
        <!--req, int-->1
      </normalizedScreenHeight>
    </normalizedScreenSize>
    <LineItemList size="4">
      <!--opt, array, subType:object, attr:size{req, int}-->
      <LineItem>
        <!--opt, object-->
        <id>
          <!--req, string-->1
        </id>
        <enabled>
          <!--req, bool, whether to enable the function-->true
        </enabled>
        <sensitivityLevel>
          <!--req, int, detection sensitivity, range:[0,100]-->1
        </sensitivityLevel>
        <directionSensitivity>
          <!--opt, enum, detection direction, subType:string, desc:detection direction-->any
        </directionSensitivity>
        <CoordinatesList>
          <!--opt, array, subType:object, range:[0,2]-->
          <Coordinates>
            <!--opt, object, the origin is the lower-left corner of the screen, desc:the origin is the lower-left corner of the screen-->
            <positionX>
              <!--req, int, x-coordinate, range:[0,1000]-->1
            </positionX>
            <positionY>
              <!--req, int, y-coordinate, range:[0,1000]-->1
            </positionY>
          </Coordinates>
        </CoordinatesList>
        <detectionTarget>
          <!--opt, enum, detection target, subType:string, desc:detection target-->all
        </detectionTarget>
        <alarmConfidence>
          <!--opt, enum, subType:string-->low
        </alarmConfidence>
        <LineItem>
      </LineItemList>
      <recogRuleType>
        <!--opt, enum, subType:string-->vectorMode
      </recogRuleType>
    </LineDetection>
  </LineDetectionList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code description-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.2.16 Get line crossing detection parameters of all channels

## Request URL

GET /ISAPI/Smart/LineDetection

## Query Parameter

None

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<LineDetectionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, subType:object, attr:version{req, string, protocolVersion}-->
  <LineDetection>
    <!--ro, opt, object-->
    <id>
      <!--ro, req, string, channel No.-->test
    </id>
    <enabled>
      <!--ro, req, bool, whether to enable the function-->true
    </enabled>
    <normalizedScreenSize>
      <!--ro, opt, object-->
      <normalizedScreenWidth>
        <!--ro, req, int-->1
      </normalizedScreenWidth>
      <normalizedScreenHeight>
        <!--ro, req, int-->1
      </normalizedScreenHeight>
    </normalizedScreenSize>
    <lineItemList size="4">
      <!--ro, opt, array, subType:object, attr:size{req, int}-->
      <LineItem>
        <!--ro, opt, object-->
        <id>
          <!--ro, req, string, ID-->1
        </id>
        <enabled>
          <!--ro, req, bool, whether to enable the function-->true
        </enabled>
        <sensitivityLevel>
          <!--ro, req, int, detection sensitivity, range:[0,100]-->1
        </sensitivityLevel>
        <directionSensitivity>
          <!--ro, opt, enum, detection direction, subType:string, desc:detection direction-->any
        </directionSensitivity>
        <CoordinatesList>
          <!--ro, opt, array, rectangle, subType:object, range:[0,2], desc:rectangle-->
          <Coordinates>
            <!--ro, opt, object, coordinates, desc:the origin is the lower-left corner of the screen-->
            <positionX>
              <!--ro, req, int, X-coordinate, range:[0,1000]-->1
            </positionX>
            <positionY>
              <!--ro, req, int, Y-coordinate, range:[0,1000]-->1
            </positionY>
          </Coordinates>
        </CoordinatesList>
        <detectionTarget>
          <!--ro, opt, enum, detection target, subType:string, desc:detection target-->all
        </detectionTarget>
        <alarmConfidence>
          <!--ro, opt, enum, subType:string-->low
        </alarmConfidence>
      </LineItem>
    </lineItemList>
    <recogRuleType>
      <!--ro, opt, enum, subType:string-->vectorMode
    </recogRuleType>
  </LineDetection>
</LineDetectionList>
```

## 16.10.2.17 Set line crossing detection parameters by channel

### Request URL

PUT /ISAPI/Smart/LineDetection/<channelID>

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<LineDetection xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, object, line crossing detection parameters, attr:version{req, string, protocolVersion}-->
</LineDetection>
```

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
    <requestURL>
        <!--ro, req, string, request URL-->null
    </requestURL>
    <statusCode>
        <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6
        (Invalid XML Content), 7 (Reboot Required)-->0
    </statusCode>
    <statusString>
        <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format",
        "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusString>
    <subStatusCode>
        <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
</ResponseStatus>
```

## 16.10.2.18 Get configuration capability of line crossing detection by channel

### Request URL

GET /ISAPI/Smart/LineDetection/<channelID>/capabilities

### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<LineDetection xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string, channel No.-->test
  </id>
  <enabled opt="true,false">
    <!--ro, req, bool, whether to enable the function, attr:opt{req, string}-->true
  </enabled>
  <normalizedScreenSize>
    <!--ro, opt, object-->
    <normalizedScreenWidth>
      <!--ro, req, int-->1
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--ro, req, int-->1
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <LineItemList size="4">
    <!--ro, opt, array, subType:object, attr:size{req, int}-->
    <LineItem>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->test
      </id>
      <enabled opt="true,false">
        <!--ro, req, bool, whether to enable the detection, attr:opt{req, string}-->true
      </enabled>
      <sensitivityLevel min="1" max="100">
        <!--ro, req, int, detection sensitivity, attr:min{req, int},max{req, int}-->1
      </sensitivityLevel>
      <directionSensitivity opt="left-right,right-left,any">
        <!--ro, opt, enum, detection direction, subType:string, attr:opt{req, string}-->any
      </directionSensitivity>
      <detectionTarget opt="all,human,vehicle">
        <!--ro, opt, enum, detection target, subType:string, attr:opt{req, string}-->all
      </detectionTarget>
      <alarmConfidence opt="low,mediumLow,mediumHigh,high" def="low">
        <!--ro, opt, enum, subType:string, attr:opt{req, string},def{req, string}-->low
      </alarmConfidence>
    </LineItem>
  </LineItemList>
  <isSupportMultiScene opt="true,false">
    <!--ro, opt, bool, whether supports multiple scene detection, attr:opt{req, string}-->true
  </isSupportMultiScene>
  <recogRuleType>
    <!--ro, opt, enum, subType:string-->vectorMode
  </recogRuleType>
  <isSupportHumanMisinfoFilter>
    <!--ro, opt, bool, whether supports false human body alarm prevention, desc:If this function is supported, this node will be returned and its value is true; otherwise, this node will not be returned.-->true
  </isSupportHumanMisinfoFilter>
  <isSupportVehicleMisinfoFilter>
    <!--ro, opt, bool, whether supports false vehicle alarm prevention, desc:If this function is supported, this node will be returned and its value is true; otherwise, this node will not be returned.-->true
  </isSupportVehicleMisinfoFilter>
  <isSupportTargetMultiSelect>
    <!--ro, opt, bool, whether supports selecting multiple detection targets, desc:If this function is supported, this node will be returned and its value is true; otherwise, this node will not be returned.-->true
  </isSupportTargetMultiSelect>
  <isSupportAllDayUpload>
    <!--ro, opt, bool, whether supports uploading all-day event-->true
  </isSupportAllDayUpload>
</LineDetection>

```

### 16.10.2.19 Get parameters of a detection line of a specific channel

#### Request URL

GET /ISAPI/Smart/LineDetection/<channelID>/lineItem/<ruleID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
ruleID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<LineItem xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
</LineItem>

```

### 16.10.2.20 Set parameters of a detection line of a specified channel

#### Request URL

PUT /ISAPI/Smart/LineDetection/<channelID>/lineItem/<ruleID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
ruleID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<LineItem xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, parameters of a detection line, attr:version{req, string, protocolVersion}-->
</LineItem>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
  <!-->
</ResponseStatus>

```

### 16.10.2.21 Get configuration capability of line crossing detection by channel

#### Request URL

GET /ISAPI/Smart/LineDetection/<channelID>/lineItem/<ruleID>/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
ruleID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<LineItem xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
</LineItem>

```

## 16.10.2.22 Line crossing detection

### EventType:linedetection

```
<?xml version="1.0" encoding="UTF-8"?>

<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, event alarm, attr:version{opt, string, protocolVersion}-->
    <ipAddress>
        <!--ro, req, string, IPv4 address of the device that triggers the alarm-->172.6.64.7
    </ipAddress>
    <ipv6Address>
        <!--ro, opt, string, IPv6 address of the device that triggers the alarm-->1080:0:0:0:8:800:200C:417A
    </ipv6Address>
    <portNo>
        <!--ro, opt, int, communication port No. of the device that triggers the alarm-->80
    </portNo>
    <protocol>
        <!--ro, opt, enum, transmission communication protocol type, subType:string, desc:when ISAPI protocol is transmitted via HCNetSDK, the channel No. is the video channel No. of private protocol. When ISAPI protocol is transmitted via EZ protocol, the channel No. is the video channel No. of EZ protocol. When ISAPI protocol is transmitted via ISUP, the channel No. is the video channel No. of ISUP-->HTTP
    </protocol>
    <macAddress>
        <!--ro, opt, string, MAC address-->01:17:24:45:D9:F4
    </macAddress>
    <dynChannelID>
        <!--ro, opt, string, digital channel No.-->test
    </dynChannelID>
    <channelID>
        <!--ro, opt, int, channel No. of the device that triggers the alarm, desc:video channel No. that triggers the alarm-->1
    </channelID>
    <relatedChannelList>
        <!--ro, opt, string, list of alarm related channels, which are of the same camera with channelID, desc:this parameter is used for live view or playback on the platform; multiple channel No.s are separated by commas-->1,2,3
    </relatedChannelList>
    <dateTime>
        <!--ro, req, datetime, alarm trigger time-->2004-05-03T17:30:08+08:00
    </dateTime>
    <activePostCount>
        <!--ro, opt, int, times that the same alarm has been uploaded, desc:event triggering frequency-->1
    </activePostCount>
    <eventType>
        <!--ro, req, enum, event type, subType:string, desc:"Linedetection" (Line Crossing Detection)-->linedetection
    </eventType>
    <eventState>
        <!--ro, req, enum, event status, subType:string, desc:for durative event: "active" (valid), "inactive" (invalid)-->active
    </eventState>
    <eventDescription>
        <!--ro, req, string, event description-->test
    </eventDescription>
    <deviceID>
        <!--ro, opt, string, device ID, desc:it should be returned for ISUP alarms, e.g., test0123 (Ehome2.0, Ehome4.0, and ISUP5.0)-->12345
    </deviceID>
    <DetectionRegionList>
        <!--ro, opt, array, detection area list, subType:object-->
        <DetectionRegionEntry>
            <!--ro, opt, object, detection area-->
            <regionID>
                <!--ro, req, string, area ID-->test
            </regionID>
            <sensitivityLevel>
                <!--ro, opt, int, sensitivity, range:[0,100]-->1
            </sensitivityLevel>
            <RegionCoordinatesList>
                <!--ro, opt, array, area coordinates, subType:object, range:[0,4], desc:rectangle-->
                <RegionCoordinates>
                    <!--ro, opt, object, area point coordinates, desc:the origin is the upper-left corner of the screen-->
                    <positionX>
                        <!--ro, req, int, X-coordinate, range:[0,1000]-->50
                    </positionX>
                    <positionY>
                        <!--ro, req, int, Y-coordinate, range:[0,1000]-->50
                    </positionY>
                </RegionCoordinates>
            </RegionCoordinatesList>
            <detectionTarget>
                <!--ro, opt, enum, target type, subType:string, desc:"human", "vehicle", "others"-->human
            </detectionTarget>
            <TargetRect>
                <!--ro, opt, object, target area, desc:the origin is the upper-left corner of the screen-->
                <x>
                    <!--ro, req, float, target X-coordinate, range:[0.000,1.000]-->0.000
                </x>
                <y>
                    <!--ro, req, float, target Y-coordinate, range:[0.000,1.000]-->0.000
                </y>
                <width>
                    <!--ro, req, float, target width, range:[0.000,1.000]-->1.000
                </width>
                <height>
                    <!--ro, req, float, target height, range:[0.000,1.000]-->1.000
                </height>
            </TargetRect>
        </DetectionRegionEntry>
    </DetectionRegionList>

```

```

</TargetRect>
<sceneID>
  <!--ro, opt, int, scene ID-->1
</sceneID>
</DetectionRegionEntry>
</DetectionRegionList>
<channelName>
  <!--ro, opt, string, channel name, range:[1,64]-->test
</channelName>
<Extensions>
  <!--ro, opt, object, extended information-->
<serialNumber>
  <!--ro, opt, string, serial No.-->test
</serialNumber>
<eventPush>
  <!--ro, opt, string, event notification-->test
</eventPush>
</Extensions>
<detectionPictureTransType>
  <!--ro, opt, enum, picture transmission mode, subType:string, desc:"url", "binary", "base64"-->url
</detectionPictureTransType>
<detectionPicturesNumber>
  <!--ro, opt, int, number of pictures contained in the alarm received in arming mode, desc:this field will not be returned when there is no picture contained in the alarm-->1
</detectionPicturesNumber>
<bkgUrl>
  <!--ro, opt, string, background picture URL, desc:background picture URL-->test
</bkgUrl>
<URLCertificationType>
  <!--ro, opt, enum, picture URL authentication method, subType:string, desc:authentication mode of picture URL: "no"-no authentication, "digest"-digest authentication (it is used for Local picture URL returned by NVR or DVR)-->no
</URLCertificationType>
<pId>
  <!--ro, opt, string, picture ID, desc:the field value is the picture Content-ID when the value of detectionPictureTransType is "binary"-->test
</pId>
<visibleLightURL>
  <!--ro, opt, string, visible Light picture URL-->test
</visibleLightURL>
<thermalURL>
  <!--ro, opt, string, thermal imaging picture URL-->test
</thermalURL>
<targetAttrs>
  <!--ro, opt, object, transmitted parameters-->{ /*可选,透传字段,推荐使用以下三个字段*/ "deviceId": "1", /*可选,设备ID, string类型, 最大长度为64*/ "deviceChannel": 1, /*可选,设备通道号, integer32类型*/ "deviceName": "shebei", /*可选,设备名称, string类型, 最大长度为128*/ }
</targetAttrs>
<TaskInfo>
  <!--ro, opt, object, task information submitted by data center products-->
<taskID>
  <!--ro, req, string, task ID, range:[0,64]-->test
</taskID>
<algorithmType>
  <!--ro, opt, enum, algorithm type, subType:int, desc:1-"highQualityVehicle" (high-quality vehicle),2-"highPerformanceVehicle" (high-performance vehicle),3-"personStructModel" (person structural modeling),4-"vehicleAndPersonStructModel" (person and vehicle structural modeling),5-"videoFaceDetect" (video face detection),6-"videoFaceDetectModel" (video face detection and modeling),7-"smart" (face detection (output frame without recognizing properties)+human body detection (output properties and frame)+behavior detection)-->1
</algorithmType>
<taskName>
  <!--ro, opt, string, task name-->test
</taskName>
<algorithmID>
  <!--ro, opt, string, algorithm package ID-->test
</algorithmID>
<monitorPointID>
  <!--ro, opt, string, camera No.-->test
</monitorPointID>
<monitorPointName>
  <!--ro, opt, string, camera name-->test
</monitorPointName>
<streamType>
  <!--ro, opt, enum, stream type, subType:string, desc:"realtime" (real-time stream), "historyvideo" (history stream), "Localvideo" (local stream)-->realtime
</streamType>
<ruleID>
  <!--ro, opt, string, rule ID, desc:it is used to distinguish different tasks with the same name, and the ruleID in the event uploading message indicates that the event is related to this task-->test
</ruleID>
<ruleCustomName>
  <!--ro, req, string, user-defined event name, range:[0,128], desc:it is defined by the caller, and it is included in the message when the alarm is uploaded. Up to 128 characters are allowed-->test
</ruleCustomName>
<algorithmInfo>
  <!--ro, opt, object, algorithm version information, desc:this field is required for secondary analysis-->
<algorithmId>
  <!--ro, opt, string, algorithm package ID-->test
</algorithmId>
<algoVendorId>
  <!--ro, opt, string, manufacturer ID-->test
</algoVendorId>
<algoVersionId>
  <!--ro, opt, string, algorithm version No.-->test
</algoVersionId>
<algoChipType>
  <!--ro, opt, string, chip type-->test

```

```

</algoChipType>
<algoModelVersion>
    <!--ro, opt, string, model version-->test
</algoModelVersion>
</algorithmInfo>
</TaskInfo>
<timeStamp>
    <!--ro, opt, string, timestamp of the frame, which triggers the alarm-->test
</timeStamp>
<targetSpeed>
    <!--ro, opt, datetime-->2019-12-30T15:29:23+08:00
</targetSpeed>
<targetDistance>
    <!--ro, opt, int, unit:m-->2
</targetDistance>
<triggerType>
    <!--ro, opt, enum, subType:string-->video
</triggerType>
<uid>
    <!--ro, opt, string, the unique ID of event uploading, range:[0,64], desc:it can be composed of time (accurate to the millisecond) and random digits-->test
</uid>
<isDataRetransmission>
    <!--ro, opt, bool, whether it is a resending data (if the real-time detection data upload fails due to the network condition, the data will be uploaded again after the device recovers.)-->true
</isDataRetransmission>
<ruleID>
    <!--ro, opt, int, rule No., range:[0,7], desc:rule No.-->0
</ruleID>
<ruleName>
    <!--ro, opt, string, rule name, range:[0,32], desc:rule name-->test
</ruleName>
<applicationScene>
    <!--ro, opt, enum, Application Scene, subType:string-->indoor
</applicationScene>
<visibleLightAbsoluteHigh>
    <!--ro, opt, object-->
    <elevation>
        <!--ro, opt, float, range:[-90.00,270.00], step:0.001-->0.00
    </elevation>
    <azimuth>
        <!--ro, opt, float, range:[0,360.00], step:0.001-->0.00
    </azimuth>
    <absoluteZoom>
        <!--ro, opt, float, zooming ratio, range:[0,10000.00]-->0.00
    </absoluteZoom>
    <focus>
        <!--ro, opt, int, range of the focus, range:[1,100000], step:1-->1
    </focus>
</visibleLightAbsoluteHigh>
<thermalAbsoluteHigh>
    <!--ro, opt, object-->
    <elevation>
        <!--ro, opt, float, range:[-90.00,270.00], step:0.001-->0.00
    </elevation>
    <azimuth>
        <!--ro, opt, float, range:[0,360.00], step:0.001-->0.00
    </azimuth>
    <absoluteZoom>
        <!--ro, opt, float, zooming ratio, range:[0,10000.00]-->0.00
    </absoluteZoom>
    <focus>
        <!--ro, opt, int, range of the focus, range:[1,100000], step:1-->1
    </focus>
</thermalAbsoluteHigh>
<deviceLocation>
    <!--ro, opt, object-->
    <longitudeType>
        <!--ro, req, enum, subType:string-->W
    </longitudeType>
    <latitudeType>
        <!--ro, req, enum, subType:string, desc:"S"-south, "N"-north-->S
    </latitudeType>
    <longitude>
        <!--ro, req, object-->
        <degree>
            <!--ro, req, int, degree(s), range:[0,180]-->1
        </degree>
        <minute>
            <!--ro, req, float, minute(s), range:[0,60]-->0.0
        </minute>
        <sec>
            <!--ro, req, float, range:[0,60]-->0.0
        </sec>
    </longitude>
    <latitude>
        <!--ro, req, object-->
        <degree>
            <!--ro, req, int, range:[0,90]-->1
        </degree>
        <minute>
            <!--ro, req, float, minute(s), range:[0,60]-->0.0
        </minute>
        <--->
    </latitude>

```

```

<sec>
  <!--ro, req, float, range:[0,60]-->0.0
</sec>
</latitude>
</deviceLocation>
<laserRanging>
  <!--ro, opt, int, range:[1,10000], step:1, unit:m-->1
</laserRanging>
</EventNotificationAlert>

```

Parameter Name	Parameter Value	Parameter Type(Content-Type)	Content-ID	File Name	Description
linedetection	[Message content]	application/xml	--	--	--
visibleLightImage	[Binary picture data]	image/jpeg	visibleLightImage	visibleLightImage.jpg	--
thermallImage	[Binary picture data]	image/jpeg	thermallImage	thermallImage.jpg	--
targetImage	[Binary picture data]	image/jpeg	targetImage	targetImage.jpg	--

**Note:** The protocol is transmitted in form format. See Chapter 4.5.1.4 for form framework description, as shown in the instance below.

```

--<frontier>
Content-Disposition: form-data; name=Parameter Name;filename=File Name
Content-Type: Parameter Type
Content-Length: ****
Content-ID: Content ID
Parameter Value

```

- Parameter Name: the name property of Content-Disposition in the header of form unit; it refers to the form unit name.
- Parameter Type (Content-Type): the Content-Type property in the header of form unit.
- File Name (filename): the filename property of Content-Disposition of form unit Headers. It exists only when the transmitted data of form unit is file, and it refers to the file name of form unit body.
- Parameter Value: the body content of form unit.

### 16.10.3 Region Entrance Detection

#### 16.10.3.1 Set region entrance detection arming schedules of all channels

##### Request URL

PUT /ISAPI/Event/schedules/regionEntrances

##### Query Parameter

None

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionEntranceScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, refer to the message XML_Schedule for details, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--opt, object-->
    <id>
      <!--req, string-->test
    </id>
    <eventType>
      <!--opt, string, event type-->regionEntrance
    </eventType>
    <videoInputChannelID>
      <!--opt, string, video input channel ID-->test
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--req, object, arming time list, attr:size{req, int}-->
      <TimeBlock>
        <!--opt, object, arming time-->
        <dayOfWeek>
          <!--opt, enum, day of a week, subType:int, desc:day of a week-->1
        </dayOfWeek>
        <TimeRange>
          <!--req, object, time range-->
          <beginTime>
            <!--req, time, start time-->00:00:00
          </beginTime>
          <endTime>
            <!--req, time, end time-->00:00:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>
</RegionEntranceScheduleList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code description-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.3.2 Get arming schedules of all channels for region entrance detection

#### Request URL

GET /ISAPI/Event/schedules/regionEntrances

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionEntranceScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, refer to the message XML_Schedule for details, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--ro, opt, object, configuration of region entrance detection arming schedules-->
    <id>
      <!--ro, req, string, ID, desc:format: event type-channel No.-->1
    </id>
    <eventType>
      <!--ro, opt, string, event type-->regionEntrance
    </eventType>
    <videoInputChannelID>
      <!--ro, opt, string, video input channel ID-->test
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--ro, req, object, list of arming schedules, attr:size{req, int}-->
      <TimeBlock>
        <!--ro, opt, object, arming schedules-->
        <dayOfWeek>
          <!--ro, opt, enum, day of a week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->1
        </dayOfWeek>
        <TimeRange>
          <!--ro, req, object, time range-->
          <beginTime>
            <!--ro, req, time, start time-->00:00:00
          </beginTime>
          <endTime>
            <!--ro, req, time, end time-->00:00:00
          </endTime>
        </TimeRange>
        </TimeBlock>
      </TimeBlockList>
    </Schedule>
  </RegionEntranceScheduleList>

```

### 16.10.3.3 Set the arming schedule of region entrance detection for a specified channel

#### Request URL

PUT /ISAPI/Event/schedules/regionEntrances/regionEntrance-<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema", version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->1
  </id>
  <eventType>
    <!--opt, string-->regionEntrance
  </eventType>
  <videoInputChannelID>
    <!--opt, string-->test
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--req, object, attr:size{req, int}-->
    <TimeBlock>
      <!--opt, object-->
      <dayOfWeek>
        <!--opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--req, object-->
        <beginTime>
          <!--req, time-->00:00:00
        </beginTime>
        <endTime>
          <!--req, time-->00:00:00
        </endTime>
      </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

#### 16.10.3.4 Get the arming schedule of region entrance detection for a specified channel

##### Request URL

GET /ISAPI/Event/schedules/regionEntrances/regionEntrance-<channelID>

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->1
  </id>
  <eventType>
    <!--ro, opt, string-->regionEntrance
  </eventType>
  <videoInputChannelID>
    <!--ro, opt, string-->test
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--ro, req, object, attr:size{req, int}-->
    <TimeBlock>
      <!--ro, opt, object-->
      <dayOfWeek>
        <!--ro, opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--ro, req, object-->
        <beginTime>
          <!--ro, req, time-->00:00:00
        </beginTime>
        <endTime>
          <!--ro, req, time-->00:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>

```

#### 16.10.3.5 Get the linkage triggering parameters of region entrance for a specified channel

##### Request URL

GET /ISAPI/Event/triggers/regionEntrance-<channelID>/notifications

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
<!--ro, req, array, subType:object, attr:version{req, string, protocolVersion}-->
<EventTriggerNotification>
<!--ro, opt, object-->
<id>
<!--ro, req, string-->test
</id>
<notificationMethod>
<!--ro, req, enum, subType:string-->email
</notificationMethod>
<notificationRecurrence>
<!--ro, opt, enum, subType:string-->beginning
</notificationRecurrence>
<videoInputID>
<!--ro, opt, string-->1
</videoInputID>
</EventTriggerNotification>
</EventTriggerNotificationList>
```

### 16.10.3.6 Set the linkage triggering parameters of region entrance for a specified channel

#### Request URL

PUT /ISAPI/Event/triggers/regionEntrance-<channelID>/notifications

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
<!--req, array, subType:object, attr:version{req, string, protocolVersion}-->
<EventTriggerNotification>
<!--opt, object-->
<id>
<!--req, string-->test
</id>
<notificationMethod>
<!--req, enum, subType:string-->email
</notificationMethod>
<notificationRecurrence>
<!--opt, enum, subType:string-->beginning
</notificationRecurrence>
<videoInputID>
<!--opt, string-->1
</videoInputID>
</EventTriggerNotification>
</EventTriggerNotificationList>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
<!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
<requestURL>
<!--ro, req, string-->null
</requestURL>
<statusCode>
<!--ro, req, enum, subType:int-->0
</statusCode>
<statusString>
<!--ro, req, enum, subType:string-->OK
</statusString>
<subStatusCode>
<!--ro, req, string-->OK
</subStatusCode>
</ResponseStatus>
```

### 16.10.3.7 Get the linkage parameters of region entrance for a specified channel

#### Request URL

GET /ISAPI/Event/triggers/regionEntrance-<channelID>?triggersEnabledStatus=<triggersEnabledStatus>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
triggersEnabledStatus	string	--

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->1
  </id>
  <eventType>
    <!--ro, req, enum, subType:string-->regionEntrance
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subType:object-->
    <EventTriggerNotification>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->test
      </id>
      <notificationMethod>
        <!--ro, req, enum, subType:string-->email
      </notificationMethod>
      <notificationRecurrence>
        <!--ro, opt, enum, subType:string-->beginning
      </notificationRecurrence>
      <videoInputID>
        <!--ro, opt, string-->1
      </videoInputID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>
```

### 16.10.3.8 Set the linkage parameters of region entrance for a specified channel

#### Request URL

PUT /ISAPI/Event/triggers/regionEntrance-<channelID>?triggersEnabledStatus=<triggersEnabledStatus>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
triggersEnabledStatus	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->1
  </id>
  <eventType>
    <!--req, enum, subType:string-->regionEntrance
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
    <EventTriggerNotification>
      <!--opt, object-->
      <id>
        <!--req, string-->test
      </id>
      <notificationMethod>
        <!--req, enum, subType:string-->email
      </notificationMethod>
      <notificationRecurrence>
        <!--opt, enum, subType:string-->beginning
      </notificationRecurrence>
      <videoInputID>
        <!--opt, string-->1
      </videoInputID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.3.9 Set region entrance detection parameters of all channels

#### Request URL

PUT /ISAPI/Smart/regionEntrance

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionEntranceList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, refer to the message XML_RegionEntrance for details, attr:version{req, string, protocolVersion}-->
  <RegionEntrance>
    <!--opt, object-->
    <id>
      <!--req, string-->test
    </id>
    <enabled>
      <!--req, bool-->true
    </enabled>
    <normalizedScreenSize>
      <!--opt, object, normalized size-->
      <normalizedScreenWidth>
        <!--req, int, normalized width-->1
      </normalizedScreenWidth>
      <normalizedScreenHeight>
        <!--req, int, normalized height-->1
      </normalizedScreenHeight>
    </normalizedScreenSize>
    <RegionEntranceRegionList>
      <!--opt, object, area List-->
      <RegionEntranceRegion>
        <!--opt, object, intrusion region-->
        <id>
          <!--req, string, detection area ID-->1
        </id>
        <sensitivityLevel>
          <!--opt, int, sensitivity, range:[0,100]-->1
        </sensitivityLevel>
        <RegionCoordinatesList>
          <!--opt, array, region coordinates List, subType:object, range:[3,10]-->
        </RegionCoordinatesList>
        <detectionTarget>
          <!--opt, enum, detection target, subType:string, desc:detection target-->all
        </detectionTarget>
        <alarmConfidence>
          <!--opt, enum, subType:string-->low
        </alarmConfidence>
      </RegionEntranceRegion>
    </RegionEntranceRegionList>
    <isSupportMultiScene>
      <!--opt, bool, whether supports multiple scene detection-->true
    </isSupportMultiScene>
  </RegionEntrance>
</RegionEntranceList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code description-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.3.10 Get region entrance detection parameters of all channels

#### Request URL

GET /ISAPI/Smart/regionEntrance

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionEntranceList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, refer to the message XML_RegionEntrance for details, attr:version{req, string, protocolVersion}-->
  <RegionEntrance>
    <!--ro, opt, object-->
    <id>
      <!--ro, req, string, channel No.-->test
    </id>
    <enabled>
      <!--ro, req, bool, whether to enable the function-->true
    </enabled>
    <normalizedScreenSize>
      <!--ro, opt, object, normalized size-->
      <normalizedScreenWidth>
        <!--ro, req, int, normalized width-->1
      </normalizedScreenWidth>
      <normalizedScreenHeight>
        <!--ro, req, int, normalized height-->1
      </normalizedScreenHeight>
    </normalizedScreenSize>
    <RegionEntranceRegionList>
      <!--ro, opt, array, area List, subType:object-->
      <RegionEntranceRegion>
        <!--ro, opt, object-->
        <id>
          <!--ro, req, string-->1
        </id>
        <sensitivityLevel>
          <!--ro, opt, int, sensitivity, range:[1,100]-->1
        </sensitivityLevel>
        <RegionCoordinatesList>
          <!--ro, opt, array, region coordinates List, subType:object, range:[3,10]-->
        </RegionCoordinatesList>
        <detectionTarget>
          <!--ro, opt, enum, detection target, subType:string, desc:detection target-->all
        </detectionTarget>
        <alarmConfidence>
          <!--ro, opt, enum, subType:string-->low
        </alarmConfidence>
      </RegionEntranceRegion>
    </RegionEntranceRegionList>
    <isSupportMultiScene>
      <!--ro, opt, bool, whether supports multiple scene detection-->true
    </isSupportMultiScene>
  </RegionEntrance>
</RegionEntranceList>

```

### 16.10.3.11 Set parameters of the region entrance detection rule by channel

#### Request URL

PUT /ISAPI/Smart/regionEntrance/<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionEntrance xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, region entrance detection configuration, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string, channel No.-->test
  </id>
  <enabled>
    <!--req, bool, whether to enable the function-->true
  </enabled>
  <normalizedScreenSize>
    <!--opt, object, normalized size-->
    <normalizedScreenWidth>
      <!--req, int, normalized width-->1
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--req, int, normalized height-->1
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <RegionEntranceRegionList>
    <!--opt, object, region List-->
    <RegionEntranceRegion>
      <!--opt, object, region entrance detection region-->
      <id>
        <!--req, string, region ID-->1
      </id>
      <sensitivityLevel>
        <!--opt, int, sensitivity, range:[0,100]-->1
      </sensitivityLevel>
      <RegionCoordinatesList>
        <!--opt, object, region coordinate list-->
      </RegionCoordinatesList>
      <detectionTarget>
        <!--opt, enum, detection target, subType:string, desc:"all", "human", "vehicle", "others" (targets except human bodies and vehicles). The value "all" is mutually exclusive with "others"-->all
      </detectionTarget>
      <alarmConfidence>
        <!--opt, enum, alarm confidence, subType:string, desc:"Low" (default value), "mediumLow", "mediumHigh", "high"-->low
      </alarmConfidence>
    </RegionEntranceRegion>
  </RegionEntranceRegionList>
  <isSupportMultiScene>
    <!--opt, bool, whether the device supports multiple scenes detection-->true
  </isSupportMultiScene>
</RegionEntrance>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.3.12 Get region entrance detection parameters by video input channel

#### Request URL

GET /ISAPI/Smart/regionEntrance/<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<RegionEntrance xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, read-only, "PDC"-people counting statistics, which is not supported during region entrance detection, attr:version{req, string, protocolVersion}-->
    <id>
        <!--ro, req, string-->test
    </id>
    <enabled>
        <!--ro, req, bool, whether to enable the function-->true
    </enabled>
    <normalizedScreenSize>
        <!--ro, opt, object, normalized size-->
        <normalizedScreenWidth>
            <!--ro, req, int, normalized width-->1
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!--ro, req, int, normalized height-->1
        </normalizedScreenHeight>
    </normalizedScreenSize>
    <RegionEntranceRegionList>
        <!--ro, opt, object-->
        <RegionEntranceRegion>
            <!--ro, opt, object, Intrusion Region-->
            <id>
                <!--ro, req, string-->1
            </id>
            <sensitivityLevel>
                <!--ro, opt, int, range:[0,100]-->1
            </sensitivityLevel>
            <RegionCoordinatesList>
                <!--ro, opt, array, area coordinate list, subType:object, range:[3,10]-->
            </RegionCoordinatesList>
            <detectionTarget>
                <!--ro, opt, enum, detection target, subType:string-->all
            </detectionTarget>
            <alarmConfidence>
                <!--ro, opt, enum, subType:string-->low
            </alarmConfidence>
        </RegionEntranceRegion>
    </RegionEntranceRegionList>
    <isSupportMultiScene>
        <!--ro, opt, bool, whether supports multiple-scene detection-->true
    </isSupportMultiScene>
</RegionEntrance>
```

### 16.10.3.13 Get configuration capability of region entrance detection by channel

#### Request URL

GET /ISAPI/Smart/regionEntrance/<channelID>/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionEntrance xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, ro, mutex capability, it indicates that which function is mutually exclusive with the region entrance detection function: "PDC,videoFrameRate50,videoFrameRate0,19-1920*1080@50fps,20-1920*1080@0fps,scheduled recording", attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string, channel No.-->test
  </id>
  <enabled opt="true,false">
    <!--ro, req, bool, whether to enable, attr:opt{req, string}-->true
  </enabled>
  <normalizedScreenSize>
    <!--ro, opt, object, normalized screen size-->
    <normalizedScreenWidth>
      <!--ro, req, int, normalized screen width-->1
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--ro, req, int, normalized screen height-->1
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <RegionEntranceRegionList size="4">
    <!--ro, opt, array, subType:object, attr:size{req, int}-->
    <RegionEntranceRegion>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->1
      </id>
      <sensitivityLevel min="1" max="100">
        <!--ro, req, int, sensitivity, attr:min{req, int},max{req, int}-->1
      </sensitivityLevel>
      <RegionCoordinatesList size="10">
        <!--ro, opt, array, region coordinate list, subType:object, range:[3,10], attr:size{req, int}-->
      </RegionCoordinatesList>
      <detectionTarget>
        <!--ro, opt, enum, detection target, subType:string-->all
      </detectionTarget>
      <alarmConfidence opt="low,mediumLow,mediumHigh,high" def="low">
        <!--ro, opt, enum, subType:string, attr:opt{req, string},def{req, string}-->low
      </alarmConfidence>
    </RegionEntranceRegion>
  </RegionEntranceRegionList>
  <isSupportMultiScene opt="true,false">
    <!--ro, req, bool, whether supports multiple scenes detection, attr:opt{req, string}-->true
  </isSupportMultiScene>
  <isSupportTargetMultiSelect>
    <!--ro, opt, bool, whether supports selecting multiple detection targets, desc:if this function is supported, this node will be returned and its value is true; otherwise, this node will not be returned-->true
  </isSupportTargetMultiSelect>
  <isSupportAllDayUpload>
    <!--ro, opt, bool, whether supports uploading all-day event-->true
  </isSupportAllDayUpload>
</RegionEntrance>

```

#### 16.10.3.14 Get parameters of all region entrance detection regions by channel

##### Request URL

GET /ISAPI/Smart/regionEntrance/<channelID>/regions

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionEntranceRegionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, refer to the message XML_RegionEntranceRegion for details, attr:version{req, string, protocolVersion}-->
</RegionEntranceRegionList>

```

#### 16.10.3.15 Set parameters of all region entrance detection regions by channel

##### Request URL

PUT /ISAPI/Smart/regionEntrance/<channelID>/regions

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<RegionEntranceRegionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, region parameters list, attr:version{req, string, protocolVersion}-->
</RegionEntranceRegionList>
```

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6
(Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format",
"Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
  <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.10.3.16 Set parameters of a region entrance detection region by channel

#### Request URL

PUT /ISAPI/Smart/regionEntrance/<channelID>/regions/<regionID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<RegionEntranceRegion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, region configuration, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string, region ID-->1
  </id>
  <sensitivityLevel>
    <!--req, int, sensitivity-->1
  </sensitivityLevel>
  <RegionCoordinatesList>
    <!--opt, array, region coordinate list, subType:object, range:[3,10]-->
  </RegionCoordinatesList>
  <detectionTarget>
    <!--opt, enum, detection target, subType:string, desc:"all", "human", "vehicle", "others" (targets except human bodies and vehicles). The value "all" is
mutually exclusive with "others"-->all
  </detectionTarget>
  <alarmConfidence>
    <!--opt, enum, alarm confidence, subType:string, desc:"Low", "mediumLow", "mediumHigh", "high"-->low
  </alarmConfidence>
</RegionEntranceRegion>
```

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

### 16.10.3.17 Get parameters of a region entrance detection region of a specific channel

#### Request URL

GET /ISAPI/Smart/regionEntrance/<channelID>/regions/<regionID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionEntranceRegion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string, region ID-->1
  </id>
  <sensitivityLevel>
    <!--ro, req, int, sensitivity-->1
  </sensitivityLevel>
  <RegionCoordinatesList>
    <!--ro, opt, array, List of area coordinates, subType:object, range:[3,10]-->
  </RegionCoordinatesList>
  <detectionTarget>
    <!--ro, opt, enum, detection target, subType:string, desc:"all", "human", "vehicle", "others"-->all
  </detectionTarget>
  <alarmConfidence>
    <!--ro, opt, enum, confidence of alarm notification, subType:string, desc:confidence of alarm notification: "low", "mediumLow", "mediumHigh", "high", "Low"-->low
  </alarmConfidence>
</RegionEntranceRegion>

```

## 16.10.4 Region Exiting Detection

### 16.10.4.1 Get arming schedules of all channels for region exiting detection

#### Request URL

GET /ISAPI/Event/schedules/regionExitings

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionExitingScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, List of arming schedules, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--ro, opt, object-->
    <id>
      <!--ro, req, string, ID, desc:format: event type-channel No.-->test
    </id>
    <eventType>
      <!--ro, opt, string, event type-->regionExiting
    </eventType>
    <videoInputChannelID>
      <!--ro, opt, string, video input channel ID-->test
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--ro, req, object, List of arming schedules, attr:size{req, int}-->
      <TimeBlock>
        <!--ro, opt, object, arming schedules-->
        <dayOfWeek>
          <!--ro, opt, enum, day of a week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->1
        </dayOfWeek>
        <TimeRange>
          <!--ro, req, object, time range-->
          <beginTime>
            <!--ro, req, time, start time-->00:00:00
          </beginTime>
          <endTime>
            <!--ro, req, time, end time-->00:00:00
          </endTime>
        </TimeRange>
        </TimeBlock>
      </TimeBlockList>
    </Schedule>
  </RegionExitingScheduleList>

```

#### 16.10.4.2 Set region exiting detection arming schedules of all channels

##### Request URL

PUT /ISAPI/Event/schedules/regionExitings

##### Query Parameter

None

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionExitingScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, refer to the message XML_Schedule for details, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--opt, object-->
    <id>
      <!--req, string-->test
    </id>
    <eventType>
      <!--opt, string, event type-->regionExiting
    </eventType>
    <videoInputChannelID>
      <!--opt, string, video input channel ID-->test
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--req, object, arming time list, attr:size{req, int}-->
      <TimeBlock>
        <!--opt, object, arming time-->
        <dayOfWeek>
          <!--opt, enum, day of the week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7 (Sunday)-->1
        </dayOfWeek>
        <TimeRange>
          <!--req, object-->
          <beginTime>
            <!--req, time, start time-->00:00:00
          </beginTime>
          <endTime>
            <!--req, time, end time-->00:00:00
          </endTime>
        </TimeRange>
        </TimeBlock>
      </TimeBlockList>
    </Schedule>
  </RegionExitingScheduleList>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, desc:sub status code description-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, desc:sub status code description-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

#### 16.10.4.3 Get the arming schedule of region exiting detection for a specified channel

##### Request URL

GET /ISAPI/Event/schedules/regionExitings/regionExiting-<channelID>

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->test
  </id>
  <eventType>
    <!--ro, opt, string-->regionExiting
  </eventType>
  <videoInputChannelID>
    <!--ro, opt, string-->test
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--ro, req, object, attr:size{req, int}-->
    <TimeBlock>
      <!--ro, opt, object-->
      <dayOfWeek>
        <!--ro, opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--ro, req, object-->
        <beginTime>
          <!--ro, req, time-->00:00:00
        </beginTime>
        <endTime>
          <!--ro, req, time-->00:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>

```

#### 16.10.4.4 Set the arming schedule of region exiting detection for a specified channel

##### Request URL

PUT /ISAPI/Event/schedules/regionExitings/regionExiting-<channelID>

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<Schedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->test
  </id>
  <eventType>
    <!--opt, string-->regionExiting
  </eventType>
  <videoInputChannelID>
    <!--opt, string-->test
  </videoInputChannelID>
  <TimeBlockList size="8">
    <!--req, object, attr:size{req, int}-->
    <TimeBlock>
      <!--opt, object-->
      <dayOfWeek>
        <!--opt, enum, subType:int-->1
      </dayOfWeek>
      <TimeRange>
        <!--req, object-->
        <beginTime>
          <!--req, time-->00:00:00
        </beginTime>
        <endTime>
          <!--req, time-->00:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>
```

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.10.4.5 Get the linkage triggering parameters of region exiting for a specified channel

#### Request URL

GET /ISAPI/Event/triggers/regionExiting-<channelID>/notifications

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

### Request Message

None

### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, subType:object, attr:version{req, string, protocolVersion}-->
  <EventTriggerNotification>
    <!--ro, opt, object-->
    <id>
      <!--ro, req, enum, subType:string-->regionExiting
    </id>
    <notificationMethod>
      <!--ro, req, enum, subType:string-->email
    </notificationMethod>
    <notificationRecurrence>
      <!--ro, opt, enum, subType:string-->beginning
    </notificationRecurrence>
    <videoInputID>
      <!--ro, opt, string-->1
    </videoInputID>
  </EventTriggerNotification>
</EventTriggerNotificationList>

```

#### 16.10.4.6 Set the linkage triggering parameters of region exiting for a specified channel

##### Request URL

PUT /ISAPI/Event/triggers/regionExiting-<channelID>/notifications

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, subType:object, attr:version{req, string, protocolVersion}-->
  <EventTriggerNotification>
    <!--opt, object-->
    <id>
      <!--req, enum, subType:string-->regionExiting
    </id>
    <notificationMethod>
      <!--ro, req, enum, subType:string-->email
    </notificationMethod>
    <notificationRecurrence>
      <!--opt, enum, subType:string-->beginning
    </notificationRecurrence>
    <videoInputID>
      <!--opt, string-->1
    </videoInputID>
  </EventTriggerNotification>
</EventTriggerNotificationList>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

#### 16.10.4.7 Set the linkage parameters of region exiting for a specified channel

##### Request URL

PUT /ISAPI/Event/triggers/regionExiting-<channelID> ?triggersEnabledStatus=<triggersEnabledStatus>

## Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
triggersEnabledStatus	string	--

## Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string-->1
  </id>
  <eventType>
    <!--req, string-->regionExiting
  </eventType>
  <eventDescription>
    <!--opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, subType:object-->
    <EventTriggerNotification>
      <!--opt, object-->
      <id>
        <!--req, string-->test
      </id>
      <notificationMethod>
        <!--req, enum, subType:string-->email
      </notificationMethod>
      <notificationRecurrence>
        <!--opt, enum, subType:string-->beginning
      </notificationRecurrence>
      <videoInputID>
        <!--opt, string-->1
      </videoInputID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>
```

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.10.4.8 Get the linkage parameters of region exiting for a specified channel

#### Request URL

GET /ISAPI/Event/triggers/regionExiting-<channelID>?triggersEnabledStatus=<triggersEnabledStatus>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
triggersEnabledStatus	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->1
  </id>
  <eventType>
    <!--ro, req, string-->regionEntrance
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->test
  </eventDescription>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subType:object-->
    <EventTriggerNotification>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->test
      </id>
      <notificationMethod>
        <!--ro, req, enum, subType:string-->email
      </notificationMethod>
      <notificationRecurrence>
        <!--ro, opt, enum, subType:string-->beginning
      </notificationRecurrence>
      <videoInputID>
        <!--ro, opt, string-->1
      </videoInputID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>
```

### 16.10.4.9 Get region exiting detection parameters of all channels

#### Request URL

GET /ISAPI/Smart/regionExiting

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionExitingList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, subType:object, attr:version{req, string, protocolVersion}-->
  <RegionExiting>
    <!--ro, opt, object, region exiting-->
    <id>
      <!--ro, req, string, channel No.-->1
    </id>
    <enabled>
      <!--ro, req, bool, whether to enable the function-->true
    </enabled>
    <normalizedScreenSize>
      <!--ro, opt, object, normalized size-->
      <normalizedScreenWidth>
        <!--ro, req, int, normalized width-->1
      </normalizedScreenWidth>
      <normalizedScreenHeight>
        <!--ro, req, int, normalized height-->1
      </normalizedScreenHeight>
    </normalizedScreenSize>
    <RegionExitingRegionList>
      <!--ro, opt, array, subType:object-->
      <RegionExitingRegion>
        <!--ro, opt, object, region exiting-->
        <id>
          <!--ro, req, string, detection area ID-->1
        </id>
        <sensitivityLevel>
          <!--ro, req, int, sensitivity, range:[0,100]-->1
        </sensitivityLevel>
        <RegionCoordinatesList>
          <!--ro, opt, array, region coordinates List, subType:object, range:[3,10]-->
        </RegionCoordinatesList>
        <detectionTarget>
          <!--ro, opt, enum, detection target, subType:string, desc:detection target-->all
        </detectionTarget>
        <alarmConfidence>
          <!--ro, opt, enum, subType:string-->low
        </alarmConfidence>
      </RegionExitingRegion>
    </RegionExitingRegionList>
    <isSupportMultiScene>
      <!--ro, opt, bool, whether supports multiple scene detection-->true
    </isSupportMultiScene>
  </RegionExiting>
</RegionExitingList>

```

#### 16.10.4.10 Set region exiting detection parameters of all channels

##### Request URL

PUT /ISAPI/Smart/regionExiting

##### Query Parameter

None

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionExitingList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, subType:object, attr:version{req, string, protocolVersion}-->
  <RegionExiting>
    <!--opt, object-->
    <id>
      <!--req, string-->1
    </id>
    <enabled>
      <!--req, bool-->true
    </enabled>
    <normalizedScreenSize>
      <!--opt, object, normalized size-->
      <normalizedScreenWidth>
        <!--req, int, normalized width-->1
      </normalizedScreenWidth>
      <normalizedScreenHeight>
        <!--req, int, normalized height-->1
      </normalizedScreenHeight>
    </normalizedScreenSize>
    <RegionExitingRegionList>
      <!--opt, array, subType:object-->
      <RegionExitingRegion>
        <!--opt, object-->
        <id>
          <!--req, string-->1
        </id>
        <sensitivityLevel>
          <!--req, int, sensitivity, range:[0,100]-->1
        </sensitivityLevel>
        <RegionCoordinatesList>
          <!--opt, array, region coordinates list, subType:object, range:[3,10]-->
        </RegionCoordinatesList>
        <detectionTarget>
          <!--opt, enum, detection target, subType:string, desc:detection target-->all
        </detectionTarget>
        <alarmConfidence>
          <!--opt, enum, subType:string-->low
        </alarmConfidence>
      </RegionExitingRegion>
    </RegionExitingRegionList>
    <isSupportMultiScene>
      <!--opt, bool, whether supports multiple scene detection-->true
    </isSupportMultiScene>
  </RegionExiting>
</RegionExitingList>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code description-->OK
  </subStatusCode>
  <!--ro, req, string, sub status code, desc:sub status code description-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.4.11 Set parameters of region exiting detection rule by channel

#### Request URL

PUT /ISAPI/Smart/regionExiting/<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionExiting xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, parameters of region exiting detection rule, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string, channel No.-->1
  </id>
  <enabled>
    <!--req, bool, whether to enable the function-->true
  </enabled>
  <normalizedScreenSize>
    <!--opt, object, normalized size-->
    <normalizedScreenWidth>
      <!--req, int, normalized width-->1
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--req, int, normalized height-->1
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <RegionExitingRegionList>
    <!--opt, array, region list of region exiting detection, subType:object-->
    <RegionExitingRegion>
      <!--opt, object, region exiting detection region-->
      <id>
        <!--req, string, region ID-->1
      </id>
      <sensitivityLevel>
        <!--req, int, sensitivity, range:[0,100]-->1
      </sensitivityLevel>
      <RegionCoordinatesList>
        <!--opt, array, region coordinate List, subType:object, range:[3,10]-->
      </RegionCoordinatesList>
      <detectionTarget>
        <!--opt, enum, detection target, subType:string, desc:"all", "human", "vehicle", "others" (targets except human bodies and vehicles). The value "all" is mutually exclusive with "others"-->all
      </detectionTarget>
      <alarmConfidence>
        <!--opt, enum, alarm confidence, subType:string, desc:"Low" (default value), "mediumLow", "mediumHigh", "high"-->low
      </alarmConfidence>
    </RegionExitingRegion>
  </RegionExitingRegionList>
  <isSupportMultiScene>
    <!--opt, bool, whether supports multi-scene detection-->true
  </isSupportMultiScene>
</RegionExiting>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.10.4.12 Get parameters of region exiting detection rule by channel

#### Request URL

GET /ISAPI/Smart/regionExiting/<channelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

## Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<RegionExiting xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, read-only, "PDC"-people counting statistics, which is not supported during region exiting detection, attr:version{req, string, protocolVersion}-->
    <id>
        <!--ro, req, string, channel No.-->1
    </id>
    <enabled>
        <!--ro, req, bool, whether to enable the function-->true
    </enabled>
    <normalizedScreenSize>
        <!--ro, opt, object, normalized size-->
        <normalizedScreenWidth>
            <!--ro, req, int, normalized width-->1
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!--ro, req, int, normalized height-->1
        </normalizedScreenHeight>
    </normalizedScreenSize>
    <RegionExitingRegionList>
        <!--ro, opt, array, List of region exiting, subType:object-->
        <RegionExitingRegion>
            <!--ro, opt, object, region exiting-->
            <id>
                <!--ro, req, string, region ID-->1
            </id>
            <sensitivityLevel>
                <!--ro, req, int, sensitivity, range:[0,100]-->1
            </sensitivityLevel>
            <RegionCoordinatesList>
                <!--ro, opt, array, region coordinate list, subType:object, range:[3,10]-->
            </RegionCoordinatesList>
            <detectionTarget>
                <!--ro, opt, enum, detection target, subType:string, desc:"all", "human", "vehicle", "others" (targets except human bodies and vehicles). The value "all" is mutually exclusive with "others"-->all
            </detectionTarget>
            <alarmConfidence>
                <!--ro, opt, enum, alarm confidence, subType:string, desc:"low" (default value), "mediumLow", "mediumHigh", "high"-->low
            </alarmConfidence>
        </RegionExitingRegion>
    </RegionExitingRegionList>
    <isSupportMultiScene>
        <!--ro, opt, bool, whether supports multi-scene detection-->true
    </isSupportMultiScene>
</RegionExiting>
```

### 16.10.4.13 Get configuration capability of region exiting detection by channel

#### Request URL

GET /ISAPI/Smart/regionExiting/<channelID>/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionExiting xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, ro, mutex capability, it indicates that which function is mutually exclusive with the region exiting detection:
    "PDC,videoFrameRate50,videoFrameRate0,19-1920*1080@50fps,20-1920*1080@0fps,scheduled recording", attr:version{req, string, protocolVersion}-->
    <id>
        <!--ro, req, string, channel No.-->test
    </id>
    <enabled opt="true,false">
        <!--ro, req, bool, whether to enable, attr:opt{req, string}-->true
    </enabled>
    <normalizedScreenSize>
        <!--ro, opt, object, normalized screen size-->
        <normalizedScreenWidth>
            <!--ro, req, int, normalized screen width-->1
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!--ro, req, int, normalized screen height-->1
        </normalizedScreenHeight>
    </normalizedScreenSize>
    <RegionExitingRegionList size="4">
        <!--ro, opt, array, subType:object, attr:size{req, int}-->
        <RegionExitingRegion>
            <!--ro, opt, object-->
            <id>
                <!--ro, req, string-->1
            </id>
            <sensitivityLevel min="1" max="100">
                <!--ro, req, int, sensitivity, attr:min{req, int},max{req, int}-->1
            </sensitivityLevel>
            <RegionCoordinatesList>
                <!--ro, opt, array, region coordinate list, subType:object, range:[3,10]-->
            </RegionCoordinatesList>
            <detectionTarget opt="all,human,vehicle,others">
                <!--ro, opt, enum, detection target, subType:string, attr:opt{req, string}-->all
            </detectionTarget>
            <alarmConfidence opt="low,mediumLow,mediumHigh,high" def="low">
                <!--ro, opt, enum, subType:string, attr:opt{req, string},def{req, string}-->low
            </alarmConfidence>
        </RegionExitingRegion>
    </RegionExitingRegionList>
    <isSupportMultiScene>
        <!--ro, req, bool, whether supports multiple scenes.-->true
    </isSupportMultiScene>
    <isSupportTargetMultiSelect>
        <!--ro, opt, bool, whether supports selecting multiple detection targets, desc:if this function is supported, this node will be returned and its value
        is true; otherwise, this node will not be returned-->true
    </isSupportTargetMultiSelect>
    <isSupportAllDayUpload>
        <!--ro, opt, bool, whether supports uploading all-day event-->true
    </isSupportAllDayUpload>
</RegionExiting>

```

#### 16.10.4.14 Set parameters of all region exiting detection regions by channel

##### Request URL

PUT /ISAPI/Smart/regionExiting/<channelID>/regions

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionExitingRegionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, array, parameter list of all region exiting detection regions, subType:object, attr:version{req, string, protocolVersion}-->
</RegionExitingRegionList>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

#### 16.10.4.15 Get parameters of all region exiting detection regions by channel

##### Request URL

GET /ISAPI/Smart/regionExiting/<channelID>/regions

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionExitingRegionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, subType:object, attr:version{req, string, protocolVersion}-->
</RegionExitingRegionList>

```

#### 16.10.4.16 Get parameters of a region exiting detection region of a specific channel

##### Request URL

GET /ISAPI/Smart/regionExiting/<channelID>/regions/<regionID>

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionID	string	--

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionExitingRegion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string, region ID-->1
  </id>
  <sensitivityLevel>
    <!--ro, req, int, sensitivity, range:[0,100]-->1
  </sensitivityLevel>
  <RegionCoordinatesList>
    <!--ro, opt, array, List of area coordinates, subType:object, range:[3,10]-->
  </RegionCoordinatesList>
  <detectionTarget>
    <!--ro, opt, enum, detection target, subType:string, desc:"all", "human", "vehicle", "others"-->all
  </detectionTarget>
  <alarmConfidence>
    <!--ro, opt, enum, confidence of alarm notification, subType:string, desc:confidence of alarm notification: "Low", "mediumLow", "mediumHigh", "high", "Low"-->low
  </alarmConfidence>
</RegionExitingRegion>

```

#### 16.10.4.17 Set parameters of a region exiting detection region by channel

##### Request URL

PUT /ISAPI/Smart/regionExiting/<channelID>/regions/<regionID>

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
regionID	string	--

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RegionExitingRegion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, parameters of a region exiting detection region, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string, region ID-->1
  </id>
  <sensitivityLevel>
    <!--req, int, sensitivity, range:[0,100]-->1
  </sensitivityLevel>
  <RegionCoordinatesList>
    <!--opt, array, region coordinate list, subType:object, range:[3,10]-->
  </RegionCoordinatesList>
  <detectionTarget>
    <!--opt, enum, detection target type, subType:string, desc:"all", "human", "vehicle", "others" (targets except human bodies and vehicles). The value "all" is mutually exclusive with "others"-->all
  </detectionTarget>
  <alarmConfidence>
    <!--opt, enum, alarm confidence, subType:string, desc:"Low" (default value), "mediumLow", "mediumHigh", "high"-->low
  </alarmConfidence>
</RegionExitingRegion>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

## 16.11 Multi-Target-Type Detection

### 16.11.1 Global Target Detection

#### 16.11.1.1 Get the arming schedule of heat map for all channels

##### Request URL

GET /ISAPI/Event/schedules/humanRecognition

##### Query Parameter

None

##### Request Message

None

##### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<HumanRecognitionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, array, subType:object, attr:version{req, string, protocolVersion}-->
</HumanRecognitionScheduleList>
```

### 16.11.2 Multi-Target-Type Detection

#### 16.11.2.1 Set the linkage triggering parameters of multi-target-type detection

##### Request URL

PUT /ISAPI/Event/triggers/mixedTargetDetection-<channelID>/notifications

##### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

##### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--req, array, subType:object, attr:version{req, string, protocolVersion}-->
<EventTriggerNotification>
    <!--opt, object-->
    <id>
        <!--req, string-->test
    </id>
    <notificationMethod>
        <!--req, enum, subType:string-->FTP
    </notificationMethod>
    <notificationRecurrence>
        <!--opt, enum, subType:string-->beginning
    </notificationRecurrence>
</EventTriggerNotification>
</EventTriggerNotificationList>
```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.11.2.2 Get the linkage triggering parameters of multi-target-type detection

#### Request URL

GET /ISAPI/Event/triggers/mixedTargetDetection-<channelID>/notifications

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTriggerNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, subType:object, attr:version{req, string, protocolVersion}-->
  <EventTriggerNotification>
    <!--ro, opt, object-->
    <id>
      <!--ro, req, string-->test
    </id>
    <notificationMethod>
      <!--ro, req, enum, subType:string-->FTP
    </notificationMethod>
    <notificationRecurrence>
      <!--ro, opt, enum, subType:string-->beginning
    </notificationRecurrence>
  </EventTriggerNotification>
</EventTriggerNotificationList>

```

## 16.12 Algorithm Scheduling

### 16.12.1 Scheduling and Managing Local Smart Resources

#### 16.12.1.1 Get parameters of intelligent resources switch

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/VCAResource

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<VCAResource xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, intelligent resources, attr:version{req, string, protocolVersion}-->
  <type>
    <!--ro, req, enum, "basicBehavior", subType:string, desc:"basicBehavior"-->basicBehavior
  </type>
</VCAResource>

```

### 16.12.1.2 Set parameters of intelligent resources switch

#### Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/VCAResource

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<VCAResource xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, VCA Resource, attr:version{req, string, protocolVersion}-->
  <type>
    <!--req, enum, type, subType:string,
desc:"basicBehavior,fulLBehavior,facesnapBehavior,facesnap,TFS,smartVehicleDetection,smartHVDetection,smart,judicial,smart264AndRoadDetection,smart264AndFa
ceDetection,smart264AndHeatMap,smartIntelligentMonitor,smartTrafficDataCollection,roadDetection,humanRecognition,perimeterCapture,vehicleDetection,HVDetect
ion,mixedTargetDetection,trackingCaptureMode,nonTrackingCaptureMode,close,faceHumanModeling,contrast,cityManagement,teacherBehavior,12MPLiveView,personQueueD
etection,verticalPeopleCounting,safetyHelmet,faceCounting,personArming,AIOpenPlatform"-->basicBehavior
  </type>
</VCAResource>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL, range:[0,1024]-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6
(Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format",
"Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
  <description>
    <!--ro, opt, string, range:[0,1024]-->badXmlFormat
  </description>
</ResponseStatus>

```

### 16.12.1.3 Get capability of switching intelligent resources

#### Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>/VCAResource/capabilities

#### Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<VCAResource xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, intelligent resources, attr:version{req, string, protocolVersion}-->
  <type>
opt="basicBehavior,fullBehavior,facesnapBehavior,facesnap,TFS,smartVehicleDetection,smartHVTDetection,smart,judicial,smart264AndRoadDetection,smart264AndFaceDetection,smart264AndHeatMap,smartVehicleIllegalParkingDetection,smartIntelligentMonitor,smartTrafficDataCollection,roadDetection,humanRecognition,perimeterCapture,vehicleDetection,HVTDetection,mixedTargetDetection,trackingCaptureMode,nonTrackingCaptureMode,close,faceHumanModelingContrast,cityManagement,teacherBehavior,12MPLiveView,personQueueDetection,verticalPeopleCounting,AIOpenPlatform,mixedTargetFaceHuman,mixedTargetAll,safetyHelmet,faceContrast,none,faceMonitor,vehicleMonitor,faceCounting,heatPDC,personDensity,behaviorMixedTargetAll,operationMonitor,fielddetection,personArming,smokeDetectAlarm,smokeDetection,personDensityQueueLeavePosition,bannerDetection,ATMSurround,ATMPanel,ATMSafetyCabin,intelligentTraffic,heatmap,teacherBehaviorAndFaceContrast,mixedTargetAllCapture,riverwayGaugeReading,MixedTargetAllAuxiliaryCapture,radarHearNurse,smokeDetectionAndFlameCombustion,PackingSpaceRecognition" def="basicBehavior">
  <!--ro, req, enum, resource type, subType:string, attr:opt{req, string},def{opt, string}, desc:resource type-->basicBehavior
  </type>
  <isSupportAPPWorking>
    <!--ro, opt, bool-->true
  </isSupportAPPWorking>
  <isSupportTest>
    <!--ro, opt, bool-->true
  </isSupportTest>
</VCAResource>

```

## 16.13 VCA

### 16.13.1 Hard Hat Detection

#### 16.13.1.1 Get arming schedules for hard hat detection of all channels

##### Request URL

GET /ISAPI/Event/schedules/safetyHelmetDetection

##### Query Parameter

None

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<SafetyHelmetDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, arming schedules for hard hat detection of all channels, subType:object, attr:version{req, string, protocolVersion}-->
</SafetyHelmetDetectionScheduleList>

```

#### 16.13.1.2 Set arming schedules for hard hat detection of all channels

##### Request URL

PUT /ISAPI/Event/schedules/safetyHelmetDetection

##### Query Parameter

None

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<SafetyHelmetDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, see details in the structure XML_Schedule, subType:object, attr:version{req, string, protocolVersion}-->
</SafetyHelmetDetectionScheduleList>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, desc:sub status code description-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, desc:sub status code description-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

## 16.13.2 VCA Abnormal Event Recognition

### 16.13.2.1 Get the intelligent channel list

#### Request URL

GET /ISAPI/Intelligent/intelliChannelList

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<IntelliChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, intelligent channel list, subType:object, attr:version{req, string, protocolVersion}-->
</IntelliChannelList>

```

## 16.14 Two-Way Audio

### 16.14.1 Two-Way Audio

#### 16.14.1.1 Get the parameters of all two-way audio channels

#### Request URL

GET /ISAPI/System/TwoWayAudio/channels

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<TwoWayAudioChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, parameters configuration of the two-way audio channel, subType:object, attr:version{opt, string, protocolVersion}-->
  <TwoWayAudioChannel>
    <!--ro, opt, object, configuration of the two-way audio channel-->
    <id>
      <!--ro, req, string, audio channel ID-->1
    </id>
    <enabled>
      <!--ro, req, bool, whether to enable or not-->true
    </enabled>
    <audioCompressionType>
      <!--ro, req, enum, encoding type of the audio output, subType:string, desc:"G.711alaw", "G.711ulaw", "G.726", "G.729", "G.729a", "G.729b", "PCM", "MP3", "AC3", "AAC", "ADPCM", "MP2L2", "Opus"-->G.711alaw
    </audioCompressionType>
    <speakerVolume>
      <!--ro, opt, int, input volume, range:[1,100]-->100
    </speakerVolume>
    <noisereduce>
      <!--ro, opt, bool, whether to enable the environmental noise filter or not-->true
    </noisereduce>
    <audioInputType>
      <!--ro, opt, enum, audio input type, subType:string, desc:"MicIn" (microphone-Level input), "LineIn" (Line-Level input), "selfAdaptive" (self-adaptive), "wirelessPickUp" (wireless audio pickup)-->MicIn
    </audioInputType>
    <lineOutForbidden>
      <!--ro, opt, bool, whether the audio output is not supported. If this node is not returned or its value is "false", audio output is supported; if the value is "true", audio output is not supported-->true
    </lineOutForbidden>
  </TwoWayAudioChannel>
</TwoWayAudioChannelList>

```

### 16.14.1.2 Set audio parameters for all two-way audio channels

#### Request URL

PUT /ISAPI/System/TwoWayAudio/channels

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<TwoWayAudioChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, parameters configuration of the two-way audio channel, subType:object, attr:version{opt, string, protocolVersion}-->
  <TwoWayAudioChannel>
    <!--opt, object, configuration of the two-way audio channel-->
    <id>
      <!--req, string, audio channel ID-->1
    </id>
    <enabled>
      <!--req, bool, whether to enable the function or not-->true
    </enabled>
    <audioCompressionType>
      <!--req, enum, encoding type of the audio output, subType:string, desc:"G.711alaw", "G.711ulaw", "G.726", "G.729", "G.729a", "G.729b", "PCM", "MP3", "AC3", "AAC", "ADPCM", "MP2L2", "Opus"-->G.711alaw
    </audioCompressionType>
    <speakerVolume>
      <!--opt, int, input volume-->100
    </speakerVolume>
    <noisereduce>
      <!--opt, bool, whether to enable the environmental noise filter or not-->true
    </noisereduce>
    <audioInputType>
      <!--req, enum, audio input type, subType:string, desc:"MicIn" (microphone-Level input), "LineIn" (Line-Level input), "selfAdaptive" (self-adaptive), "wirelessPickUp" (wireless audio pickup)-->MicIn
    </audioInputType>
    <lineOutForbidden>
      <!--opt, bool, whether the audio output is not supported. If this node is not returned or its value is "false", audio output is supported; if the value is "true", audio output is not supported-->true
    </lineOutForbidden>
  </TwoWayAudioChannel>
</TwoWayAudioChannelList>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

### 16.14.1.3 Set audio parameters of a specified two-way audio channel

#### Request URL

PUT /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>

#### Query Parameter

Parameter Name	Parameter Type	Description
twoWayAudioChannelID	string	--

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<TwoWayAudioChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, configuration of the two-way audio channel, attr:version{opt, string, protocolVersion}-->
  <id>
    <!--req, string, audio channel ID-->1
  </id>
  <enabled>
    <!--req, bool, whether to enable or not-->true
  </enabled>
  <audioCompressionType>
    <!--req, enum, encoding type of the audio output, subType:string, desc:"G.711alaw", "G.711ulaw", "G.726", "G.729", "G.729a", "G.729b", "PCM", "MP3", "AC3", "AAC", "ADPCM", "MP2L2", "Opus"-->G.711alaw
  </audioCompressionType>
  <speakerVolume>
    <!--opt, int, input volume, range:[1,100]-->100
  </speakerVolume>
  <noisereduce>
    <!--opt, bool, whether to enable the environmental noise filter or not-->true
  </noisereduce>
  <audioInputType>
    <!--opt, enum, audio input type, subType:string, desc:"MicIn" (microphone-Level input), "LineIn" (Line-level input), "selfAdaptive" (self-adaptive), "wirelessPickUp" (wireless audio pickup)-->MicIn
  </audioInputType>
  <lineOutForbidden>
    <!--opt, bool, whether the audio output is not supported. If this node is not returned or its value is "false", audio output is supported; if the value is "true", audio output is not supported-->true
  </lineOutForbidden>
</TwoWayAudioChannel>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
    <statusCode>
      <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
    </statusCode>
    <statusString>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

#### 16.14.1.4 Get the capability of configuring audio parameters for a specified two-way audio channel

##### Request URL

GET /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>/capabilities?audioInputType=<audioInputType>

##### Query Parameter

Parameter Name	Parameter Type	Description
twoWayAudioChannelID	string	--
audioInputType	string	--

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<TwoWayAudioChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, capability of configuring audios for the two-way audio channel, attr:version{opt, string, protocolVersion}-->
  <id>
    <!--ro, req, string, audio channel ID-->1
  </id>
  <enabled>
    <!--ro, req, bool, whether to enable or not-->true
  </enabled>
  <audioCompressionType opt="G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM,MP2L2,Opus,G.722.1">
    <!--ro, req, enum, encoding type of the audio output, subType:string, attr:opt{opt, string}, desc:"G.711alaw", "G.711ulaw", "G.726", "G.729", "G.729a", "G.729b", "PCM", "MP3", "AC3", "AAC", "ADPCM", "MP2L2", "Opus", "G.722.1"-->G.711alaw
  </audioCompressionType>
  <speakerVolume min="0" max="100">
    <!--ro, opt, int, input volume, attr:min{opt, int},max{opt, int}-->100
  </speakerVolume>
  <noisereduce opt="true,false">
    <!--ro, opt, bool, whether to enable the environmental noise filter or not, attr:opt{opt, string}-->true
  </noisereduce>
  <audioInputType opt="MicIn,LineIn,selfAdaptive,wirelessPickUp">
    <!--ro, opt, enum, audio input type, subType:string, attr:opt{opt, string}, desc:"MicIn" (microphone-level input), "LineIn" (line-level input), "selfAdaptive" (self-adaptive), "wirelessPickUp" (wireless audio pickup)-->MicIn
  </audioInputType>
  <isSupportEchoCancellation>
    <!--ro, opt, bool, whether the device supports echo cancellation-->true
  </isSupportEchoCancellation>
</TwoWayAudioChannel>

```

#### 16.14.1.5 Stop two-way audio

##### Request URL

PUT /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>/close?sessionId=<audioSessionID>

##### Query Parameter

Parameter Name	Parameter Type	Description
twoWayAudioChannelID	string	Two-Way Audio Channel ID
audioSessionID	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL, range:[0,1024]-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, which describes the error in details, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>
```

### 16.14.1.6 Start two-way audio

#### Request URL

PUT /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>/open?type=<type>

#### Query Parameter

Parameter Name	Parameter Type	Description
twoWayAudioChannelID	string	--
type	string	--

### Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<TwoWayAudio xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, parameters for this two-way audio, attr:version{req, string, protocolVersion}-->
</TwoWayAudio>
```

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<TwoWayAudioSession xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, result returned when starting two-way audio succeeds, attr:version{opt, string, protocolVersion}-->
  <sessionId>
    <!--ro, req, string, two-way audio session ID-->1
  </sessionId>
</TwoWayAudioSession>
```

### 16.14.1.7 Get the audio parameters of a specified two-way audio channel

#### Request URL

GET /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>?audioInputType=<audioInputType>

#### Query Parameter

Parameter Name	Parameter Type	Description
twoWayAudioChannelID	string	--
audioInputType	string	--

### Request Message

None

### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<TwoWayAudioChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, configuration of the two-way audio channel, attr:version{opt, string, protocolVersion}-->
  <id>
    <!--ro, req, string, audio channel ID-->
  </id>
  <enabled>
    <!--ro, req, bool, whether to enable or not-->true
  </enabled>
  <audioCompressionType>
    <!--ro, req, enum, encoding type of the audio output, subType:string, desc:"G.711alaw", "G.711ulaw", "G.726", "G.729", "G.729a", "G.729b", "PCM", "MP3", "AC3", "AAC", "ADPCM", "MP2L2", "Opus", "G.722.1"-->G.711alaw
  </audioCompressionType>
  <speakerVolume>
    <!--ro, opt, int, input volume, range:[1,100]-->100
  </speakerVolume>
  <noisereduce>
    <!--ro, opt, bool, whether to enable the environmental noise filter or not-->true
  </noisereduce>
  <audioInputType>
    <!--ro, opt, enum, audio input type, subType:string, desc:"MicIn" (microphone-level input), "LineIn" (line-level input), "selfAdaptive" (self-adaptive), "wirelessPickUp" (wireless audio pickup)-->MicIn
  </audioInputType>
  <lineOutForbidden>
    <!--ro, opt, bool, whether the audio output is not supported. If this node is not returned or its value is "false", audio output is supported; if the value is "true", audio output is not supported-->true
  </lineOutForbidden>
</TwoWayAudioChannel>
```

### 16.14.1.8 Get the capability of configuring parameters for all two-way audio channels

#### Request URL

GET /ISAPI/System/TwoWayAudio/channels/capabilities

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<TwoWayAudioChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, capability of configuring audios for all two-way audio channels, attr:version{opt, string, protocolVersion}-->
  <TwoWayAudioChannel>
    <!--ro, opt, object, capability of configuring audios for the two-way audio channel-->
    <id>
      <!--ro, req, string, audio channel ID-->1
    </id>
    <enabled>
      <!--ro, req, bool, whether to enable or not-->true
    </enabled>
    <audioCompressionType opt="G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM,MP2L2,Opus">
      <!--ro, req, enum, encoding type of the audio output, subType:string, attr:opt{opt, string}, desc:"G.711alaw", "G.711ulaw", "G.726", "G.729", "G.729a", "G.729b", "PCM", "MP3", "AC3", "AAC", "ADPCM", "MP2L2", "Opus"-->G.711alaw
    </audioCompressionType>
    <speakerVolume min="0" max="100">
      <!--ro, opt, int, input volume, attr:min{opt, int},max{opt, int}-->100
    </speakerVolume>
    <noisereduce opt="true,false">
      <!--ro, opt, bool, whether to enable the environmental noise filter or not, attr:opt{opt, string}-->true
    </noisereduce>
    <audioInputType opt="MicIn,LineIn,selfAdaptive,wirelessPickUp">
      <!--ro, opt, enum, audio input type, subType:string, attr:opt{opt, string}, desc:"MicIn" (microphone-level input), "LineIn" (line-level input), "selfAdaptive" (self-adaptive), "wirelessPickUp" (wireless audio pickup)-->MicIn
    </audioInputType>
    <isSupportEchoCancellation>
      <!--ro, opt, bool, whether the device supports echo cancellation-->true
    </isSupportEchoCancellation>
  </TwoWayAudioChannel>
</TwoWayAudioChannelList>

```

## 16.15 Audio Recognition

### 16.15.1 Audio Exception Detection

#### 16.15.1.1 Get the arming schedule of audio detection for all channels

##### Request URL

GET /ISAPI/Event/schedules/audioDetections

##### Query Parameter

None

##### Request Message

None

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<AudioDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, subType:object, attr:version{req, string, protocolVersion}-->
</AudioDetectionScheduleList>

```

#### 16.15.1.2 Set the arming schedule of audio detection for all channels

##### Request URL

PUT /ISAPI/Event/schedules/audioDetections

##### Query Parameter

None

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<AudioDetectionScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, subType:object, attr:version{req, string, protocolVersion}-->
</AudioDetectionScheduleList>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

## 16.16 Human Body Detection and PIR Sensing

### 16.16.1 PIR Detection

#### 16.16.1.1 Set the arming schedule of passive infrared (PIR) detection for all channels

##### Request URL

PUT /ISAPI/Event/schedules/PIR

##### Query Parameter

None

##### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PIRScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, subType:object, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--opt, object-->
    <id>
      <!--req, string-->PIR-1
    </id>
    <eventType>
      <!--opt, enum, subType:string-->PIR
    </eventType>
    <videoInputChannelID>
      <!--opt, string-->1
    </videoInputChannelID>
    <timeBlockList size="8">
      <!--req, array, subType:object, attr:size{opt, int}-->
      <TimeBlock>
        <!--opt, object-->
        <dayOfWeek>
          <!--req, enum, subType:int-->1
        </dayOfWeek>
        <TimeRange>
          <!--req, object-->
          <beginTime>
            <!--req, time-->10:00:00
          </beginTime>
          <endTime>
            <!--req, time-->10:00:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </timeBlockList>
  </Schedule>
</PIRScheduleList>

```

##### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.16.1.2 Get the arming schedule of passive infrared (PIR) detection for all channels

#### Request URL

GET /ISAPI/Event/schedules/PIR

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PIRScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, subType:object, attr:version{req, string, protocolVersion}-->
  <Schedule>
    <!--ro, opt, object-->
    <id>
      <!--ro, req, string-->PIR-1
    </id>
    <eventType>
      <!--ro, opt, enum, subType:string-->PIR
    </eventType>
    <videoInputChannelID>
      <!--ro, opt, string-->1
    </videoInputChannelID>
    <TimeBlockList size="8">
      <!--ro, req, array, subType:object, attr:size{opt, int}-->
      <TimeBlock>
        <!--ro, opt, object-->
        <dayOfWeek>
          <!--ro, opt, enum, subType:int-->1
        </dayOfWeek>
        <TimeRange>
          <!--ro, req, object-->
          <beginTime>
            <!--ro, req, time-->10:00:00
          </beginTime>
          <endTime>
            <!--ro, req, time-->10:00:00
          </endTime>
        </TimeRange>
      </TimeBlock>
    </TimeBlockList>
  </Schedule>
</PIRScheduleList>

```

### 16.16.1.3 Set the linkage parameters of passive infrared (PIR) event

#### Request URL

PUT /ISAPI/Event/triggers/PIR

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, event linkage, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string, index-->PIR
  </id>
  <eventType>
    <!--req, enum, event type, subType:string, desc:"PIR"-->PIR
  </eventType>
  <eventDescription>
    <!--opt, string, event description-->PIR Event trigger Information
  </eventDescription>
  <videoInputChannelID>
    <!--opt, string, video input channel ID-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string, dynamic video input channel ID-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--opt, array, linkage action List, subType:object-->
    <EventTriggerNotification>
      <!--opt, object, linkage action-->
      <id>
        <!--req, string, index-->test
      </id>
      <notificationMethod>
        <!--req, enum, Linkage action, subType:string, desc:"FTP" (uploading via FTP), "HTTP" (uploading via HTTP), "IM" (instant messaging), "IO" (I/O output), "LightAudioAlarm" (Light and audio alarm), "audio" (audio), "beep" (buzzer), "center" (uploading to the center), "Cloud" (Cloud storage), "email", "focus" (focus), "monitorAlarm" (alarm on monitor), "ptz" (PTZ), "record" (recording), "syslog" (system log), "trace" (tracking), "whiteLight" (white Light), "SMS"-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--opt, enum, Linkage cycle, subType:string, desc:"beginning" (start), "beginningandend" (start and end), "recurring" (occur again)-->beginning
      </notificationRecurrence>
      <videoInputID>
        <!--opt, string, video channel ID-->1
      </videoInputID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

## Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
</ResponseStatus>

```

### 16.16.1.4 Get the linkage parameters of passive infrared (PIR) event

#### Request URL

GET /ISAPI/Event/triggers/PIR

#### Query Parameter

None

#### Request Message

None

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EventTrigger xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string-->PIR
  </id>
  <eventType>
    <!--ro, req, enum, subType:string-->PIR
  </eventType>
  <eventDescription>
    <!--ro, opt, string-->PIR Event trigger Information
  </eventDescription>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <EventTriggerNotificationList>
    <!--ro, opt, array, subType:object-->
    <EventTriggerNotification>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, string-->test
      </id>
      <notificationMethod>
        <!--ro, req, enum, subType:string-->FTP
      </notificationMethod>
      <notificationRecurrence>
        <!--ro, opt, enum, subType:string-->beginning
      </notificationRecurrence>
      <videoInputID>
        <!--ro, opt, string-->1
      </videoInputID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

### 16.16.1.5 Set the passive infrared (PIR) alarm parameters

#### Request URL

PUT /ISAPI/WLAlarm/PIR

#### Query Parameter

None

#### Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PIRALarm xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, passive infrared (PIR) alarm, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool, whether to enable passive infrared (PIR) alarm-->true
  </enabled>
</PIRALarm>

```

#### Response Message

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, response message, attr:version{ro, req, string, protocolVersion}-->
  <requestURL>
    <!--ro, req, string, request URL, range:[0,1024]-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, status code, subType:int, desc:0 (OK), 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid XML Format), 6 (Invalid XML Content), 7 (Reboot Required)-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, status information, subType:string, desc:"OK" (succeeded), "Device Busy", "Device Error", "Invalid Operation", "Invalid XML Format", "Invalid XML Content", "Reboot" (reboot device)-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, sub status code, desc:sub status code, which describes the error in details-->OK
  </subStatusCode>
  <description>
    <!--ro, opt, string, range:[0,1024]-->badXmlFormat
  </description>
</ResponseStatus>

```

### 16.16.1.6 Get the passive infrared (PIR) alarm parameters

#### Request URL

GET /ISAPI/WLAlarm/PIR

#### Query Parameter

None

#### Request Message

None

#### Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<PIRAlarm xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, passive infrared (PIR) alarm, attr:version{req, string, protocolVersion}-->
    <enabled>
        <!--ro, req, bool, whether to enable passive infrared (PIR) alarm-->true
    </enabled>
</PIRAlarm>
```

### 16.16.1.7 PIR event details

#### EventType:PIR

```

<?xml version="1.0" encoding="UTF-8"?>

<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, event details, attr:version{req, string, protocolVersion}-->
    < ipAddress>
        <!--ro, req, string, IPv4 address of the device that triggers the alarm-->172.6.64.7
    </ipAddress>
    < ipv6Address>
        <!--ro, opt, string, IPv6 address of the device that triggers the alarm-->1080:0:0:0:8:800:200C:417A
    </ipv6Address>
    <portNo>
        <!--ro, opt, int, communication port No. of the device that triggers the alarm-->80
    </portNo>
    <protocol>
        <!--ro, opt, enum, transmission communication protocol type, subType:string, desc:when ISAPI protocol is transmitted via HCNetSDK, the channel No. is the video channel No. of private protocol. When ISAPI protocol is transmitted via EZ protocol, the channel No. is the video channel No. of EZ protocol. When ISAPI protocol is transmitted via ISUP, the channel No. is the video channel No. of ISUP-->HTTP
    </protocol>
    <macAddress>
        <!--ro, opt, string, MAC address-->01:17:24:45:D9:F4
    </macAddress>
    <dynChannelID>
        <!--ro, opt, string, digital channel number-->1
    </dynChannelID>
    <channelID>
        <!--ro, opt, int, channel No. of the device that triggers the alarm, desc:video channel No. that triggers the alarm-->1
    </channelID>
    <dateTime>
        <!--ro, req, datetime, alarm trigger time-->2004-05-03T17:30:08+08:00
    </dateTime>
    <activePostCount>
        <!--ro, opt, int, times that the same alarm has been uploaded, desc:event triggering frequency-->1
    </activePostCount>
    <eventType>
        <!--ro, req, enum, event type, subType:string, desc:PIR: PIR-->PIR
    </eventType>
    <eventState>
        <!--ro, req, enum, event status, subType:string, desc:for durative event: active (valid), inactive (invalid)-->active
    </eventState>
    <eventDescription>
        <!--ro, req, string, event description-->PIR alarm
    </eventDescription>
    <channelName>
        <!--ro, opt, string, channel name-->test
    </channelName>
    <deviceID>
        <!--ro, opt, string, device ID, desc:it should be returned for EHome alarms, e.g., test0123 (EHome2.0, EHome4.0, ISUP5.0)-->12345
    </deviceID>
    <CaptureList>
        <!--ro, opt, array, captured picture list, subtype:object-->
        <Capture>
            <!--ro, opt, object, captured picture-->
            <resourcesContentType>
                <!--ro, req, enum, resource type, subType:string, desc:"url" (the resource is transmitted via URL), "binary" (the resource is transmitted in binary format)-->url
            </resourcesContentType>
            <resourcesContent>
                <!--ro, req, string, resource ID, desc: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 resource URL-->http://10.65.95.131:6120/pic?D970A95367F0C94295698366FB149EA7
            </resourcesContent>
            <time>
                <!--ro, req, datetime, capture time-->2017-04-22T15:39:01+08:00
            </time>
        </Capture>
    </CaptureList>
    <UUID>
        <!--ro, opt, string, unique ID of event uploading, range:[1,64], desc:it can be composed of time (accurate to the millisecond) and random digits-->2020082619390183400q3xC2gHjlszZ76Dx219817RA62TwNeW4j2w52JMB1W63
    </UUID>
    <recordChannelList>
        <!--ro, opt, array, list of channels that are recording, subType:object, range:[0,64]-->
        <recordChannel>
            <!--ro, req, int, No. of the channel which is recording, range:[0,256]-->1
        </recordChannel>
    </recordChannelList>
    <isDataRetransmission>
        <!--ro, opt, bool, whether the data is replenished, desc:if the real-time detection data upload fails due to the network condition, the data will be uploaded again after the device recovers-->true
    </isDataRetransmission>
</EventNotificationAlert>

```

## 17 How-To Video Guidance

If you need access to corresponding video guidance for device integration, please register on <https://tpp.hikvision.com> and visit our Training Center: <https://tpp.hikvision.com/tpp/Training>. The Training Center is specifically designed to provide technical training and guidance resources for our partners. On this platform, you can find integration video tutorials for various devices, enabling better understanding and learning of the integration process. To offer more personalized service, our Training Center also supports filtering by integration protocols, devices, and applications.

margoum.contact@gmail.com  
IA Sport Vision