



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

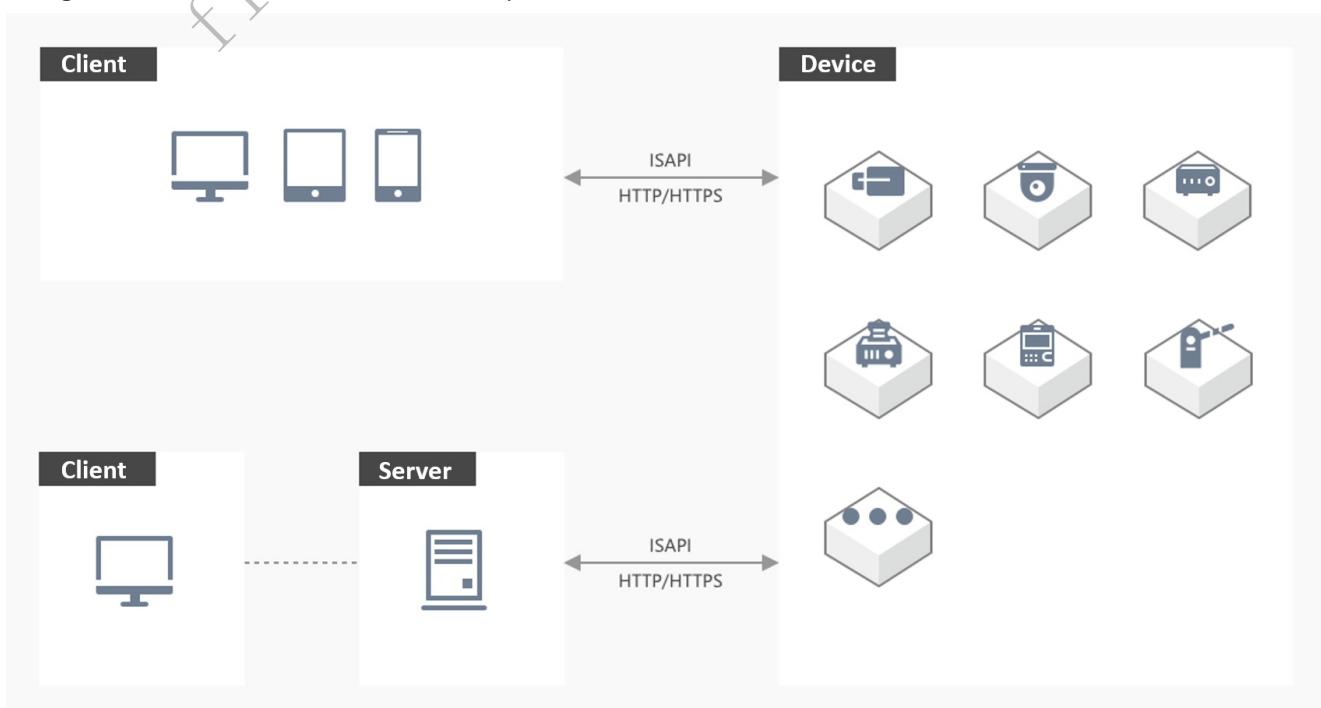
2.1 Introduction

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

- Vehicle Access Control Management

- ANPR Cameras

DS-2CD9046-ESUL, DS-TCG204-DI, DS-TCG205-B, DS-TCG205-B(Module), DS-TCG205-E, DS-TCG225-JIR, DS-TCG227, DS-TCG227-A, DS-TCG227-AIR, DS-TCG2A5-B, DS-TCG400-E, DS-TCG405-E, DS-TCG405-E(MOD), DS-TCG405-E/H, DS-TCG405-EP, DS-TCG406-E, DS-TMC403-E, DS-TMC403-EL

2.3 Terms And Definitions

2.3.1 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.

2.3.2 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.

2.3.3 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.

2.3.4 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.

2.3.5 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

ISAPI: Intelligent Security API.

ISUP: Intelligent Security Uplink Protocol.

HTTP: Hypertext Transfer 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)

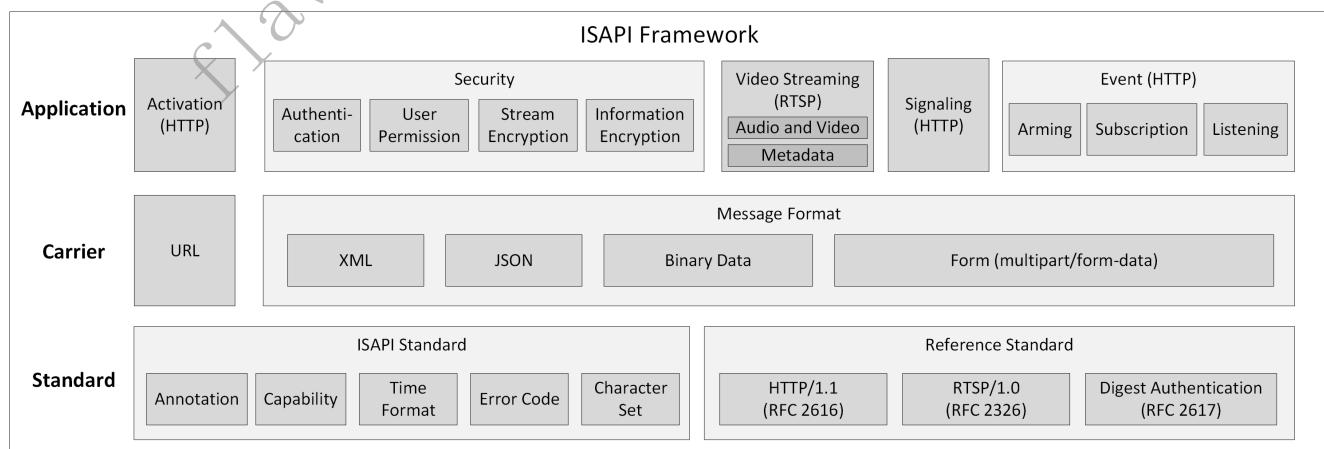
AID: Automatic (Traffic) Incident Detection

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.

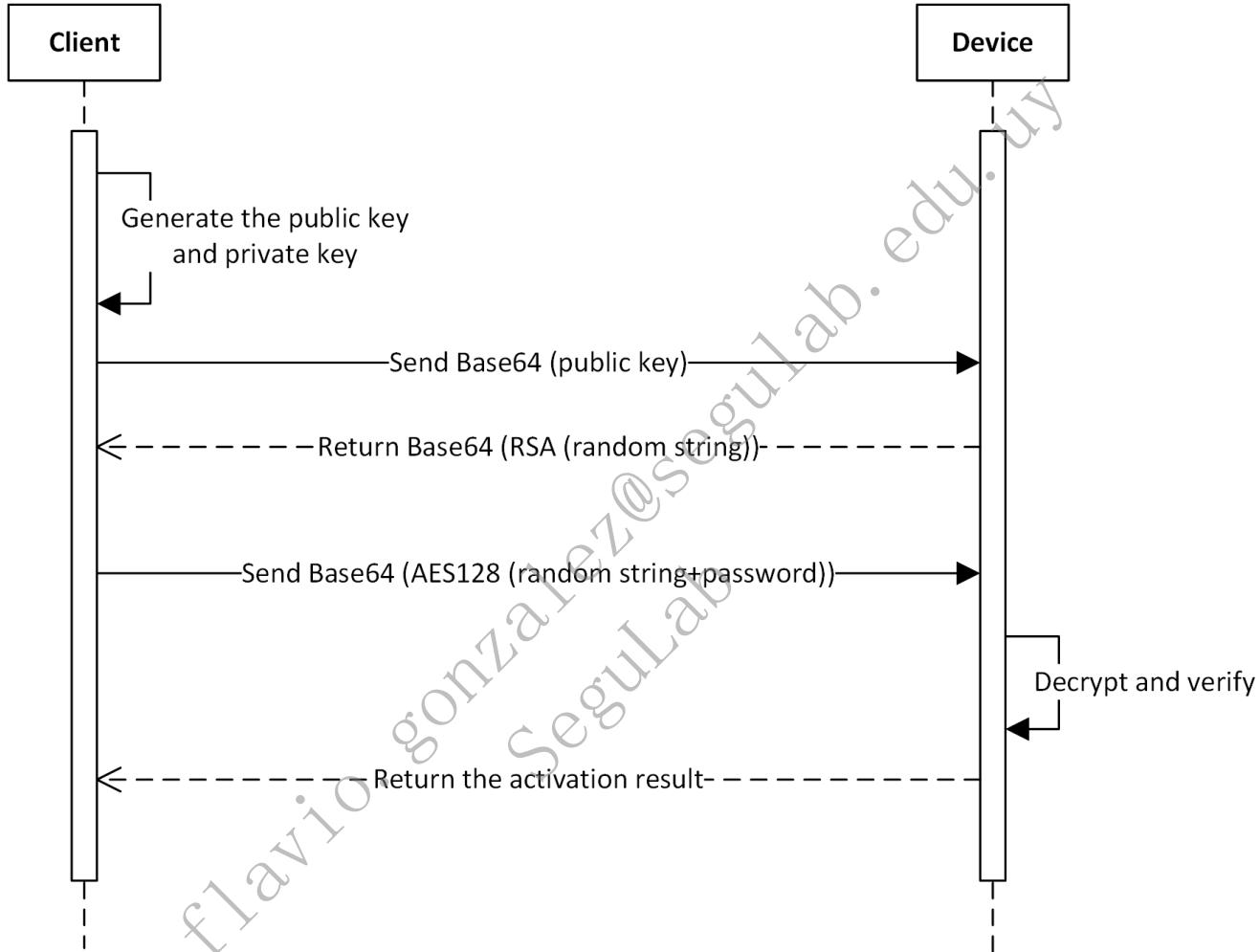
3.2 Activation

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 URL: `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 aaaabbbbccccddddd and the real password is Abc12345, the data before encryption is aaaabbbbccccdddddAbc12345. 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-Type: multipart/form-data; name="humanBackgroundImage.jpg";
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"
}
```

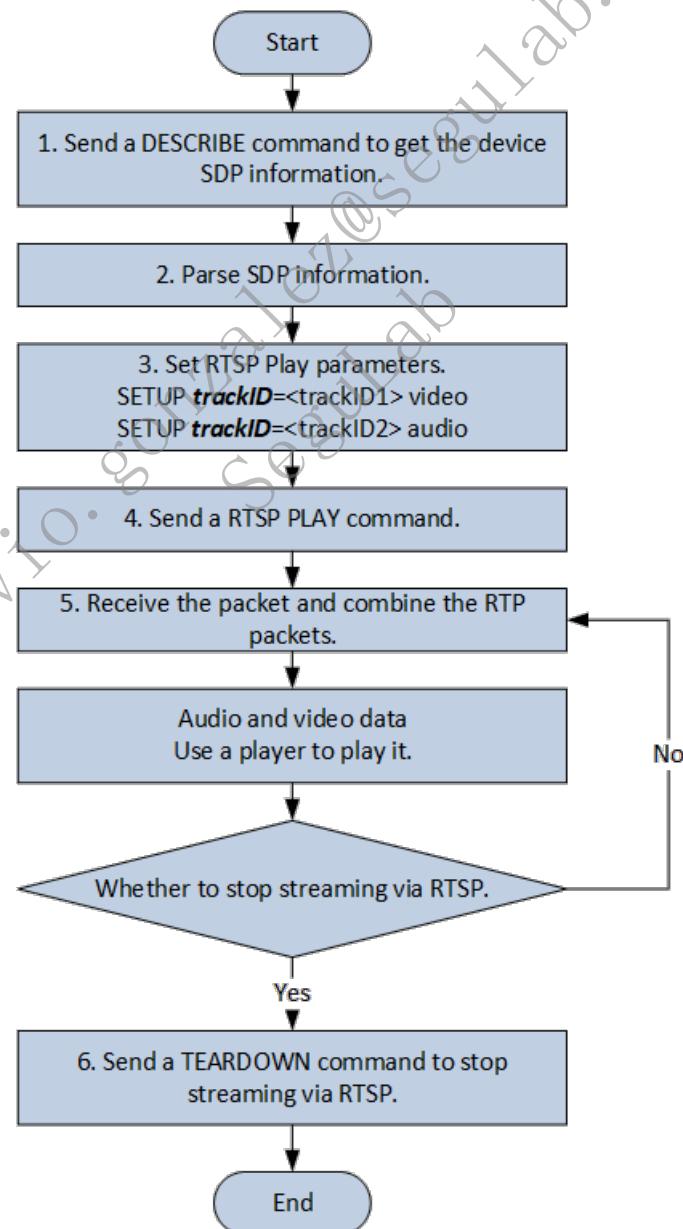
4.3 Real-Time Live View

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 with 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="ff343f5ff82de028dd9b4932cc44201"
Transport:RTP/AVP/TCP;unicast;interleaved=0-1:ssrc=0
User-Agent:NKPlayer-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="ff343f5ff82de0028dd904932cc44201"
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

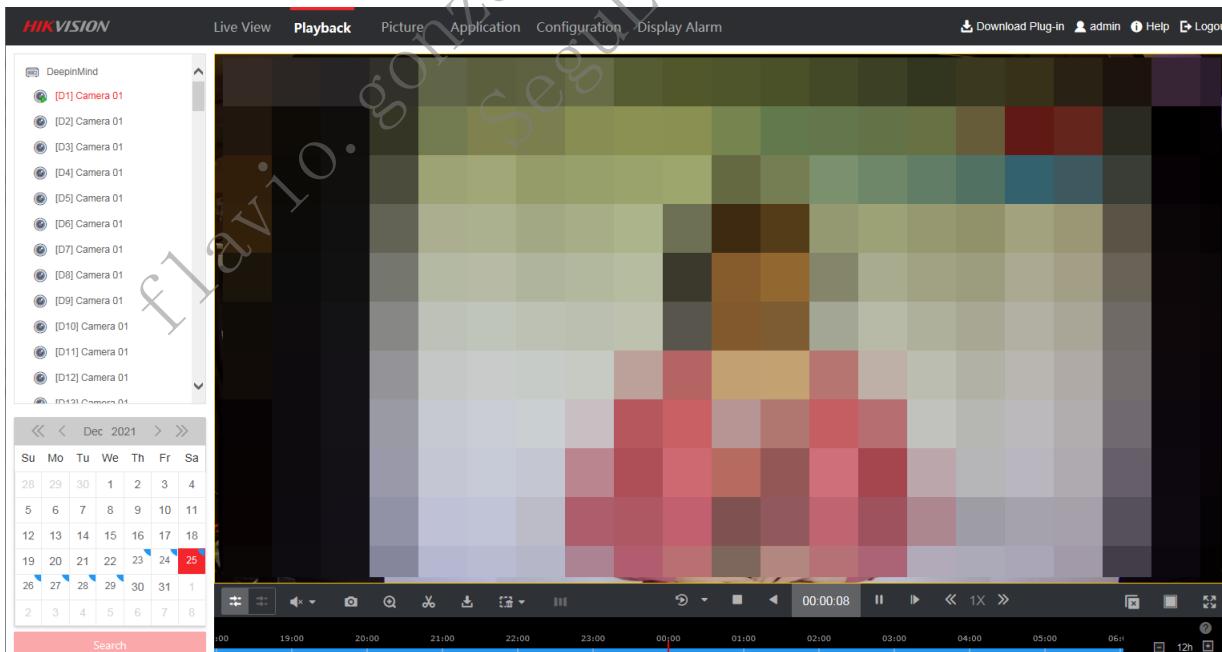
```

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=3b451c9d37cb637827da0815086e7ecfd9984b0095b3cb7198e8197a424aa279

<?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="55e6895b9", 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=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 Arming and Subscription

5.1.1 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.

5.1.2 API Calling Flow

5.1.2.1 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.

5.1.2.1.2 Syntax

```

GET /ISAPI/Event/notification/alertStream HTTP/1.1
Host: <data_gateway_ip>
Connection: Keep-Alive

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"

GET /ISAPI/Event/notification/alertStream 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: 10.6.165.192

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

--<frontier>
Content-Type: application/xml; charset="UTF-8" <!-- Some alarms are in JSON format, so the upper Layer should parse based on the Content-Type field -->
Content-Length: text_length

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

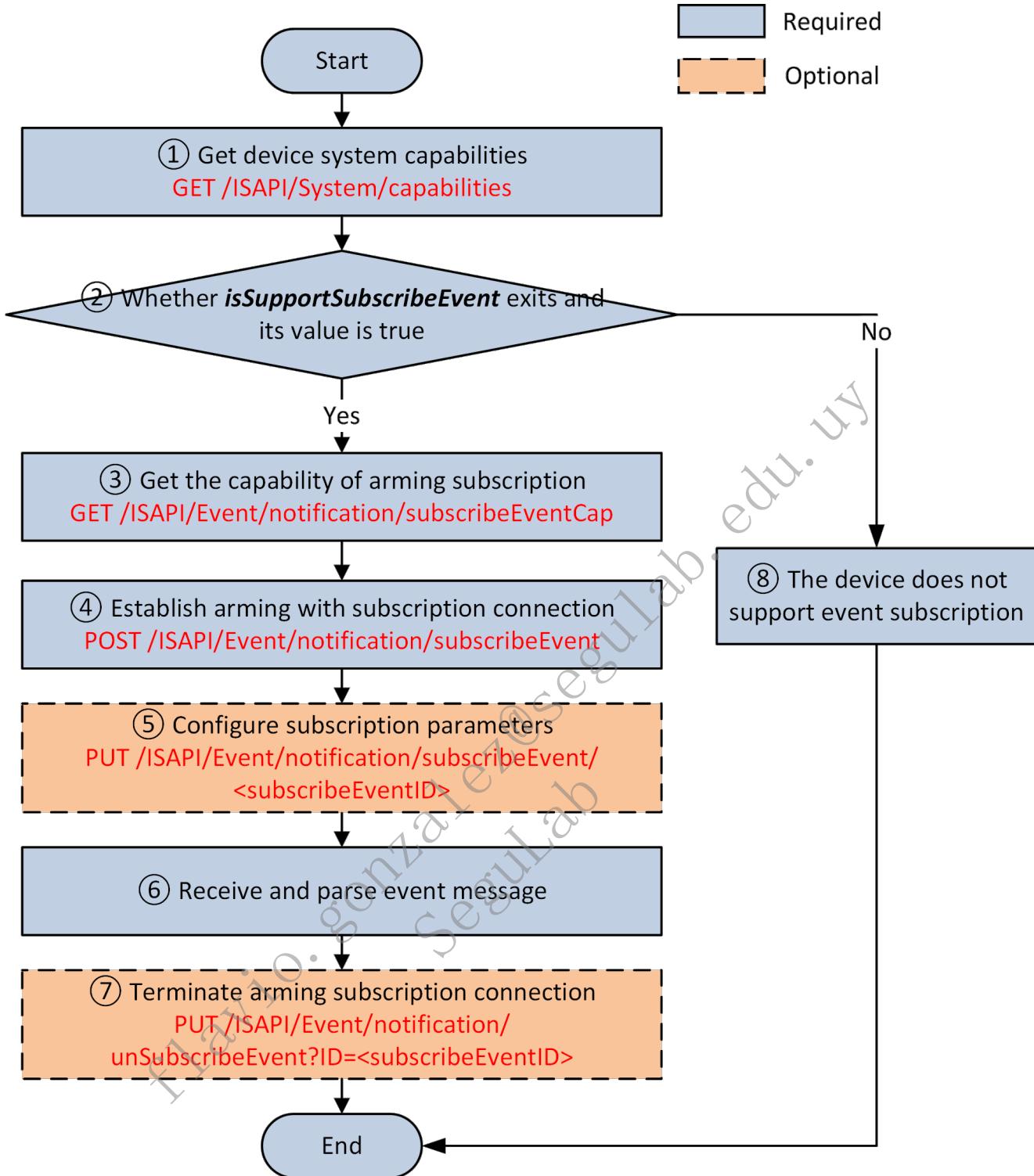
[Image Data]
--<frontier>

```

Note: <data_gateway_ip> and <frontier> are variables, and [Image Data] is an abbreviated representation indicating the raw data of an image at this location.

5.1.2.2 Subscription

5.1.2.2.1 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.

5.1.2.2 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>
```

5.1.2.3 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. Under normal circumstances, the device will not actively close the arming connection, so the device will not send the form format end symbol --AaB03x-- on the arming connection.

```

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.

5.1.3 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.

5.1.4 Sample Messages

5.1.4.1 Establish Arming Subscription

```

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/>

```

5.1.4.2 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>

```

5.2 Calling Flow of Device Packet Capture

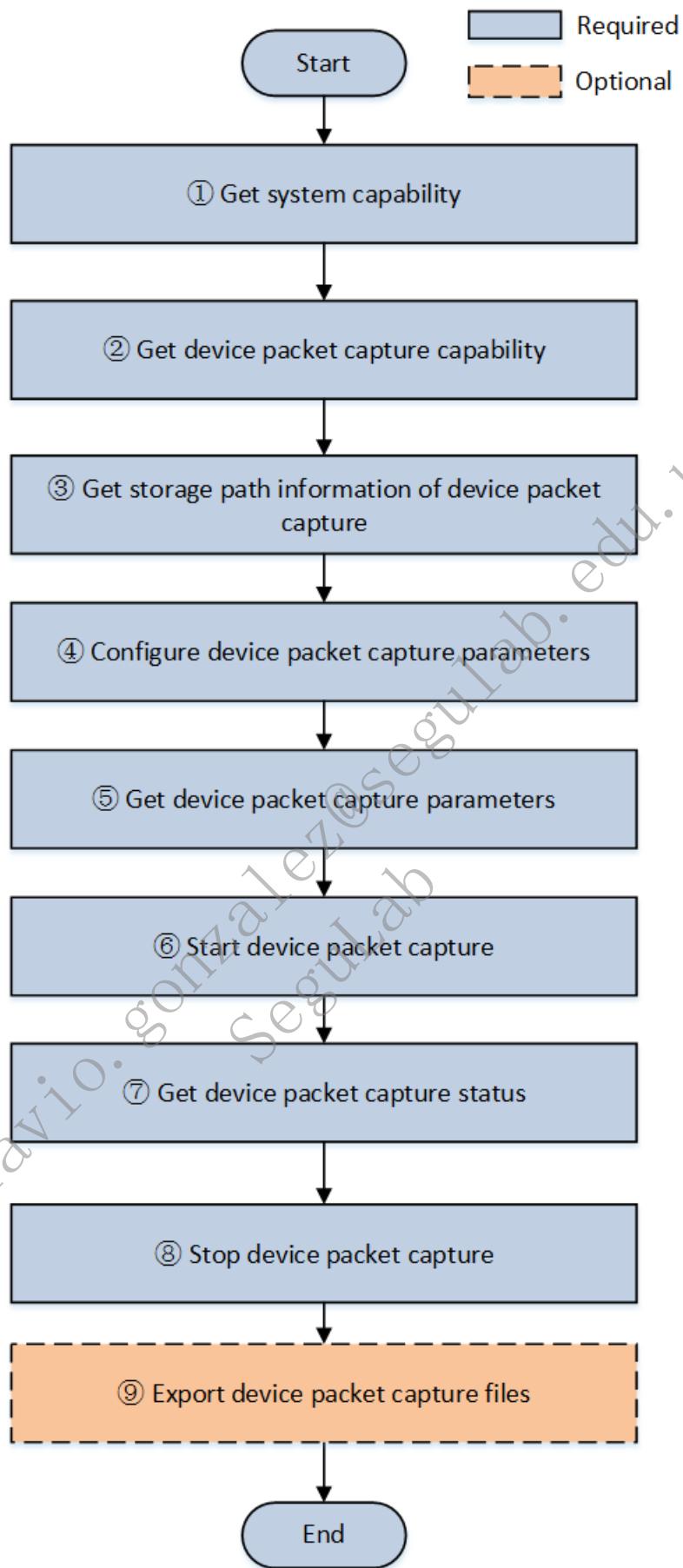
5.2.1 Function Introduction

When problems arise after a device is deployed on-site, interaction message between the device and the external network is necessary to help developers for troubleshooting. Packet capture can be stored on the local device, and packet capture files can be exported after capture is complete. Also, packet capture files can be uploaded to cloud storage, and packet capture data can be obtained in real-time even if the device does not have the storage space.

1. Device packet capture: save packet capture files on the local device, and export the files after capture is complete. Also, uploading packet capture files to cloud storage after capture is complete is supported. Then the client can obtain the storage URL and download packet capture files from the cloud storage.
2. Device real-time packet capture: after it is enabled, the device returns an URI for downloading packet capture data. The client can submit this URI to the browser to download the packet data. The device transmits packet data via HTTP Chunked, and users can store the packet data through the browser.

5.2.2 API Calling Flow

5.2.2.1 Device Packet Capture



1. Get device system capabilities: `GET /ISAPI/System/capabilities`. Get to know if the device supports packet capture by the field: `<isSupportNetworkCapture>true</isSupportNetworkCapture>`.
2. Check if the device supports packet captures: `GET /ISAPI/System/networkCapture/capabilities?format=json`. If `isSupportManualControl` is true, the device supports packet capture. If `isSupportManualControlAsyn` is true, the device supports asynchronous packet capture.

3. Get storage path information of device packet capture: GET /ISAPI/System/networkCapture/StoragePathInfo?format=json.

4. Configure device packet capture parameters such as capture duration, storage path, port, and address: PUT /ISAPI/System/networkCapture/captureParams?format=json.

5. Get device packet capture parameters such as capture duration, storage path, port, and address: GET /ISAPI/System/networkCapture/captureParams?format=json.

6. Start device packet capture: depending on the parameters, packet capture files can be saved on the local device for export after capture is complete, or packet capture files can be uploaded to cloud storage and downloaded via the storage URL, or the packet capture data can be returned in real-time.

Start device packet capture: PUT /ISAPI/System/networkCapture/manualStart?format=json&asyn=<asyn>&realTime=<realTime>.

Note:

- If the API does not contain URL parameters, the packet capture file is saved on the local device.
- If the API contains asyn=true, the packet capture file is automatically uploaded to cloud storage after capture is complete.

7. After starting capture, you can repeatedly get capture status, including whether the capture is ongoing, the size of the packet capture data, and the progress and storage URL for uploading the data to cloud storage.

Get status of device packet capture: GET /ISAPI/System/networkCapture/manualStatus?format=json.

8. Packet capture can be stopped at any time by calling the interface of stopping packet capture.

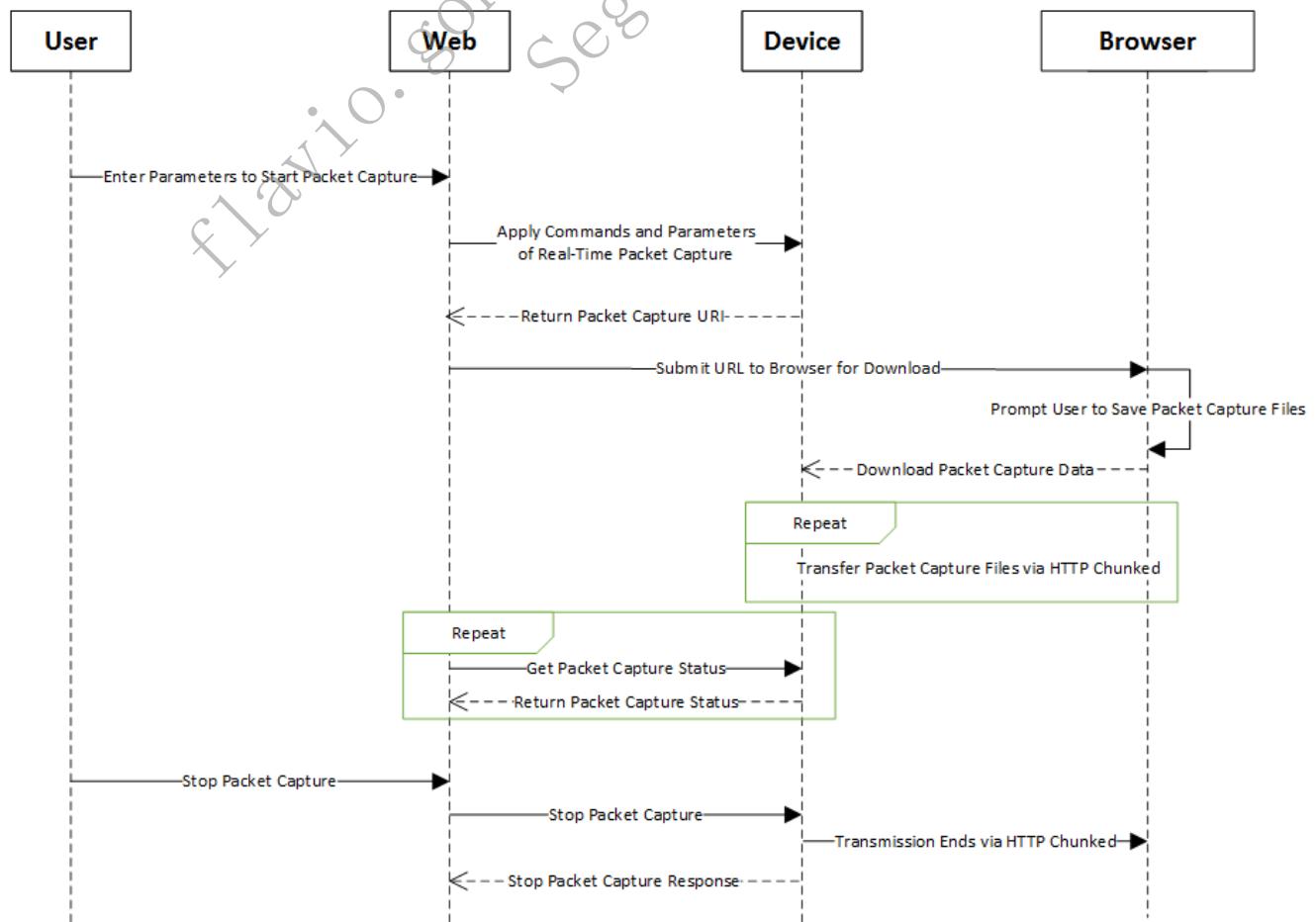
Stop device packet capture: PUT /ISAPI/System/networkCapture/manualStop?format=json.

9. (Optional) If packet capture data is stored on the local device, packet capture files need to be exported. If packet capture files are saved to cloud storage or captured in real-time, there is no need to export packet capture files.

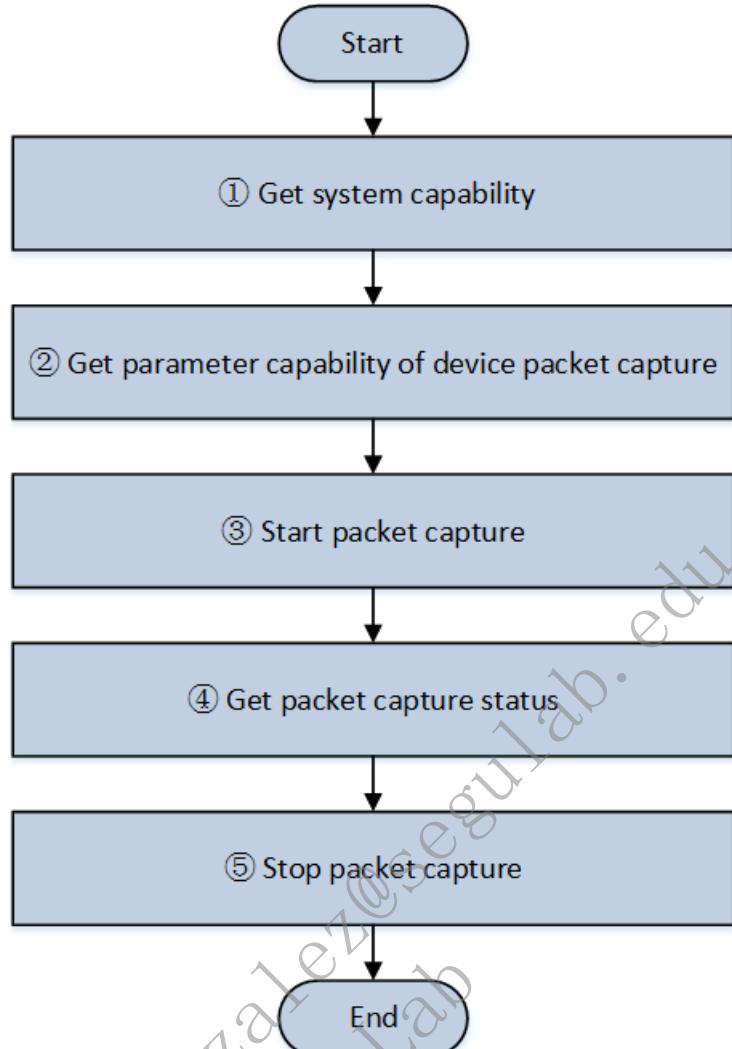
Export device packet capture files: GET /ISAPI/System/networkCapture/exportFile?format=json.

5.2.2.2 Device Real-Time Packet Capture

The process of real-time packet capture is shown as the following:



See the following figure for the calling flow:



1. Get device system capabilities: GET /ISAPI/System/capabilities.

<isSupportStartNetworkCapture>true</isSupportStartNetworkCapture> indicates the device supports starting packet capture. <isSupportStopNetworkCapture>true</isSupportStopNetworkCapture> indicates the device supports stopping packet capture. <isSupportGetNetworkCaptureStatus>true</isSupportGetNetworkCaptureStatus> indicates the device supports getting packet capture status.

2. Get capabilities of packet capture parameters: GET /ISAPI/System/NetworkCaptureParams/capabilities?format=json. The realTimeEnabled field indicates whether the device supports real-time packet capture.

3. Set the field realTimeEnabled as true in the parameters applied to the device to start device packet capture. The device returns an URI, and the client can download the real-time packet capture data from the device through a browser.

Start packet capture: POST /ISAPI/System/StartNetworkCapture?format=json&security=<security>&iv=<iv>.

Note:

- If `realTimeEnabled=true` is contained when starting device packet capture, it indicates packet capture data is uploaded in real-time by HTTP Chunked. The URL for downloading the returned packet capture data is valid for 30 seconds by default. If the download is attempted after this time, the device should return an HTTP 404 status code.

4. After starting packet capture, you can repeatedly get packet capture status, including whether packet capture is ongoing and the size of the packet capture data.

Get packet capture status: GET /ISAPI/System/GetNetworkCaptureStatus?format=json.

5. Packet capture can be stopped at any time by calling the interface of stopping packet capture.

Stop device packet capture: POST /ISAPI/System/StopNetworkCapture?format=json.

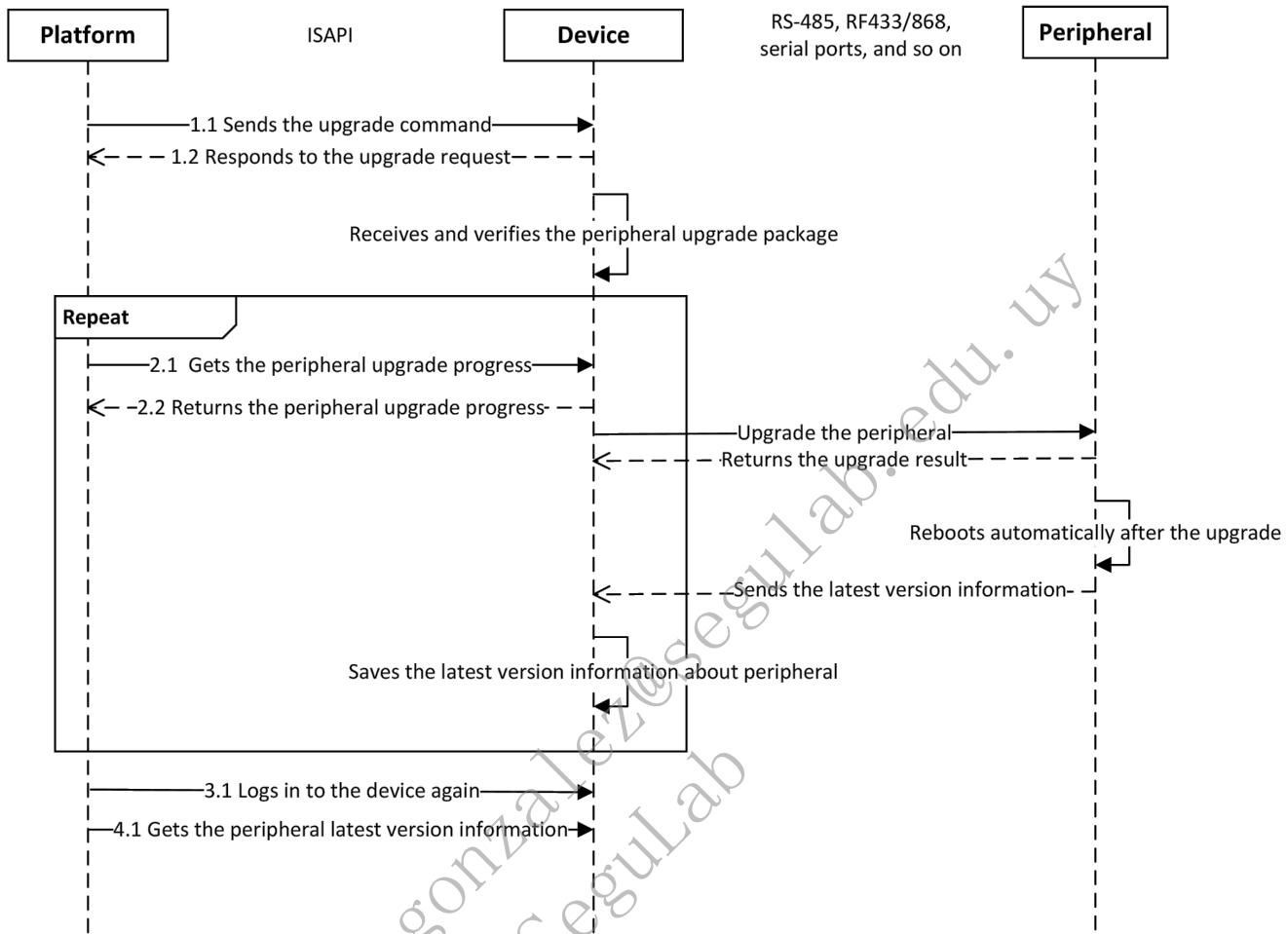
5.3 Device Peripherals Upgrade

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

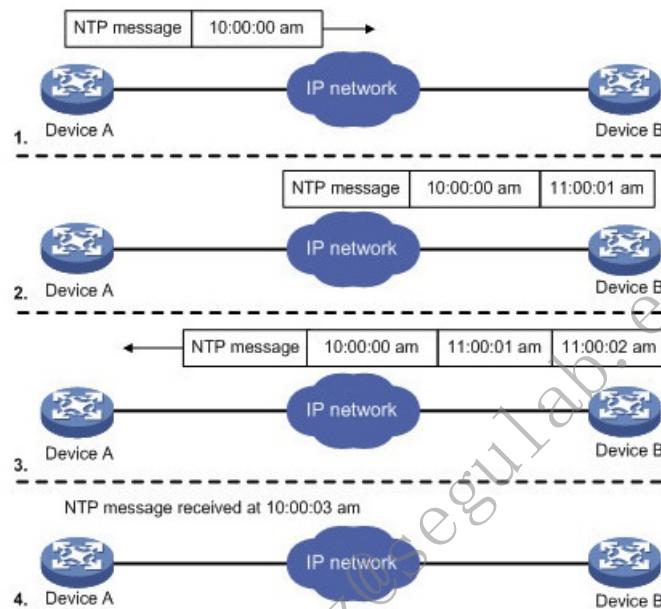
Time sync is a method to synchronize the time of all devices connecting to the NTP server, so that all devices can share the same clock time for providing related functions based on time. Supported time sync types: NTP time sync, manual sync, satellite time sync, platform time synchronization, etc. The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

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

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

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`.

Note: For manual time synchronization (time offset including time zone offset): Set manual time synchronization: `localTime` refers to the local time on device (time offset excluded, in format like `2019-02-28T10:50:44`); `timeZone` refers to the time offset of local time on device (time offset format with DST disabled: `CST-8:00:00`; time offset format with DST enabled: `CST-8:00:00DST00:30:00,M4.1.0/02:00:00,M10.5.0/02:00:00`); Get manual time synchronization: `localTime` refers to the local time on device (time offset included, in format like `2019-02-28T10:50:44+8:30`); `timeZone` refers to the time offset of local time on device (time offset format with DST disabled: `CST-8:00:00`; time offset format with DST enabled: `CST-8:00:00DST00:30:00,M4.1.0/02:00:00,M10.5.0/02:00:00`).

5.4.2.2 Time Zone Configuration

1. Get device time zone configuration capability

Call `GET /ISAPI/System/capabilities` to get the system capability. When `isSupportTimeZone` is returned, the time zone configuration is supported by the device.

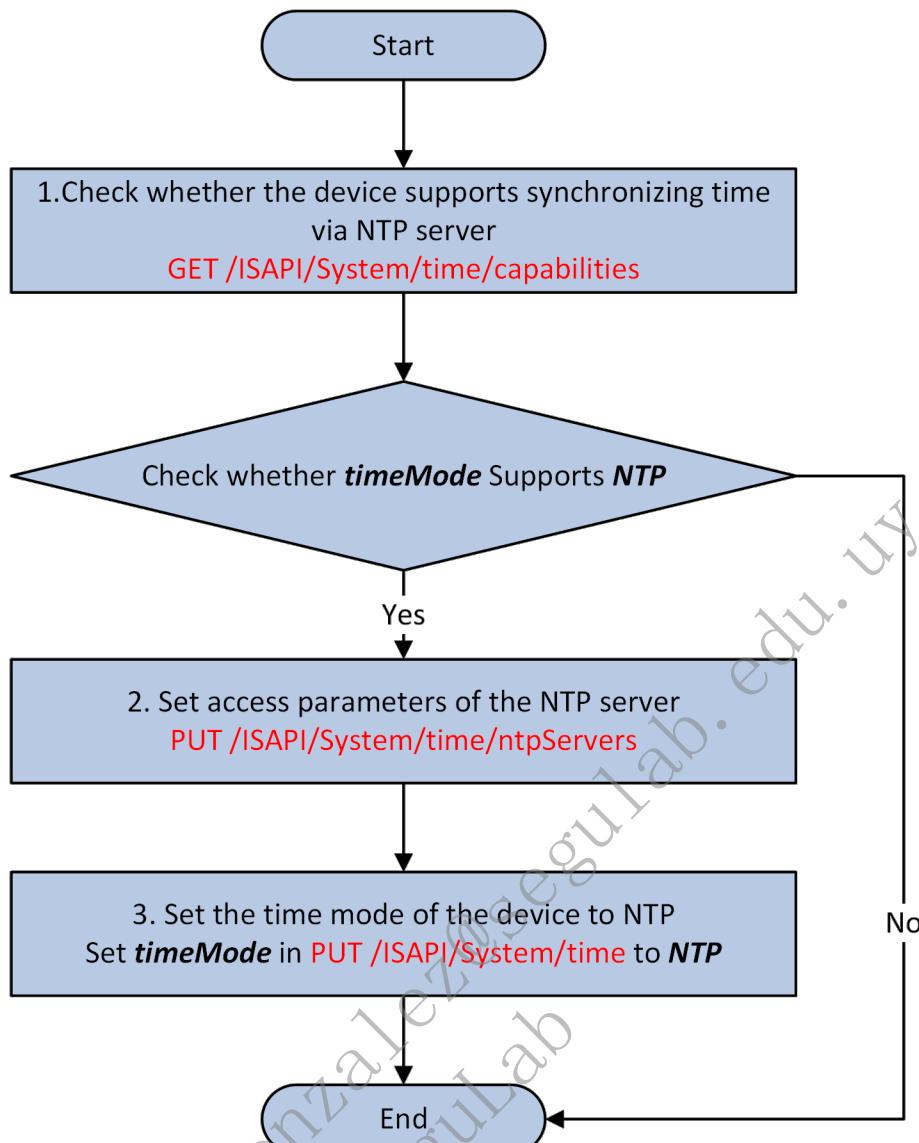
2. Configure time zone parameters

Get the device's time zone parameters: `GET /ISAPI/System/time/timeZone`. Set the device's time zone parameters: `PUT /ISAPI/System/time/timeZone`.

If DST (Daylight Saving Time) is disabled, the example of returned time zone parameters is: `CST-8:00:00`. It refers to UTC+8, and `-8:00:00` is the UTC local time. If DST (Daylight Saving Time) is enabled, the example of returned time zone parameters is: `CST-8:00:00DST00:30:00,M4.1.0/02:00:00,M10.5.2/02:00:00`. It refers to UTC+8, the DST time is 30 minutes ahead of local time, the DST starts at 02:00:00 on the first Sunday of April and ends at 02:00:00 on the fifth Tuesday of October. MX.Y.Z: X is the month, Y is the week number in the month, Z is the day of a week (0-Sunday, 1-Monday, 2-Tuesday, 3-Wednesday, 4-Thursday, 5-Friday, 6-Saturday).

5.4.2.3 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`

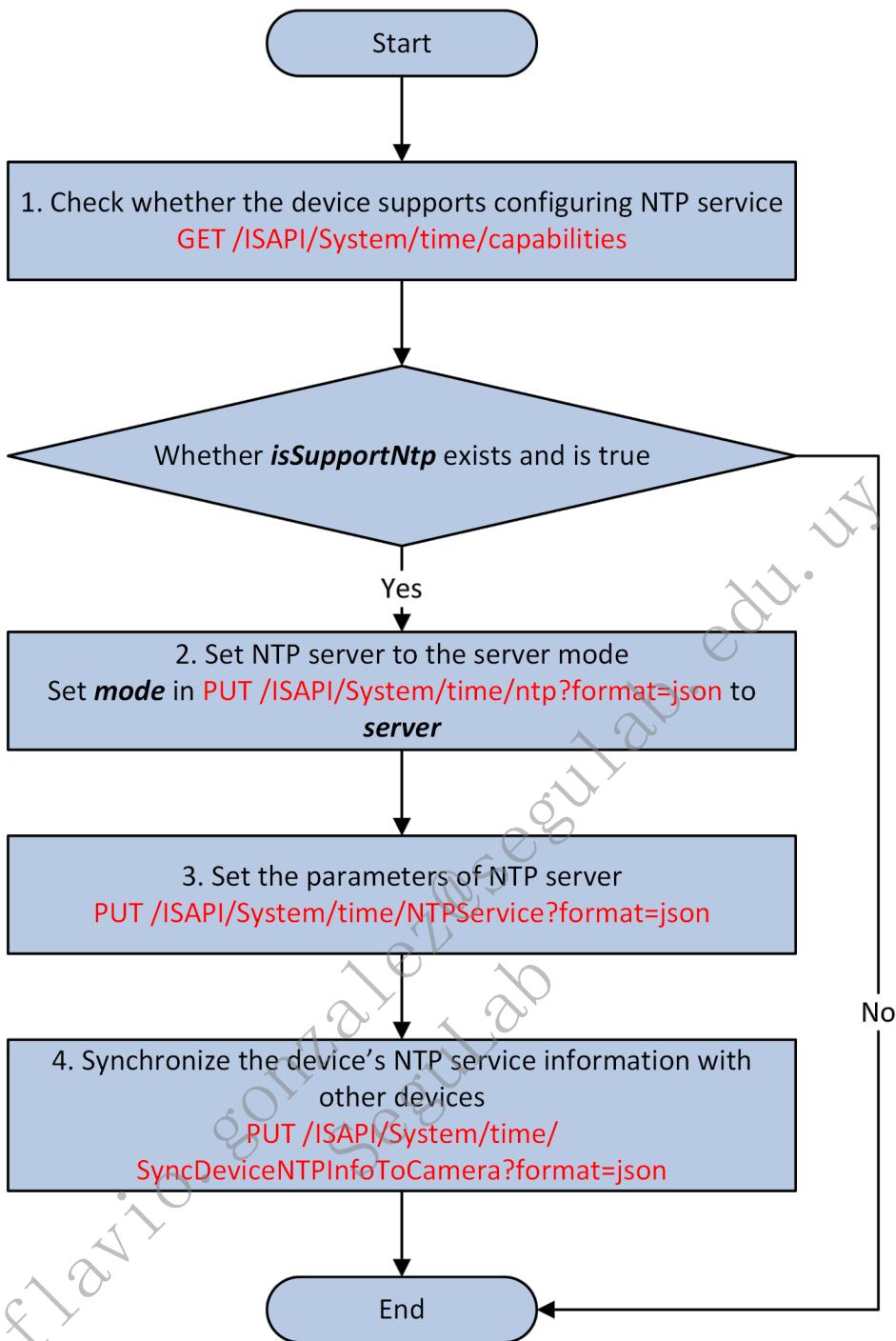
3. Set the time mode of the device to NTP

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`

5.4.2.4 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/SyncDeviceNTPInfoToCamera/capabilities?format=json

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

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

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

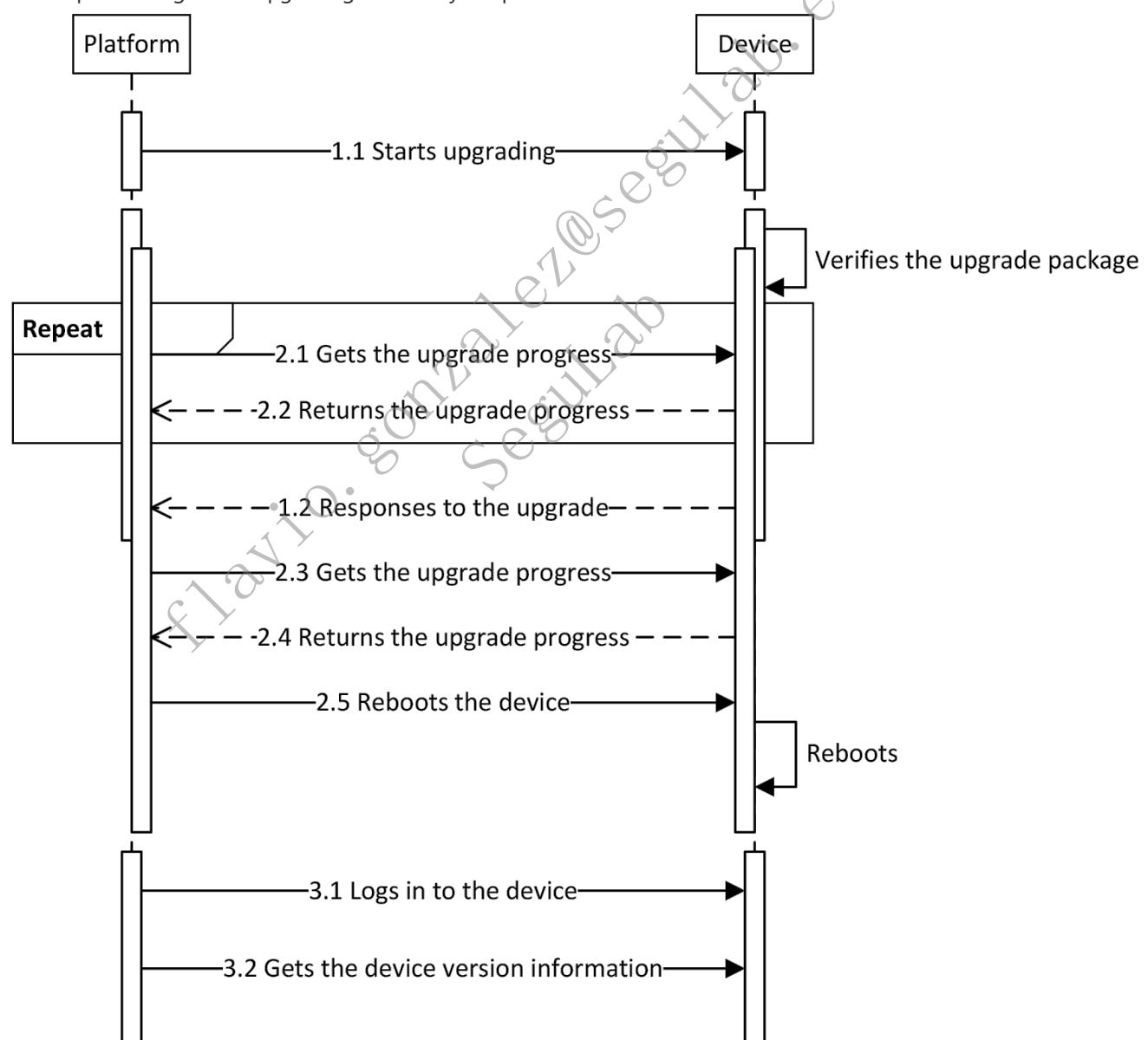
5.5 Device Upgrade

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 Mutually Exclusive Functions

5.6.1 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, i.e, only one of them is allowed at one time).

5.6.2 API Calling Flow

The following three APIs are available for the integration of mutually exclusive functions:

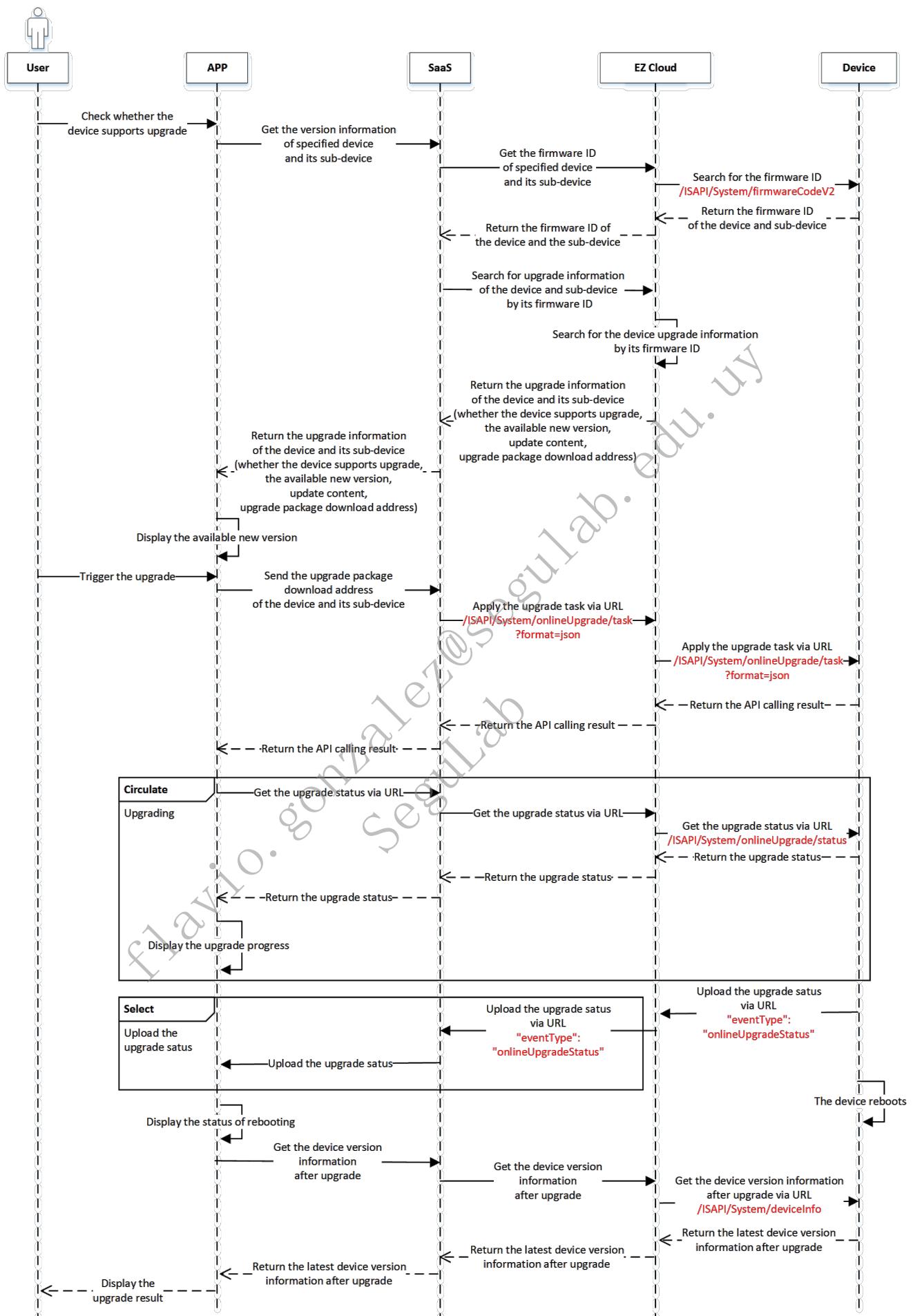
1. Get the information of mutually exclusive functions: GET /ISAPI/System/mutexFunction/capabilities?format=json. Call this URL to get the list of existing mutually exclusive functions supported by the device. Note: NVR devices only support setting exclusive function "perimeter" (perimeter protection), and do not support "linedetection" (line crossing detection), "fielddetection" (intrusion detection), "regionEntrance" (region entrance), or "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 and see whether to change the settings and disable the mutually exclusive 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.

5.7 Remote Online Upgrade Based on The EZ2.0

5.7.1 Introduction to the Function

Considering that the device is connected to EZ via EZ 2.0 Protocol, EZ OTA upgrade solution is not supported. This solution applies to specified version upgrade.

5.7.2 API Calling Flow



1. Search for the device firmware ID: GET `/ISAPI/System/firmwareCodeV2`.

2. Search for the device upgrade information by its firmware ID. If there is an available new version, apply the upgrade task to the device: POST `/ISAPI/System/onlineUpgrade/task?format=json`. You can specify to upgrade

device (e.g., NVR) or sub-device (e.g., IPC) via the node <channel>.

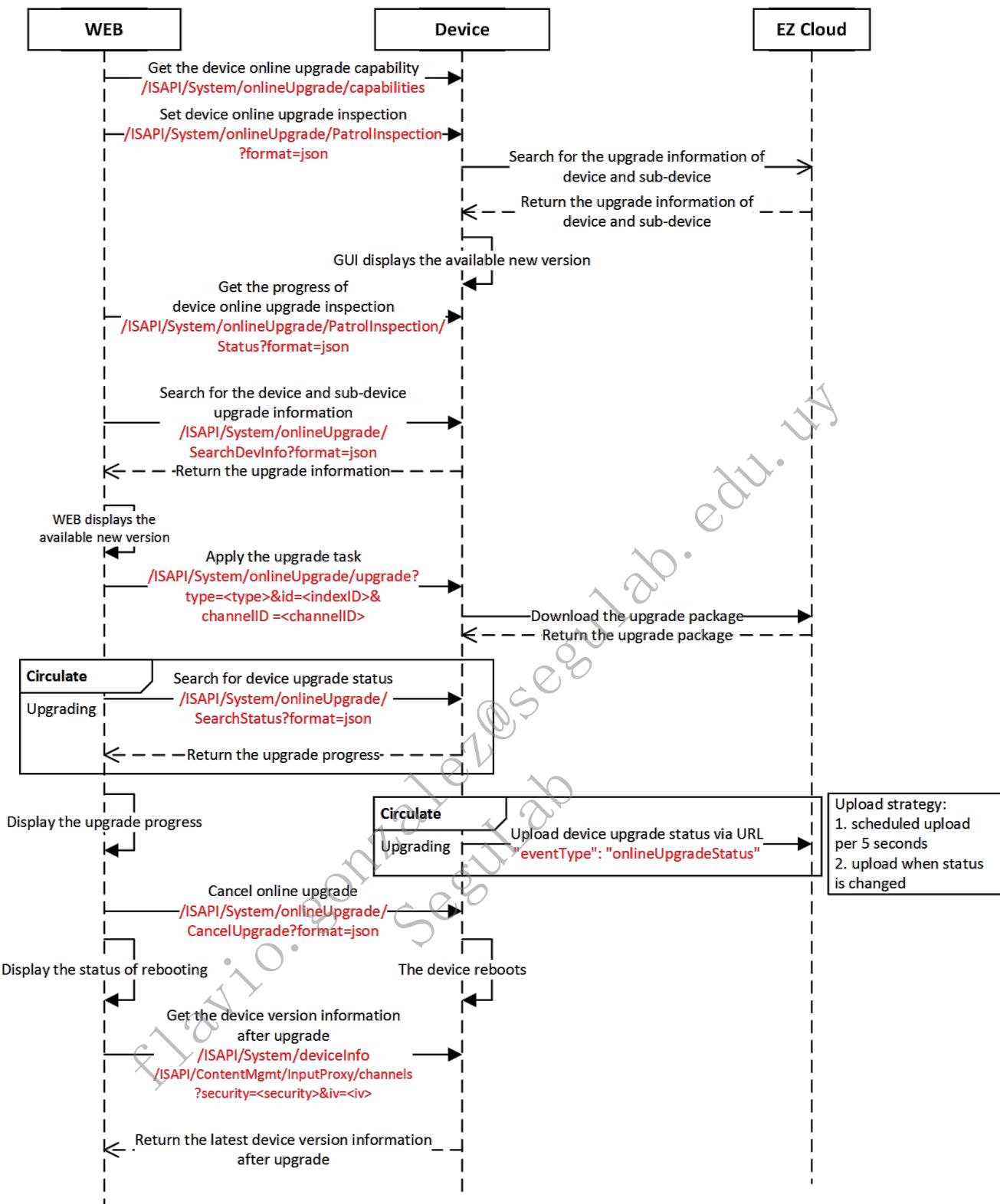
3. During the device upgrade, you can get the real-time upgrade progress on the platform: GET /ISAPI/System/onlineUpgrade/status.
4. After the device upgrade is finished, the device will upload the "onlineUpgradeStatus" and reboot automatically.
5. After both upgrade and reboot are finished, you can get the latest device version information on the platform.
 - Get the device (e.g., NVR) version information: GET /ISAPI/System/deviceInfo.
 - Get the sub-device (e.g., IPC) version information: GET /ISAPI/ContentMgmt/InputProxy/channels?security=<security>&iv=<iv>.

5.8 Remote Online Upgrade Based on the WEB Client

5.8.1 Introduction to the Function

This solution applies to remote online upgrade of the devices and sub-devices connected to WEB client.

5.8.2 API Calling Flow



1. Get device online upgrade capability: `GET /ISAPI/System/onlineUpgrade/capabilities`.
2. Manually trigger the device online upgrade inspection: `POST /ISAPI/System/onlineUpgrade/PatrolInspection?format=json`. After it is enabled, the device will search for the available new version.
3. During the inspection, the WEB client can get the real-time inspection progress: `GET /ISAPI/System/onlineUpgrade/PatrolInspection/Status?format=json`.
4. After the device inspection is finished, search for the device and sub-device upgrade information: `GET /ISAPI/System/onlineUpgrade/SearchDevInfo?format=json`.
5. Apply the upgrade task: `PUT /ISAPI/System/onlineUpgrade/upgrade?type=<type>&id=<indexID>&channelID = <channelID>`. You can upgrade a specified device via the node `<channelID>` in the URL. Specify sub-device upgrade by configuring `<type>`.

6. You can search for online upgrade status during the upgrade: GET /ISAPI/System/onlineUpgrade/SearchStatus?format=json.

7. You can cancel the upgrade during the upgrade: PUT /ISAPI/System/onlineUpgrade/CancelUpgrade?format=json.

8. After the upgrade and reboot are finished, you can get the latest device version information.

- Get the device (e.g., NVR) version information: GET /ISAPI/System/deviceInfo.
- Get the sub-device (e.g., IPC) version information: GET /ISAPI/ContentMgmt/InputProxy/channels?security=<security>&iv=<iv>.

5.9 Serial Port Accessed External Device Management

5.9.1 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.

5.9.2 API Calling Flow

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

2. Set the information of 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.

3. 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.

4. 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.

5.10 Serial Port Data Transparent Transmission

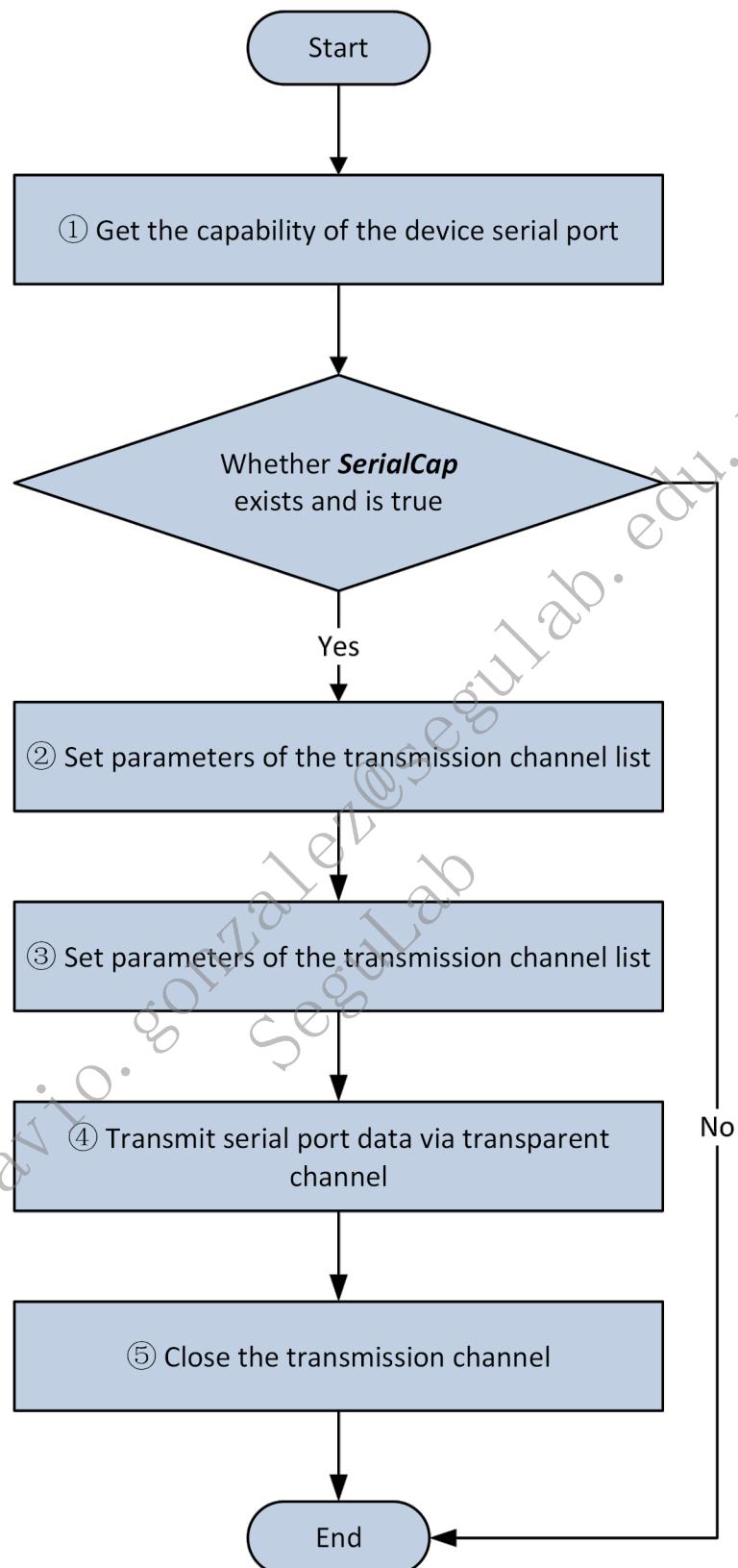
5.10.1 Introduction to the Function

RS485, RS422 and RS232 serial ports external to the device are used as transparent channels to transmit serial port data. Supports the client sending serial data to the device, which then forwards the data to the serial port; conversely, when the serial port sends data to the device, which transparently transmits the data to the client. Note that the device's transparent transmission of serial data is half-duplex communication, meaning that bidirectional communication can be implemented between the device and the client, but the communication data is not necessarily in a question-and-answer format, and the client must match the request and response relationship on its own.

Since most standard HTTP request libraries do not support the persistent connection for receiving and sending data in real time, it is recommended that the client uses two TCP clients to implement the sending and receiving of

transparently transmitted serial data.

5.10.2 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.

5.11 Serial Port Parameter Configuration

5.11.1 Introduction to the Function

Serial port parameter configuration.

5.11.2 API Calling Flow

1. Check whether the device supports configuring serial port parameters.

Get the capability of device serial port: `GET /ISAPI/System/capabilities`. If is returned and its value is true, 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 a 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.

5.12 Sub-device Batch Upgrade

5.12.1 Introduction to the Function

Sub-device Batch Upgrade (Single Task): applicable to situations of upgrading multiple sub-devices using one upgrade package. Application scenarios include: when the UWB positioning anchor connects to the web, upgrading multiple tags (sub-devices) through the positioning engine (gateway), with the same upgrade package.

Management of Sub-device Batch Upgrade Tasks: There are many types of sub-devices, such as LoRa nodes, which have a slow upgrade process and different models of LoRa nodes in the field also require different upgrade packages. Therefore, it is necessary to support the creation of multiple upgrade tasks at once, entrusting the upgrade to the devices, to enhance the practicality of batch upgrading sub-devices.

5.12.2 API Calling Flow

Sub-device Batch Upgrade (Single Task)

Get the device capability: /ISAPI/System/capabilities.

If the node `isSupportBulkUpgradeChildDevice` exists and is true, it indicates that the device support the function.

If the node `isSupportSearchBulkUpgradeChildDeviceProgress` exists and is true, it indicates that the device support getting the progress of batch upgrading sub-devices.

Sub-device batch upgrade: `POST /ISAPI/System/BulkUpgradeChildDeviceList?format=json`.

Batch sub-device upgrade progress search: `POST /ISAPI/System/BulkUpgradeChildDeviceList/Search?format=json`.

Sub-device Batch Upgrade Task Management

Get device system capabilities `/ISAPI/System/capabilities`, If the node `isSupportBulkUpgradeChildDevice` exists and is true, it indicates that batch upgrade of sub devices is supported,

If the node `isSupportSearchBulkUpgradeChildDeviceProgress` exists and is true, it indicates that progress of batch upgrading sub devices can be searched,

If the node `isSupportSearchBulkUpgradeChildDeviceTask` exists and is true, it indicates that batch upgrading devices can be searched,

If the node `isSupportModifyBulkUpgradeChildDeviceTask` exists and is true, it indicates that batch upgrading sub devices can be edited,

If the node `isSupportDeleteBulkUpgradeChildDeviceTask` exists and is true, it indicates that task of batch upgrading sub devices can be deleted.

Get the capability of batch upgrading sub-device: `GET /ISAPI/System/BulkUpgradeChildDevice/capabilities?format=json`.

Batch upgrade sub devices: `POST /ISAPI/System/BulkUpgradeChildDeviceList?format=json`.****

Search for tasks of batch sub-device upgrade: `POST /ISAPI/System/ModifyBulkUpgradeChildDeviceTask?format=json`.

Edit tasks of batch sub-device upgrade: `PUT /ISAPI/System/ModifyBulkUpgradeChildDeviceTask?format=json`.****

Delete tasks of batch sub-device upgrade: `PUT /ISAPI/System/DeleteBulkUpgradeChildDeviceTask?format=json`.

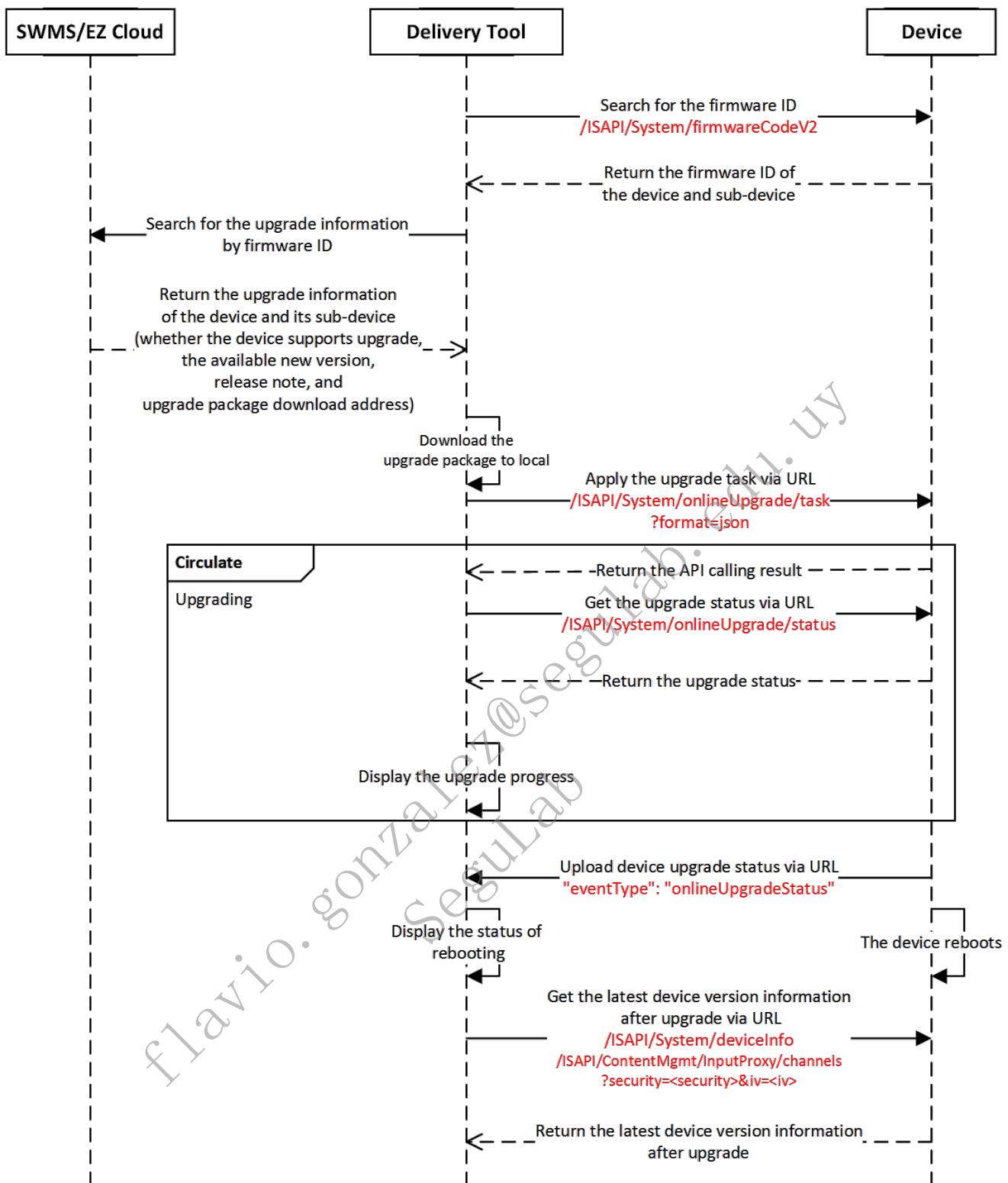
Note: 1. The two methods for batch upgrading child devices are the same API: `POST /ISAPI/System/BulkUpgradeChildDeviceList?format=json`. If the `BulkUpgradeChildDeviceListCap.taskID` is returned in `GET /ISAPI/System/BulkUpgradeChildDevice/capabilities?format=json`, it indicates that the task method is supported in the sub device upgrade. 2. To determine which sub devices can be upgraded, you can use the API `POST /ISAPI/IoTGateway/Childmanage/SearchChild?format=json`. Sub devices with the tag `isUpgradable` and value `true` indicate that the sub devices support upgrades.

5.13 Upgrade Based on Interactive Tools Local Deployment

5.13.1 Introduction to the Function

This solution applies to the upgrade of devices and sub-devices connected via ISAPI. If the device is connected via ISAPI protocols, it is required to set up HTTP(s) service to store device upgrade package and generate URL for downloading upgrade package. If the device is connected via HCNetSDK, you can implement the device upgrade via transmitting ISAPI.

5.13.2 API Calling Flow



1. Search for the device firmware ID: `GET /ISAPI/System/firmwareCodeV2`.
2. Search for the device upgrade information by firmware ID on the SWMS/EZ Cloud, download upgrade package to local storage, and apply the upgrade task to the device: `POST /ISAPI/System/onlineUpgrade/task?format=json`. You can specify to upgrade device (e.g., NVR) or sub-device (e.g., camera) via the node `<channel>`.
3. During the upgrade, you can get the real-time upgrade progress: `GET /ISAPI/System/onlineUpgrade/status`.
4. After upgrade is completed, the device will upload the `"onlineUpgradeStatus"` (upgrade status), and reboot automatically.
5. After the device rebooting, you can get the latest device version information.
 - Get the device (e.g., NVR) version information: `GET /ISAPI/System/deviceInfo`.
 - Get the sub-device (e.g., IPC) version information: `GET /ISAPI/ContentMgmt/InputProxy/channels?security=`

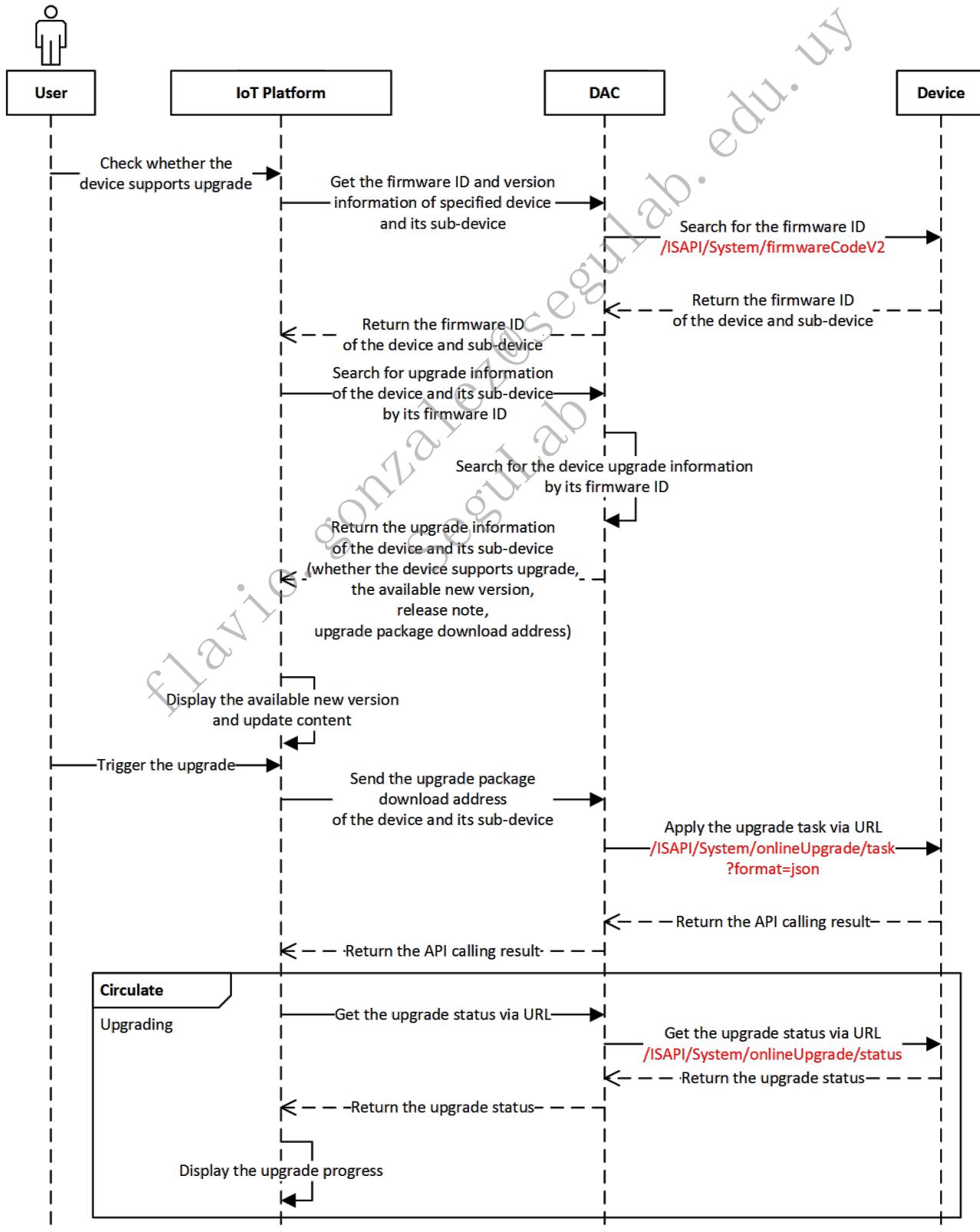
<security>&iv=<iv>.

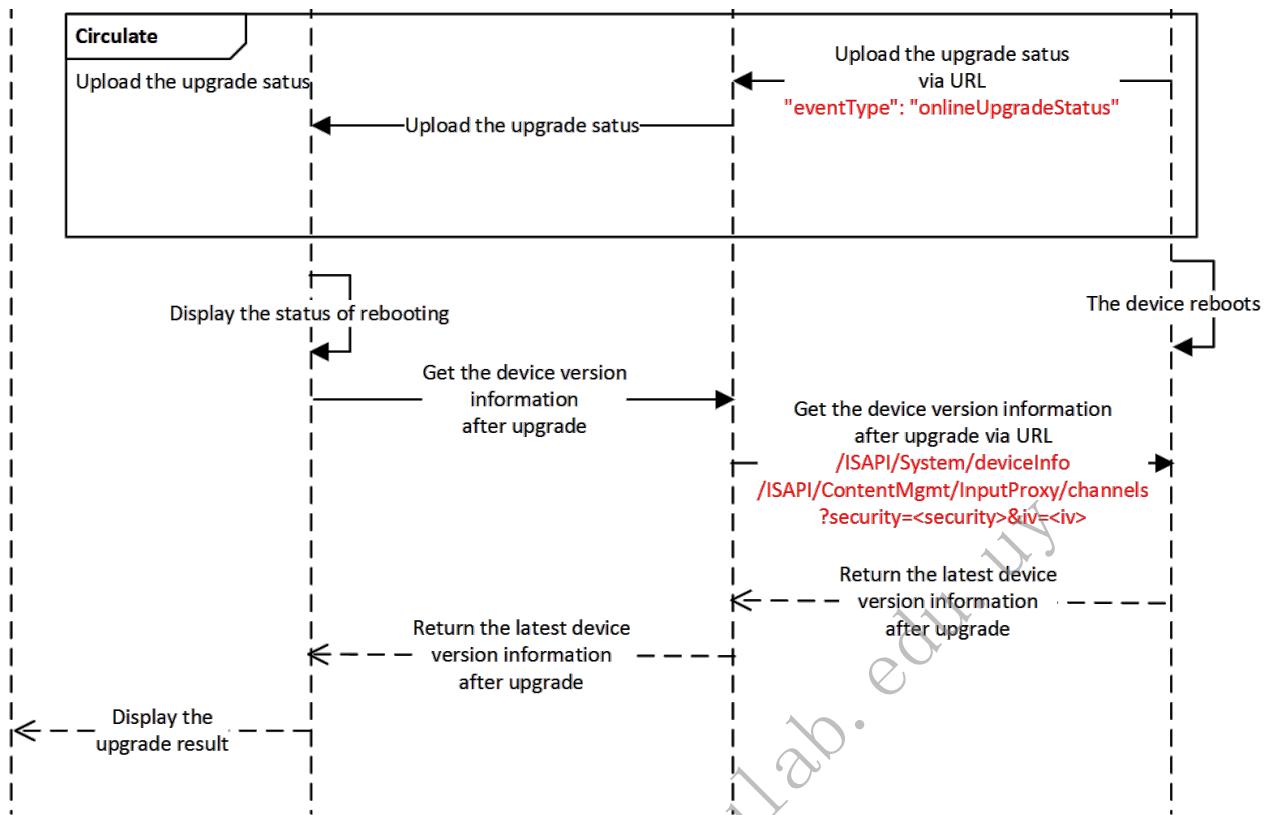
5.14 Upgrade Based on Platform Local Deployment

5.14.1 Introduction to the Function

This solution applies to the upgrade of devices and sub-devices connected via ISAPI and ISUP. If via ISAPI, it is required to set up HTTP(s) service or transfer binary upgrade package files so as to store device upgrade package and generate URL for downloading upgrade package. If via ISUP, it is required to set up FTP service to store device upgrade package and generate URL for downloading upgrade package. If the device is connected via HCNetSDK protocols, you can realize device upgrade via transmitting ISAPI.

5.14.2 API Calling Flow





1. Search for device firmware ID on the platform: `GET /ISAPI/System/firmwareCodeV2`.
2. Search for the device upgrade information in DAC by device firmware ID on the platform. If there is an available new version, DAC will trigger an upgrade task and apply it to the device: `POST /ISAPI/System/onlineUpgrade/task?format=json`. You can specify to upgrade device (e.g., NVR) or sub-device (e.g., IPC) via the node `<channel>`.
3. During the upgrade, you can get the real-time upgrade progress on the platform: `GET /ISAPI/System/onlineUpgrade/status`.
4. After the device upgrade is finished, the device will upload the "onlineUpgradeStatus" (upgrade status), and will reboot automatically.
5. After the upgrade and reboot are finished, you can get the latest device version information on the platform.
 - Get the device (e.g., NVR) version information: `GET /ISAPI/System/deviceInfo`.
 - Get the sub-device (e.g., IPC) version information: `GET /ISAPI/ContentMgmt/InputProxy/channels?security=<security>&iv=<iv>`.

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>/overlays/capabilities.`
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:

- o 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.

- o 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 Async Capturing Manually

6.2.1 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.

6.2.2 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.

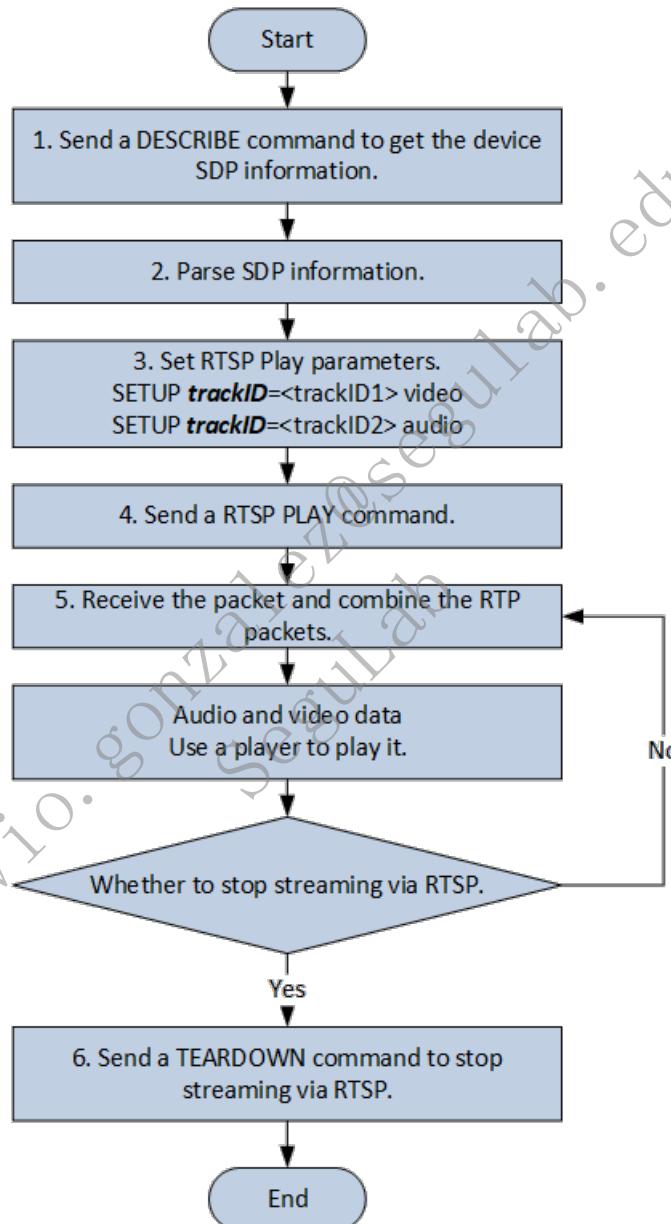
6.3 Live View

6.3.1 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](#)).

6.3.2 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.
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 URLs 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`.

6.3.3. 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:NKPlayer-1.0.00.081112
```

2. Server responds that authentication is required.

```
RTSP/1.0 401 Unauthorized
CSeq: 0
WWW-Authenticate: Digest realm="3521781c29ach312330dd668", 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="3521781c29actb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/channels/101",
response="76a2c9c5b8edbd49838013cf1cf27941"
User-Agent:NJKPlayer-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:NKPlayer-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-1.00.00.081112
```

```
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=0.00000-0.00000
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
```

6.4 Manual Capturing Events

6.4.1 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.

6.4.2 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.

6.5 Motion Detection

6.5.1 Introduction to the Function

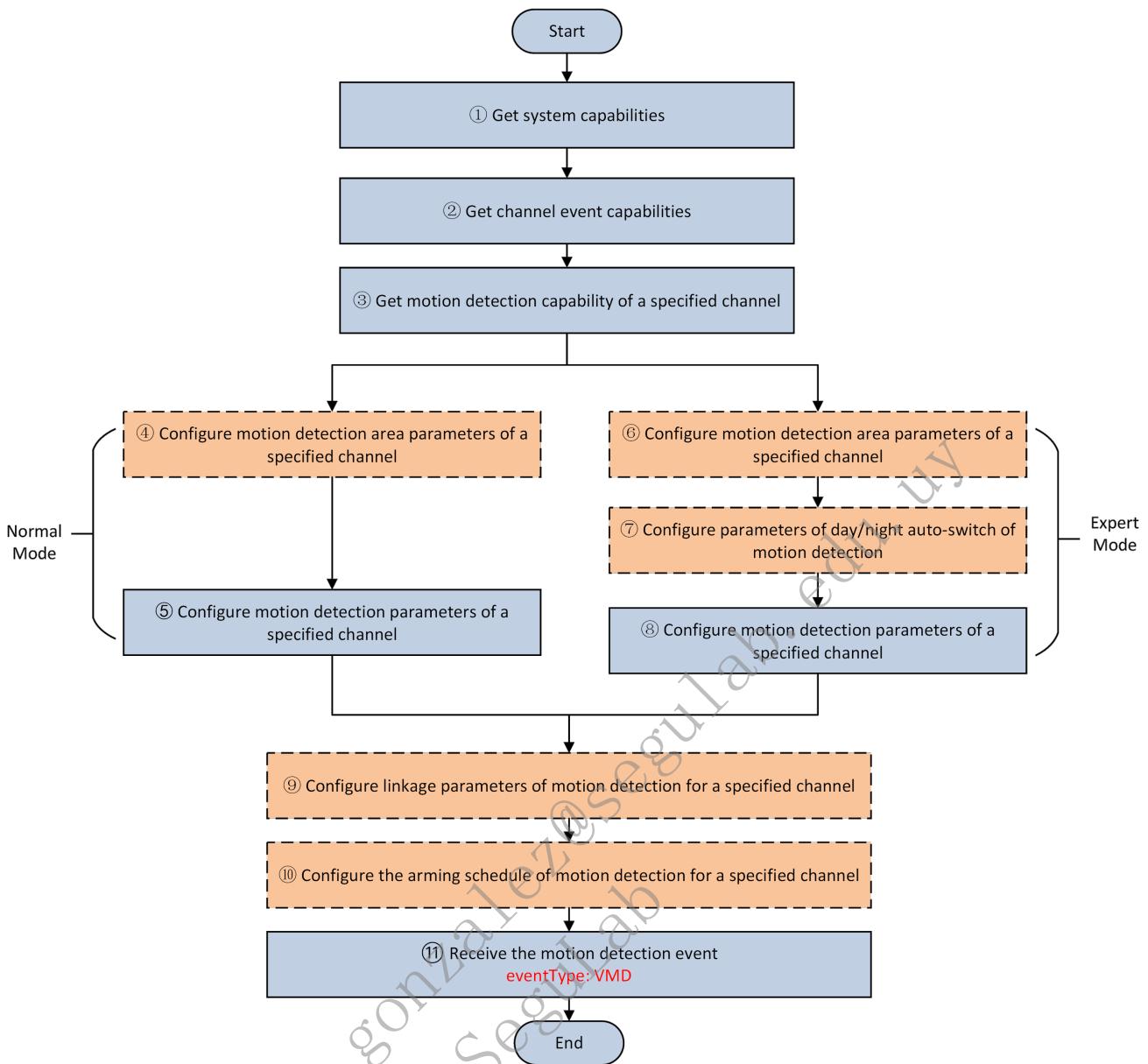
Commonly used in unattended camera recording and automatic alarm, motion detection can help to reduce the cost of manual monitoring for public institutions and companies. Besides, it can greatly improve monitoring efficiency and accuracy by avoiding the errors of manual monitoring due to fatigue after 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.

6.5.2 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.

6.5.3 Message Example

6.5.3.1 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>
```

6.5.3.2 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>
```

6.6 Sync Capturing Manually

6.6.1 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.

6.6.2 API Calling Flow

- (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`.

7 Video Management

7.1 . Scheduled Recording

7.1.1 Introduction to the Function

The web interface for reference is shown below.

The screenshot shows the 'Record Schedule' configuration for camera [D6] Camera_01. The 'Continuous' recording mode is selected. The schedule grid shows a repeating pattern of 24-hour cycles from Monday to Sunday, with each hour marked from 00 to 24. The 'Save' and 'Copy to...' buttons are at the bottom.

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>.

7.2 Search for Event Video

7.2.1 Introduction to the Function

This function supports searching for the videos of linked cameras when an event occurs.

7.2.2 API Calling Flow

- Check whether the device supports searching for event videos.

Get the device sysgem capability: GET /ISAPI/System/capabilities. If the node <SerialCap> is returned and its value is true, the device supports searching for event videos.

- Search for event videos: POST /ISAPI/ContentMgmt/eventRecordSearch?format=json.

7.3 Search for Videos and Pictures

7.3.1 Introduction to the Function

This function supports searching for the video/picture data stored in the device. Video data source includes scheduled recording, manual recording and event recording, and picture data source includes scheduled capture and event linkage capture.

7.3.2 API Calling Flow

- Check the device's supported time format. Get device's system capabilities: GET /ISAPI/System/capabilities.
 - If node `isSupportTimeTypeSTD` is returned with value `true`, it indicates that the device supports the ISO 8601 standard time, and URLs of functions related to ISO 8601 time parameters support additional query parameter `timeType=STD`.
 - If node `isSupportTimeTypeSTD` is not returned. Call GET /ISAPI/System/time/timeType?format=json to get the time type. If the `type` field returns `local`, it indicates that the device video search and playback use local time. If the `type` field returns `UTC`, it indicates that the device video search and playback use ISO 8601 time format.
 - If the device does not support API GET /ISAPI/System/time/timeType?format=json. Call GET /ISAPI/System/deviceInfo and check the time type of device via field `deviceType`. If `deviceType` (device type) is CVR, NVR, DVR, or ITCCAM, the device video search and playback use local time. Otherwise, follow the protocol specifications and use the TZ format, for example, `2024-05-13T16:00:00Z`.
- Check whether the device supports video/picture search. Get the device's video search capability via GET /ISAPI/ContentMgmt/capabilities. If `<isSupportRecordSearch>` is returned, it indicates that the device supports video/picture search.
- Search for videos and pictures:
 - Get the capability of searching for videos and pictures: GET /ISAPI/ContentMgmt/capabilities. Get the picture search types that the device supports searching via the node `<pictureSearchType>` returned in the message, including pictures of event capture, scheduled capture and manual capture. Get the video types that

the device supports searching for via the node <recordSearchType> returned in the message.

- ii. Get the parameters capability of searching for videos and pictures: GET /ISAPI/ContentMgmt/search/capabilities. Get the periods when the device supports searching for videos and pictures via the node <timeSpanList> returned in the message. Get the file tags that the device supports searching via the node <labelSearch> returned in the message.
- iii. Search for videos and pictures: POST /ISAPI/ContentMgmt/search?security=<security>&iv=<iv>.
 - o The URL containing timeType=STD indicates that the time in the input and output messages are according to the ISO 8601 time format. The URL without timeType=STD indicates to use device's original time.
 - o Check whether it is picture or video to be searched via the node <trackID> in the request message <CMSearchDescription>. "101" (search for the main stream videos of channel 1), "103" (search for the pictures of channel 1), and it is channel 1 by default .

Note:

- o The value of CMSearchDescription.trackIDList.trackID is obtained from the id value returned in the response message of GET /ISAPI/ContentMgmt/record/tracks. The meaning of id: 101 is the main stream of channel 1, 102 is the sub-stream of channel 1, 103 is the capture schedule of channel 1, 201 is the main stream of channel 2, and so on. When the device saves only one copy of channel videos, it does not distinguish between the main and sub stream, and only supports setting trackID to xx1.
- o If <trackIDList> is returned in response message of GET /ISAPI/ContentMgmt/search/capabilities, it indicates that the device supports accurate search, and the returned value in <trackIDList> indicates the channels that support independent video search.

7.4 Search for Videos by Target Type

7.4.1 Introduction to the Function

This function realizes persons, vehicles or mixed targets distinguishing when searching for motion detection event videos. Searching for videos by target type is newly added. The search conditions include target type information, and the returned result of the search contains the start and end time as well as the file name of the video.

7.4.2 API Calling Flow

1. Check whether the device supports searching by target type:

Get the device capability: GET /ISAPI/System/capabilities. If the node is returned and its value is true, it indicates that the device supports searching by target type. Currently, the device supports searching by person and by vehicle.

2. Get the capability of searching by target type: GET /ISAPI/ContentMgmt/SearchByTargetType/capabilities?format=json.

3. Search for videos by target type: POST /ISAPI/ContentMgmt/SearchByTargetType?format=json.

8 Vehicle Recognition

8.1 Motor Vehicle Recognition

8.1.1 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.

8.1.2. Device Event Upload

8.1.2.1 Check Whether Device Supports Baseline Vehicle Recognition Event

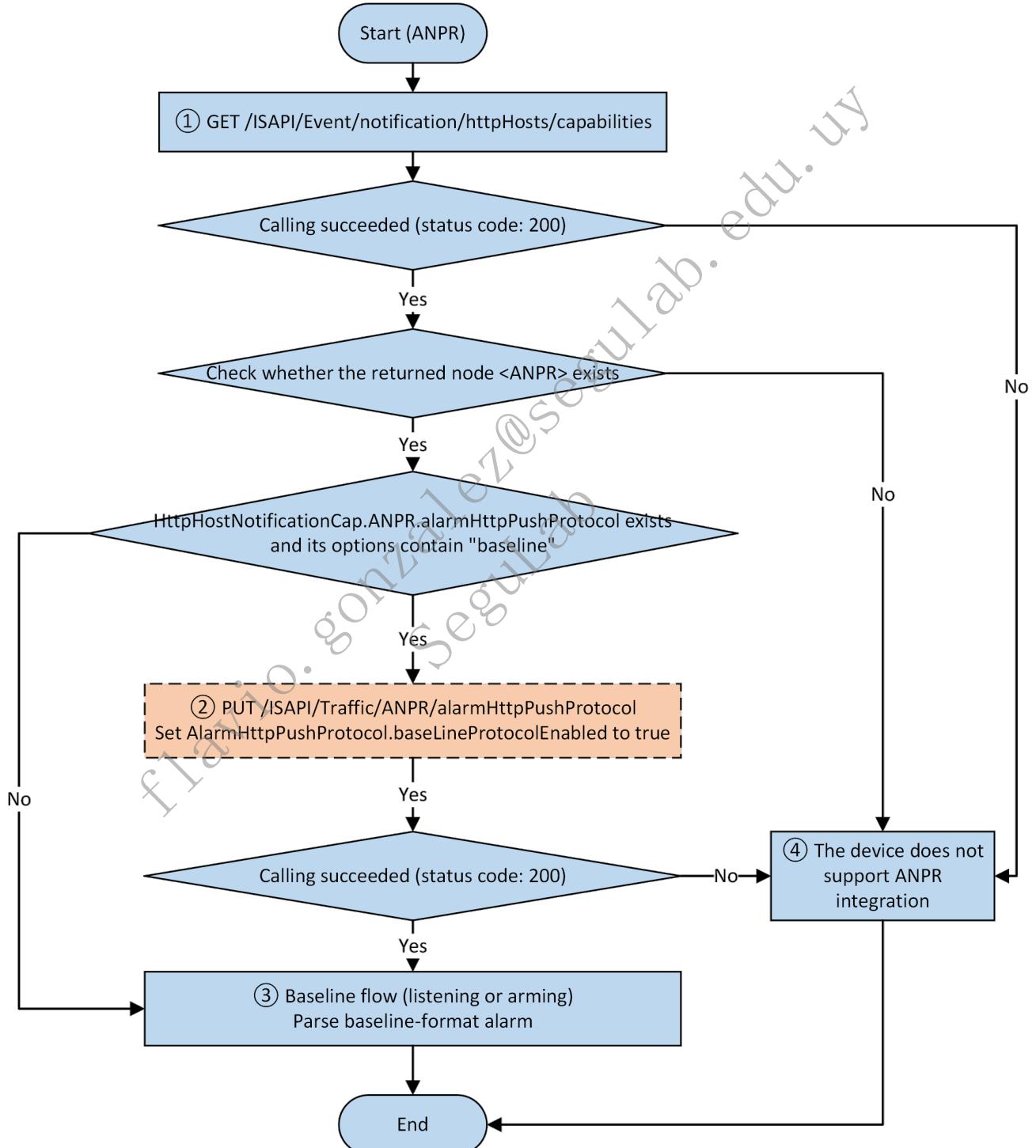
The integrator get the capability from device.

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.

8.1.2.2 Check Whether Device Supports Baseline Flow



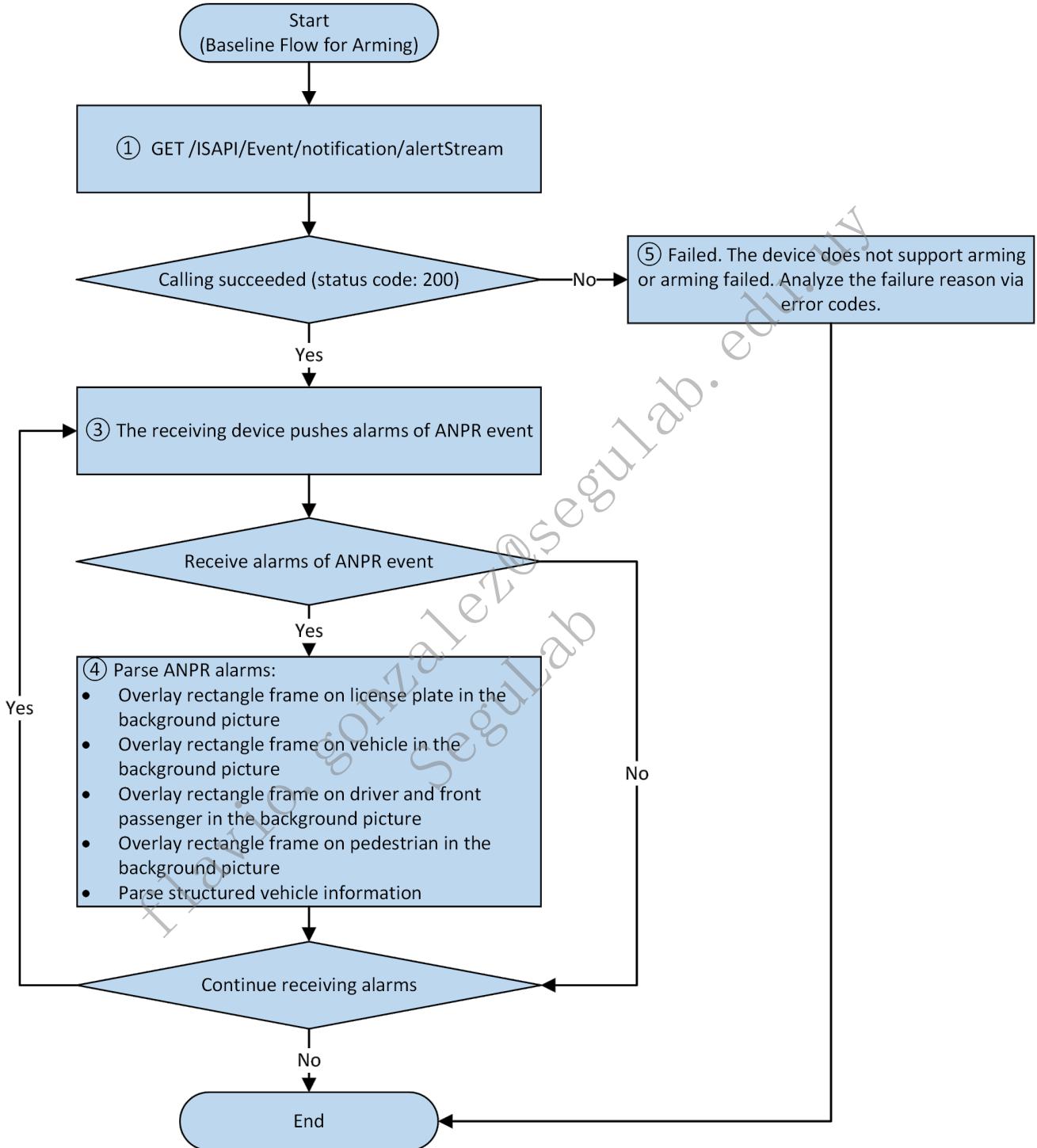
1. (Client) Get the configuration capability of listening host via GET /ISAPI/Event/notification/httpHosts/capabilities.
2. (Client) Configure the alarm mode switch for device via PUT /ISAPI/Traffic/ANPR/alarmHttpPushProtocol and set <AlarmHttpPushProtocol>.baseLineProtocolEnabled to true.
3. Receive 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.

4. The ANPR function integration is not supported by device.

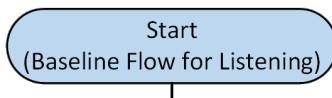
8.1.3 Baseline Flow Event Upload

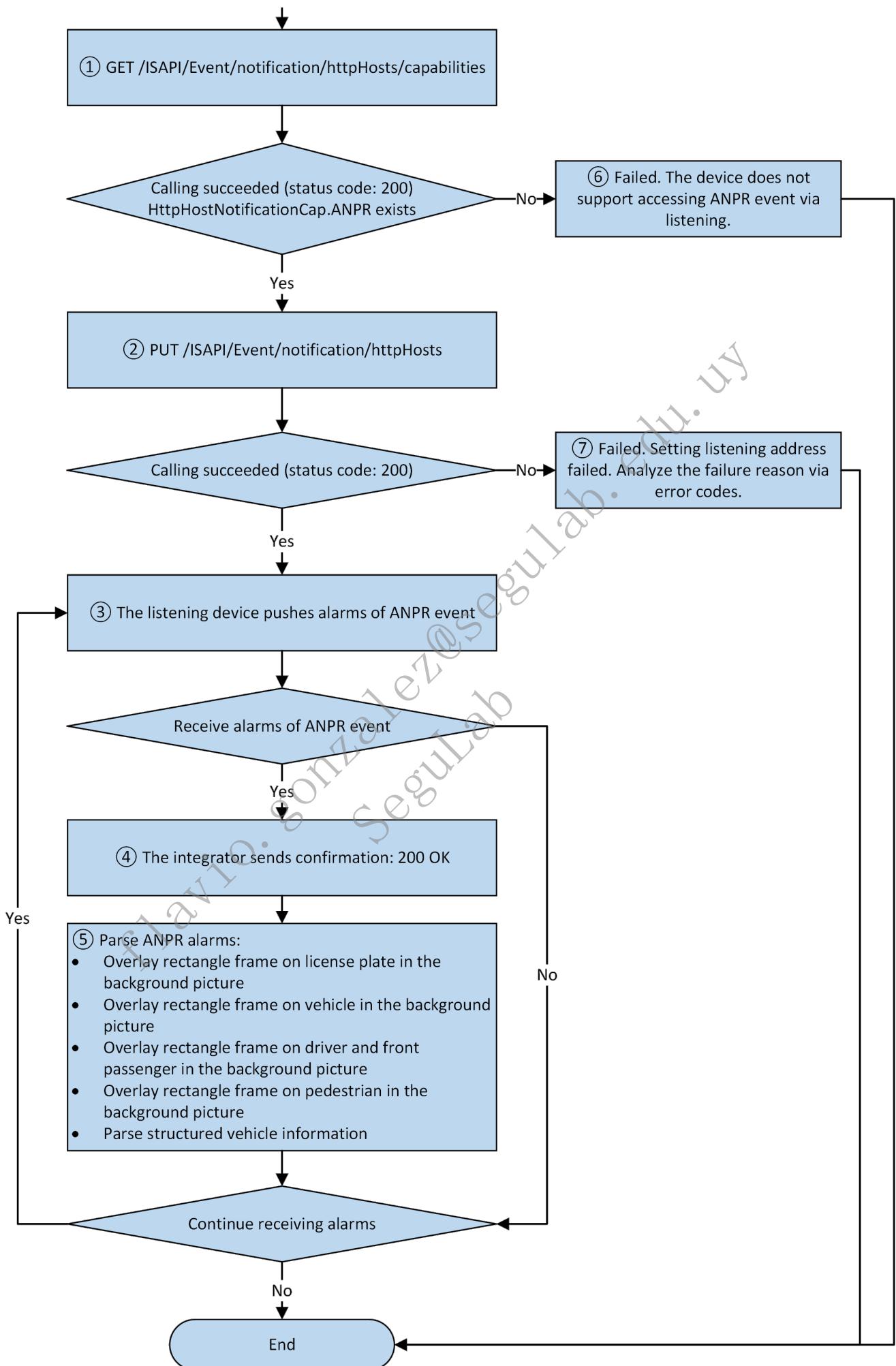
8.1.3.1 Arming Flow



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.

8.1.3.2 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.

8.1.4 Baseline Flow Message Format and Example

8.1.4.1 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>
```

8.1.2.2 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

9 Traffic Control

9.1 Traffic Enforcement of Vehicles

9.1.1 Introduction to the Function

9.1.1.1 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.

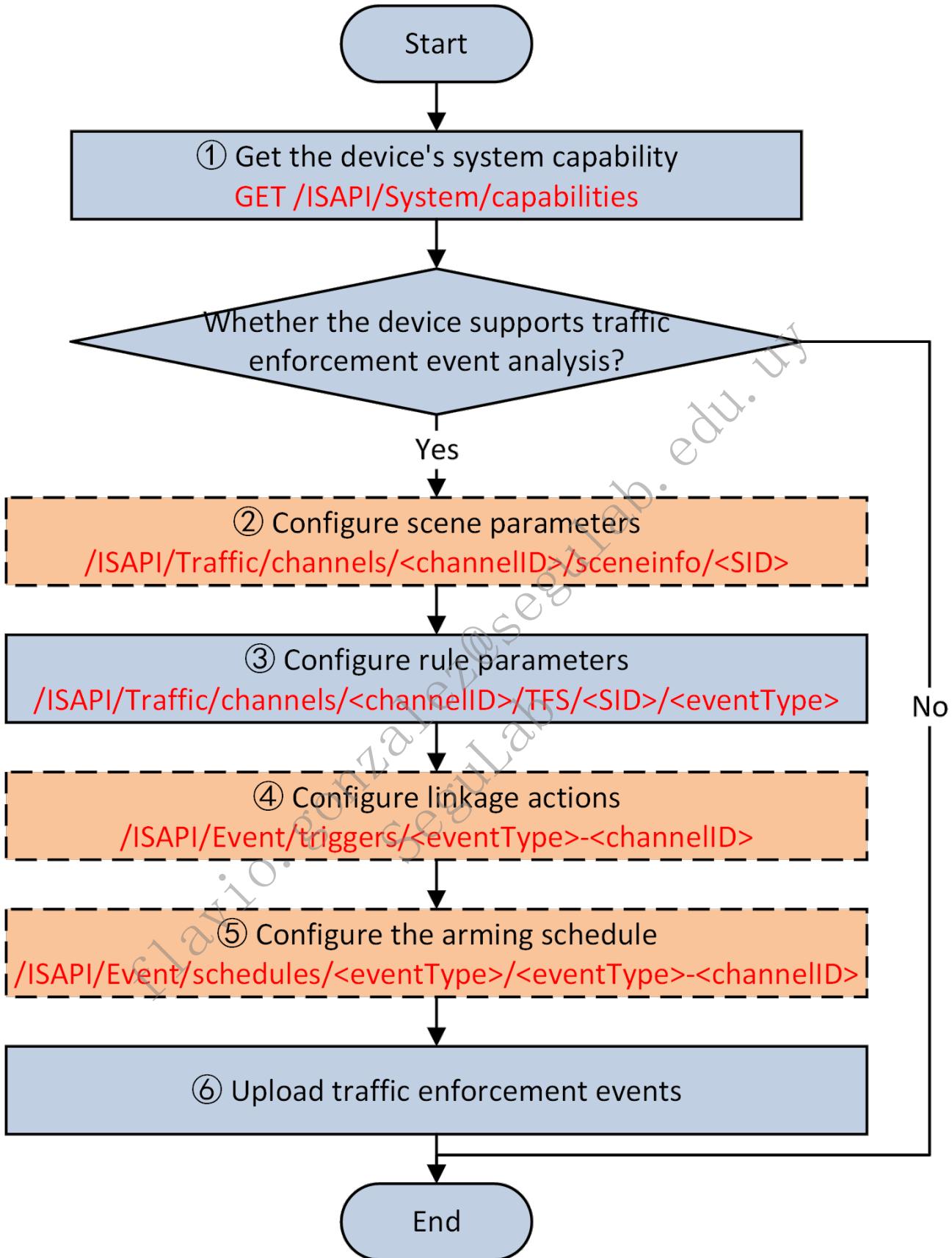
9.1.1.2 Smart Linkage System for Parking Management

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.

9.1.2 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.

9.1.3 Exception Handling

9.1.3.1 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.	

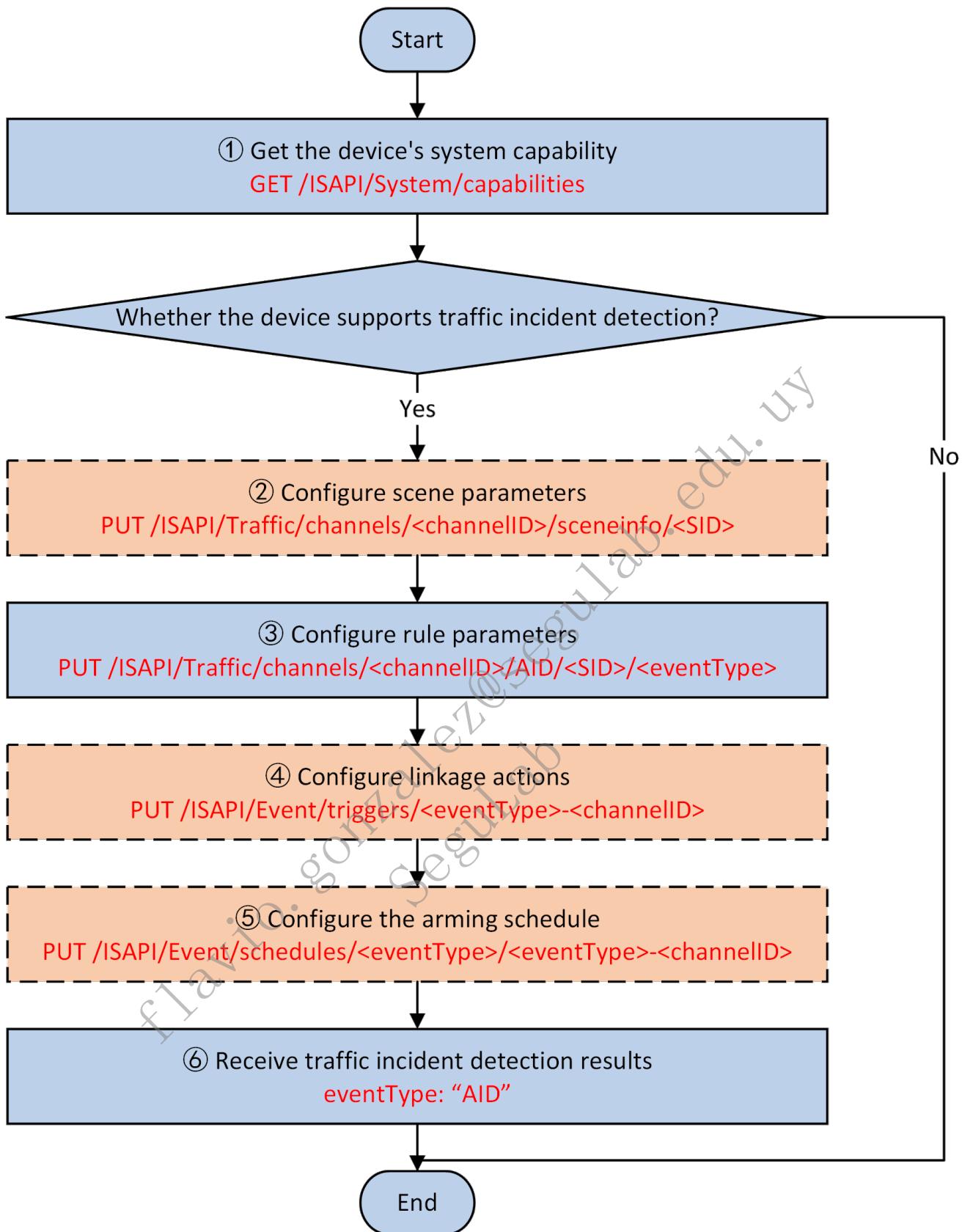
9.2 Traffic Incidents of Vehicles

9.2.1 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.

9.2.2 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.

9.2.3 Exception Handling

9.2.3.1 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.	

10 Video Recognition

10.1 Professional Smart Abnormal Event Detection

10.1.1 Introduction to the Function

Professional Smart Abnormal Event Detection includes the following features:

- Line Crossing Detection
- Intrusion Detection
- Region Entrance Detection
- Region Exiting Detection
- Loitering Detection
- People Gathering Detection
- Rapid Movement Detection
- Parking Detection
- Unattended Baggage Detection
- Object Removal Detection

- Running Detection
- Abnormal Number of People Detection
- Person Spacing Change Detection
- Violent Motion Detection
- On/Off Duty Detection
- Falling Down Detection
- Lingering Detection
- Person Density Detection
- Climbing Detection
- Trend Analysis
- In-Toilet Overtime Detection
- Sudden Change of Sound Intensity Detection
- Student Standing Up Detection
- Getting Up from Bed Detection

Professional smart abnormal event detection is designed for specific scenarios: workplace activity detection and pedestrian activity detection.

Workplace activity detection includes but not limited to:

- Using a Mobile Phone
- Number of People Changes
- On/Off Duty
- Person Lingering

Pedestrian activity detection includes but not limited to:

- Running Detection
- People Gathering Detection
- Violent Motion Detection
- Person Falling Detection

You can configure workplace activity detection and pedestrian activity detection rules and alarm linkage actions to suit your needs.

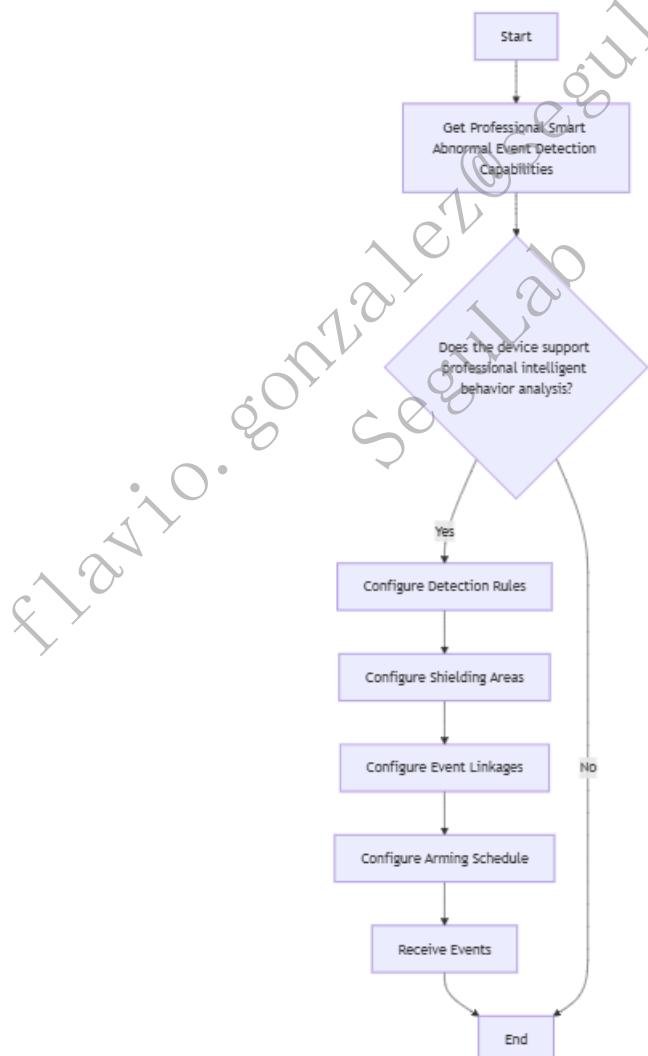
Abnormal event detection can be categorized into two types: Basic smart abnormal event detection and professional smart abnormal event detection. Their differences are as follows:

1. **Algorithms:** Traditional algorithms are used for basic smart abnormal event detection, while deep learning algorithms are used for professional smart abnormal event detection.
2. **Detection Effect:** Basic smart abnormal event detection can only analyze targets as people or vehicles. Professional smart abnormal event detection can not only distinguish between targets as people or vehicles but also identify specific behaviors, such as the number of people and human behaviors such as using a mobile phone, violent motion, and running.
3. **Supported Events:** Smart Basic Intelligent Abnormal Behavior Detection supports events such as:
 - Line Crossing Detection
 - Intrusion Detection
 - Region Entrance Detection
 - Region Exiting Detection
 - Loitering Detection
 - People Gathering Detection
 - Rapid Movement Detection
 - Parking Detection
 - Unattended Baggage Detection
 - Object Removal Detection

Professional Smart Abnormal Event Detection supports all the events mentioned above and additional events including:

- Running Detection
- Abnormal Number of People Detection
- Person Spacing Change Detection
- Violent Motion Detection
- On/Off Duty Detection
- Falling Down Detection
- Lingering Detection
- Person Density Detection
- Climbing Detection
- Trend Analysis
- In-Toilet Overtime Detection
- Sudden Change of Sound Intensity Detection
- Student Standing Up Detection
- Getting Up from Bed Detection

10.1.2 API Calling Flow



Before taking the following steps, ensure that this smart detection has been enabled for devices.

1. Get Professional Smart Abnormal Event Detection Capabilities

- Call `GET /ISAPI/Intelligent/capabilities` to get professional smart abnormal event detection capabilities and the field `isBehaviorSupport` is returned as `true` in the response message.
- Call `GET /ISAPI/Intelligent/channels/<channelID>/capabilities` to get professional smart abnormal

event detection capabilities for a specified channel and the field `isBehaviorSupport` is returned as `true` in the response message.

Note: `isBehaviorSupport` as `true` indicates support for professional smart abnormal event detection. The fields of supported events are as follows:

- `isLineDetectionSupport` indicates support for **Line Crossing Detection**.
- `isFieldDetectionSupport` indicates support for **Intrusion Detection**.
- `isRegionEntranceSupport` indicates support for **Region Entrance Detection**.
- `isRegionExitingSupport` indicates support for **Region Exiting Detection**.
- `isLoiteringSupport` indicates support for **Loitering Detection**.
- `isGroupSupport` indicates support for **People Gathering Detection**.
- `isRapidMoveSupport` indicates support for **Rapid Movement Detection**.
- `isParkingSupport` indicates support for **Parking Detection**.
- `isUnattendedBaggageSupport` indicates support for **Unattended Baggage Detection**.
- `isAttendedBaggageSupport` indicates support for **Object Removal Detection**.
- `isRunningSupport` indicates support for **Running Detection**.
- `isPeopleNumChangeSupport` indicates support for **Abnormal Number of People Detection**.
- `isSpacingChangeSupport` indicates support for **Person Spacing Change Detection**.
- `isViolentMotionSupport` indicates support for **Violent Movement Detection**.
- `isLeavePositionSupport` indicates support for **On/Off Duty Detection**.
- `isFallDownSupport` indicates support for ** Falling Down Detection**.
- `isRetentionParamSupport` indicates support for **Lingering Detection**.
- `isSupportPersonDensityDetection` indicates support for **Person Density Detection**.
- `isVcaEventAdvReachHeightSupport` indicates support for **Climbing Detection**.
- `isSupportSituationAnalysis` indicates support for **Trend Analysis**.
- `isSupportToiletTarry` indicates support for **In-Toilet Overtime Detection**.
- `isSupportAudioAbnormal` indicates support for **Sudden Change of Sound Intensity Detection**.
- `isSupportStandUp` indicates support for **Student Standing Up Detection**.
- `isSupportGetUp` indicates support for **Getting Up from Bed Detection**.

2. Configure Detection Rules

- Call `GET /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/rule/capabilities` to get professional smart abnormal event detection capabilities.
- Call `GET /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/rule/<ruleID>` to get detection rules.
- Call `PUT /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/rule/<ruleID>` to configure detection rules.
- Call `DELETE /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/rule/<ruleID>` to delete detection rules.
- Call `GET /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>` to get all detection rules.
- Call `PUT /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>` to configure all detection rules.

Note:

- The `eventType` field in the rules indicates the corresponding event type.
- The `ruleType` field indicates the event support drawing a detection line or a polygon detection area. Some events (sudden change of sound intensity) do not support drawing a box or line, some events (line crossing, climbing) support drawing a line, and some events (except sudden change of sound intensity, line crossing, and climbing) support for polygon detection areas.

3. (Optional) Configure Shielding Areas

- Call `GET /ISAPI/Intelligent/channels/<channelID>/Shield/capabilities` to get professional smart area shielding capabilities.
- Call `GET /ISAPI/Intelligent/channels/<channelID>/Shield` to get shielding areas for all events.
- Call `PUT /ISAPI/Intelligent/channels/<channelID>/Shield` to configure shielding areas for all events.

4. (Optional) Configure Event Linkages

- Call `GET /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/notifications` to get event linkages.
- Call `PUT /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/notifications` to configure event linkages.

5. (Optional) Configure Arming Schedule

- Call `GET /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/schedules` to get the schedule to disable the professional smart abnormal event detection rules.
- Call `PUT /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/schedules` to configure the schedule to disable the professional smart abnormal event detection rules.

Note:

- Up to eight time periods can be configured per day.
- Each two time periods cannot overlap.

6. Receive Events

- Line Crossing Detection: `"eventType": "linedetection"`
- Intrusion Detection: `"eventType": "fielddetection"`
- Region Entrance Detection: `"eventType": "regionEntrance"`
- Region Exiting Detection: `"eventType": "regionExiting"`
- Loitering Detection: `"eventType": "loitering"`
- People Gathering Detection: `"eventType": "group"`
- Rapid Movement Detection: `"eventType": "rapidMove"`
- Parking Detection: `"eventType": "parking"`
- Unattended Baggage Detection: `"eventType": "unattendedBaggage"`
- Object Removal Detection: `"eventType": "attendedBaggage"`
- Running Detection: `"eventType": "running"`
- Abnormal Number of People Detection: `"eventType": "peopleNumChange"`
- Person Spacing Change Detection: `"eventType": "spacingChange"`
- Violent Movement Detection: `"eventType": "violentMotion"`
- On/Off Duty Detection: `"eventType": "leavePosition"`
- Falling Down Detection: `"eventType": "failDown"`
- Lingering Detection: `"eventType": "retention"`
- Person Density Event: `"eventType": "personDensityDetection"`
- Climbing Detection: `"eventType": "advReachHeight"`
- Trend Analysis: `"eventType": "situationAnalysis"`
- In-Toilet Overtime Detection: `"eventType": "toiletTarry"`
- Sudden Change of Sound Intensity Detection: `"eventType": "audioAbnormal"`
- Student Standing Up Detection: `"eventType": "standUp"`
- Getting Up from Bed Detection: `"eventType": "getUp"`
- Sitting Still Detection: `"eventType": "sitQuietly"`
- Key Target Standing Up Detection: `"eventType": "keyPersonGetUp"`
- Using Mobile Phone Detection: `"eventType": "playCellphone"`

- Yard Lingering: "eventType": "yardTarry"
- Key Target Standing Up Detection: "eventType": "keyPersonGetUp"
- Sitting Still Detection: "eventType": "sitQuietly"
- Yard Lingering: "eventType": "yardTarry"
- Using Mobile Phone Detection: "eventType": "playCellphone"

Note:

- The content of `targetAttrs` is in JSON format. When the event format is XML, use **CDATA (Character Data)** to include this data in the event and transmit it to the platform.
- **CDATA (Character Data)** is used to include large amounts of text data in XML documents. This text may contain special characters (such as <, >, &), which have special meanings in XML and typically need to be escaped. CDATA blocks allow these special characters to be included directly in the document without escaping, such as including JavaScript scripts or CSS styles in XML documents.

10.2 Upgrading the Deep Learning Model for Perimeter Intrusion Detection

10.2.1 Introduction to the Function

You can upgrade the deep learning model for perimeter intrusion detection. You can also upgrade both the local training model for NVRs and the training model for network camera channels.

10.2.2 API Calling Flow

1. Determine if the device supports upgrading the deep learning model for perimeter intrusion detection: Call `GET /ISAPI/Intelligent/capabilities`. If the field `isSupportUpgradeDepthPerimeterModel` is returned as `true` in the response message, this function is supported.
2. Get the deep learning model upgrade capabilities: `GET /ISAPI/Intelligent/depthPerimeter/upgradeTrainModel/capabilities?format=json`
3. Set the model version: `PUT /ISAPI/Intelligent/depthPerimeter/upgradeTrainModel/VersionOperation?format=json`
 - You cannot specify a version to upgrade. Only rolling back to the previous version and restoring to the factory model version are currently supported.
4. Get the model upgrade progress: `GET /ISAPI/Intelligent/depthPerimeter/upgradeTrainModel/process?format=json`
5. Set the model version for a channel: `PUT /ISAPI/Intelligent/depthPerimeter/channels/<channelID>/upgradeTrainModel/VersionOperation?format=json`
6. Get the model upgrade progress for a channel: `GET /ISAPI/Intelligent/depthPerimeter/channels/<channelID>/upgradeTrainModel/process?format=json`

11 Two-Way Audio

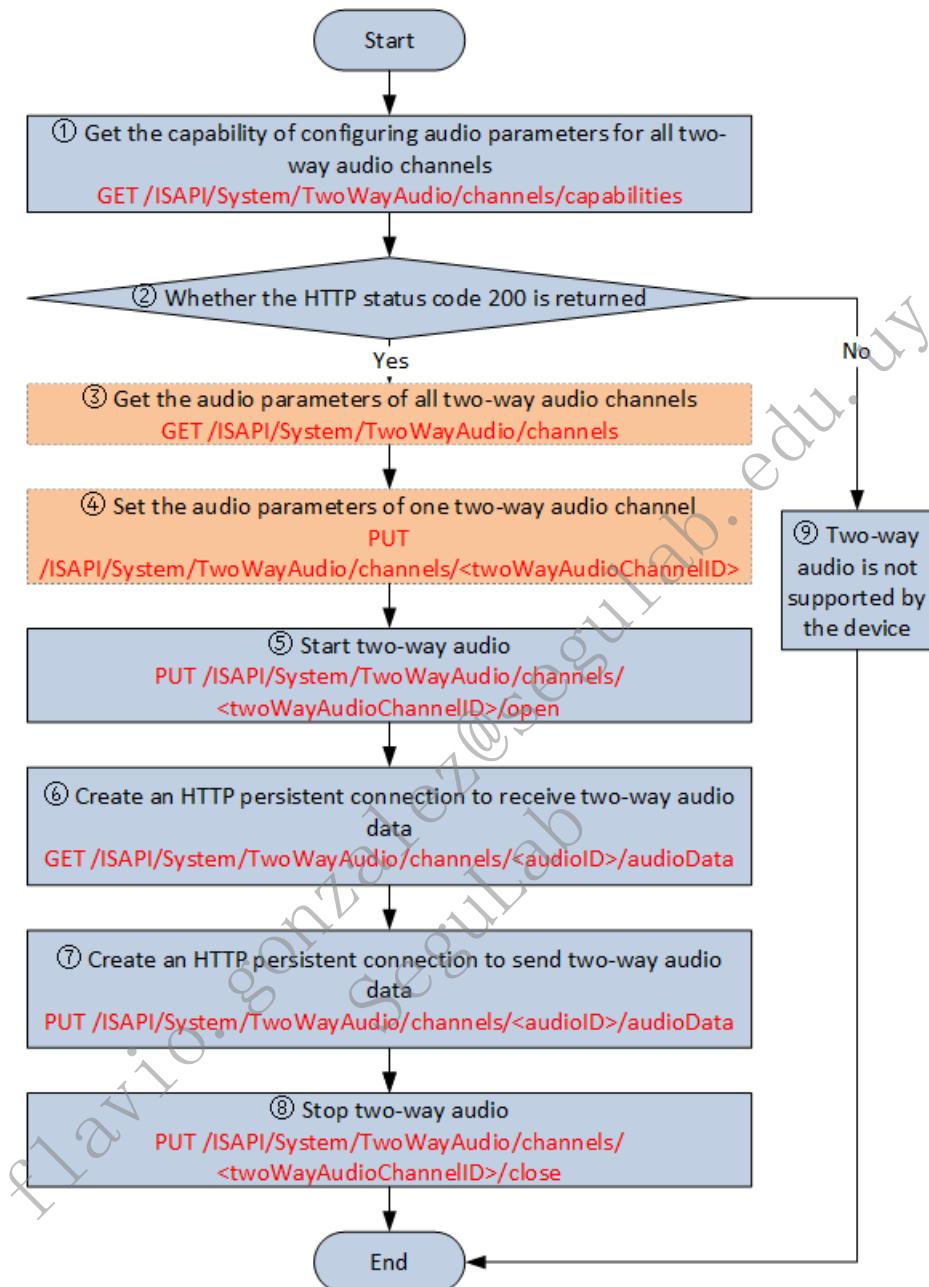
11.1 Two-Way Audio

11.1.1 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.

11.1.2 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

11.1.3 Exception Handling

11.1.3.1 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.	Th or is all Tv au in pr Pl ac af tw au st ►

11.1.4 Message Format and Example

11.1.4.1 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.
```

11.1.4.2 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.
```

12 API Reference

12.1 Device (General)

12.1.1 Device Exception Detection

12.1.1.1 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>
  <inputIOPortID>
    <!--ro, opt, string-->test
  </inputIOPortID>
  <dynInputIOPortID>
    <!--ro, opt, string-->test
  </dynInputIOPortID>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <intervalBetweenEvents>
    <!--ro, opt, int, unit:s-->1
  </intervalBetweenEvents>
  <WLSensorID>
    <!--ro, opt, string-->1
  </WLSensorID>
  <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>
      <notificationInterval>
        <!--ro, opt, int, unit:ms-->1
      </notificationInterval>
      <outputIOPortID>
        <!--ro, opt, string-->test
      </outputIOPortID>
      <dynOutputIOPortID>
        <!--ro, opt, string-->test
      </dynOutputIOPortID>
      <videoInputID>
        <!--ro, opt, string-->1
      </videoInputID>
      <dynVideoInputID>
        <!--ro, opt, string-->1
      </dynVideoInputID>
      <ptzAction>
        <!--ro, opt, object-->
        <ptzChannelID>
          <!--ro, req, string-->test
        </ptzChannelID>
        <actionName>
          <!--ro, req, enum, subType:string-->preset
        </actionName>
        <actionNum>
          <!--ro, opt, int-->1
        </actionNum>
        <presetDurationTime>
          <!--ro, opt, int, unit:s, dep:and, ${$.EventTrigger.EventTriggerNotificationList[*].EventTriggerNotification.ptzAction.actionName, eq, preset}-->1
        </presetDurationTime>
      </ptzAction>
      <cellphoneNumber>
        <!--ro, opt, string, range:[0,11]-->test
      </cellphoneNumber>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

12.1.1.2 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>
    <inputIOPortID>
        <!--opt, string-->test
    </inputIOPortID>
    <dynInputIOPortID>
        <!--opt, string-->test
    </dynInputIOPortID>
    <videoInputChannelID>
        <!--opt, string-->1
    </videoInputChannelID>
    <dynVideoInputChannelID>
        <!--opt, string-->1
    </dynVideoInputChannelID>
    <intervalBetweenEvents>
        <!--opt, int, unit:s-->1
    </intervalBetweenEvents>
    <WLSensorID>
        <!--opt, string-->1
    </WLSensorID>
    <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>
            <notificationInterval>
                <!--opt, int, unit:ms-->1
            </notificationInterval>
            <outputIOPortID>
                <!--opt, string-->test
            </outputIOPortID>
            <dynOutputIOPortID>
                <!--opt, string-->test
            </dynOutputIOPortID>
            <videoInputID>
                <!--opt, string-->1
            </videoInputID>
            <dynVideoInputID>
                <!--opt, string-->1
            </dynVideoInputID>
            <ptzAction>
                <!--opt, object-->
                <ptzChannelID>
                    <!--req, string-->test
                </ptzChannelID>
                <actionName>
                    <!--req, enum, subType:string-->preset
                </actionName>
                <actionNum>
                    <!--opt, int-->1
                </actionNum>
                <presetDurationTime>
                    <!--opt, int, unit:s, dep:and, ${.EventTrigger.EventTriggerNotificationList[*].EventTriggerNotification.ptzAction.actionName, eq, preset}-->1
                </presetDurationTime>
            </ptzAction>
            <cellphoneNumber>
                <!--opt, string, range:[0,11]-->test
            </cellphoneNumber>
        </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>
```

12.1.1.3 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>
```

12.1.1.4 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>
  <inputIOPortID>
    <!--ro, opt, string-->test
  </inputIOPortID>
  <dynInputIOPortID>
    <!--ro, opt, string-->test
  </dynInputIOPortID>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <intervalBetweenEvents>
    <!--ro, opt, int, unit:s-->1
  </intervalBetweenEvents>
  <WLSensorID>
    <!--ro, opt, string-->1
  </WLSensorID>
  <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>
      <notificationInterval>
        <!--ro, opt, int, unit:ms-->1
      </notificationInterval>
      <outputIOPortID>
        <!--ro, opt, string-->test
      </outputIOPortID>
      <dynOutputIOPortID>
        <!--ro, opt, string-->test
      </dynOutputIOPortID>
      <videoInputID>
        <!--ro, opt, string-->1
      </videoInputID>
      <dynVideoInputID>
        <!--ro, opt, string-->1
      </dynVideoInputID>
      <ptzAction>
        <!--ro, opt, object-->
        <ptzChannelID>
          <!--ro, req, string-->test
        </ptzChannelID>
        <actionName>
          <!--ro, req, enum, subType:string-->preset
        </actionName>
        <actionNum>
          <!--ro, opt, int-->1
        </actionNum>
        <presetDurationTime>
          <!--ro, opt, int, unit:s, dep:and, ${$.EventTrigger.EventTriggerNotificationList[*].EventTriggerNotification.ptzAction.actionName, eq, preset}-->1
        </presetDurationTime>
      </ptzAction>
      <cellphoneNumber>
        <!--ro, opt, string, range:[0,11]-->test
      </cellphoneNumber>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

12.1.1.5 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>
```

12.1.1.6 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>
  <inputIOPortID>
    <!--opt, string-->test
  </inputIOPortID>
  <dynInputIOPortID>
    <!--opt, string-->test
  </dynInputIOPortID>
  <videoInputChannelID>
    <!--opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string-->1
  </dynVideoInputChannelID>
  <intervalBetweenEvents>
    <!--opt, int, unit:s-->1
  </intervalBetweenEvents>
  <WLSensorID>
    <!--opt, string-->1
  </WLSensorID>
  <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>
      <notificationInterval>
        <!--opt, int, unit:ms-->1
      </notificationInterval>
      <outputIOPortID>
        <!--opt, string-->test
      </outputIOPortID>
      <dynOutputIOPortID>
        <!--opt, string-->test
      </dynOutputIOPortID>
      <videoInputID>
        <!--opt, string-->1
      </videoInputID>
      <dynVideoInputID>
        <!--opt, string-->1
      </dynVideoInputID>
      <ptzAction>
        <!--opt, object-->
        <ptzChannelID>
          <!--req, string-->test
        </ptzChannelID>
        <actionName>
          <!--req, enum, subType:string-->preset
        </actionName>
        <actionNum>
          <!--opt, int-->1
        </actionNum>
        <presetDurationTime>
          <!--opt, int, unit:s, dep:and, ${$.EventTrigger.EventTriggerNotificationList[*].EventTriggerNotification.ptzAction.actionName, eq, preset}-->1
        </presetDurationTime>
      </ptzAction>
      <cellphoneNumber>
        <!--opt, string, range:[0,11]-->test
      </cellphoneNumber>
    </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>
```

12.1.1.7 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>
  <inputIOPortID>
    <!--ro, opt, string-->test
  </inputIOPortID>
  <dynInputIOPortID>
    <!--ro, opt, string-->test
  </dynInputIOPortID>
  <videoInputChannelID>
    <!--ro, opt, string-->1
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string-->1
  </dynVideoInputChannelID>
  <intervalBetweenEvents>
    <!--ro, opt, int, unit:s-->1
  </intervalBetweenEvents>
  <WLSensorID>
    <!--ro, opt, string-->1
  </WLSensorID>
  <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>
      <notificationInterval>
        <!--ro, opt, int, unit:ms-->1
      </notificationInterval>
      <outputIOPortID>
        <!--ro, opt, string-->test
      </outputIOPortID>
      <dynOutputIOPortID>
        <!--ro, opt, string-->test
      </dynOutputIOPortID>
      <videoInputID>
        <!--ro, opt, string-->1
      </videoInputID>
      <dynVideoInputID>
        <!--ro, opt, string-->1
      </dynVideoInputID>
      <ptzAction>
        <!--ro, opt, object-->
        <ptzChannelID>
          <!--ro, req, string-->test
        </ptzChannelID>
        <actionName>
          <!--ro, req, enum, subType:string-->preset
        </actionName>
        <actionNum>
          <!--ro, opt, int-->1
        </actionNum>
        <presetDurationTime>
          <!--ro, opt, int, unit:s, dep:and, ${$.EventTrigger.EventTriggerNotificationList[*].EventTriggerNotification.ptzAction.actionName, eq, preset}-->1
        </presetDurationTime>
      </ptzAction>
      <cellphoneNumber>
        <!--ro, opt, string, range:[0,11]-->test
      </cellphoneNumber>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>

```

12.1.1.8 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>
    <inputIOPortID>
        <!--opt, string-->test
    </inputIOPortID>
    <dynInputIOPortID>
        <!--opt, string-->test
    </dynInputIOPortID>
    <videoInputChannelID>
        <!--opt, string-->1
    </videoInputChannelID>
    <dynVideoInputChannelID>
        <!--opt, string-->1
    </dynVideoInputChannelID>
    <intervalBetweenEvents>
        <!--opt, int, unit:s-->1
    </intervalBetweenEvents>
    <WLSensorID>
        <!--opt, string-->1
    </WLSensorID>
    <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>
            <notificationInterval>
                <!--opt, int, unit:ms-->1
            </notificationInterval>
            <outputIOPortID>
                <!--opt, string-->test
            </outputIOPortID>
            <dynOutputIOPortID>
                <!--opt, string-->test
            </dynOutputIOPortID>
            <videoInputID>
                <!--opt, string-->1
            </videoInputID>
            <dynVideoInputID>
                <!--opt, string-->1
            </dynVideoInputID>
            <ptzAction>
                <!--opt, object-->
                <ptzChannelID>
                    <!--req, string-->test
                </ptzChannelID>
                <actionName>
                    <!--req, enum, subType:string-->preset
                </actionName>
                <actionNum>
                    <!--opt, int-->1
                </actionNum>
                <presetDurationTime>
                    <!--opt, int, unit:s, dep:and, ${.EventTrigger.EventTriggerNotificationList[*].EventTriggerNotification.ptzAction.actionName, eq, preset}-->1
                </presetDurationTime>
            </ptzAction>
            <cellphoneNumber>
                <!--opt, string, range:[0,11]-->test
            </cellphoneNumber>
        </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>
```

12.1.1.9 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>
```

12.1.1.10 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>
  <HDDList>
    <!--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>
  </HDDList>
</EventNotificationAlert>

```

12.1.1.11 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>

```

12.1.1.12 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>
  <ip6Address>
    <!--ro, opt, string, IPv6 address of the device that triggers an alarm-->1080:0:0:0:8:800:200C:417A
  </ip6Address>
  <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>

```

12.1.2 Event Subscription

12.1.2.1 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}-->
    <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, req, 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), "CDsStatus" (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), "LFPD"(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), "blindsightDetection"(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), "highHTemperature"(HDD high temperature detection event), "highTempAlarm"(HDD high temperature alarm), "hotSpare"(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), "LowHTemperature"(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), "scenedechangeDetection", "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), "vehicleRocgResult", "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>
            <pictureURLType opt="binary,localURL,cloudStorageURL,multipart" def="cloudStorageURL">
                <!--ro, opt, enum, alarm picture format, subType:string, attr:opt{req, string},def{req, string}, desc:"binary" (binary), "LocalURL" (device Local URL), "cloudStorageURL" (cloud storage URL)-->cloudStorageURL
            </pictureURLType>
        </Event>
    <EventList>
        <pictureURLType opt="binary,localURL,cloudStorageURL,multipart" def="cloudStorageURL">
            <!--ro, opt, enum, format unified configuration of alarm pictures, subType:string, attr:opt{req, string},def{req, string}, desc:"binary", "LocalURL" (device Local URL), "cloudStorageURL" (cloud storage URL). The node indicates the upload mode of all event pictures. If the node is applied, the <pictureURLType> of the <Event> will be invalid. If the node is not applied, the pictures are uploaded in the default mode. The default data type of uploaded pictures for front-end devices is binary, and for back-end devices is Local URL of the device-->cloudStorageURL
        </pictureURLType>
        <level opt="high,middle,low">
            <!--ro, opt, enum, arming Level, subType:string, attr:opt{req, string}, desc:"high" (default value), "middle", "Low"-->high
        </level>
    </EventList>
</SubscribeEventCap>
```

12.1.2.2 Cancel subscribing alarm/event

Request URL

PUT /ISAPI/Event/notification/unSubscribeEvent?ID=<subscribeEventID>

Query Parameter

Parameter Name	Parameter Type	Description
subscribeEventID	string	Subscription event ID, which should match the id field in the SubscribeEventResponse of the response message from the POST /ISAPI/Event/notification/subscribeEvent?deployID=. This field is optional. When the protocol is /ISAPI/Event/notification/unSubscribeEvent, it indicates that all subscription links are to be unsubscribed. When the protocol is /ISAPI/Event/notification/unSubscribeEvent?ID=XXX, it indicates that a specific subscription link is to be unsubscribed.

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, opt, 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>
  <MErrCode>
    <!--ro, opt, string-->0x00000000
  </MErrCode>
  <MErrDevSelfEx>
    <!--ro, opt, string-->0x00000000
  </MErrDevSelfEx>
</ResponseStatus>
```

12.1.3 Online Upgrading Management

12.1.3.1 Apply the task of online upgrade

Request URL

POST /ISAPI/System/onlineUpgrade/task?format=json&UpgradeType=<UpgradeType>

Query Parameter

Parameter Name	Parameter Type	Description
UpgradeType	string	false

Request Message

```
{
    "PackageDescription": {
        /*req, object, description of the upgrade package*/
        "name": "DSJ_N3_GML_DEBUG_STD_1.2.0_190524.zip",
        /*opt, string, upgrade package name, range:[1,256]*/
        "URL": "test",
        /*opt, string, URL of the upgrade package, range:[1,1024], desc:the task ID is contained in the URL. The uploading and downloading of the upgrade package is based on the object storage protocol (version 2.0). If this node is null, the FTP address will take effect. If both the URL and FTP address are configured, the URL has priority over FTP address*/
        "MD5": "test",
        /*opt, string, range:[0,256]*/
        "moduleType": "cardReader",
        /*opt, enum, module type, subType:string, desc:"cardReader" (card reader), "FPMModule" (fingerprint module), "securityModule" (security module), "extendModule" (I/O extended module), "channelController" (lane controller), "IRModule" (infrared module), "lampModule" (indicator module), "elevatorController" (sub elevator controller), "FPAlgorithmProgram" (fingerprint algorithm programma of card reader), "dockStationHost" (dock station host), "dockStationScreen" (dock station screen), "stationHostAndScreen" (dock station host and screen)*/
        "moduleID": 1,
        /*opt, int, module ID*/
        "spellingControllerSubBoardList": [
            /*opt, array, splicing controller sub board list, subType:object, range:[1,1024], dep:or, {$.PackageDescription.moduleType,eq,spellingControllerMainAndSubBoard},{$.PackageDescription.moduleType,eq,spellingControllerSubBoard}, desc:it is valid when the value of the node moduleType is "spellingControllerMainAndSubBoard" or "spellingControllerSubBoard". If it is not configured, it indicates that all the sub boards will be updated*/
            {
                "subBoardID": 1,
                /*opt, int, sub board ID, range:[1,1024]*/
                "ipV4Address": "test"
                /*ro, opt, string, IPv4, range:[1,16], desc:when an exception occurs on the main board, the sub board can be updated through the IP address. The default sub board interface will be used*/
            }
        ],
        "channel": 1,
        /*opt, int, channel No., desc:this node is used for upgrading the IP channel of the NVR*/
        "childDevID": "3434645611",
        /*opt, string, range:[9,32]*/
        "customTaskID": "test",
        /*opt, string, user-defined task ID, range:[1,128], desc:if this node exists, it will be returned in the result*/
        "delayedUpgrade": true,
        /*opt, bool, whether to delay upgrading the package, desc:if this node is not applied, it indicates that the package will be upgraded immediately; if this node is applied and the value is false, it indicates that the upgrade will be delayed. User can decide the time to upgrade the package of the dock station or body camera if the upgrade is delayed*/
        "delayedUpgradeDateTime": "1970-01-01T00:00:00+08:00",
        /*opt, datetime, time of starting the delayed upgrade, dep:or {$.PackageDescription.delayedUpgrade,eq,false}, desc:the related capability is "isSupportDelayedUpgradeByDateTime" in GET /ISAPI/System/onlineUpgrade/capabilities*/
        "ftpAddress": "test",
        /*opt, string, FTP address of the upgrade package, desc:this node will take effect when URL is null and the format is: ftp://[username]:[password]@[IP address]:[port No.]*/
        "downloadStrategy": {
            /*opt, object, device's upgrade package downloading strategy parameters*/
            "sequelEnabled": true,
            /*opt, bool, whether to enable ANR (Automatic Network Replenishment), desc:if enabled, when downloading is interrupted, the device will try downloading the package again; if disabled, the default strategy of the device will be adopted*/
            "sequelTimeout": 3000,
            /*opt, int, timeout of a single ANR attempt, step:1, unit:ms, desc:when a single ANR attempt exceeds the timeout period but the download does not continue, the ANR attempt will be recognized as failed*/
            "maxRetryTimes": 5
            /*opt, int, ANR maximum attempt times, desc:when the device ANR exceeds the maximum attempt times but the download does not continue, the download and the upgrade will be recognized as failed*/
        },
        "upgradePackageURLType": "full"
        /*opt, enum, subType:string*/
    }
}
```

Response Message

```
{
    "statusCode": 1,
    /*ro, opt, int, status code, desc:1-succeeded, other value-error occurred. This field is required when an error occurred*/
    "statusString": "OK",
    /*ro, opt, string, status description, range:[1,64], desc:"OK" (succeeded), other value (error occurred). It is required when an error occurred*/
    "subStatusCode": "ok",
    /*ro, opt, string, sub status code, range:[1,64], desc:"ok" (succeeded), other value (error occurred). This field 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 message, desc:this node is required when the value of statusCode is not 1*/
    "taskID": "test"
    /*ro, opt, string, ID of the upgrade task, desc:it can be used to get the status and progress of the task*/
}
```

12.1.4 Port Settings

12.1.4.1 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>
    <rs485WorkMode opt="Led, CaptureTrigger,transparent">
        <!--ro, opt, string, range of RS-485 serial port No. supported by the device, attr:opt{req, string}-->test
    </rs485WorkMode>
    <rs232SerialNumber opt="1,2,3">
        <!--ro, opt, int, range of RS-232 serial port numbers supported by the device, attr:opt{req, string}-->1
    </rs232SerialNumber>
    <rs485SerialNumber opt="4,5,6">
        <!--ro, opt, int, range of RS-485 serial port numbers supported by the device, attr:opt{req, string}-->1
    </rs485SerialNumber>
    <isSupportAuthenticationService>
        <!--ro, opt, bool, whether the device supports the serial port authentication service-->true
    </isSupportAuthenticationService>
    <isSupportDeviceInfo>
        <!--ro, opt, bool, whether the device supports configuring the information about the serial port device-->true
    </isSupportDeviceInfo>
    <isSupportSearchDeviceInfoRelations>
        <!--ro, opt, bool, whether the device supports searching the linkage information of the serial port device-->true
    </isSupportSearchDeviceInfoRelations>
</SerialCap>
```

12.1.4.2 Get the capability of a specific serial port

Request URL

GET /ISAPI/System/Serial/ports/<portID>/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
portID	string	--

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<SerialPort xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, port No., attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, int, ID-->0
  </id>
  <enabled opt="true,false">
    <!--ro, opt, bool, whether to enable, attr:opt{req, string}-->true
  </enabled>
  <serialPortType opt="RS485,RS422,RS232">
    <!--ro, opt, string, serial port type, attr:opt{req, string}-->RS485
  </serialPortType>
  <duplexMode opt="half,full">
    <!--ro, opt, string, duplex mode of the serial port, attr:opt{req, string}-->half
  </duplexMode>
  <baudRate opt="1200,2400,4800,9600,19200,38400,57600,115200">
    <!--ro, opt, int, attr:opt{req, string}-->1200
  </baudRate>
  <dataBits opt="6,7,8">
    <!--ro, opt, int, attr:opt{req, string}-->6
  </dataBits>
  <parityType opt="none,even,odd,mark,space">
    <!--ro, opt, string, attr:opt{req, string}-->none
  </parityType>
  <stopBits opt="1,1.5,2">
    <!--ro, opt, string, stop bit, attr:opt{req, string}-->1
  </stopBits>
  <flowCtrl opt="none,software,hardware">
    <!--ro, opt, string, flowCtrl, attr:opt{req, string}-->none
  </flowCtrl>
  <rs485WorkMode>
    <!--Led,sensor,CaptureTrigger,transparent,GPSS,ModbusRTU,supplementLight,audibleVisualAlarm,columnSpeaker,commModeUsing433MHz,signalLampControl-->
    <!--ro, opt, string, working mode of RS-485 serial port, dep:and,{$.SerialPortList.SerialPort.serialPortType,eq,RS485}, attr:opt{req, string}, desc:working mode of RS-485 serial port,which is used for LED display or triggering transmission of captured pictures. This node is valid only when <serialPortType> is "RS485"-->sensor
  </rs485WorkMode>
</SerialPort>

```

12.1.4.3 Get the configuration parameters of a specified transparent transmission channel

Request URL

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

Query Parameter

Parameter Name	Parameter Type	Description
portID	string	--
channelID	string	--

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<TransparentChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, transparent transmission channel, attr:version{req, string, protocolVersion}-->
  <id>
    <!--ro, req, string, channel No.-->1
  </id>
  <enabled>
    <!--ro, req, bool, whether to enable the channel-->true
  </enabled>
  <serialPortID>
    <!--ro, req, string, serial port No.-->1
  </serialPortID>
</TransparentChannel>

```

12.1.4.4 Set the parameters of a specific serial port supported by the device

Request URL

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

Query Parameter

Parameter Name	Parameter Type	Description
portID	string	--
channelID	string	--

Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<TransparentChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, string, serial port ID-->1
  </id>
  <enabled>
    <!--req, bool, whether to enable the serial port:-->true
  </enabled>
  <serialPortID>
    <!--req, string-->1
  </serialPortID>
</TransparentChannel>
```

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>
```

12.1.4.5 Close the transmission channel

Request URL

PUT /ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>/close?type=<type>

Query Parameter

Parameter Name	Parameter Type	Description
portID	string	--
channelID	string	--
type	string	Data (purpose) type "general" (general data transparent transmission, the device sends the transparent transmission data via serial port, regardless of whether there is an external component connected to the serial port), "sensorCtrl" (sensor data transparent transmission control, when the device sends the transparent transmission data via serial port, it will also stop collecting sensor data, and compared with general type, the device will perform specific control and management on the sensors externally connected to the serial port)

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, opt, string, range:[0,1024]-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
  <description>
    <!--ro, opt, string, range:[0,1024]-->badXmlFormat
  </description>
  <MErrCode>
    <!--ro, opt, string-->0x00000000
  </MErrCode>
  <MErrDevSelfEx>
    <!--ro, opt, string-->0x00000000
  </MErrDevSelfEx>
</ResponseStatus>

```

12.1.4.6 Open the transmission channel

Request URL

PUT /ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>/open?type=<type>

Query Parameter

Parameter Name	Parameter Type	Description
portID	string	--
channelID	string	Camera ID
type	string	Data (purpose) type. "general" (general data transparent transmission, the device sends the transparent transmission data via serial port, regardless of whether there is an external component connected to the serial port), "sensorCtrl" (sensor data transparent transmission control, when the device sends the transparent transmission data via serial port, it will also stop collecting sensor data, and compared with general type, the device will perform specific control and management on the sensors externally connected to the serial port)

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, opt, string, range:[0,1024]-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, subType:int-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, subType:string-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string-->OK
  </subStatusCode>
  <description>
    <!--ro, opt, string, range:[0,1024]-->badXmlFormat
  </description>
  <MErrCode>
    <!--ro, opt, string-->0x00000000
  </MErrCode>
  <MErrDevSelfEx>
    <!--ro, opt, string-->0x00000000
  </MErrDevSelfEx>
</ResponseStatus>

```

12.1.4.7 Set the parameters of a specific serial port supported by the device

Request URL

PUT /ISAPI/System/Serial/ports/<portID>?permissionController=<indexID>&childDevID=<childDevID>&deviceIndex=<deviceIndex>

Query Parameter

Parameter Name	Parameter Type	Description
portID	string	--
indexID	string	--
childDevID	string	--
deviceIndex	string	--

Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<SerialPort xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, port, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, int, serial port ID-->0
  </id>
  <enabled>
    <!--opt, bool, whether to enable the function-->true
  </enabled>
  <encryptKey>
    <!--opt, string-->test
  </encryptKey>
  <serialNumber>
    <!--opt, int, serial No.-->0
  </serialNumber>
  <serialPortType>
    <!--opt, enum, serial port type: "RS485","RS422","RS232", subType:string, desc:serial port type: "RS485","RS422","RS232"-->RS485
  </serialPortType>
  <serialAddress>
    <!--opt, int-->1
  </serialAddress>
  <duplexMode>
    <!--opt, enum, duplex mode of the serial port, subType:string, desc:"half", "full"-->half
  </duplexMode>
  <direction>
    <!--opt, enum, "monodirectional,bidirectional", subType:string, desc:"monodirectional, bidirectional"-->monodirectional
  </direction>
  <baudRate>
    <!--opt, enum, subType:int-->2400
  </baudRate>
  <dataBits>
    <!--opt, int-->6
  </dataBits>
  <parityType>
    <!--opt, enum, parity type, subType:string, desc:"none, even, odd, mark, space"-->none
  </parityType>
</SerialPort>

```

```

<stopBits>
  <!--opt, string, stop bit: "1,1.5,2"-->1
</stopBits>
<workMode>
  <!--opt, enum, work mode, subType:string, desc:working mode: "console","transparent","audiomixer","stairsControl"-elevator control,"cardReader"-card reader,"disabled","custom". This node is required only when <serialPortType> is set to "RS232"-->console
</workMode>
<flowCtrl>
  <!--opt, enum, "none,software,hardware", subType:string, desc:"none, software, hardware"-->none
</flowCtrl>
<rs485WorkMode>
  <!--opt, enum, working mode of RS-485 serial port, which is used for LED display or triggering transmission of captured pictures: "Led, CaptureTrigger". This node is valid only when <serialPortType> is "RS485", subType:string, dep:or,{$.SerialPort.serialPortType,ea,RS485}, desc:working mode of RS-485 serial port,which is used for LED display or triggering transmission of captured pictures: "Led,CaptureTrigger". This node is valid only when <serialPortType> is "RS485"-->sensor
</rs485WorkMode>
<copyToAll>
  <!--opt, bool-->true
</copyToAll>
<deviceName>
  <!--ro, opt, string-->test
</deviceName>
<deviceProtocol>
  <!--ro, opt, int-->1
</deviceProtocol>
<mode>
  <!--opt, enum, work mode, subType:string, desc:deq,working mode: "readerMode,clientMode,externMode,accessControlHost,disabled",this node is valid only when <serialPortType> is "RS485"-->readerMode
</mode>
<outputDataType>
  <!--opt, enum, output data type, subType:string, dep:and,{$.SerialPort.mode,eq,accessControlHost}, desc:"cardNo,employeeNo", this node is valid when <mode>is "accessControlHost"-->cardNo
</outputDataType>
<isVariable>
  <!--opt, bool-->true
</isVariable>
<PTZInfo>
  <!--opt, object, PTZ information-->
<bindChannelNo>
  <!--req, int-->0
</bindChannelNo>
<PTZProtocolList>
  <!--req, object-->
<PTZProtocol>
  <!--opt, object-->
  <id>
    <!--req, int, serial port ID-->0
  </id>
  <protoDesc>
    <!--req, string-->test
  </protoDesc>
</PTZProtocol>
</PTZProtocolList>
</PTZInfo>
<ScreenCtrlInfo>
  <!--opt, object-->
<ScreenCtrlProtoList>
  <!--req, object-->
<ScreenCtrlProto>
  <!--opt, object-->
  <id>
    <!--req, int, serial port ID-->0
  </id>
  <protoDesc>
    <!--req, string-->test
  </protoDesc>
</ScreenCtrlProto>
</ScreenCtrlProtoList>
</ScreenCtrlInfo>
<MatrixCtrlInfo>
  <!--opt, object-->
<MatrixCtrlProtoList>
  <!--req, object-->
<MatrixCtrlProto>
  <!--opt, object-->
  <id>
    <!--req, int, serial port ID-->0
  </id>
  <protoDesc>
    <!--req, string-->test
  </protoDesc>
</MatrixCtrlProto>
</MatrixCtrlProtoList>
</MatrixCtrlInfo>
<KeyboardCtrlInfo>
  <!--opt, object-->
<KeyboardCtrlProtoList>
  <!--req, object-->
<KeyboardCtrlProto>
  <!--opt, object-->
  <id>
    <!--req, int, serial port ID-->0
  </id>
  <protoDesc>
    <!--req, string-->test
  </protoDesc>
</KeyboardCtrlProto>
</KeyboardCtrlProtoList>

```

```

<protoDesc>
    <!--req, string-->test
    </protoDesc>
    </KeyboardCtrlProto>
</KeyboardCtrlProtoList>
</KeyboardCtrlInfo>
<ControlAddress>
    <!--opt, object-->
    <enabled>
        <!--req, bool, whether to enable the function-->true
    </enabled>
    <address>
        <!--req, int, address, range:[0,255]-->1
    </address>
</ControlAddress>
<SensorCtrlList>
    <!--opt, object-->
    <Sensor>
        <!--opt, object-->
        <id>
            <!--req, int, ID, range:[1,3]-->1
        </id>
        <name>
            <!--req, string, sensor name, range:[1,32]-->visibilitySensor
        </name>
        <enabled>
            <!--req, bool, whether to enable the function, desc:whether to enable the serial port: "true,false"-->false
        </enabled>
    </Sensor>
</SensorCtrlList>
<stairsControl>
    <!--opt, object, elevator control parameters, this node is valid when the value of the node <mode> is stairsControl, dep:and,
    {$.SerialPort.mode,eq,stairsControl}-->
    <outputDataType>
        <!--opt, enum, output data type, subType:string, desc:"cardNo,employeeNo", this node is valid when <mode>is "accessControlHost"-->floorNumber
    </outputDataType>
</stairsControl>
<comMode>
    <!--opt, enum, subType:string-->active
</comMode>
<ModbusRTU>
    <!--opt, object, dep:and,{$.SerialPort.rs485WorkMode,eq,ModbusRTU}-->
    <ModbusRTUBaudRate>
        <!--opt, enum, subType:string-->9600
    </ModbusRTUBaudRate>
    <ModbusRTUDataBits>
        <!--opt, enum, subType:int-->6
    </ModbusRTUDataBits>
    <ModbusRTUParityType>
        <!--opt, enum, subType:string-->none
    </ModbusRTUParityType>
    <ModbusRTUStopBits>
        <!--opt, enum, subType:string-->1
    </ModbusRTUStopBits>
    <ModbusRTUFlowCtrl>
        <!--opt, enum, subType:string-->none
    </ModbusRTUFlowCtrl>
    <decoderAddress>
        <!--opt, int, range:[0,255]-->0
    </decoderAddress>
</ModbusRTU>
<accessName>
    <!--opt, string, range:[1,32]-->test
</accessName>
<serialPortUse>
    <!--opt, enum, subType:string-->acquisition
</serialPortUse>
<acquisitionProtocol>
    <!--opt, enum, subType:string, dep:and,{$.SerialPort.serialPortUse,eq,acquisition}-->modbus
</acquisitionProtocol>
<serialProtocol>
    <!--opt, enum, subType:string-->GB26875
</serialProtocol>
<SupplementLightList>
    <!--opt, array, subType:object, dep:and,{$.SerialPort.rs485WorkMode,eq,supplementLight}-->
    <SupplementLight>
        <!--opt, object-->
        <SupplementLightID>
            <!--req, int, range:[1,5]-->1
        </SupplementLightID>
    </SupplementLight>
</SupplementLightList>
<cardReaderType>
    <!--opt, enum, subType:string-->K1108AM
</cardReaderType>
</SerialPort>

```

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, error reason description in detail, desc:error reason description in detail-->OK
  </subStatusCode>
  </ResponseStatus>

```

12.1.4.8 Get the parameters of a specific port supported by the device

Request URL

GET /ISAPI/System/Serial/ports/<portID>?permissionController=<indexID>&childDevID=<childDevID>&deviceIndex=<deviceIndex>

Query Parameter

Parameter Name	Parameter Type	Description
portID	string	--
indexID	string	0 (main permission controller), 1 (sub permission controller) When this parameter does not exist, it indicates getting the serial port information of the main permission controller (i.e., the device serial port information).
childDevID	string	--
deviceIndex	string	For C60S devices connected to multiple LED sending cards, the URL needs to include deviceIndex= to distinguish which LED sending card to operate on. The value of deviceIndex is the value of "id" when "portType" is LED and can be obtained via GET /ISAPI/DisplayDev/Video/outputs/channels.

Request Message

None

Response Message

```

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

```

12.1.4.9 Get the list of serial ports supported by the device

Request URL

GET /ISAPI/System/Serial/ports?permissionController=<indexID>&deviceIndex=<deviceIndex>

Query Parameter

Parameter Name	Parameter Type	Description
indexID	string	0 (main permission controller), 1 (sub permission controller) If this parameter does not exist, it indicates getting the serial port information of main permission controller, that is, the serial port information of the device.
deviceIndex	string	For C60S device connecting multiple LED sending cards, the URL needs to include the parameter deviceIndex= to specify which LED sending card to operate on. The value of deviceIndex is the value of "id" when "portType" is LED and can be obtained via GET /ISAPI/DisplayDev/Video/outputs/channels.

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, array, subType:object, attr:version{req, string, protocolVersion}-->
  <SerialPort>
    <!--ro, opt, object-->
    <id>
      <!--ro, req, int-->0
    </id>
    <enabled>
      <!--ro, opt, bool-->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, subType:string-->console
    </workMode>
    <flowCtrl>
      <!--ro, opt, enum, subType:string-->none
    </flowCtrl>
    <rs485WorkMode>
      <!--ro, opt, enum, subType:string, dep:or,{$.SerialPort.serialPortType,eq,RS485}-->sensor
    </rs485WorkMode>
  </SerialPort>
</SerialPortList>
```

12.1.5 System Maintenance

12.1.5.1 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}-->
  <isSupportSMARTTest>
    <!--ro, opt, bool, whether the device supports HDD detection, desc:related URI: /ISAPI/ContentMgmt/Storage/hdd/SMARTTest/config-->true
  </isSupportSMARTTest>
  <isSupportMainAndSubRecord>
    <!--ro, opt, bool, whether the device supports the main stream and sub-stream recording-->true
  </isSupportMainAndSubRecord>
  <isSupportLogConfig>
    <!--ro, opt, bool, whether the device supports Log configuration, desc:related URI: /ISAPI/ContentMgmt/LogConfig/capabilities-->true
  </isSupportLogConfig>
  <isSupportPictureInfo>
    <!--ro, opt, bool, whether it supports picture search on the SD card of the capture camera-->true
  </isSupportPictureInfo>
  <isSupportLogSearch>
    <!--ro, opt, bool, whether it supports Log search-->true
  </isSupportLogSearch>
  <isSupportRecordSearch>
    <!--ro, opt, bool, whether it supports video search-->true
  </isSupportRecordSearch>
</RacmCap>

```

12.1.5.2 Get the device status capability

Request URL

GET /ISAPI/ITC/ITCStatus/capabilities

Query Parameter

None

Request Message

None

Response Message

```

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

<ITCStatusCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <isSupportFortifyLinkStatus>
    <!--ro, req, bool-->true
  </isSupportFortifyLinkStatus>
  <isSupportDetectorStatus>
    <!--ro, req, bool-->true
  </isSupportDetectorStatus>
  <relatedTriggerInputs>
    <!--ro, req, int, number of linked triggering inputs-->1
  </relatedTriggerInputs>
  <relatedSyncOutputs>
    <!--ro, req, int, number of linked synchronous outputs-->1
  </relatedSyncOutputs>
</ITCStatusCap>

```

12.1.5.3 Get the live view status

Request URL

GET /ISAPI/ITC/ITCStatus/previewStatus

Query Parameter

None

Request Message

None

Response Message

```

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

<PreviewStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, Live view status, attr:version{req, string, protocolVersion}-->
  <previewCount>
    <!--ro, req, int, number of channels for Live view-->1
  </previewCount>
  <previewCfgList>
    <!--ro, opt, array, subType:object-->
    <previewCfg>
      <!--ro, opt, object-->
      <id>
        <!--ro, req, int-->1
      </id>
      <previewIP>
        <!--ro, req, string, Live view connection IP-->test
      </previewIP>
      <channelType>
        <!--ro, req, enum, channel type, subType:string, desc:"mainChannel", "subChannel", "noChannel"-->mainChannel
      </channelType>
      <frameRate>
        <!--ro, req, int, frame rate, desc:the value equals to the maximum frame rate *100. This node is set to 0 if there is no live view-->1
      </frameRate>
      <resolutionWidth>
        <!--ro, req, int, width of the video resolution-->1
      </resolutionWidth>
      <resolutionHeight>
        <!--ro, req, int, height of the video resolution-->1
      </resolutionHeight>
    </previewCfg>
  </previewCfgList>
  <snapResolutionWidth>
    <!--ro, req, int, width of the capture resolution-->1
  </snapResolutionWidth>
  <snapResolutionHeight>
    <!--ro, req, int, height of the capture resolution-->1
  </snapResolutionHeight>
  <triggerType>
    <!--ro, req, enum, trigger mode, subType:string, desc:trigger mode-->normal
  </triggerType>
</PreviewStatus>

```

12.1.5.4 Get the system security capability

Request URL

GET /ISAPI/Security/capabilities?username=<userName>&deviceIndex=<deviceIndex>

Query Parameter

Parameter Name	Parameter Type	Description
userName	string	user name
deviceIndex	string	For centralized splicing controllers connecting to LED sending cards, the URL needs to include the parameter deviceIndex= to specify LED sending card. The value of deviceIndex is the value of "id" when "portType" is LED and can be obtained via GET /ISAPI/DisplayDev/Video/outputs/channels.

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>
</SecurityCap>

```

12.1.5.5 Get device 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, device system capability, attr:version{opt, string, protocolVersion}-->
    <isSupportPreview>
        <!--ro, opt, bool, whether the device supports Live view-->true
    </isSupportPreview>
    <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>
    <isSupportConfigurationData>
        <!--ro, opt, bool, whether the device supports importing and exporting the configuration files safely, desc:for traffic devices, this node is required and must be true-->true
    </isSupportConfigurationData>
    <isSupportUpdatefirmware>
        <!--ro, opt, bool, whether the device supports upgrading, desc:for traffic devices, this node is required and must be true-->true
    </isSupportUpdatefirmware>
    <isSupportDeviceInfo>
        <!--ro, opt, bool, whether the device supports getting the device information, desc:for traffic devices, this node is required and must be true-->true
    </isSupportDeviceInfo>
    <isSupportTime>
        <!--ro, opt, bool, whether the device supports time configuration, desc:for traffic device, this node is required and must be true. For other devices, the isSupportTimeCap is used to configure time-->true
    </isSupportTime>
    <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>
            <isSupport28181>
                <!--ro, opt, bool, N/A-->true
            </isSupport28181>
            <SnmpCap>
                <!--ro, opt, object, SNMP (Simple Network Management Protocol) capability-->
                <isSupport>
                    <!--ro, req, bool, whether the device supports SNMP-->true
                </isSupport>
            </SnmpCap>
            <isSupportIPFilter>
                <!--ro, opt, bool, whether the device supports IP filtering-->true
            </isSupportIPFilter>
        
```

```

</isSupportEHome>
<!--ro, opt, bool, whether the device supports ISUP server configuration-->true
</isSupportEHome>
<isSupportWirelessDial>
<!--ro, opt, bool, whether the device supports wireless dial-up protocol-->true
</isSupportWirelessDial>
<isSupportWifiProbe>
<!--ro, opt, bool, whether the device supports Wi-Fi signal detection-->true
</isSupportWifiProbe>
<VerificationCodeModification>
<!--ro, opt, object, device verification code configuration-->
<verificationCodeType opt="normal,empty">
<!--ro, opt, string, verification code type, attr:opt{opt, string}-->empty
</verificationCodeType>
</VerificationCodeModification>
<isSupportIntegrate>
<!--ro, opt, bool, whether the device supports access protocol configuration-->true
</isSupportIntegrate>
<isSupportWebSocket>
<!--ro, opt, bool, whether the device supports WebSocket-->true
</isSupportWebSocket>
<isSupportWebSockets>
<!--ro, opt, bool, whether the device supports WebSockets-->true
</isSupportWebSockets>
<isSupportEventDataOverWebSocket opt="true,false">
<!--ro, opt, bool, whether the device supports uploading the event data via WebSocket, attr:opt{opt, string}-->true
</isSupportEventDataOverWebSocket>
<isSupportPingDeny>
<!--ro, opt, bool, whether the device supports configuring parameters to prevent the device from being tested using ping command-->true
</isSupportPingDeny>
<isSupportClusterIntercom>
<!--ro, opt, bool, whether the 7200 intercom protocol is transmitted, 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.-->true
</isSupportClusterIntercom>
<isSupportTSOverWebSocket>
<!--ro, opt, bool, whether the device supports transparent transmission via the WebSocket serial port, desc:(TS:TransportSerial)-->true
</isSupportTSOverWebSocket>
</NetworkCap>
<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>
<rs485WorkMode opt="Led,CaptureTrigger">
<!--ro, opt, string, RS-485 working modes supported by the device, attr:opt{opt, string}, desc:Led (external LED screen, which is used for LED display), CaptureTrigger (external device, which is used to trigger the transmission of the captured data)-->test
</rs485WorkMode>
</SerialCap>
<isSupportSubscribeEvent>
<!--ro, opt, bool, whether the device supports subscribing to events, desc:related URI: /ISAPI/Event/notification/subscribeEventCap-->true
</isSupportSubscribeEvent>
<isSupportSubscribeIOTInfo>
<!--ro, opt, bool, whether the device supports subscribing to the IoT information-->true
</isSupportSubscribeIOTInfo>
</SysCap>
<ITCCap>
<!--ro, opt, object, related node and URIs: the node <isSupportVehicleDetection> in /ISAPI/ITC/capability and /ISAPI/ITC/capabilities-->
<isSupportVehicleDetection>
<!--ro, opt, bool, whether the device supports vehicle detection-->true
</isSupportVehicleDetection>
<isSupportLicencePlateAuditData>
<!--ro, opt, bool, whether the device supports importing or exporting the license plate blocklist and allowlist data-->true
</isSupportLicencePlateAuditData>
<isSupportSyncSignalOutput>
<!--ro, opt, bool, whether the device supports configuring output parameters-->true
</isSupportSyncSignalOutput>
</ITCCap>
<RacmCap>
<!--ro, opt, object, UI before picture search-->
<inputProxyNums>
<!--ro, opt, int, supported number of digital channels, desc:related URI: /ISAPI/ContentMgmt/InputProxy/channels/<ID>-->1
</inputProxyNums>
</RacmCap>
<VideoIntercomCap>
<!--ro, opt, object, video intercom capability, desc:related URI: /ISAPI/VideoIntercom/capabilities-->
<platform opt="R0,R1,S29,N10,N8,N4">
<!--ro, opt, enum, platform type, subType:string, attr:opt{req, string}, desc:"R0", "R1"; it will be returned if the device supports-->R0
</platform>
</VideoIntercomCap>
<isSupportDumpData>
<!--ro, opt, bool, whether the device supports exporting the Dump file-->true
</isSupportDumpData>
<AIDEventSupport>

```

opt="abandonedObject,pedestrian,congestion,roadBlock,construction,trafficAccident,fogDetection,wrongDirection,illegalParking,SSharpDriving,lowSpeed,dragRaci

```

ng,obstacle,vehNoYieldPedest,illegalMannedVeh,illegalMannedNonMotorVeh,umbrellaTentInstall,nonMotorVehOnVehLane,wearingNoHelmet,pedestRedLightRunning,pedest
OnNonMotorVehLane,pedestOnVehLane,conflagration,smoke,VehicleSpeedDrop,lowVisibility,animalsOccupy,targetSpeedDetection,emergencyLane,gasser,changeLaneConti
nuously">
    <!--ro, opt, string, supported AID event, attr:opt{opt, string}-->test
    </AIDEventSupport>
    <TFSEventSupport opt="illegalParking
,wrongDirection,crossLane,laneChange,vehicleExist,turnRound,parallelParking,notKeepDistance,notSlowZebraCrossing,overtakeRightSide,lowSpeed,dragRacing,chang
eLaneContinuously,SSharpDriving,largeVehicleOccupyLine,jamCrossLine,obstacle,blackSmokeVehicle,turnRightStop,occupyDedicatedLane,notDriveInDedicatedLane,TFS
_nonZipperMerge,overSpeed,emergencyLane,gasser">
        <!--ro, opt, string, supported TFS, attr:opt{opt, string}-->test
    </TFSEventSupport>
    <isSupportVehicleFaceRecognition>
        <!--ro, opt, bool, whether the device supports checkpoint vehicle recognition, desc:background: the traffic devices support "facial recognition"
(driver), however, the corresponding node is not added to the device to notify the integration of this capability-->true
    </isSupportVehicleFaceRecognition>
    <isSupportLCDScreen>
        <!--ro, opt, bool, whether the device supports LCD display screen-->true
    </isSupportLCDScreen>
    <isSupportEmmcInfo>
        <!--ro, opt, bool, the capability of supporting the EMMC configuration, desc:related URI: /ISAPI/System/emmcInfo/capabilities?format=json-->true
    </isSupportEmmcInfo>
    <isSupportVehicleIllegalType opt="1,2...29">
        <!--ro, opt, string, the vehicle violation types supported by the device, attr:opt{opt, string}, desc:the content value is the same as the index value,
refer to NET_ITS_PLATE_RESULT.NET_DVR_VEHICLE_INFO.byIllegalType the capture camera is connected to the new platform, and the device supports "traffic
violation alarm"; however, the corresponding node is not added to the device to notify the integration of this capability; that is, the traffic violation
alarm capability lacks details. If later the platform needs to develop capability with details, the current device will be incompatible with the capability;
and detailed capability is hidden in the configuration of all trigger modes, which will influence overall access-->test
    </isSupportVehicleIllegalType>
    <isSupportRadar>
        <!--ro, opt, bool, whether the device supports radar functions, desc:related URI: /ISAPI/Radar/capabilities?format=json-->true
    </isSupportRadar>
    <isSupportISUPHttpPassthrough>
        <!--ro, opt, bool, whether the device supports HTTP transparent transmission via ISUP V5.0-->true
    </isSupportISUPHttpPassthrough>
    <isSupportDetectorMotionAlarm>
        <!--ro, opt, bool, whether the device supports the detector moving alarm, desc:eventType: detectorMotionAlarm-->true
    </isSupportDetectorMotionAlarm>
    <isSupportRadarVideoDetection>
        <!--ro, opt, bool, related URI: /ISAPI/Intelligent/channels/<ID>/radarVideoDetection/capabilities?format=json, desc:related URI:
/ISAPI/Intelligent/channels/<ID>/radarVideoDetection/capabilities?format=json-->true
    </isSupportRadarVideoDetection>
    <isSupportOnlineUpgradeTask>
        <!--ro, opt, bool, whether device online upgrading is supported-->true
    </isSupportOnlineUpgradeTask>
    <isSupportObjectServer>
        <!--ro, opt, bool, whether the device supports the object storage service, desc:related URI: /ISAPI/System/objectServer/capabilities?format=json-->true
    </isSupportObjectServer>
    <isSupportDrivingDirectionAtIntersection>
        <!--ro, opt, bool, whether the device supports intersection driving direction detection, desc:eventType: drivingDirectionAtIntersection; this node is
not returned if not supported-->true
    </isSupportDrivingDirectionAtIntersection>
    <supportQuickConfigType opt="icr,dayNightGate,syncSignalOutput">
        <!--ro, opt, string, the list of supported functions of the quick configuration pop-up on the web, attr:opt{opt, string},
desc:the web quick configuration does not distinguish the configuration order, and it displays all supported function lists on the pop-up on the tab page
        icr refers to ICR switch configuration (related URL: /ISAPI/Image/channels/<channelID>/); icr
        dayNightGate refers to day and night mode switching (related URL: /ISAPI/Image/channels/<channelID>/dayNightGate);
        IOoutputs refers to synchronous signal output (related URL: /ISAPI/System/syncSignalOutput)-->syncSignalOutput
    </supportQuickConfigType>
    <isSupportChildmanage>
        <!--ro, opt, bool, whether the device supports gateway sub-device topology management, desc:GET /ISAPI/IoTGateway/Childmanage/capabilities?format=json-->true
    </isSupportChildmanage>
    <isSupportPowerIndicatorLightCtrlParams>
        <!--ro, opt, bool, whether the device supports parameters of device battery indicator control, desc:GET/PUT /ISAPI/System/PowerIndicatorLightCtrlParams?
format=json-->true
    </isSupportPowerIndicatorLightCtrlParams>
    <isSupportMaintenanceServiceAuthorization>
        <!--ro, opt, bool, whether it supports device maintenance service authorization information, desc:GET/PUT /ISAPI/System/MaintenanceServiceAuthorization?
format=json-->true
    </isSupportMaintenanceServiceAuthorization>
</DeviceCap>

```

12.1.5.6 Export device configuration file

Request URL

GET /ISAPI/System/configurationData?secretkey=<secretkey>

Query Parameter

Parameter Name	Parameter Type	Description
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

Binary Data

12.1.5.7 Get the languages supported by the device

Request URL

GET /ISAPI/System/DeviceLanguage

Query Parameter

None

Request Message

None

Response Message

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

<DeviceLanguage xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--req, object, Languages supported by the device, attr:version{req, string, protocolVersion}-->
    <language>
        <!--req, enum, Language, subType:string, desc:"SimChinese" (simplified Chinese), "Traditional Chinese", "English", "Russian",
        "Bulgarian", "Hungarian", "Greek", "German", "Italian", "Czech", "Slovakia", "French", "Polish", "Dutch", "Portuguese", "Spanish", "Romanian", "Turkish",
        "Japanese", "Danish", "Swedish", "Norwegian", "Finnish", "Korean", "Thai", "Estonia", "Vietnamese", "Hebrew", "Latvian", "Arabic", "Soviet" (Slovenian),
        "Croatian", "Lithuanian", "Serbian", "BrazilianPortuguese" (Brazilian Portuguese), "Indonesian", "Ukrainian", "EURSpanish", "Soviet", "Uzbek", "Kazak",
        "Kirghiz", "Farsi", "Azerbaijani", "Burmese", "Mongolian"-->SimChinese
        </language>
    </DeviceLanguage>
```

12.1.5.8 Set device language parameters

Request URL

PUT /ISAPI/System/DeviceLanguage

Query Parameter

None

Request Message

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

<DeviceLanguage xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--req, object, Languages supported by the device, attr:version{req, string, protocolVersion}-->
    <language>
        <!--req, enum, Language, subType:string, desc:"SimChinese" (simplified Chinese), "Traditional Chinese", "English", "Russian", "Bulgarian",
        "Hungarian", "Greek", "German", "Italian", "Czech", "Slovakia", "French", "Polish", "Dutch", "Portuguese", "Spanish", "Romanian", "Turkish", "Japanese",
        "Danish", "Swedish", "Norwegian", "Finnish", "Korean", "Thai", "Estonia", "Vietnamese", "Hebrew", "Latvian", "Arabic", "Soviet" (Slovenian), "Croatian",
        "Lithuanian", "Serbian", "BrazilianPortuguese" (Brazilian Portuguese), "Indonesian", "Ukrainian", "EURSpanish"-->SimChinese
        </language>
    </DeviceLanguage>
```

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)-->0
      <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-->OK
      </subStatusCode>
    </statusCode>
  </statusCode>
</ResponseStatus>

```

12.1.5.9 Get the capability of configuring the device language

Request URL

GET /ISAPI/System/DeviceLanguage/capabilities

Query Parameter

None

Request Message

None

Response Message

```

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

<DeviceLanguage xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, device Language configuration, attr:version{req, string, protocolVersion}-->
  <language>
    opt="SimChinese,TraChinese,English,Russian,Bulgarian,Hungarian,Greek,German,Italian,Czech,Slovakia,French,Polish,Dutch,Portuguese,Spanish,Romanian,Turkish,J
    apanese,Danish,Swedish,Norwegian,Finnish,Korean,Thai,Estonia,Vietnamese,Hebrew,Latvian,Arabic,Sovenian,Croatian,Lithuanian,Serbian,BrazilianPortuguese,Indon
    esian,Ukrainian,EURSpanish,Uzbek,Kazak,Kirghiz,Farsi,Azerbaijhan,Burmese,Mongolian,Anglicism,Estonian"
    <!--ro, req, enum, Language, subtype:string, attr:opt{req, string}, desc:"SimChinese" (simplified Chinese), "TraChinese" (traditional Chinese),
    "English", "Russian", "Bulgarian", "Hungarian", "Greek", "German", "Italian", "Czech", "Slovakia", "French", "Polish", "Dutch", "Portuguese", "Spanish",
    "Romanian", "Turkish", "Japanese", "Danish", "Swedish", "Norwegian", "Finnish", "Korean", "Thai", "Estonia", "Vietnamese", "Hebrew", "Latvian", "Arabic",
    "Sovenian"-Slovenian, "Croatian", "Lithuanian", "Serbian", "BrazilianPortuguese"-Brazilian Portuguese, "Indonesian", "Ukrainian", "EURSpanish", "Sovenian",
    "Uzbek", "Kazak", "Kirghiz", "Farsi", "Azerbaijhan", "Burmese", "Mongolian"-->SimChinese
    </language>
    <upgradeFirmWareEnabled>
      <!--ro, opt, bool, whether to enable upgrading the firmware-->true
    </upgradeFirmWareEnabled>
  </DeviceLanguage>

```

12.1.5.10 Set restoring devices' default parameters

Request URL

PUT /ISAPI/System/factoryReset?mode=<mode>&childDevID=<devIndex>&loginPassword=<loginPassword>&module=<module>

Query Parameter

Parameter Name	Parameter Type	Description
mode	enum	--
devIndex	string	--
loginPassword	string	--
module	enum	--

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

12.1.5.11 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{opt, string, protocolVersion}-->
  <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>
  <isSupport28181>
    <!--ro, opt, bool, N/A-->true
  </isSupport28181>
  <SnmpCap>
    <!--ro, opt, object, SNMP (Simple Network Management Protocol) capability-->
    <isSupport>
      <!--ro, req, bool, whether the device supports SNMP-->true
    </isSupport>
  </SnmpCap>
  <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 the device supports EHome (ISUP) protocol-->true
  </isSupportEHome>
  <isSupportWirelessDial>
    <!--ro, opt, bool, whether the device supports wireless dial-up protocol-->true
  </isSupportWirelessDial>
  <isSupportWifiProbe>
    <!--ro, opt, bool, whether the device supports Wi-Fi signal detection-->true
  </isSupportWifiProbe>
  <VerificationCodeModification>
    <!--ro, opt, object, verification code editing-->
    <verificationCodeType opt="normal,empty">
      <!--ro, opt, string, verification code type, attr:opt{req, string}-->test
    </verificationCodeType>
  </VerificationCodeModification>
  <isSupportIntegrate>
    <!--ro, opt, bool, whether the device supports access protocol configuration-->true
  </isSupportIntegrate>
  <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>
  <isSupportEventDataOverWebSocket opt="true,false">
    <!--ro, opt, bool, whether the device supports uploading the event data via WebSocket, attr:opt{req, string}-->true
  </isSupportEventDataOverWebSocket>
  <isSupportGantryETCServer>
    <!--ro, opt, bool, N/A, desc:N/A-->true
  </isSupportGantryETCServer>
  <isSupportDSTPServerCfg>
    <!--ro, opt, bool, whether the device supports DTSP server address configuration, desc:GET/PUT /ISAPI/System/Network/DSTPAccess/DSTPServerCfg?format=json-->true
  </isSupportDSTPServerCfg>
  <isSupportDSTPProtocolsCap>
    <!--ro, opt, bool, whether the device supports DTSP capability, desc:GET /ISAPI/System/Network/DSTPAccess/DSTPProtocolsCap?format=json-->true
  </isSupportDSTPProtocolsCap>
  <isSupportTSDSTPServer>
    <!--ro, opt, bool, whether the device supports transparent transmission via the WebSocket serial port, desc:(TS:TransportSerial)-->true
  </isSupportTSDSTPServer>
</NetworkCap>

```

12.1.5.12 Set SSH parameters

Request URL

PUT /ISAPI/System/Network/ssh?readerID=<readerID>&type=<type>&SOCChipID=<SOCChipID>

Query Parameter

Parameter Name	Parameter Type	Description
readerID	string	--
type	enum	--
SOCChipID	string	--

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>
  <statusString>
    <!--ro, req, enum, read-only,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>
  <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>
```

12.1.5.13 Reboot device

Request URL

PUT /ISAPI/System/reboot?childDevID=<devIndex>&module=<module>&loginPassword=<loginPassword>

Query Parameter

Parameter Name	Parameter Type	Description
devIndex	string	Reboot a sub-device via specifying sub-device ID. The ID of the sub-device connected to the current device can be searched via /ISAPI/IoTGateway/Childmanage/SearchChild?format=json.
module	enum	The "module" specifies a module to restart, without requiring a device reboot, corresponding to the capability (/ISAPI/System/capabilities/deviceReboot->module). "algorithmProgram (traffic terminal radar-assisted algorithm program, supports independent algorithm program startup), "radar" (restart radar module of the radar-assisted camera).
loginPassword	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, opt, 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-->0
  </subStatusCode>
  <description>
    <!--ro, opt, string, range:[0,1024]-->badXmlFormat
  </description>
  <MErrCode>
    <!--ro, opt, string-->0x00000000
  </MErrCode>
  <MErrDevSelEx>
    <!--ro, opt, string-->0x00000000
  </MErrDevSelEx>
</ResponseStatus>
```

12.1.5.14 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>
  <menuNums>
    <!--ro, opt, int, number of Local menus that can be displayed-->0
  </menuNums>
```

```

</menuNums>
<externalChannelNum>
    <!--ro, opt, bool, number of extended analog channels-->true
</externalChannelNum>
<isSupportHeatmap>
    <!--ro, opt, bool, whether the device supports heat map-->true
</isSupportHeatmap>
<isSupportCounting>
    <!--ro, opt, bool, whether the device supports people counting-->true
</isSupportCounting>
<countingType>
    <!--ro, opt, enum, statistics type, subType:string, desc:statistics type-->human
</countingType>
<isSupportPicture>
    <!--ro, opt, bool, whether the device supports OSD picture overlay, desc:related URI: /ISAPI/System/Video/inputs/channels/<channelID>/image/picture-->true
</isSupportPicture>
<isSupportPreviewSwitch>
    <!--ro, opt, bool, whether the device supports switching live view-->true
</isSupportPreviewSwitch>
<isSupportRecordStatus>
    <!--ro, opt, bool, whether the device supports searching for the recording status-->true
</isSupportRecordStatus>
<isSupportPrivacyMask>
    <!--ro, opt, bool, whether the device supports privacy mask-->true
</isSupportPrivacyMask>
<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>
<isSupportFocusVideoMode>
    <!--ro, opt, bool-->true
</isSupportFocusVideoMode>
<isSupportExternalChannel>
    <!--ro, opt, bool, whether the device supports extended analog channels-->true
</isSupportExternalChannel>
<isSupportMultiChannelCounting>
    <!--ro, opt, bool-->true
</isSupportMultiChannelCounting>
<isSupportCountingCollection>
    <!--ro, opt, bool, whether the device supports people counting ANR-->true
</isSupportCountingCollection>
<isSupportHeatmapCollection>
    <!--ro, opt, bool, whether the device supports heat map data ANR-->true
</isSupportHeatmapCollection>
<channelFlexible opt="name,enable,online,linknum">
    <!--ro, opt, enum, subType:string, attr:opt{req, string}-->name
</channelFlexible>
<isSupportOutputsResource>
    <!--ro, opt, bool, related URI: /ISAPI/System/Video/outputs/resource?format=json, desc:related URI: /ISAPI/System/Video/outputs/resource?format=json-->true
</isSupportOutputsResource>
<OSDlanguage opt="GBK,EUC-KR,Hebrew">
    <!--ro, opt, enum, subType:string, attr:opt{req, string},def{req, string}-->GBK
</OSDlanguage>
<isSupportMixedChannel>
    <!--ro, opt, bool-->true
</isSupportMixedChannel>
<isSupportMixedChannelStatus>
    <!--ro, opt, bool-->true
</isSupportMixedChannelStatus>
<isSupportOutputCourseware>
    <!--ro, opt, bool, related URI: /ISAPI/System/Video/outputs/courseware/capabilities?format=json, desc:related URI: /ISAPI/System/Video/outputs/courseware/capabilities?format=json-->true
</isSupportOutputCourseware>
<isSupportVideoInputMode>
    <!--ro, opt, bool, related URI: /ISAPI/System/Video/inputs/mode/capabilities, desc:related URI: /ISAPI/System/Video/inputs/mode/capabilities-->true
</isSupportVideoInputMode>
<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>
<PreviewMode>
    <!--ro, opt, object-->
<PIPType>
    <!--ro, opt, object-->
<mainScreenChannelID>
    <!--ro, opt, int-->0
</mainScreenChannelID>
<subScreenChannelID>
    <!--ro, opt, int-->0
</subScreenChannelID>
</PIPType>
</PreviewMode>
<isSupportViewTag>
    <!--ro, opt, bool-->true
</isSupportViewTag>
<isSupportMenuStatus>
    <!--ro, opt, bool-->true

```

```
</isSupportMenuStatus>
</VideoCap>
```

12.1.6 System Upgrade

12.1.6.1 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)*/
        }
    ]
}
```

12.1.6.2 Get the device upgrading status and progress

Request URL

GET /ISAPI/System/upgradeStatus?type=<type>&childDevID=<childDevID>&id=<id>

Query Parameter

Parameter Name	Parameter Type	Description
type	string	"usbmic" (array microphone module)
childDevID	string	--
id	string	The id indicates the peripheral No. to be upgraded.

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>
  <reboot>
    <!--ro, opt, bool, dep:and,{$.upgradeStatus.percent,eq,100}-->false
  </reboot>
  <upgradeStage>
    <!--ro, opt, enum, subType:string-->software
  </upgradeStage>
</upgradeStatus>

```

12.1.7 Time Settings

12.1.7.1 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},{$.Time.timeMode,eq,NTP}-->CST-8:00:00DST00:30:00,M4.1.0/02:00:00,M10.5.0/02:00:00
  </timeZone>
  <satelliteInterval>
    <!--ro, opt, int, satellite time synchronization interval, step:1, unit:min, desc:unit: minute-->60
  </satelliteInterval>
  <isSummerTime>
    <!--ro, opt, bool, whether the device time returned currently is in DST (Daylight Saving Time) system-->true
  </isSummerTime>
  <platformType>
    <!--ro, opt, enum, platform type, subType:string, dep:and,{$.Time.timeMode,eq,platform}, desc:exists only when the timeMode is selected as platform-->EZVIZ
  </platformType>
  <platformNo>
    <!--ro, opt, int, platform No., range:[1,2], dep:and,{$.Time.timeMode,eq,GB28181}, desc:it is the only ID, which is configured via platformNo in GB28181List, related URI: /ISAPI/System/Network/SIP/<SIPServerID>-->1
  </platformNo>
</Time>

```

12.1.7.2 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},{$.Time.timeMode,eq,NTP}-->CST-8:00:00DST00:30:00,M4.1.0/02:00:00,M10.5.0/02:00:00
    </timeZone>
    <satelliteInterval>
      <!--opt, int, satellite time synchronization interval, step:1, unit:min, desc:unit: minute-->60
    </satelliteInterval>
    <isSummerTime>
      <!--opt, bool, whether the time returned by the current device is that in the DST (daylight saving time)-->true
    </isSummerTime>
    <platformType>
      <!--opt, enum, platform type, subType:string, dep:and,{$.Time.timeMode,eq,platform}, desc:exists only when the timeMode is selected as platform, related URI: /ISAPI/System/Network/EZVIZ-->EZVIZ
    </platformType>
    <platformNo>
      <!--opt, int, platform No., range:[1,2], dep:and,{$.Time.timeMode,eq,GB28181}, desc:it is the only ID, which is configured via platformNo in GB28181List, related URI: /ISAPI/System/Network/SIP/<SIPServerID>-->1
    </platformNo>
  </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>

```

12.1.7.3 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>
    <platformType opt="EZVIZ">
        <!--ro, opt, enum, platform type, subType:string, dep:and,{$.Time.timeMode,eq,platform}, attr:opt{opt, string}, desc:platform type-->EZVIZ
    </platformType>
    <platformNo min="1" max="2">
        <!--ro, opt, int, platform No., range:[1,2], dep:and,{$.Time.timeMode,eq,GB28181}, attr:min{req, int},max{req, int}, desc:it is the only ID, which is configured via platformNo in GB28181List, related URI: /ISAPI/System/Network/SIP/<SIPServerID>-->1
    </platformNo>
    <isSupportHistoryTime>
        <!--ro, opt, bool, supported capability of the historical time synchronization list, desc:related URI: /ISAPI/System/time/historyInfo?format=json-->true
    </isSupportHistoryTime>
    <isSupportTimeFilter>
        <!--ro, opt, bool, supported capability of filtering time synchronization, desc:related URI: /ISAPI/System/time/filter/capabilities?format=json-->true
    </isSupportTimeFilter>
    <displayFormat opt="MM/dd/yyyy hh:mm,mm,dd-MM-yyyy hh:MM-dd-yyyy hh:mm,yyyy-MM-dd hh:mm">
        <!--ro, opt, enum, time display format, subType:string, attr:opt{req, string}, desc:if this node is returned, it indicates that the device supports configuration time display format, related URI: /ISAPI/System/time/timeType?format=json-->MM/dd/yyyy hh:mm
    </displayFormat>
    <isSupportSyncDeviceNTPInfoToCamera>
        <!--ro, opt, bool, the capability of synchronizing device's NTP service information with the camera, desc:related URI: /ISAPI/System/time/SyncDeviceNTPInfoToCamera/capabilities?format=json-->true
    </isSupportSyncDeviceNTPInfoToCamera>
    <isSupportNTPService>
        <!--ro, opt, bool-->true
    </isSupportNTPService>
</Time>

```

12.1.7.4 Set device local time

Request URL

PUT /ISAPI/System/time/localTime

Query Parameter

None

Request Message

Binary Data

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>

```

12.1.7.5 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>
    <ipAddress>
      <!--opt, string, IPv4 address, range:[1,32]-->192.168.1.112
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPv6 address, range:[1,128]-->1030:C9B4:FF12:48AA:1A2B
    </ipv6Address>
    <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>
    <enabled>
      <!--opt, bool, whether to enable, desc:disabled (by default)-->false
    </enabled>
  </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>
```

12.1.7.6 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>
    <ipAddress>
      <!--ro, opt, string, IPv4 address, range:[1,32]-->192.168.1.112
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPv6 address, range:[1,128]-->1030:C9B4:FF12:48AA:1A2B
    </ipv6Address>
    <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>
    <enabled>
      <!--ro, opt, bool, whether to enable, desc:disabled (by default)-->false
    </enabled>
  </NTPServer>
</NTPServerList>

```

12.1.7.7 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>
  <ipAddress>
    <!--opt, string, IPv4 address, range:[1,32], desc:IPv4 address-->192.168.1.112
  </ipAddress>
  <ipv6Address>
    <!--opt, string, IPv6 address, range:[1,128], desc:IPv6 address-->1030:C9B4:FF12:48AA:1A2B
  </ipv6Address>
  <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>
  <enabled>
    <!--opt, bool-->false
  </enabled>
</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>
  <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>

```

12.1.7.8 Get the parameters of a NTP server

Request URL

GET /ISAPI/System/time/ntpServers/<NTPServerID>

Query Parameter

Parameter Name	Parameter Type	Description
NTPServerID	string	NTP server No.

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], dep:and,{$.NTPServer.addressingFormatType,eq,hostname}-->xxx12345
  </hostName>
  <ipAddress>
    <!--ro, opt, string, IPv4 address, range:[1,32], dep:and,{$.NTPServer.addressingFormatType,eq,ipAddress}-->192.168.1.112
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPv6 address, range:[1,128]-->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>
  <enabled>
    <!--ro, opt, bool-->false
  </enabled>
  <devicePortNoValid>
    <!--ro, opt, enum, subType:string-->yes
  </devicePortNoValid>
  <portType>
    <!--ro, opt, enum, subType:string-->auto
  </portType>
  <customPortNo>
    <!--ro, opt, int, range:[1,65535], step:1, dep:and,{$.NTPServer.portType,eq,custom}-->1
  </customPortNo>
  <hostNameExampleList>
    <!--ro, opt, array, subType:object-->
    <hostNameExample>
      <!--ro, opt, object-->
      <hostName>
        <!--ro, opt, string-->xxx12345
      </hostName>
    </hostNameExample>
  </hostNameExampleList>
</NTPServer>

```

12.1.7.9 Get the configuration capability of a specific NTP server

Request URL

GET /ISAPI/System/time/ntpServers/<NTPServerID>/capabilities

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 min="1" max="1">
    <!--ro, req, string, ID, range:[0,1], attr:min{req, int},max{req, int}-->1
  </id>
  <addressingFormatType opt="ipaddress,hostname">
    <!--ro, req, enum, IP address type of NTP server, subType:string, attr:opt{req, string}, desc:"ipaddress" (IP address), "hostname" (domain name)-->
  >hostname
    </addressingFormatType>
    <hostName min="0" max="68">
      <!--ro, opt, string, NTP server domain name, range:[0,68], attr:min{req, int},max{req, int}-->time.windows.com
    </hostName>
    <ipAddress min="0" max="32">
      <!--ro, opt, string, IPv4 address, range:[0,32], attr:min{req, int},max{req, int}, desc:IPv4 address-->192.168.1.112
    </ipAddress>
    <ipv6Address min="0" max="128">
      <!--ro, opt, string, IPv6 address, range:[0,128], attr:min{req, int},max{req, int}, desc:IPv6 address-->1030::C9B4:FF12:48AA:1A2B
    </ipv6Address>
    <portNo min="0" max="65535">
      <!--ro, opt, int, port No., range:[0,65535], attr:min{req, int},max{req, int}, desc:port No.-->123
    </portNo>
    <synchronizeInterval min="0" max="10800">
      <!--ro, opt, int, time synchronization interval, range:[0,10800], unit:min, attr:min{req, int},max{req, int}, desc:NTP time synchronization interval, unit: minute-->1440
    </synchronizeInterval>
    <enabled opt="true,false">
      <!--ro, opt, bool, attr:opt{req, string}-->true
    </enabled>
    <portType opt="auto,custom" def="auto">
      <!--ro, opt, enum, subType:string, attr:opt{req, string},def{req, string}-->auto
    </portType>
    <customPortNo min="1" max="65535">
      <!--ro, opt, int, range:[0,65535], dep:and,{$.NTPServer.portType,eq,custom}, attr:min{req, int},max{req, int}-->123
    </customPortNo>
  </NTPServer>

```

12.1.7.10 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>
    <ipAddress min="1" max="32">
      <!--ro, opt, string, IPv4 address, range:[1,32], attr:min{req, int},max{req, int}, desc:IPv4 address-->192.168.1.112
    </ipAddress>
    <ipv6Address min="1" max="128">
      <!--ro, opt, string, IPv6 address, range:[1,128], attr:min{req, int},max{req, int}, desc:IPv6 address-->1030:C9B4:FF12:48AA:1A2B
    </ipv6Address>
    <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>
    <enabled opt="true,false">
      <!--ro, opt, bool, whether to enable, attr:opt{req, string}, desc:true (enable), false (disable)-->true
    </enabled>
  </NTPServer>
</NTPServerList>

```

12.2 Video (General)

12.2.1 Backlight Compensation

12.2.1.1 Set the WDR (Wide Dynamic Range) parameters of a specific channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/hdr

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<Hdr xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <hdrMode>
    <!--req, enum, WDR (Wide Dynamic Range) mode, subType:string, desc:"close", "realHdr"-WDR, "digitalHdr"-digital WDR, "Sdr"-super WDR, "auto"-->Sdr
  </hdrMode>
  <hdrSwitch>
    <!--opt, object-->
    <hdrSwitchMode>
      <!--req, enum, switch mode, subType:string, desc:"open", "timeCtrl"-enabled by time, "lightCtrl"-enabled by brightness. <timeSwitch> nad <lightSwitch>
      are invalid when this node is set to "open"-->lightCtrl
    </hdrSwitchMode>
    <timeSwitch>
      <!--req, object, enable WDR by time-->
      <startHour>
        <!--opt, int, start time (hour)-->1
      </startHour>
      <startMinute>
        <!--opt, int, start time (minute)-->1
      </startMinute>
      <endHour>
        <!--opt, int, start time (hour)-->1
      </endHour>
      <endMinute>
        <!--opt, int, start time (minute)-->1
      </endMinute>
    </timeSwitch>
  </HdrSwitch>
  <hdrLevel>
    <!--opt, enum, WDR Level, subType:string, desc:"Level1", "Level2", "Level3", "Level4", "Level5", "Level6", "Level7"-->level7
  </hdrLevel>
  <sdrLevel>
    <!--opt, int-->1
  </sdrLevel>
</Hdr>

```

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>

```

12.2.1.2 Get the WDR (Wide Dynamic Range) parameters of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/hdr

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<Hdr xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <hdrMode>
    <!--ro, req, enum, WDR mode, subType:string, desc:"close", "realHdr"-WDR, "digitalHdr"-digital WDR, "Sdr"-super WDR, "auto".-->Sdr
  </hdrMode>
  <HdrSwitch>
    <!--ro, opt, object-->
    <hdrSwitchMode>
      <!--ro, req, enum, switch mode, subType:string, desc:"open", "timeCtrl"-enabled by time, "LightCtrl"-enabled by brightness. <timeSwitch> nad
      <LightSwitch> are invalid when this node is set to "open"-->lightCtrl
    </hdrSwitchMode>
    <timeSwitch>
      <!--ro, req, object, enable WDR by time-->
      <startHour>
        <!--ro, opt, int, start time (hour)-->1
      </startHour>
      <startMinute>
        <!--ro, opt, int, start time (minute)-->1
      </startMinute>
      <endHour>
        <!--ro, opt, int, start time (hour)-->1
      </endHour>
      <endMinute>
        <!--ro, opt, int, start time (minute)-->1
      </endMinute>
    </timeSwitch>
  </HdrSwitch>
  <hdrLevel>
    <!--ro, opt, enum, WDR Level, subType:string, desc:"Level1", "Level2", "Level3", "Level4", "Level5", "Level6", "Level7"-->level1
  </hdrLevel>
  <SdrLevel>
    <!--ro, opt, int-->1
  </SdrLevel>
</Hdr>

```

12.2.1.3 Get the WDR (Wide Dynamic Range) capability of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/hdr/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Channel No.

Request Message

None

Response Message

```

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

<Hdr xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <hdrMode opt="close,realHdr,digitalHdr,Sdr,BLC">
    <!--ro, req, enum, WDR mode, subType:string, attr:opt{req, string}, desc:"close", "realHdr"-WDR, "digitalHdr"-digital WDR, "Sdr"-super WDR, "auto".-->
  >Sdr
    </hdrMode>
    <HdrSwitch>
      <!--ro, opt, object-->
      <hdrSwitchMode opt="open,timeCtrl,lightCtrl">
        <!--ro, req, enum, switch mode, subType:string, attr:opt{req, string}, desc:"open", "timeCtrl"-enabled by time, "LightCtrl"-enabled by brightness.>
        <timeSwitch> nad <lightSwitch> are invalid when this node is set to "open"-->lightCtrl
      </hdrSwitchMode>
      <timeSwitch>
        <!--ro, req, object, enable WDR by time-->
        <startHour min="0" max="23">
          <!--ro, opt, int, start time (hour), attr:min{req, int},max{req, int}-->1
        </startHour>
        <startMinute min="0" max="59">
          <!--ro, opt, int, start time (minute), attr:min{req, int},max{req, int}-->1
        </startMinute>
        <endHour min="0" max="23">
          <!--ro, opt, int, start time (hour), attr:min{req, int},max{req, int}-->1
        </endHour>
        <endMinute min="0" max="59">
          <!--ro, opt, int, start time (minute), attr:min{req, int},max{req, int}-->1
        </endMinute>
      </timeSwitch>
    </HdrSwitch>
    <HdrLevel opt="level1,level2,level3,level4,level5,level6,level7">
      <!--ro, opt, enum, WDR Level, subType:string, attr:opt{req, string}, desc:"Level1", "Level2", "Level3", "Level4", "Level5", "Level6", "Level7"-->level7
    </HdrLevel>
    <SdrLevel min="0" max="100" def="50">
      <!--ro, opt, int, dep:and,{$.Hdr.hdrMode,eq,Sdr}, attr:min{req, int},max{req, int},def{req, int}-->1
    </SdrLevel>
  </Hdr>

```

12.2.1.4 Get the halo inhibition parameters of the traffic lights of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/lightHal

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

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

<lightHal xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, halo inhibition parameters of the traffic Lights, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool, whether to enable the function-->true
  </enabled>
  <halLevel>
    <!--ro, req, int, halo inhibition intensity, range:[0,100]-->0
  </halLevel>
  <halRange>
    <!--ro, req, int, halo inhibition range, range:[0,2064]-->0
  </halRange>
</lightHal>

```

12.2.1.5 Set the halo inhibition parameters of the traffic lights of a specific channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/lightHal

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

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

<lightHal xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, halo inhibition parameters of the traffic lights, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool, whether to enable the function-->true
  </enabled>
  <halLevel>
    <!--req, int, halo inhibition intensity, range:[0,100]-->0
  </halLevel>
  <halRange>
    <!--req, int, halo inhibition range, range:[0,2064]-->0
  </halRange>
</lightHal>
```

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>
```

12.2.2 Exposure Settings

12.2.2.1 Get the configuration parameters of the low illumination electronic shutter of a specified channel

Request URL

GET /ISAPI/Image/channels/<channelID>/DSS

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

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

<DSS xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, Low illumination electronic shutter, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool, whether to enable-->true
  </enabled>
  <DSSLevel>
    <!--ro, opt, enum, Low illumination electronic shutter Level in exposure, subType:string,
desc:"*1.25,*1.5,*2,*3*4,*6,*8,*12,*16,*24,*32,*48,*64,*96,*128,*256,*512,auto". "*2" indicates that the speed of the Low illumination electronic shutter is
2 times of the normal shutter speed-->*1.25
  </DSSLevel>
  <mode>
    <!--ro, opt, enum, mode, subType:string, desc:"frame" (frame mode, it indicates that node DSSLevel is valid), "Level" (level mode, it indicates that
slowShutterLevel is valid). If this node does not exist, the default mode is frame mode-->frame
  </mode>
  <slowShutterLevel>
    <!--ro, req, int, shutter level, range:[1,6]-->1
  </slowShutterLevel>
</DSS>

```

12.2.2.2 Set the configuration parameters of the low illumination electronic shutter for a specified channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/DSS

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<DSS xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, Low illumination electronic shutter, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool, whether to enable-->true
  </enabled>
  <DSSLevel>
    <!--opt, enum, Low illumination Level, subType:string, desc:Low illumination Level-->*1.25
  </DSSLevel>
  <mode>
    <!--opt, enum, mode, subType:string, desc:"frame" (frame mode, it indicates that node DSSLevel is valid), "Level" (level mode, it indicates that
slowShutterLevel is valid). If this node does not exist, the default mode is frame mode-->frame
  </mode>
  <slowShutterLevel>
    <!--req, int, slow shutter level in exposure, range:[1,6]-->1
  </slowShutterLevel>
</DSS>

```

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>

```

12.2.2.3 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>
  <autoIrisLevel>
    <!--opt, int, automatic aperture level-->1
  </autoIrisLevel>
  <OverexposeSuppress>
    <!--opt, object-->
    <enabled>
      <!--req, bool-->true
    </enabled>
    <Type>
      <!--opt, enum, "AUTO", "MANUAL", subType:string, desc:"AUTO", "MANUAL"-->AUTO
    </Type>
    <DistanceLevel>
      <!--opt, int, focal length-->1
    </DistanceLevel>
    <shortIRDistanceLevel>
      <!--opt, int, this node depends on <Type>-->1
    </shortIRDistanceLevel>
    <longIRDistanceLevel>
      <!--opt, int, this node depends on <Type>-->1
    </longIRDistanceLevel>
    <supplementLightIntensity1>
      <!--opt, int, range:[1,100], dep:and, ${.Exposure.OverexposeSuppress.Type,eq,MANUAL}-->1
    </supplementLightIntensity1>
    <supplementLightIntensity2>
      <!--opt, int, range:[1,100], dep:and, ${.Exposure.OverexposeSuppress.Type,eq,MANUAL}-->1
    </supplementLightIntensity2>
    <supplementLightIntensity3>
      <!--opt, int, range:[1,100], dep:and, ${.Exposure.OverexposeSuppress.Type,eq,MANUAL}-->1
    </supplementLightIntensity3>
    <supplementLightIntensity4>
      <!--opt, int, range:[1,100], dep:and, ${.Exposure.OverexposeSuppress.Type,eq,MANUAL}-->1
    </supplementLightIntensity4>
  </OverexposeSuppress>
  <pIris>
    <!--opt, object, aperture-->
    <pIrisType>
      <!--opt, enum, aperture type, subType:string, desc:aperture type-->AUTO
    </pIrisType>
    <IrisLevel>
      <!--opt, int, aperture level-->1
    </IrisLevel>
  </pIris>
  <faceExposure>
    <!--opt, object, face exposure-->
    <enabled>
      <!--opt, bool, whether to enable face exposure-->true
    </enabled>
    <sensitivity>
      <!--opt, int, sensitivity of face exposure-->1
    </sensitivity>
  </faceExposure>
  <pIrisGeneral>
    <!--opt, object, it is available when <ExposureType> is set to "pIris-General", desc:it is available when ExposureType is set to "pIris-General"-->
    <irisLevel>
      <!--opt, int, aperture level of general lens, range:[1,100]-->1
    </irisLevel>
  </pIrisGeneral>
</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-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

12.2.2.4 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	Camera ID
parameterType	string	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it wil be the value of the parameter in the current PUT URL.

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>
  <autoIrisLevel>
    <!--ro, opt, int-->1
  </autoIrisLevel>
  <OverexposeSuppress>
    <!--ro, opt, object-->
    <enabled>
      <!--ro, req, bool-->true
    </enabled>
    <Type>
      <!--ro, opt, enum, this node depends on <enabled>, "AUTO,MANUAL", subType:string, desc:this node depends on <enabled>, "AUTO,MANUAL"-->AUTO
    </Type>
    <DistanceLevel>
      <!--ro, opt, int, this node depends on <Type>-->1
    </DistanceLevel>
    <shortIRDistanceLevel>
      <!--ro, opt, int, this node depends on <Type>-->1
    </shortIRDistanceLevel>
    <longIRDistanceLevel>
      <!--ro, opt, int, this node depends on <Type>-->1
    </longIRDistanceLevel>
    <supplementLightIntensity1>
      <!--ro, opt, int, range:[1,100], dep:and,{$.Exposure.OverexposeSuppress.Type,eq,MANUAL}-->1
    </supplementLightIntensity1>
    <supplementLightIntensity2>
      <!--ro, opt, int, range:[1,100], dep:and,{$.Exposure.OverexposeSuppress.Type,eq,MANUAL}-->1
    </supplementLightIntensity2>
    <supplementLightIntensity3>
      <!--ro, opt, int, range:[1,100], dep:and,{$.Exposure.OverexposeSuppress.Type,eq,MANUAL}-->1
    </supplementLightIntensity3>
    <supplementLightIntensity4>
      <!--ro, opt, int, range:[1,100], dep:and,{$.Exposure.OverexposeSuppress.Type,eq,MANUAL}-->1
    </supplementLightIntensity4>
    <highLightDistanceLevel>
      <!--ro, opt, int, range:[1,100]-->50
    </highLightDistanceLevel>
    <lowLightDistanceLevel>
      <!--ro, opt, int, range:[1,100]-->50
    </lowLightDistanceLevel>
  </OverexposeSuppress>
  <pIris>
    <!--ro, opt, object, this node depends on <ExposureType>, "AUTO,MANUAL"-->
    <pIrisType>
      <!--ro, opt, enum, subType:string-->AUTO
    </pIrisType>
    <IrisLevel>
      <!--ro, opt, int, this node depends on <pIrisType>-->1
    </IrisLevel>
  </pIris>
  <faceExposure>
    <!--ro, opt, object, face exposure-->
    <enabled>
      <!--ro, opt, bool-->true
    </enabled>
    <sensitivity>
      <!--ro, opt, int, sensitivity-->1
    </sensitivity>
  </faceExposure>
  <pIrisGeneral>
    <!--ro, opt, object, it is available when <ExposureType> is set to "pIris-General", desc:it is valid when <ExposureType> is set to "pIris-General"-->
    <irisLevel>
      <!--ro, opt, int, iris level of general lens, range:[1,100]-->1
    </irisLevel>
  </pIrisGeneral>
  <combinationParams>
    <!--ro, opt, int, range:[0,100]-->50
  </combinationParams>
  <opticalFilter>
    <!--ro, opt, enum, subType:string-->none
  </opticalFilter>
  <lowGhostingExposure>
    <!--ro, opt, object-->
    <lowGhostingShutterLevel>
      <!--ro, opt, enum, subType:string-->1/100
    </lowGhostingShutterLevel>
  </lowGhostingExposure>
  <exposureCompensateLevel>
    <!--ro, opt, int, range:[-100,100]-->0
  </exposureCompensateLevel>
  <exposureControl>
    <!--ro, opt, enum, subType:string-->center
  </exposureControl>
</Exposure>

```

12.2.2.5 Set the multi-shutter parameters for a specified channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/multishut

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

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

<ImageMultishut xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <doubleShutEnable>
    <!--req, bool, whether to enable dual-shutter-->true
  </doubleShutEnable>
  <codecType>
    <!--req, enum, type, subType:string, desc:type-->all_ia
  </codecType>
  <threeShutEnable>
    <!--opt, bool, whether to enable three-shutter-->true
  </threeShutEnable>
  <doubleVideoEnable>
    <!--opt, bool, whether to enable dual-video-->true
  </doubleVideoEnable>
</ImageMultishut>
```

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>
```

12.2.2.6 Get the multi-shutter parameters of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/multishut?parameterType=<parameterType>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it will be the value of the parameter in the current PUT URL.

Request Message

None

Response Message

```

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

<ImageMultishut xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <doubleShutEnable>
    <!--ro, req, bool, whether to enable dual-shutter-->true
  </doubleShutEnable>
  <codecType>
    <!--ro, req, enum, type, subType:string, desc:type-->all_ia
  </codecType>
  <threeShutEnable>
    <!--ro, opt, bool-->true
  </threeShutEnable>
  <doubleVideoEnable>
    <!--ro, opt, bool-->true
  </doubleVideoEnable>
  <multiCapture>
    <!--ro, opt, object-->
    <iaCaptureEnabled>
      <!--ro, opt, bool-->true
    </iaCaptureEnabled>
    <recordCaptureEnabled>
      <!--ro, opt, bool-->true
    </recordCaptureEnabled>
    <captureCombineType>
      <!--ro, opt, enum, subType:string-->none
    </captureCombineType>
  </multiCapture>
</ImageMultishut>

```

12.2.2.7 Set the regional exposure parameters of a specified channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/regionalExposure

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<RegionalExposure xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <StartPoint>
    <!--opt, object-->
    <positionX>
      <!--req, int, range:[0,1000]-->0
    </positionX>
    <positionY>
      <!--req, int, range:[0,1000]-->0
    </positionY>
  </StartPoint>
  <EndPoint>
    <!--opt, object-->
    <positionX>
      <!--req, int, range:[0,1000]-->0
    </positionX>
    <positionY>
      <!--req, int, range:[0,1000]-->0
    </positionY>
  </EndPoint>
  <enabled>
    <!--opt, bool-->false
  </enabled>
  <showRuleEnabled>
    <!--opt, bool-->false
  </showRuleEnabled>
</RegionalExposure>

```

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>

```

12.2.2.8 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>
  <scheduleEnabled>
    <!--opt, bool-->true
  </scheduleEnabled>
  <scheduleShutterLevel>
    <!--opt, int-->1
  </scheduleShutterLevel>
  <ScheduleList>
    <!--opt, array, subType:object-->
    <Schedule>
      <!--opt, object-->
      <startHour>
        <!--opt, int, start time (hour)-->1
      </startHour>
      <startMinute>
        <!--opt, int, start time (minute)-->1
      </startMinute>
      <endHour>
        <!--opt, int, end time (hour)-->1
      </endHour>
      <endMinute>
        <!--opt, int, end time (minute)-->1
      </endMinute>
    </Schedule>
  </ScheduleList>
  <redShutterLevel>
    <!--opt, int, range:[0,20000]-->1
  </redShutterLevel>
</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>

```

12.2.2.9 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	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it wil be the value of the parameter in the current PUT URL.

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>
  <scheduleEnabled>
    <!--ro, opt, bool-->true
  </scheduleEnabled>
  <scheduleShutterLevel>
    <!--ro, opt, int-->1
  </scheduleShutterLevel>
  <ScheduleList>
    <!--ro, opt, array, subType:object-->
    <Schedule>
      <!--ro, opt, object-->
      <startHour>
        <!--ro, opt, int, start time (hour)-->1
      </startHour>
      <startMinute>
        <!--ro, opt, int, start time (minute)-->1
      </startMinute>
      <endHour>
        <!--ro, opt, int, end time (hour)-->1
      </endHour>
      <endMinute>
        <!--ro, opt, int, end time (minute)-->1
      </endMinute>
    </Schedule>
  </ScheduleList>
  <redShutterLevel>
    <!--ro, opt, int, range:[0,20000]-->1
  </redShutterLevel>
  <maxShutterLevelLimit>
    <!--ro, opt, enum, subType:string-->1/1
  </maxShutterLevelLimit>
  <minShutterLevelLimit>
    <!--ro, opt, enum, subType:string-->1/1
  </minShutterLevelLimit>
</Shutter>

```

12.2.2.10 Set the lens aperture parameters for a specified video input channel

Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/iris

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<IrisData xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, aperture, attr:version{req, string, protocolVersion}-->
  <iris>
    <!--opt, int, aperture adjusting speed, range:[-100,100], desc:aperture adjusting speed-->1
  </iris>
</IrisData>

```

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>

```

12.2.3 Focus Settings

12.2.3.1 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>
  <RegionalExposure>
    <!--ro, opt, object-->
    <enabled opt="true,false" def="false">
      <!--ro, opt, bool, attr:opt{req, string},def{req, string}-->true
    </enabled>
    <showRlueEnabled opt="true,false" def="false">
      <!--ro, opt, bool, attr:opt{req, string},def{req, string}-->true
    </showRlueEnabled>
  </RegionalExposure>
</ImageCap>

```

12.2.3.2 Set the regional focus parameters for a specified channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/regionalFocus

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

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

<RegionalFocus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, regional focus parameters, attr:version{req, string, protocolVersion}-->
  <StartPoint>
    <!--opt, object, start point, desc:the origin is the upper-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>
  </StartPoint>
  <EndPoint>
    <!--opt, object, end point, desc:the origin is the upper-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>
  </EndPoint>
</RegionalFocus>
```

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>
```

12.2.3.3 Set the focus parameters for a specified video input channel

Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>/focus

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

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

<FocusData xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, parameters of focus control, attr:version{req, string, protocolVersion}-->
  <focus>
    <!--req, int, focus control, range:[-100,100], desc:focus control-->0
  </focus>
</FocusData>
```

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>

```

12.2.4 Image Adjustment Settings

12.2.4.1 Set the gamma correction parameters for a specified channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/gammaCorrection

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<gammaCorrection xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, gamma correction parameters, attr:version{req, string, protocolVersion}-->
  <gammaCorrectionEnabled>
    <!--opt, bool, whether to enable gamma correction-->true
  </gammaCorrectionEnabled>
  <gammaCorrectionLevel>
    <!--opt, int, gamma correction Level-->1
  </gammaCorrectionLevel>
</gammaCorrection>

```

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>

```

12.2.4.2 Get the gamma correction parameters of a specified channel

Request URL

GET /ISAPI/Image/channels/<channelID>/gammaCorrection?parameterType=<parameterType>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it wil be the value of the parameter in the current PUT URL.

Request Message

None

Response Message

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

<gammaCorrection xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <gammaCorrectionEnabled>
    <!--ro, opt, bool-->true
  </gammaCorrectionEnabled>
  <gammaCorrectionLevel>
    <!--ro, opt, int-->1
  </gammaCorrectionLevel>
</gammaCorrection>
```

12.2.4.3 Get the capability of configuring parameters for visual and thermal image fusion

Request URL

GET /ISAPI/Image/dualLightFusion/capabilities?format=json

Query Parameter

None

Request Message

None

Response Message

```
{
  "enabled": {
    /*ro, opt, object, whether to enable visual and thermal image fusion*/
    "@opt": [true, false]
    /*ro, req, array, range, subType:bool*/
  },
  "switchingThreshold": {
    /*ro, opt, object*/
    "@min": 0,
    /*ro, req, int, range, range:[0,7]*/
    "@max": 7
    /*ro, req, int, range, range:[0,7], step:1*/
  },
  "channelID": {
    /*ro, opt, object*/
    "@min": 1,
    /*ro, opt, int, range:[0,7]*/
    "@max": 8
    /*ro, opt, int, range:[0,7], step:1*/
    "@opt": [1, 4]
    /*ro, opt, array, subType:int*/
  }
}
```

12.2.4.4 Set the parameters for visual and thermal image fusion

Request URL

PUT /ISAPI/Image/dualLightFusion?format=json&channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```
{
  "enabled": true,
  /*opt, bool, whether to enable visual and thermal image fusion, desc:when the illumination is low, the visual and thermal image fusion can improve the live view effect*/
  "switchingThreshold": 2
  /*opt, int, range:[0,7], step:1*/
}
```

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, it corresponds to subStatusCode*/
  "errorMsg": "ok"
  /*ro, opt, string, error description, desc:this field is required when the value of statusCode is not 1*/
}
```

12.2.4.5 Get the parameters for visual and thermal image fusion

Request URL

GET /ISAPI/Image/dualLightFusion?format=json&channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Required if this field is supported.

Request Message

None

Response Message

```
{
  "enabled": true,
  /*ro, opt, bool, whether to enable visual and thermal image fusion, desc:when the illumination is low, the visual and thermal image fusion can improve the live view effect*/
  "switchingThreshold": 2
  /*ro, opt, int, range:[0,7], step:1*/
}
```

12.2.5 Image Adjustment

12.2.5.1 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>
  <DSS>
    <!--opt, object, DSS-->
    <enabled>
      <!--req, bool, whether to enable the function-->true
    </enabled>
    <mode>
      <!--opt, enum, subType:string-->frame
    </mode>
    <slowShutterLevel>
      <!--opt, int, range:[1,6]-->1
    </slowShutterLevel>
  </DSS>
  <WhiteBalance>
    <!--opt, object, white balance-->
    <WhiteBalanceStyle>
      <!--opt, enum, subType:string-->auto
    </WhiteBalanceStyle>
    <whiteBalanceLevel>
      <!--opt, int, range:[0,100]-->1
    </whiteBalanceLevel>
  </WhiteBalance>
  <Exposure>
    <!--opt, object-->
    <ExposureType>
      <!--req, enum, subType:string-->auto
    </ExposureType>
    <pIris>
      <!--opt, object, aperture-->
      <pIrisType>
        <!--opt, enum, subType:string-->MANUAL
      </pIrisType>
      <IrisLevel>
        <!--opt, int, range:[1,100]-->1
      </IrisLevel>
    </pIris>
  </Exposure>
  <Sharpness>
    <!--opt, object-->
    <SharpnessLevel>
      <!--req, int, range:[0,100]-->1
    </SharpnessLevel>
  </Sharpness>
  <gammaCorrection>
    <!--opt, object-->
    <gammaCorrectionEnabled>
      <!--opt, bool-->true
    </gammaCorrectionEnabled>
    <gammaCorrectionLevel>
      <!--opt, int, range:[0,100]-->1
    </gammaCorrectionLevel>
  </gammaCorrection>
  <Shutter>
    <!--opt, object-->
    <ShutterLevel>
      <!--opt, enum, subType:string-->1/1
    </ShutterLevel>
  </Shutter>
  <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>
  </Color>
</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>

```

12.2.5.2 Set the brightness enhancement parameters for a specified channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/brightEnhance

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<BrightEnhance xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, brightness enhancement parameters, attr:version{req, string, protocolVersion}-->
  <brightEnhanceEnabled>
    <!--req, bool, whether to enable-->true
  </brightEnhanceEnabled>
  <brightEnhanceLevel>
    <!--req, int, brightness enhancement level, range:[0,100]-->1
  </brightEnhanceLevel>
</BrightEnhance>

```

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>

```

12.2.5.3 Get the brightness enhancement parameters of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/brightEnhance?parameterType=<parameterType>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it wil be the value of the parameter in the current PUT URL.

Request Message

None

Response Message

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

<BrightEnhance xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <brightEnhanceEnabled>
    <!--ro, req, bool, whether to enable brightness enhancement-->true
  </brightEnhanceEnabled>
  <brightEnhanceLevel>
    <!--ro, req, int, brightness enhancement Level, range:[0,100]-->1
  </brightEnhanceLevel>
</BrightEnhance>
```

12.2.5.4 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	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, 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>
  <DSS>
    <!--ro, opt, object, DSS-->
    <enabled opt="true,false">
      <!--ro, req, bool, whether to enable the function, attr:opt{req, string}-->true
    </enabled>
    <mode opt="frame,level">
      <!--ro, opt, enum, mode, subType:string, attr:opt{req, string}, desc:WDR mode: "open,close,auto"-->frame
    </mode>
    <slowShutterLevel min="1" max="6">
      <!--ro, opt, int, range:[1,6], attr:min{req, int},max{req, int}-->1
    </slowShutterLevel>
  </DSS>
  <whiteBalance>
    <!--ro, opt, object, white balance-->
    <whiteBalanceLevel min="0" max="100">
      <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
    </whiteBalanceLevel>
  </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,SF5014M-12MP,SF3018M-10MP,MZ1140M,MZ2514M-10MP,MZ7018M-8MP,MZ2516M-10MPIR,indoor50HZ,indoor60HZ,lowGhosting,customize">
```

```

<!--ro, req, string, attr:opt{req, string}-->test
</ExposureType>
<pIris>
    <!--ro, opt, object, aperture-->
    <pIrisType opt="AUTO,MANUAL">
        <!--ro, opt, enum, subType:string, attr:opt{req, string}-->MANUAL
    </pIrisType>
    <IrisLevel min="1" max="100">
        <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
    </IrisLevel>
</pIris>
</Exposure>
<Sharpness>
    <!--ro, opt, object-->
    <SharpnessLevel min="0" max="255">
        <!--ro, req, int, attr:min{req, int},max{req, int}-->1
    </SharpnessLevel>
</Sharpness>
<gammaCorrection>
    <!--ro, opt, object-->
    <gammaCorrectionEnabled opt="true,false">
        <!--ro, opt, bool, attr:opt{req, string}-->true
    </gammaCorrectionEnabled>
    <gammaCorrectionLevel min="0" max="10">
        <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
    </gammaCorrectionLevel>
</gammaCorrection>
<Shutter>
    <!--ro, opt, object-->
    <ShutterLevel>
        <!--ro, opt, string, attr:opt{req, string}-->test
    </ShutterLevel>
</Shutter>
<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="255">
        <!--ro, opt, int, brightness, attr:min{req, int},max{req, int}-->0
    </brightnessLevel>
    <contrastLevel min="0" max="255">
        <!--ro, opt, int, contrast, attr:min{req, int},max{req, int}-->0
    </contrastLevel>
    <saturationLevel min="0" max="255">
        <!--ro, opt, int, saturation, attr:min{req, int},max{req, int}-->0
    </saturationLevel>
</Color>
<DayNightGate>
    <!--ro, opt, object-->
    <brightnessThreld min="0" max="100">
        <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
    </brightnessThreld>
</DayNightGate>
<NoiseReduceExt>
    <!--ro, opt, object-->
    <mode opt="close,general,trailingNoiseBalance,advanced">
        <!--ro, req, enum, subType:string, attr:opt{req, string}-->close
    </mode>
</NoiseReduceExt>
<GeneralMode>
    <!--ro, opt, object-->
    <generalLevel min="0" max="255">
        <!--ro, opt, int, attr:min{req, int},max{req, int}-->50
    </generalLevel>
</GeneralMode>
<AdvancedMode>
    <!--ro, opt, object-->
    <spatialLevel min="0" max="100">
        <!--ro, opt, int, attr:min{req, int},max{req, int}-->50
    </spatialLevel>
    <temporalLevel min="0" max="100">
        <!--ro, opt, int, attr:min{req, int},max{req, int}-->50
    </temporalLevel>
</AdvancedMode>
<NoiseReduceExt>
<isSupportJPEGParam>
    <!--ro, opt, bool-->true
</isSupportJPEGParam>
<isSupportIcr>
    <!--ro, opt, bool-->true
</isSupportIcr>
<isSupportMultishut>
    <!--ro, opt, bool-->true
</isSupportMultishut>
<isSupportPlateBright>
    <!--ro, opt, bool-->true
</isSupportPlateBright>
<NoiseReduce2D>
    <!--ro, opt, object-->

```

```

<!--ro, opt, object-->
<noiseReduce2DEnable>
  <!--ro, opt, bool-->true
</noiseReduce2DEnable>
<noiseReduce2DLevel min="0" max="255">
  <!--ro, opt, int, attr:min{req, int},max{req, int}-->50
</noiseReduce2DLevel>
</NoiseReduce2D>
<BrightEnhance>
  <!--ro, opt, object-->
  <brightEnhanceLevel min="0" max="100">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->50
  </brightEnhanceLevel>
</BrightEnhance>
</ImageChannel>

```

12.2.5.5 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>
  <hueLevel>
    <!--opt, int, hue-->55
  </hueLevel>
  <grayScale>
    <!--opt, object, gray scale-->
    <grayScaleMode>
      <!--opt, enum, gray scale mode, subType:string, desc:"indoor", "outdoor"-->outdoor
    </grayScaleMode>
  </grayScale>
  <nightMode>
    <!--opt, bool, whether to enable night mode, when its value is "true", the saturation can be adjusted, otherwise, the saturation cannot be adjusted-->
    <true>
      <nightMode>
        <redBrightnessLevel>
          <!--opt, int, range:[0,100]-->0
        </redBrightnessLevel>
        <sharpnessLevel>
          <!--opt, int, range:[0,100]-->0
        </sharpnessLevel>
      </nightMode>
    </true>
  </nightMode>
</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>
      <!--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>

```

12.2.5.6 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	Channel No.
parameterType	string	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it will be the value of the parameter in the current PUT URL.

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>
  <hueLevel>
    <!--ro, opt, int, hue-->55
  </hueLevel>
  <grayScale>
    <!--ro, opt, object-->
    <grayScaleMode>
      <!--ro, opt, enum, gray scale mode, subType:string, desc:"indoor", "outdoor"-->outdoor
    </grayScaleMode>
  </grayScale>
  <nightMode>
    <!--ro, opt, bool, whether to enable night mode-->true
  </nightMode>
  <redBrightnessLevel>
    <!--ro, opt, int, range:[0,100]-->0
  </redBrightnessLevel>
  <sharpnessLevel>
    <!--ro, opt, int, sharpness, range:[0,100]-->0
  </sharpnessLevel>
  <hueMode>
    <!--ro, opt, enum, subType:string-->warm
  </hueMode>
</Color>

```

12.2.5.7 Set the gray scale range parameters of a specific channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/grayRange

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

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

<GrayRange xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, gray scale range, attr:version{req, string, protocolVersion}-->
  <grayValueType>
    <!--req, enum, gray scale value, subType:string, desc:"hightContrast"-high contrast with less details in shadows, the value ranges from 0 to 255;
    "moreDetail"-more detail in shadows but low contrast, the value ranges from 16 to 235. "hightContrast" (high contrast with less details in shadows),
    "moreDetail" (more detail in shadows but low contrast)-->hightContrast
  </grayValueType>
</GrayRange>
```

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>
```

12.2.5.8 Get the grayscale range capability of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/grayRange/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

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

<GrayRange xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <grayValueType opt="hightContrast,moreDetail">
    <!--ro, req, string, grayscale value, attr:opt{req, string}, desc:"hightContrast"-high contrast with less details in shadows, the value ranges from 0 to
    255; "moreDetail"-more detail in shadows but low contrast,-->test
  </grayValueType>
</GrayRange>
```

12.2.5.9 Get the grayscale range parameters of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/grayRange?parameterType=<parameterType>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it will be the value of the parameter in the current PUT URL.

Request Message

None

Response Message

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

<GrayRange xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <grayValueType>
    <!--ro, req, enum, grayscale value, subType:string, desc:"highContrast"-high contrast with less details in shadows, the value ranges from 0 to 255;
    "moreDetail"-more detail in shadows but low contrast, the value ranges from 16 to 235. "highContrast" (high contrast with less details in shadows),
    "moreDetail" (more detail in shadows but low contrast)-->highContrast
  </grayValueType>
</GrayRange>
```

12.2.5.10 Get the contrast enhancement parameters of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/lse

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

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

<LSE xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <lseLevel>
    <!--ro, opt, int, contrast enhancement level, range:[0,100]-->1
  </lseLevel>
  <lseHaloLevel>
    <!--ro, opt, int, halo inhibition intensity, range:[0,100]-->1
  </lseHaloLevel>
  <switchMode>
    <!--ro, opt, enum, switch mode, subType:string, desc:"open", "timeCtrl"-enabled by time, "LightCtrl"-enabled by brightness-->open
  </switchMode>
  <timeSwitch>
    <!--ro, req, object, switch time, dep:and,{$.LSE.switchMode,eq,timeCtrl}-->
    <startTime>
      <!--ro, opt, datetime, start time-->1970-01-01T00:00:00+08:00
    </startTime>
    <endTime>
      <!--ro, opt, datetime, end time-->1970-01-01T00:00:00+08:00
    </endTime>
  </timeSwitch>
  <lightLevel>
    <!--ro, req, int, brightness level, dep:and,{$.LSE.switchMode,eq,LightCtrl}-->1
  </lightLevel>
</LSE>
```

12.2.5.11 Set the contrast enhancement parameters of a specific channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/lse

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

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

<LSE xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <lseLevel>
    <!--opt, int, contrast enhancement level, range:[0,100]-->1
  </lseLevel>
  <lseHaloLevel>
    <!--opt, int, halo inhibition intensity, range:[0,100]-->1
  </lseHaloLevel>
  <switchMode>
    <!--opt, enum, LSE switch mode, subType:string, desc:"open", "timeCtrl"-enabled by time, "LightCtrl"-enabled by brightness-->open
  </switchMode>
  <timeSwitch>
    <!--req, object, switch time, dep:and,{$.LSE.switchMode,eq,timeCtrl}-->
    <startTime>
      <!--opt, datetime, start time-->1970-01-01T00:00:00+08:00
    </startTime>
    <endTime>
      <!--opt, datetime, end time-->1970-01-01T00:00:00+08:00
    </endTime>
  </timeSwitch>
  <lightLevel>
    <!--req, int, brightness Level, dep:and,{$.LSE.switchMode,eq,LightCtrl}-->1
  </lightLevel>
</LSE>
```

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>
```

12.2.5.12 Get the contrast enhancement capability of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/lse/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

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

<LSE xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <lseLevel>
    <!--ro, req, int, contrast enhancement level, range:[0,100]-->50
  </lseLevel>
  <lseHaloLevel>
    <!--ro, req, int, halo inhibition intensity, range:[0,100]-->1
  </lseHaloLevel>
  <switchMode>
    <!--ro, opt, enum, switch mode, subType:string, desc:"open", "timeCtrl"-enabled by time, "LightCtrl"-enabled by brightness-->open
  </switchMode>
  <timeSwitch>
    <!--ro, req, object, switch time, dep:and,{$.LSE.switchMode,eq,timeCtrl}-->
    <startTime>
      <!--ro, opt, datetime, start time-->1970-01-01T00:00:00+08:00
    </startTime>
    <endTime>
      <!--ro, opt, datetime, end time-->1970-01-01T00:00:00+08:00
    </endTime>
  </timeSwitch>
  <lightLevel>
    <!--ro, opt, int, brightness level, range:[0,100], dep:and,{$.LSE.switchMode,eq,LightCtrl}-->1
  </lightLevel>
</LSE>

```

12.2.5.13 Set the license plate brightness compensation parameters of a specific channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/plateBright

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<PlateBright xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, license plate brightness compensation, attr:version{req, string, protocolVersion}-->
  <plateBrightEnabled>
    <!--req, bool, whether to enable license plate brightness compensation-->true
  </plateBrightEnabled>
  <plateBrightSensitivity>
    <!--opt, int, license plate brightness compensation sensitivity, range:[1,100]-->1
  </plateBrightSensitivity>
  <correctFactorEnabled>
    <!--req, bool, whether to enable supplement light correction factor-->true
  </correctFactorEnabled>
  <correctFactor>
    <!--opt, int, supplement light correction factor, range:[0,100]-->1
  </correctFactor>
</PlateBright>

```

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>

```

12.2.5.14 Get the license plate brightness compensation capability of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/plateBright/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

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

<PlateBright xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, license plate brightness compensation parameters, attr:version{req, string, protocolVersion}-->
  <plateBrightEnabled>
    <!--ro, req, bool, whether to enable license plate brightness compensation-->true
  </plateBrightEnabled>
  <plateBrightSensitivity min="0" max="100">
    <!--ro, opt, int, license plate brightness compensation sensitivity, range:[0,100], attr:min{req, int},max{req, int}-->1
  </plateBrightSensitivity>
  <correctFactorEnabled>
    <!--ro, opt, bool, whether to enable supplement light correction factor-->true
  </correctFactorEnabled>
  <correctFactor min="0" max="100">
    <!--ro, opt, int, supplement light correction factor, range:[0,100], attr:min{req, int},max{req, int}-->1
  </correctFactor>
</PlateBright>
```

12.2.5.15 Get the license plate brightness compensation parameters of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/plateBright?parameterType=<parameterType>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it wil be the value of the parameter in the current PUT URL.

Request Message

None

Response Message

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

<PlateBright xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, license plate brightness compensation parameters, attr:version{req, string, protocolVersion}-->
  <plateBrightEnabled>
    <!--ro, req, bool, whether to enable license plate brightness compensation-->true
  </plateBrightEnabled>
  <plateBrightSensitivity>
    <!--ro, opt, int, license plate brightness compensation sensitivity, range:[0,100]-->1
  </plateBrightSensitivity>
  <correctFactorEnabled>
    <!--ro, opt, bool, whether to enable supplement light correction factor-->true
  </correctFactorEnabled>
  <correctFactor>
    <!--ro, opt, int, supplement light correction factor, range:[0,100]-->1
  </correctFactor>
</PlateBright>
```

12.2.5.16 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	Channel ID

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}-->
    <requestURL>
        <!--ro, opt, 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>
    <MErrCode>
        <!--ro, opt, string-->0x00000000
    </MErrCode>
    <MErrDevSelfEx>
        <!--ro, opt, string-->0x00000000
    </MErrDevSelfEx>
</ResponseStatus>
```

12.2.5.17 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}-->
    <SharpnessMode>
        <!--opt, enum, mode, subType:string, desc:"manual", "auto"-->manual
    </SharpnessMode>
    <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>
      <!--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>

```

12.2.5.18 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	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it wil be the value of the parameter in the current PUT URL.

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}-->
  <SharpnessMode>
    <!--ro, opt, enum, mode, subType:string, desc:"manual", "auto"-->manual
  </SharpnessMode>
  <SharpnessLevel>
    <!--ro, req, int, sharpness Level-->1
  </SharpnessLevel>
</Sharpness>

```

12.2.5.19 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	Video channel No.
parameterType	string	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it wil be the value of the parameter in the current PUT URL.

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>
  <DSS>
    <!--ro, opt, object-->
    <enabled>
      <!--ro, req, bool-->true
    </enabled>
    <mode>
      <!--ro, opt, enum, subType:string-->frame
    </mode>
    <slowShutterLevel>
      <!--ro, opt, int, range:[1,6]-->1
    </slowShutterLevel>
  </DSS>
  <WhiteBalance>
    <!--ro, opt, object-->
    <WhiteBalanceStyle>
      <!--ro, opt, enum, subType:string-->auto
    </WhiteBalanceStyle>
    <whiteBalanceLevel>
      <!--ro, opt, int, range:[0,100]-->1
    </whiteBalanceLevel>
  </WhiteBalance>
  <Exposure>
    <!--ro, opt, object-->
    <ExposureType>
      <!--ro, req, enum, subType:string-->auto
    </ExposureType>
    <pIris>
      <!--ro, opt, object-->
      <pIrisType>
        <!--ro, opt, enum, subType:string-->MANUAL
      </pIrisType>
      <IrisLevel>
        <!--ro, opt, int, range:[1,100]-->1
      </IrisLevel>
    </pIris>
  </Exposure>
  <Sharpness>
    <!--ro, opt, object-->
    <SharpnessLevel>
      <!--ro, req, int, range:[0,100]-->1
    </SharpnessLevel>
  </Sharpness>
  <gammaCorrection>
    <!--ro, opt, object-->
    <gammaCorrectionEnabled>
      <!--ro, opt, bool-->true
    </gammaCorrectionEnabled>
    <gammaCorrectionLevel>
      <!--ro, opt, int, range:[0,100]-->1
    </gammaCorrectionLevel>
  </gammaCorrection>
  <Shutter>
    <!--ro, opt, object-->
    <ShutterLevel>
      <!--ro, opt, enum, subType:string-->1/1
    </ShutterLevel>
  </Shutter>
  <Gain>
    <!--ro, opt, object-->
    <GainLevel>
      <!--ro, opt, string, range:[0,10]-->test
    </GainLevel>
  </Gain>
  <Color>
    <!--ro, opt, object-->
    <brightnessLevel>
      <!--ro, opt, int, range:[0,255]-->0
    </brightnessLevel>
    <contrastLevel>
      <!--ro, opt, int, range:[0,255]-->0
    </contrastLevel>
    <saturationLevel>
      <!--ro, opt, int, range:[0,255]-->0
    </saturationLevel>
  </Color>
  <DayNightGate>
    <!--ro, opt, object-->
    <brightnessThreld>
      <!--ro, opt, int-->1
    </brightnessThreld>
  </DayNightGate>
```

```

<!-- ro, opt, object-->
</brightnessThreshold>
</DayNightGate>
<NoiseReduce2D>
    <!--ro, opt, object-->
    <noiseReduce2DEnable>
        <!--ro, req, bool-->true
    </noiseReduce2DEnable>
    <noiseReduce2DLevel>
        <!--ro, opt, int, range:[0,255]-->1
    </noiseReduce2DLevel>
</NoiseReduce2D>
<BrightEnhance>
    <!--ro, opt, object-->
    <brightEnhanceLevel>
        <!--ro, req, int, range:[0,100]-->1
    </brightEnhanceLevel>
</BrightEnhance>
<NoiseReduceExt>
    <!--ro, opt, object-->
    <mode>
        <!--ro, req, enum, subType:string-->close
    </mode>
    <GeneralMode>
        <!--ro, opt, object-->
        <generalLevel>
            <!--ro, opt, int, range:[0,255]-->1
        </generalLevel>
    </GeneralMode>
    <AdvancedMode>
        <!--ro, opt, object-->
        <spatialLevel>
            <!--ro, opt, int, range:[0,100]-->1
        </spatialLevel>
        <temporalLevel>
            <!--ro, opt, int, range:[0,100]-->1
        </temporalLevel>
    </AdvancedMode>
</NoiseReduceExt>
<PlateBright>
    <!--ro, opt, object-->
    <plateBrightEnabled>
        <!--ro, req, bool-->true
    </plateBrightEnabled>
    <plateBrightSensitivity>
        <!--ro, opt, int, range:[1,100]-->1
    </plateBrightSensitivity>
</PlateBright>
<ImageMultishut>
    <!--ro, opt, object-->
    <doubleShutEnable>
        <!--ro, req, bool-->true
    </doubleShutEnable>
    <codecType>
        <!--ro, req, enum, subType:string-->all_ia
    </codecType>
</ImageMultishut>
<JPEGParam>
    <!--ro, opt, object-->
    <JPEGSize>
        <!--ro, opt, int, range:[64,8196], unit:KB-->64
    </JPEGSize>
    <mergeJPEGSize>
        <!--ro, opt, int, range:[64,8196], unit:KB-->64
    </mergeJPEGSize>
</JPEGParam>
<ImageChannel>

```

12.2.5.20 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>
  <DSS>
    <!--ro, opt, object-->
    <enabled opt="true,false">
      <!--ro, req, bool, attr:opt{req, string}-->true
    </enabled>
    <mode opt="frame,level">
      <!--ro, opt, enum, mode, subType:string, attr:opt{req, string}, desc:WDR mode: "open,close,auto"-->frame
    </mode>
    <slowShutterLevel min="1" max="6">
      <!--ro, req, int, range:[1,6], attr:min{req, int},max{req, int}-->1
    </slowShutterLevel>
  </DSS>
  <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,HZ0618P-4MPSG,MZ1050P-12MP">
      <!--ro, req, enum, subType:string, attr:opt{req, string}-->auto
    </ExposureType>
    <pIris>
      <!--ro, opt, object, iris-->
      <pIrisType opt="AUTO,MANUAL">
        <!--ro, opt, enum, iris type, subType:string, attr:opt{req, string}-->AUTO
      </pIrisType>
      <IrisLevel min="0" max="10">
        <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
      </IrisLevel>
    </pIris>
  </Exposure>
  <Sharpness>
    <!--ro, opt, object-->
    <SharpnessLevel min="0" max="100">
      <!--ro, req, int, attr:min{req, int},max{req, int}-->1
    </SharpnessLevel>
  </Sharpness>
  <gammaCorrection>
    <!--ro, opt, object-->
    <gammaCorrectionEnabled opt="auto,manual">
      <!--ro, opt, bool, attr:opt{req, string}-->true
    </gammaCorrectionEnabled>
    <gammaCorrectionLevel min="0" max="10">
      <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
    </gammaCorrectionLevel>
  </gammaCorrection>
  <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, dep:and,{$.Color.nightMode,eq,true}, attr:min{req, int},max{req, int}-->33
    </saturationLevel>
    <nightMode opt="true,false">
      <!--ro, opt, bool, attr:opt{req, string}-->true
    </nightMode>
  </Color>
</ImageChannel>

```

12.2.5.21 Get the image mode parameters of all channels

Request URL

GET /ISAPI/Image/channels/imageModes

Query Parameter

None

Request Message

None

Response Message

```

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

<ImageModeList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, image mode configurations, subType:object, attr:version{req, string, protocolVersion}-->
  <ImageMode>
    <!--ro, opt, object, image mode-->
    <type>
      <!--ro, req, enum, type, subType:string, desc:"standard", "indoor", "outdoor", "dimLight", "bright", "soft", "vivid"-->standard
    </type>
    <recommendation>
      <!--ro, req, object, recommended configurations-->
      <brightnessLevel>
        <!--ro, opt, int, brightness, range:[0,100]-->1
      </brightnessLevel>
      <contrastLevel>
        <!--ro, opt, int, contrast, range:[0,100]-->1
      </contrastLevel>
      <sharpnessLevel>
        <!--ro, opt, int, sharpness, range:[0,100]-->1
      </sharpnessLevel>
      <saturationLevel>
        <!--ro, opt, int, saturation, range:[0,100]-->1
      </saturationLevel>
      <hueLevel>
        <!--ro, opt, int, hue, range:[0,100]-->1
      </hueLevel>
      <deNoiseLevel>
        <!--ro, opt, int, noise reduction, range:[0,100]-->1
      </deNoiseLevel>
    </recommendation>
  </ImageMode>
</ImageModeList>

```

12.2.6 Image Enhancement

12.2.6.1 Get the electronic defogging parameters of a specified channel

Request URL

GET /ISAPI/Image/channels/<channelID>/defog

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

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

<Defog xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool-->true
  </enabled>
</Defog>

```

12.2.6.2 Set the electronic defogging parameters for a specified channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/defog

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<Defog xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, defog configurations, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool, whether to enable-->true
  </enabled>
</Defog>

```

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
</ResponseStatus>

```

12.2.6.3 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>
  <dehazeIntensity>
    <!--ro, opt, int, range:[0,100], dep:and, ${.Dehaze.DehazeMode, eq, advancedMode}-->0
  </dehazeIntensity>
  <smoothIntensity>
    <!--ro, opt, int, range:[0,100], dep:and, ${.Dehaze.DehazeMode, eq, advancedMode}-->0
  </smoothIntensity>
</Dehaze>

```

12.2.6.4 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>
```

12.2.6.5 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>
  <GainWindow>
    <!--opt, object-->
    <RegionCoordinatesList>
      <!--opt, array, subType:object, range:[3,10]-->
      <RegionCoordinates>
        <!--opt, object, region coordinates, desc:the origin is the upper-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>
  </GainWindow>
  <scheduleEnabled>
    <!--opt, bool-->true
  </scheduleEnabled>
  <scheduleGainLevel>
    <!--opt, int, dep:and,{$.Gain.scheduleEnabled,eq,true}-->1
  </scheduleGainLevel>
  <Schedule>
    <!--opt, object, dep:and,{$.Gain.scheduleEnabled,eq,true}-->
    <startHour min="0" max="23">
      <!--opt, int, start time, range:[0,23], attr:min{req, int},max{req, int}-->1
    </startHour>
    <startMinute min="0" max="59">
      <!--opt, int, start time (minute), range:[0,59], attr:min{req, int},max{req, int}-->1
    </startMinute>
    <endHour min="0" max="23">
      <!--opt, int, end time, range:[0,23], attr:min{req, int},max{req, int}-->1
    </endHour>
    <endMinute min="0" max="59">
      <!--opt, int, end time (minute), range:[0,59], attr:min{req, int},max{req, int}-->1
    </endMinute>
  </Schedule>
  <redGainLevel>
    <!--opt, int, range:[0,100]-->1
  </redGainLevel>
</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-->0
  </subStatusCode>
</ResponseStatus>

```

12.2.6.6 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	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it wil be the value of the parameter in the current PUT URL.

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>
  <GainWindow>
    <!--ro, opt, object-->
    <RegionCoordinatesList>
      <!--ro, opt, array, subType:object, range:[3,10]-->
      <RegionCoordinates>
        <!--ro, opt, object, the origin is the upper-left corner of the screen, 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>
  </GainWindow>
  <scheduleEnabled>
    <!--ro, opt, bool-->true
  </scheduleEnabled>
  <scheduleGainLevel>
    <!--ro, opt, int, dep:and,{$.Gain.scheduleEnabled,eq,true}-->1
  </scheduleGainLevel>
  <Schedule>
    <!--ro, opt, object, time range, dep:and,{$.Gain.scheduleEnabled,eq,true}-->
    <startHour min="0" max="23">
      <!--ro, opt, int, start time, range:[0,23], attr:min{req, int},max{req, int}-->1
    </startHour>
    <startMinute min="0" max="59">
      <!--ro, opt, int, start time (minute), range:[0,59], attr:min{req, int},max{req, int}-->1
    </startMinute>
    <endHour min="0" max="23">
      <!--ro, opt, int, end time, range:[0,23], attr:min{req, int},max{req, int}-->1
    </endHour>
    <endMinute min="0" max="59">
      <!--ro, opt, int, end time (minute), range:[0,59], attr:min{req, int},max{req, int}-->1
    </endMinute>
  </Schedule>
  <redGainLevel>
    <!--ro, opt, int, range:[0,100]-->1
  </redGainLevel>
  <minExposureGainLevel>
    <!--ro, opt, int, range:[0,10]-->0
  </minExposureGainLevel>
  <maxExposureGainLevel>
    <!--ro, opt, int, range:[0,10]-->0
  </maxExposureGainLevel>
</Gain>
```

12.2.6.7 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}-->
  <node>
    <!--req, enum, mode, subType:string, desc:"close", "general", "advanced"-->close
  </node>
  <GeneralMode>
    <!--opt, object, mode-->
    <generalLevel>
      <!--req, int, level-->1
    </generalLevel>
  </GeneralMode>
  <AdvancedMode>
    <!--opt, object, advanced mode-->
    <FrameNoiseReduceLevel>
      <!--req, int-->1
    </FrameNoiseReduceLevel>
    <InterFrameNoiseReduceLevel>
      <!--req, int-->1
    </InterFrameNoiseReduceLevel>
    <SecondarySpatialNoiseReduceLevel>
      <!--opt, int, range:[0,100]-->1
    </SecondarySpatialNoiseReduceLevel>
    <SecondaryTemporalNoiseReduceLevel>
      <!--opt, int, range:[0,100]-->1
    </SecondaryTemporalNoiseReduceLevel>
  </AdvancedMode>
</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>
```

12.2.6.8 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>
  <AdvancedMode>
    <!--ro, opt, object, advanced mode-->
    <FrameNoiseReduceLevel>
      <!--ro, req, int-->1
    </FrameNoiseReduceLevel>
    <InterFrameNoiseReduceLevel>
      <!--ro, req, int-->1
    </InterFrameNoiseReduceLevel>
    <SecondarySpatialNoiseReduceLevel>
      <!--ro, opt, int, range:[0,100]-->1
    </SecondarySpatialNoiseReduceLevel>
    <SecondaryTemporalNoiseReduceLevel>
      <!--ro, opt, int, range:[0,100]-->1
    </SecondaryTemporalNoiseReduceLevel>
  </AdvancedMode>
</NoiseReduce>

```

12.2.6.9 Set the 2D noise reduction parameters of a specific channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/NoiseReduce2D

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<NoiseReduce2D xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 2D noise reduction parameters, attr:version{req, string, protocolVersion}-->
  <noiseReduce2DEnable>
    <!--req, bool, whether to enable 2D noise reduction-->true
  </noiseReduce2DEnable>
  <noiseReduce2DLevel min="0" max="100">
    <!--opt, int, 2D noise reduction level, attr:min{req, int},max{req, int}-->1
  </noiseReduce2DLevel>
</NoiseReduce2D>

```

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>

```

12.2.6.10 Get 2D noise reduction parameters of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/NoiseReduce2D?parameterType=<parameterType>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it will be the value of the parameter in the current PUT URL.

Request Message

None

Response Message

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

<NoiseReduce2D xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <noiseReduce2DEnable>
    <!--ro, req, bool-->true
  </noiseReduce2DEnable>
  <noiseReduce2DLevel min="0" max="255">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
  </noiseReduce2DLevel>
</NoiseReduce2D>
```

12.2.6.11 Set the 3D noise reduction parameters of a specific channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/noiseReduceExt

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

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

<NoiseReduceExt 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, general mode-->
    <generalLevel>
      <!--opt, int, general Level, range:[0,100]-->1
    </generalLevel>
  </GeneralMode>
  <AdvancedMode>
    <!--opt, object, advanced mode-->
    <spatialLevel>
      <!--opt, int, spatial Level, range:[0,100]-->1
    </spatialLevel>
    <temporalLevel>
      <!--opt, int, level, range:[0,100]-->1
    </temporalLevel>
  </AdvancedMode>
</NoiseReduceExt>
```

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>

```

12.2.6.12 Get 3D noise reduction parameters of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/noiseReduceExt?parameterType=<parameterType>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
parameterType	string	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it wil be the value of the parameter in the current PUT URL.

Request Message

None

Response Message

```

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

<NoiseReduceExt 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, general mode-->
    <generalLevel>
      <!--ro, opt, int, general level, range:[0,100]-->1
    </generalLevel>
  </GeneralMode>
  <AdvancedMode>
    <!--ro, opt, object, advanced mode-->
    <spatialLevel>
      <!--ro, opt, int, spatial level, range:[0,100]-->1
    </spatialLevel>
    <temporalLevel>
      <!--ro, opt, int, temporal level, range:[0,100]-->1
    </temporalLevel>
  </AdvancedMode>
  <trailingNoiseBalance>
    <!--ro, opt, object-->
    <level>
      <!--ro, opt, int, range:[0,100]-->1
    </level>
  </trailingNoiseBalance>
</NoiseReduceExt>

```

12.2.7 Image Light Supplement

12.2.7.1 Set the day/night mode switching parameters for a specified channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/dayNightGate

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

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

<DayNightGate xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, day/night mode switching parameters, attr:version{req, string, protocolVersion}-->
  <brightnessThreld>
    <!--opt, int, brightness-->1
  </brightnessThreld>
  <nightModeCtrl>
    <!--opt, object, day/night mode control-->
    <nightModeCtrlMode>
      <!--opt, enum, day/night switching mode, subType:string, desc:"manual" (manual switch), "time" (timed switch), "auto" (auto switch)-->manual
    </nightModeCtrlMode>
    <manualMode>
      <!--opt, object, manual switch-->
      <manualModeVal>
        <!--opt, enum, manual switching mode, subType:string, desc:"day", "night"-->day
      </manualModeVal>
    </manualMode>
    <timeMode>
      <!--opt, object, timed switch-->
      <switchList>
        <!--opt, array, subType:object-->
        <switch>
          <!--opt, object-->
          <nightModeTmeId>
            <!--opt, int-->1
          </nightModeTmeId>
          <nightModeStartHour>
            <!--opt, int, range:[0,23]-->1
          </nightModeStartHour>
          <nightModeStartMinute>
            <!--opt, int, range:[0,59]-->1
          </nightModeStartMinute>
          <nightModeEndHour>
            <!--opt, int, range:[0,23]-->1
          </nightModeEndHour>
          <nightModeEndMinute>
            <!--opt, int, range:[0,59]-->1
          </nightModeEndMinute>
        </switch>
      </switchList>
    </timeMode>
    <autoMode>
      <!--opt, object, auto switch-->
      <autoSwtichThresholdVal>
        <!--opt, int, threshold for switching, range:[1,100]-->1
      </autoSwtichThresholdVal>
    </autoMode>
  </nightModeCtrl>
</DayNightGate>
```

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>
```

12.2.7.2 Get the day/night switch parameters of the specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/dayNightGate

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

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

<DayNightGate xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, day/night switch, attr:version{req, string, protocolVersion}-->
  <brightnessThreld>
    <!--ro, opt, int, brightness-->1
  </brightnessThreld>
  <nightModeCtrl>
    <!--ro, opt, object, day/night mode switching control-->
    <nightModeCtrlMode>
      <!--ro, opt, enum, day/night switch mode, subType:string, desc:"manual" (manual switch mode), "time" (timed switch mode), "auto" (automatic switch mode)-->manual
    </nightModeCtrlMode>
    <manualMode>
      <!--ro, opt, object, manual switch-->
      <manualModeVal>
        <!--ro, opt, enum, "day", "night", subType:string, desc:"day", "night"-->day
      </manualModeVal>
    </manualMode>
    <timeMode>
      <!--ro, opt, object, timed switch-->
      <scheduleType>
        <!--ro, opt, enum, subType:string-->day
      </scheduleType>
      <switchList>
        <!--ro, opt, array, subType:object-->
        <switch>
          <!--ro, opt, object, timed switch settings-->
          <nightModeMeId>
            <!--ro, opt, int, night mode ID-->1
          </nightModeMeId>
          <nightModeStartHour>
            <!--ro, opt, int, night mode starting hour, range:[0,23]-->1
          </nightModeStartHour>
          <nightModeStartMinute>
            <!--ro, opt, int, night mode starting minute, range:[0,59]-->1
          </nightModeStartMinute>
          <nightModeEndHour>
            <!--ro, opt, int, night mode ending hour, range:[0,23]-->1
          </nightModeEndHour>
          <nightModeEndMinute>
            <!--ro, opt, int, night mode ending minute, range:[0,59]-->1
          </nightModeEndMinute>
        </switch>
      </switchList>
      <timeScheduleList>
        <!--ro, opt, array, subType:object, range:[0,2]-->
        <timeSchedule>
          <!--ro, opt, object-->
          <triggerData>
            <!--ro, opt, object-->
            <startTime>
              <!--ro, opt, object-->
              <monthOfYear>
                <!--ro, opt, int, range:[1,12]-->0
              </monthOfYear>
            </startTime>
            <endTime>
              <!--ro, opt, object-->
              <monthOfYear>
                <!--ro, opt, int, range:[1,12]-->0
              </monthOfYear>
            </endTime>
          </triggerData>
          <timeSwitchList>
            <!--ro, opt, array, subType:object, range:[0,1]-->
            <timeSwitch>
              <!--ro, opt, object-->
              <startHour>
                <!--ro, opt, int, range:[0,23]-->1
              </startHour>
              <startMinute>
                <!--ro, opt, int, range:[0,59]-->1
              </startMinute>
            </timeSwitch>
          </timeSwitchList>
        </timeSchedule>
      </timeScheduleList>
    </timeMode>
  </nightModeCtrl>
</DayNightGate>
```

```

</startMinute>
<endHour>
    <!--ro, opt, int, range:[0,23]-->1
</endHour>
<endMinute>
    <!--ro, opt, int, range:[0,59]-->1
</endMinute>
</timeSwitch>
</timeSwitchList>
</timeSchedule>
</timeScheduleList>
<dayNightStatus>
    <!--ro, opt, enum, subType:string-->day
</dayNightStatus>
</timeMode>
<autoMode>
    <!--ro, opt, object, automatic switch-->
<autoSwitchThresholdVal>
    <!--ro, opt, int, threshold for automatic switch, range:[0,100]-->1
</autoSwitchThresholdVal>
<autoNightSwitchThresholdVal>
    <!--ro, opt, int, range:[0,100]-->1
</autoNightSwitchThresholdVal>
<autoDayToNightDelayTime>
    <!--ro, opt, int, range:[0,120]-->1
</autoDayToNightDelayTime>
<autoNightToDayDelayTime>
    <!--ro, opt, int, range:[0,120]-->1
</autoNightToDayDelayTime>
<ICRChange>
    <!--ro, opt, bool-->false
</ICRChange>
<lightChange>
    <!--ro, opt, bool-->false
</lightChange>
<WDRChange>
    <!--ro, opt, bool-->false
</WDRChange>
<dayNightStatus>
    <!--ro, opt, enum, subType:string-->day
</dayNightStatus>
</autoMode>
</nightModeCtrl>
</DayNightGate>

```

12.2.7.3 Set the IR-cut filter parameters for cameras of a specific channel

Request URL

PUT /ISAPI/Image/channels/<channelID>/icr

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<ImageIcrE xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, object, ICR configuration parameters, attr:version{req, string, protocolVersion}-->
    <ICRCtrl>
        <!--opt, object, IR-cut filter-->
        <ICRctrlMode>
            <!--req, enum, "close", "manual"-manually switch,"time"-switch by schedule,"auto"-automatically switch, subType:string, desc:"close", "manual"-manually switch, "time"-switch by schedule, "auto"-automatically switch-->close
        </ICRctrlMode>
        <ManualMode>
            <!--opt, object, manually switch mode-->
            <ManualPresetVal>
                <!--req, enum, manually switch, subType:string, desc:"day", "night"-->day
            </ManualPresetVal>
        </ManualMode>
        <TimeMode>
            <!--opt, object, switch by schedule-->
            <SwitchList>
                <!--opt, array, schedule list for switching, subType:object, desc:up to two time period control can be supported. For example: time period is from 6:00 to 18:00, the preset ID is 1-->
                <Preset>
                    <!--opt, object, preset-->
                    <PresetId>
                        <!--req, int, preset ID-->1
                    </PresetId>
                    <PresetVal>
                        <!--opt, enum, "day","night", subType:string, desc:"day", "night"-->day
                    </PresetVal>
                </Preset>
            </SwitchList>
        </TimeMode>
    </ICRCtrl>
</ImageIcrE>

```

```

        </PresetVal>
    </Preset>
    <TimeSwitch>
        <!--opt, array, time period, subType:object-->
        <timeId>
            <!--req, int, time period ID, range:[0,2]-->1
        </timeId>
        <startHour>
            <!--opt, int, start hour, range:[0,23]-->1
        </startHour>
        <startMinute>
            <!--opt, int, start minute, range:[0,59]-->1
        </startMinute>
        <endHour>
            <!--opt, int, end hour, range:[0,23]-->1
        </endHour>
        <endMinute>
            <!--opt, int, end minute, range:[0,59]-->1
        </endMinute>
    </TimeSwitch>
</SwitchList>
<TimeMode>
<AutoMode>
    <!--opt, object, automatically switch according to the Light change-->
    <DayNightFilterTh>
        <!--req, int, day/night auto switch sensitivity, range:[0,100]-->1
    </DayNightFilterTh>
    <ICRAutoSwitch min="0" max="100">
        <!--opt, int, ICR auto-switch and threshold, range:[0,100], attr:min{req, int},max{req, int}-->1
    </ICRAutoSwitch>
</AutoMode>
<ICRCtrl>
<ECtrl>
    <!--opt, object, E device-->
    <ECtrlMode>
        <!--req, enum, "close", "manual"-manually switch,"time"-switch by schedule,"auto"-automatically switch, subType:string, desc:"close", "manual" (manually switch), "time" (switch by schedule), "auto" (automatically switch)-->close
    </ECtrlMode>
    <ManualMode>
        <!--opt, object, manually switch-->
        <ManualPresetVal>
            <!--req, int, manually switch: "day", "night", range:[0,100]-->1
        </ManualPresetVal>
    </ManualMode>
    <TimeMode>
        <!--opt, object, switch by schedule-->
        <SwitchList>
            <!--opt, array, schedule List, subType:object, desc:up to two time period control can be supported. For example: time period is from 6:00 to 18:00, the preset ID is 1-->
            <Preset>
                <!--opt, object, preset-->
                <PresetId>
                    <!--req, int, preset ID,which ranges from 0 to 1-->1
                </PresetId>
                <PresetVal>
                    <!--opt, enum, "day", "night", subType:string, desc:"day", "night"-->day
                </PresetVal>
            </Preset>
            <TimeSwitch>
                <!--opt, array, time period, subType:object-->
                <timeId>
                    <!--req, int, time period ID, range:[0,2]-->1
                </timeId>
                <startHour>
                    <!--opt, int, start hour, range:[0,23]-->1
                </startHour>
                <startMinute>
                    <!--opt, int, start minute, range:[0,59]-->1
                </startMinute>
                <endHour>
                    <!--opt, int, end hour, range:[0,23]-->1
                </endHour>
                <endMinute>
                    <!--opt, int, end minute, range:[0,59]-->1
                </endMinute>
            </TimeSwitch>
        </SwitchList>
    </TimeMode>
    <AutoMode>
        <!--opt, object, automatically switch according to the Light change-->
        <DayNightFilterTh>
            <!--req, int, day/night auto switch sensitivity, range:[0,100]-->1
        </DayNightFilterTh>
        <ExceptionCatchTh>
            <!--req, int, exception detection sensitivity, range:[0,100]-->1
        </ExceptionCatchTh>
    </AutoMode>
</ECtrl>
<ImageIcrE>

```

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>

```

12.2.7.4 Get the IR-cut filter parameters for cameras of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/icr

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

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

<ImageIcrE xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, ICR parameters configuration, attr:version{req, string, protocolVersion}-->
  <ICRCtrl>
    <!--ro, opt, object, IR-cut filter-->
    <ICRCtrlMode>
      <!--ro, req, enum, "close", "manual"-manually switch, "time"-switch by schedule, "auto"-automatically switch, subType:string, desc:"close", "manual"-manually switch, "time"-switch by schedule, "auto"-automatically switch-->close
      <ICRCtrlMode>
        <ManualMode>
          <!--ro, opt, object, manually switch, dep:and,{$.ImageIcrE.ICRCtrl.ICRCtrlMode,eq,manual}-->
          <ManualPresetVal>
            <!--ro, req, enum, manually switch, subType:string, desc:"day", "night"-->day
          </ManualPresetVal>
        </ManualMode>
        <TimeMode>
          <!--ro, opt, object, switch by schedule, dep:and,{$.ImageIcrE.ICRCtrl.ICRCtrlMode,eq,time}-->
          <scheduleType>
            <!--ro, opt, enum, subType:string-->day
          </scheduleType>
          <SwitchList>
            <!--ro, opt, array, schedule list for switching, subType:object, range:[0,2], dep:and,{$.ImageIcrE.ICRCtrl.TimeMode.scheduleType,eq,day}, desc:up to two time period control can be supported. For example: time period is from 6:00 to 18:00, the preset ID is 1-->
            <Preset>
              <!--ro, opt, object, preset-->
              <PresetId>
                <!--ro, req, int, preset ID-->1
              </PresetId>
              <PresetVal>
                <!--ro, opt, enum, "day", "night", subType:string, desc:"day", "night"-->day
              </PresetVal>
            </Preset>
            <TimeSwitch>
              <!--ro, opt, object-->
              <timeId>
                <!--ro, req, int, range:[0,2]-->1
              </timeId>
              <startHour>
                <!--ro, opt, int, range:[0,23]-->1
              </startHour>
              <startMinute>
                <!--ro, opt, int, range:[0,59]-->1
              </startMinute>
              <endHour>
                <!--ro, opt, int, range:[0,23]-->1
              </endHour>
              ...
            </TimeSwitch>
          </SwitchList>
        </TimeMode>
      </ICRCtrlMode>
    </ICRCtrl>
  </ImageIcrE>

```

```

<endMinute>
  <!--ro, opt, int, range:[0,59]-->1
</endMinute>
<PresetVal>
  <!--ro, opt, enum, subType:string-->day
</PresetVal>
</TimeSwitch>
</SwitchList>
<dayNightStatus>
  <!--ro, opt, enum, subType:string-->day
</dayNightStatus>
</timeScheduleList>
<!--ro, opt, array, subType:object, range:[0,2], dep:and,{$.ImageIcrE.ICRCtrl.TimeMode.scheduleType,eq,range}-->
<timeSchedule>
  <!--ro, opt, object-->
<triggerData>
  <!--ro, opt, object-->
<startTime>
  <!--ro, opt, object-->
<monthOfYear>
  <!--ro, opt, int, range:[1,12]-->1
</monthOfYear>
</startTime>
<endTime>
  <!--ro, opt, object-->
<monthOfYear>
  <!--ro, opt, int, range:[1,12]-->1
</monthOfYear>
</endTime>
</triggerData>
</timeSwitchList>
<!--ro, opt, array, subType:object, range:[0,2]-->
<timeSwitch>
  <!--ro, opt, object-->
<timeID>
  <!--ro, req, int, range:[0,1]-->1
</timeID>
<startHour>
  <!--ro, opt, int, range:[0,23]-->1
</startHour>
<startMinute>
  <!--ro, opt, int, range:[0,59]-->1
</startMinute>
<endHour>
  <!--ro, opt, int, range:[0,23]-->1
</endHour>
<endMinute>
  <!--ro, opt, int, range:[0,59]-->1
</endMinute>
<dayNightVal>
  <!--ro, opt, enum, subType:string-->day
</dayNightVal>
</timeSwitch>
</timeSwitchList>
</timeSchedule>
</timeScheduleList>
</TimeMode>
<AutoMode>
  <!--ro, opt, object, automatically switch according to the Light change, dep:and,{$.ImageIcrE.ICRCtrl.ICRCtrlMode,eq,auto}-->
<DayNightFilterTh>
  <!--ro, req, int, day/night auto switch sensitivity, range:[0,100]-->1
</DayNightFilterTh>
<ICRAutoSwitch min="0" max="100">
  <!--ro, opt, int, ICR auto-switch and threshold, range:[0,100], attr:min{req, int},max{req, int}-->1
</ICRAutoSwitch>
<dayNightStatus>
  <!--ro, opt, enum, subType:string-->day
</dayNightStatus>
</AutoMode>
</ICRCtrl>
<ECtrl>
  <!--ro, opt, object, -E device,halo inhibition filter,which can be used to inhibit the halo-->
<ECtrlMode>
  <!--ro, req, enum, "close","manual"-manually switch,"time"-switch by schedule,"auto"-automatically switch, subType:string, desc:"close", "manual"-manually switch, "time"-switch by schedule, "auto"-automatically switch-->close
</ECtrlMode>
<ManualMode>
  <!--ro, opt, object, manually switch, dep:and,{$.ImageIcrE.ECtrl.ECtrlMode,eq,manual}-->
<ManualPresetVal>
  <!--ro, req, int, manually switch, range:[0,100]-->1
</ManualPresetVal>
</ManualMode>
<TimeMode>
  <!--ro, opt, object, switch by schedule, dep:and,{$.ImageIcrE.ECtrl.ECtrlMode,eq,time}-->
<SwitchList>
  <!--ro, opt, array, schedule list for switching, subType:object, range:[0,2], desc:up to two time period control can be supported. For example: time period is from 6:00 to 18:00, the preset ID is 1-->
<Preset>
  <!--ro, opt, object, preset-->
<PresetId>
  <!--ro, req, int, preset ID-->1
</PresetId>
<PresetVal>
  <!--ro, opt, enum, "day", "night", subType:string, desc:"day", "night"-->day

```

```

        </PresetVal>
    </Preset>
    <TimeSwitch>
        <!--ro, opt, object-->
        <timeId>
            <!--ro, req, int, range:[0,2]-->1
        </timeId>
        <startHour>
            <!--ro, opt, int, range:[0,23]-->1
        </startHour>
        <startMinute>
            <!--ro, opt, int, range:[0,59]-->1
        </startMinute>
        <endHour>
            <!--ro, opt, int, range:[0,23]-->1
        </endHour>
        <endMinute>
            <!--ro, opt, int, range:[0,59]-->1
        </endMinute>
        <PresetVal>
            <!--ro, opt, enum, subType:string-->day
        </PresetVal>
    </TimeSwitch>
    <SwitchList>
</TimeMode>
<AutoMode>
    <!--ro, opt, object, automatically switch according to the Light change, dep:and, ${.ImageIcrE.ECtrl.EctrlMode.eq,auto}-->
    <DayNightFilterTh>
        <!--ro, req, int, day/night auto switch sensitivity, range:[0,100]-->1
    </DayNightFilterTh>
    <ExceptionCatchTh>
        <!--ro, req, int, exception detection sensitivity, range:[0,100]-->1
    </ExceptionCatchTh>
</AutoMode>
</ECtrl>
</ImageIcrE>

```

12.2.7.5 Get the IR-cut filter capability for cameras of a specific channel

Request URL

GET /ISAPI/Image/channels/<channelID>/icr/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	

Request Message

None

Response Message

```

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

<ImageIcrE xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, ICR parameters configuration, attr:version{req, string, protocolVersion}-->
    <ICRCtrl>
        <!--ro, opt, object, IR-cut filter-->
        <ICRCtrlMode opt="close,manual,time,auto">
            <!--ro, req, string, "close","manual"-manually switch,"time"-switch by schedule,"auto"-automatically switch, attr:opt{req, string}-->test
        </ICRCtrlMode>
        <ManualMode>
            <!--ro, opt, object, manually switch-->
            <ManualPresetVal opt="day,night">
                <!--ro, req, string, manually switch, attr:opt{req, string}-->test
            </ManualPresetVal>
        </ManualMode>
        <TimeMode>
            <!--ro, opt, object, switch by schedule-->
            <scheduleType opt="day,range">
                <!--ro, opt, string, attr:opt{req, string}-->day
            </scheduleType>
            <SwitchList size="2">
                <!--ro, opt, array, subType:object, dep:and, ${.ImageIcrE.ICRCtrl.TimeMode.scheduleType,eq,day}, attr:size{req, int}-->
                <Preset>
                    <!--ro, req, object-->
                    <PresetId min="0" max="1">
                        <!--ro, req, int, attr:min{req, int},max{req, int}-->1
                    </PresetId>
                    <PresetVal opt="day,night">
                        <!--ro, opt, enum, subType:string, attr:opt{req, string}-->day
                    </PresetVal>
                </Preset>
            </TimeMode>

```

```

<!-->
<!--ro, opt, object-->
<timeId min="0" max="1">
  <!--ro, req, int, attr:min{req, int},max{req, int}-->1
</timeId>
<startHour min="0" max="23">
  <!--ro, opt, int, range:[0,23], attr:min{req, int},max{req, int}-->1
</startHour>
<startMinute min="0" max="59">
  <!--ro, opt, int, range:[0,59], attr:min{req, int},max{req, int}-->1
</startMinute>
<endHour min="0" max="23">
  <!--ro, opt, int, range:[0,23], attr:min{req, int},max{req, int}-->1
</endHour>
<endMinute min="0" max="59">
  <!--ro, opt, int, range:[0,59], attr:min{req, int},max{req, int}-->1
</endMinute>
<PresetVal opt="day,night">
  <!--ro, opt, enum, subType:string, attr:opt{req, string}-->day
</PresetVal>
</TimeSwitch>
</SwitchList>
<timeScheduleList size="2">
  <!--ro, opt, object, dep:and,{$.ImageIcrE.ICRCtrl.TimeMode.scheduleType,eq,range}, attr:size{req, int}-->
<timeSchedule>
  <!--ro, opt, object-->
  <triggerData>
    <!--ro, opt, object-->
    <startTime>
      <!--ro, opt, object-->
      <monthOfYear min="1" max="12">
        <!--ro, opt, int, range:[1,12], attr:min{req, int},max{req, int}-->0
      </monthOfYear>
    </startTime>
    <endTime>
      <!--ro, opt, object-->
      <monthOfYear min="1" max="12">
        <!--ro, opt, int, range:[1,12], attr:min{req, int},max{req, int}-->0
      </monthOfYear>
    </endTime>
  </triggerData>
  <timeSwitchList size="2">
    <!--ro, opt, object, attr:size{req, int}-->
    <timeSwitch>
      <!--ro, opt, object-->
      <timeID min="0" max="1">
        <!--ro, req, int, range:[0,1], attr:min{req, int},max{req, int}-->1
      </timeID>
      <startHour min="0" max="23">
        <!--ro, opt, int, range:[0,23], attr:min{req, int},max{req, int}-->1
      </startHour>
      <startMinute min="0" max="59">
        <!--ro, opt, int, range:[0,59], attr:min{req, int},max{req, int}-->1
      </startMinute>
      <endHour min="0" max="23">
        <!--ro, opt, int, range:[0,23], attr:min{req, int},max{req, int}-->1
      </endHour>
      <endMinute min="0" max="59">
        <!--ro, opt, int, range:[0,59], attr:min{req, int},max{req, int}-->1
      </endMinute>
      <dayNightVal opt="day,night">
        <!--ro, opt, enum, subType:string, attr:opt{req, string}-->day
      </dayNightVal>
    </timeSwitch>
  </timeSwitchList>
</timeSchedule>
</timeScheduleList>
</TimeMode>
<AutoMode>
  <!--ro, opt, object, automatically switch according to the light change-->
  <DayNightFilterTh min="0" max="100">
    <!--ro, req, int, day/night auto switch sensitivity, range:[0,100], attr:min{req, int},max{req, int}-->1
  </DayNightFilterTh>
  <ICRAutoSwitch min="0" max="100">
    <!--ro, opt, int, ICR auto-switch and threshold, range:[0,100], attr:min{req, int},max{req, int}-->1
  </ICRAutoSwitch>
</AutoMode>
</ICRCtrl>
<ECtrl1>
  <!--ro, opt, object, -E device,halo inhibition filter,which can be used to inhibit the halo-->
  <ECtrl1Mode opt="close,manual,time,auto">
    <!--ro, req, string, "close","manual"-manually switch,"time"-switch by schedule,"auto"-automatically switch, attr:opt{req, string}-->test
  </ECtrl1Mode>
  <ManualMode>
    <!--ro, opt, object, manually switch-->
    <ManualPresetVal min="0" max="100">
      <!--ro, req, int, manually switch, range:[0,100], attr:min{req, int},max{req, int}-->1
    </ManualPresetVal>
  </ManualMode>
  <TimeMode>
    <!--ro, opt, object, switch by schedule-->
    <SwitchList size="2">
      <!--ro, opt, array, subType:object, attr:size{req, int}-->
      <Preset>

```

```

<!--ro, req, object-->
<PresetId min="0" max="1">
    <!--ro, req, int, attr:min{req, int},max{req, int}-->1
</PresetId>
<PresetVal opt="day,night">
    <!--ro, opt, enum, subType:string, attr:opt{req, string}-->day
</PresetVal>
</Preset>
<TimeSwitch>
    <!--ro, opt, object-->
    <timeId min="0" max="1">
        <!--ro, req, int, attr:min{req, int},max{req, int}-->1
    </timeId>
    <startHour min="0" max="23">
        <!--ro, opt, int, range:[0,23], attr:min{req, int},max{req, int}-->1
    </startHour>
    <startMinute min="0" max="59">
        <!--ro, opt, int, range:[0,59], attr:min{req, int},max{req, int}-->1
    </startMinute>
    <endHour min="0" max="23">
        <!--ro, opt, int, range:[0,23], attr:min{req, int},max{req, int}-->1
    </endHour>
    <endMinute min="0" max="59">
        <!--ro, opt, int, range:[0,59], attr:min{req, int},max{req, int}-->1
    </endMinute>
    <PresetVal opt="day,night">
        <!--ro, opt, enum, subType:string, attr:opt{req, string}-->day
    </PresetVal>
</TimeSwitch>
</SwitchList>
</TimeMode>
<AutoMode>
    <!--ro, opt, object, automatically switch according to the Light change-->
    <DayNightFilterTh min="0" max="100">
        <!--ro, req, int, day/night auto switch sensitivity, range:[0,100], attr:min{req, int},max{req, int}-->1
    </DayNightFilterTh>
    <ExceptionCatchTh min="0" max="100">
        <!--ro, req, int, "0-100", range:[0,100], attr:min{req, int},max{req, int}-->1
    </ExceptionCatchTh>
    </AutoMode>
</ECtrl>
</ImageICrE>

```

12.2.7.6 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>
      <!--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>

```

12.2.7.7 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,riverSeaProtection,perimeterProtection,forestFirePrevention,foggy,faceFirst,plateFirst,moveFirst,autoDR,sportSnap,dustMoisture,rampWay"
    <!--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>
    <isSupportRecommendation>
      <!--ro, opt, bool-->true
    </isSupportRecommendation>
    <modelNameList size="2">
      <!--ro, opt, object, attr:size{req, int}-->
      <modelNameItem>
        <!--ro, opt, object-->
        <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,riverSeaProtection,perimeterProtection,forestFirePrevention,foggy,faceFirst,plateFirst,moveFirst,autoDR"
          <!--ro, req, enum, subType:string, attr:opt{req, string}-->indoor
          </mode>
          <modelName>
            <!--ro, req, string-->test
          </modelName>
        </modelNameItem>
      </modelNameList>
    </MountingScenario>

```

12.2.7.8 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	Parameter type. When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured, it will be the value of the parameter configured on the device.

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}-->
  <node>
    <!--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
    </node>
    <modelNameList>
      <!--ro, opt, array, subType:object-->
      <modelNameItem>
        <!--ro, opt, object-->
        <node>
          <!--ro, req, enum, subType:string-->indoor
        </node>
        <modelName>
          <!--ro, req, string-->test
        </modelName>
      </modelNameItem>
    </modelNameList>
  </MountingScenario>
```

12.2.8 Manual Capture

12.2.8.1 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>&livingBodyDetect=<livingBodyDetect>

Query Parameter

Parameter Name	Parameter Type	Description
trackStreamID	string	Channel No. * 100 + stream type (-1: main stream, -2: sub-stream).
snapShotImageType	string	Image formats: "bayer" (image not processed by ISP, which can be used to test whether the image clarity of the factory equipment meets the standard), "JPEGWITHRULE" (this type of image is specially for HIKMICRO devices, indicating a JPEG image captured with temperature rule information overlaid), "JPEGWITHOUTRULE" (when this type is used by HIKMICRO devices, it indicates a JPEG image captured without temperature rule information overlaid), "JPEG" (when this type is used by HIKMICRO devices, whether the captured JPEG image includes rule information depends on the enabling status of overlay temperature information on stream function. If it is enabled, the image returned will have rule information overlaid; otherwise, it will not. If the HIKMICRO device calls the URL without using this parameter, the logic for the returned JPEG image is consistent with the above JPEG type, depending on the enabling status of overlay temperature information on stream function).
videoResolutionWidth	string	Resolution width, and if not configured, it is 704 by default (D1 resolution width).
videoResolutionHeight	string	Resolution height, and if not configured, it is 576 by default (D1 resolution height).
imageQuality	string	Image quality: "best", "better", "general"
x	string	x-coordinate of the specified area, the value ranges from 0.000 to 1.000. The origin is the upper-left corner of the screen
y	string	y-coordinate of the specified area, the value ranges from 0.000 to 1.000. The origin is the upper-left corner of the screen
height	string	Height of the specified area. The value ranges from 0.000 to 1.000. The origin is the upper-left corner of the screen.
width	string	Width of the specified area. The value ranges from 0.000 to 1.000. The origin is the upper-left corner of the screen.
maxSnapResolution	enum	Capture images using the maximum image capture resolution. For certain devices, such as low-power devices, the actual maximum image capture resolution differs from the video encoding resolution. During certain intelligent detections, images are captured at the maximum image capture resolution. If not configured, the default value is 1.
livingBodyDetect	enum	Whether to perform face anti-spoofing detection when capturing images.

Request Message

None

Response Message

Binary Data

12.2.9 Motion Detection

12.2.9.1 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>
        <inputIOPortID>
            <!--ro, opt, string-->test
        </inputIOPortID>
        <outputIOPortID>
            <!--ro, opt, string-->test
        </outputIOPortID>
        <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)-->
                </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>
        <HolidayBlockList>
            <!--ro, opt, array, holiday arming time, subType:object-->
            <TimeBlock>
                <!--ro, opt, object, arming time-->
                <TimeRange>
                    <!--ro, req, object-->
                    <beginTime>
                        <!--ro, req, time, start time-->10:00:00
                    </beginTime>
                    <endTime>
                        <!--ro, req, time, end time-->10:00:00
                    </endTime>
                </TimeRange>
            </TimeBlock>
        </HolidayBlockList>
    </Schedule>
</MotionDetectionScheduleList>
```

12.2.9.2 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_video01
    </id>
    <eventType>
      <!--opt, enum, event type, subType:string, desc:"VMD" (motion detection)-->VMD
    </eventType>
    <inputIOPortID>
      <!--opt, string-->test
    </inputIOPortID>
    <outputIOPortID>
      <!--opt, string-->test
    </outputIOPortID>
    <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>
          <!--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>
          <!--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>
    <HolidayBlockList>
      <!--opt, array, holiday arming time, subType:object-->
      <TimeBlock>
        <!--opt, object, arming time-->
        <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>
    </HolidayBlockList>
  </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-->
</ResponseStatus>

```

12.2.9.3 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_video1
  </id>
  <eventType>
    <!--ro, opt, string-->VMD
  </eventType>
  <inputIOPortID>
    <!--ro, opt, string-->test
  </inputIOPortID>
  <outputIOPortID>
    <!--ro, opt, string-->test
  </outputIOPortID>
  <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>
  <HolidayBlockList>
    <!--ro, opt, array, subType:object-->
    <TimeBlock>
      <!--ro, opt, object-->
      <TimeRange>
        <!--ro, req, object-->
        <beginTime>
          <!--ro, req, time-->10:00:00
        </beginTime>
        <endTime>
          <!--ro, req, time-->10:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </HolidayBlockList>
</Schedule>
```

12.2.9.4 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>
  <inputIOPortID>
    <!--opt, string-->test
  </inputIOPortID>
  <outputIOPortID>
    <!--opt, string-->test
  </outputIOPortID>
  <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>
  <HolidayBlockList>
    <!--opt, array, subType:object-->
    <TimeBlock>
      <!--opt, object-->
      <TimeRange>
        <!--req, object-->
        <beginTime>
          <!--req, time-->10:00:00
        </beginTime>
        <endTime>
          <!--req, time-->10:00:00
        </endTime>
      </TimeRange>
    </TimeBlock>
  </HolidayBlockList>
</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>

```

12.2.9.5 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)-->
    <normalizedScreenWidth>
      <!--ro, req, int, normalized width-->0
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--ro, req, int, normalized height-->0
    </normalizedScreenHeight>
  </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>
      <objectSize>
        <!--ro, opt, int, percentage, range:[0,100]-->0
      </objectSize>
      <dayObjectSize>
        <!--ro, opt, int, percentage (during day time), range:[0,100]-->0
      </dayObjectSize>
      <nightObjectSize>
        <!--ro, opt, int, percentage (during night), range:[0,100]-->0
      </nightObjectSize>
      <RegionCoordinatesList>
        <!--ro, req, array, rule region list, subType:object, range:[0,4], desc:rectangle-->
        <RegionCoordinates>

```

```

<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>
<enableWithMoving>
    <!--ro, opt, bool, whether to enable dynamic analysis-->true
</enableWithMoving>
</MotionDetectionExt>

```

12.2.9.6 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>
        <!--req, int, minimum target resolution-->0
    </minObjectSize>
    <maxObjectSize>
        <!--req, int, maximum target resolution-->0
    </maxObjectSize>
    <ROI>
        <!--opt, object, region of interest (ROI)-->
        <normalizedScreenWidth>
            <!--req, int, normalized width-->0
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!--req, int, normalized height-->0
        </normalizedScreenHeight>
    </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-->

```

```

<channel>
    <!--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>
<objectSize>
    <!--opt, int, percentage, range:[0,100]-->0
</objectSize>
<dayObjectSize>
    <!--opt, int, percentage (during day time), range:[0,100]-->0
</dayObjectSize>
<nightObjectSize>
    <!--opt, int, percentage (during night), range:[0,100]-->0
</nightObjectSize>
<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>
<enableWithMoving>
    <!--opt, bool, whether to enable dynamic analysis-->true
</enableWithMoving>
</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>
    <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>
    <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>

```

12.2.10 OSD Settings

12.2.10.1 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}-->
  <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>
  </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>
    </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>
    <displayWeek>
      <!--ro, opt, bool, whether to display week information-->true
    </displayWeek>
  </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>
  </channelNameOverlay>
  <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>
  <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>
<boundary>
<!--ro, opt, int, the minimum boundary, range:[0,2], dep:and,{$.VideoOverlay.alignment,eq,GB}, desc:unit: number of characters-->1
</boundary>
<upDownBoundary>
<!--ro, opt, int, range:[0,2]-->1
</upDownBoundary>
<PackingSpaceRecognitionOverlay>
<!--ro, opt, object, display of parking space detection result-->
<isParkEnabled>
<!--ro, opt, bool, whether to display the parking space status (occupied or free)-->true
</isParkEnabled>
<PackingSpaceRecognitionOverlay>
<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>
</VideoOverlay>

```

12.2.10.2 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-->test
    </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>
    <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)-->
    </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)-->
    </OverlayRegion>

```

```

        <!--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>
</TiltAngle>
<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>
<PackingSpaceRecognitionOverlay>
  <!--opt, object, display of parking space detection result-->
  <isParkedEnabled>
    <!--opt, bool, whether to display the parking space status (occupied or free)-->true
  </isParkedEnabled>
</PackingSpaceRecognitionOverlay>
<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 informaiton 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>
<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>
<positionY>
    <!--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
</positionY>
<positionX>
    <!--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
</positionX>
<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>
    <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:the detailed information of custom error returned by device applications, it is used for fast debugging-->badXmlFormat
    </description>
</ResponseStatus>

```

12.2.10.3 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}-->
    <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 min="0" max="64">
                <!--ro, req, string, attr:min{req, int},max{req, int}-->test
            </displayText>
            <isPersistentText>
                <!--ro, opt, bool-->true
            </isPersistentText>
        </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>
        <displayWeek opt="true,false">
            <!--ro, opt, bool, attr:opt{req, string}-->true
        </displayWeek>
    </DateTimeOverlay>
    <channelNameOverlay>
        <!--ro, opt, object-->
        <enabled opt="true,false">
            <!--ro, req, bool, attr:opt{req, string}-->true
        </enabled>
        <positionX>
            <!--ro, opt, int, range:[0,1920]-->1
        </positionX>
        <positionY>
            <!--ro, opt, int, range:[0,1920]-->1
        </positionY>
        <name min="0" max="128">
            <!--ro, opt, string, range:[0,128], attr:min{req, int},max{req, int}-->Camera 01
        </name>
    </channelNameOverlay>
    <fontSize opt="16*16,32*32,48*48,64*64,80*80,96*96,128*128,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(opt, string) -->outline
    </frontColorMode>
    <frontColor>
        <!--ro, opt, string, font color, desc:font color-->#000000
    </frontColor>

```

```

<alignment opt="customize,alignRight,alignLeft,GB,GB2025,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>
<upDownBoundary min="0" max="5">
  <!--ro, opt, int, attr:min{opt, int},max{opt, int}-->1
</upDownBoundary>
</VideoOverlay>

```

12.2.10.4 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	Channel No.

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}-->
  <id>
    <!--ro, opt, int, ID-->1
  </id>
  <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>
  <name>
    <!--ro, opt, string, channel name, range:[0,128]-->Camera 01
  </name>
  <OverlayRegion>
    <!--ro, opt, object, coordinates of overlaid string frame (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>

```

12.2.10.5 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}-->
  <id>
    <!--opt, int, ID-->1
  </id>
  <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>
  <name>
    <!--opt, string, channel name, range:[0,32]-->Camera 01
  </name>
  <OverlayRegion>
    <!--opt, object-->
    <width>
      <!--req, int, width, range:[0,1000]-->0
    </width>
    <height>
      <!--req, int, height, range:[0,1000]-->0
    </height>
  </OverlayRegion>
</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>

```

12.2.10.6 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}-->
  <enabled>
    <!--req, bool, whether to enable date and time information overlay-->true
  </enabled>
  <positionX>
    <!--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>
    <!--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>
  <dateStyle>
    <!--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-YYYY/MM/DD", "CHR-MM/DD/YYYY", "CHR-DD/MM/YYYY"-->YYYY-MM-DD
  </dateStyle>
  <timeStyle>
    <!--opt, enum, time format, subType:string, desc:"12hour" (12-hour system), "24hour" (24-hour system)-->12hour
  </timeStyle>
  <displayWeek>
    <!--opt, bool, whether to display week information-->true
  </displayWeek>
  <displayDate>
    <!--opt, bool, whether to display date-->true
  </displayDate>
  <displayTime>
    <!--opt, bool, whether to display time-->true
  </displayTime>
  <OverlayRegion>
    <!--opt, object-->
    <width>
      <!--req, int, width, range:[0,1000]-->0
    </width>
    <height>
      <!--req, int, height, range:[0,1000]-->0
    </height>
  </OverlayRegion>
</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>

```

12.2.10.7 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	Channel No.

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}-->
  <enabled>
    <!--ro, req, bool, whether to enable date and time overlay-->true
  </enabled>
  <positionX>
    <!--ro, req, int, X-coordinate of the upper-left corner of the overlay area, 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 overlay area, range:[0,576], desc:the origin is the lower-left corner of the screen-->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"-->YYYY-MM-DD
  </dateStyle>
  <timeStyle>
    <!--ro, opt, enum, time format, subType:string, desc:time format-->12hour
  </timeStyle>
  <displayWeek>
    <!--ro, opt, bool, whether to display week information-->true
  </displayWeek>
  <displayDate>
    <!--ro, opt, bool-->true
  </displayDate>
  <displayTime>
    <!--ro, opt, bool-->true
  </displayTime>
  <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>
</DateTimeOverlay>

```

12.2.10.8 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>

```

12.2.10.9 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	Channel No.

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>

```

12.2.11 White Balance

12.2.11.1 Set the white balance parameters of a 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", "atuo1", "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>
      <!--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>

```

12.2.11.2 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	When parameterType is recommendation, the obtained value is recommended by device. If parameterType is not configured as an input parameter, it wil be the value of the parameter in the current PUT URL.

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", "atuo2", "manual", "indoor", "outdoor", "autotrace", "onece", "sodiumlight", "mercuryLight"-->auto
  </WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--ro, req, int, white balance, range:[0,100], desc:white balance-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--ro, req, int, white balance, range:[0,100], desc:white balance-->1
  </WhiteBalanceBlue>
  <WhiteBalanceLevel>
    <!--ro, opt, int, range:[0,100], dep:or, ${$.WhiteBalance.WhiteBalanceStyle, eq, auto}, ${$.WhiteBalance.WhiteBalanceStyle, eq, auto1}, ${$.WhiteBalance.WhiteBalanceStyle, eq, auto2}-->1
  </WhiteBalanceLevel>
  <customColorTemperatureLevel>
    <!--ro, opt, int, range:[1800,10000]-->1
  </customColorTemperatureLevel>
  <WhiteBalanceRedFineTuning>
    <!--ro, opt, int, range:[0,100]-->1
  </WhiteBalanceRedFineTuning>
  <WhiteBalanceBlueFineTuning>
    <!--ro, opt, int, range:[0,100]-->1
  </WhiteBalanceBlueFineTuning>
  <autoMode>
    <!--ro, opt, enum, subType:string, dep:and, ${$.WhiteBalance.WhiteBalanceStyle, eq, auto}-->video
  </autoMode>
  <manualMode>
    <!--ro, opt, enum, subType:string, dep:and, ${$.WhiteBalance.WhiteBalanceStyle, eq, manual}-->colorFineTuning
  </manualMode>
</WhiteBalance>

```

12.3 Video Management

12.3.1 Scheduled Recording

12.3.1.1 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.

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-033595353625}
  </TrackGUID>
  <Size>
    <!--ro, opt, int, video 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>
    <!--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), "fallDown" (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, 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>
  
```

```

<SrcUrlMethods>
    <!--ro, req, enum, methods supported by the source, subType:string, 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), "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>
<PreRecordTimeSeconds>
    <!--ro, opt, int, pre-record duration, unit:s, desc:from 0 to 10 seconds; default: 5 seconds-->5
</PreRecordTimeSeconds>
<PostRecordTimeSeconds>
    <!--ro, opt, int, post-record duration, unit:s, desc:from 0 to 20 seconds; default: 5 seconds-->5
</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.aaa.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>

```

```

<!--RecordTimeSeconds>
<!--ro, opt, int, pre-record, unit:s-->1
</PreRecordTimeSeconds>
<PostRecordTimeSeconds>
<!--ro, opt, int, post-record, unit:s-->1
</PostRecordTimeSeconds>
<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, string, schedule module type-->www.xxx.com/racm/schedule/ver10
</ScheduleBlockType>
</ScheduleBlock>
</HolidaySchedule>
</CustomExtension>
</CustomExtensionList>
<IntelligentRecord>
<!--ro, opt, bool, whether to enable VCA recording-->false
</IntelligentRecord>
<delayTime>
<!--ro, opt, int, capture delay time, unit:s-->1
</delayTime>
<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>
<redundancyRec>
<!--ro, opt, bool, whether it is redundant recording-->false
</redundancyRec>
<passbackRecord>
<!--ro, opt, bool, whether to copy back recording-->false
</passbackRecord>
<lockDuration>
<!--ro, opt, int, recording locking duration, unit:h, 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>
<!--ro, opt, bool, whether to enable video archiving-->false
</recordBackup>
<SVCLevel>
<!--ro, opt, enum, SVC frame extracting type, subType:int, desc:0-not extracted, 1-half extracted, 2-three quarters extracted-->0
</SVCLevel>
<recordManage>
<!--ro, opt, bool, whether to enable video scheduling, desc:true (enabled scheduled recording), false (no scheduled recording but the recording schedule is still in use, e.g., motion detection and copy-back)-->false
</recordManage>
<extraSaveAudio>
<!--ro, opt, bool, whether to enable separate audio storage-->false
</extraSaveAudio>
</Track>

```

12.3.1.2 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.

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-DDDD-033595353625}
</TrackGUID>
<Size>
  <!--opt, int, recording size-->1
</Size>
<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>
<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>
<SrcUrlMethods>
  <!--req, enum, methods supported by the source, subType:string, desc:"DESCRIBE" (subscribe), "SETUP", "PLAY", "TEARDOWN" (stop)-->DESCRIBE
</SrcUrlMethods>
<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-4520-8BB8-7135789C8790}
</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>
<ScheduleDSTEnable>
  <!--req, bool, whether to enable Daylight Saving Time (DST)-->true

```

```

        ... -->, base, method to change daylight saving time (DST) /> see
    </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>
    <Log>
        <!--opt, bool, log, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
    </Log>
    <SaveImg>
        <!--opt, bool, save image, desc:this node is retained, and returned by some devices, but has no practical meaning-->true
    </SaveImg>
    <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), "failDown" (getting up or falling down detection)-->CMR
        </ActionRecordingMode>
    <PreRecordTimeSeconds>
        <!--opt, int, pre-record duration, unit:s, desc:from 0 to 10 seconds; default: 5 seconds-->5
    </PreRecordTimeSeconds>
    <PostRecordTimeSeconds>
        <!--opt, int, post-record duration, unit:s, desc:from 0 to 20 seconds; default: 5 seconds-->5
    </PostRecordTimeSeconds>
    <Actions>
        <ScheduleAction>
    </ScheduleAction>
    <ScheduleBlock>
    <ScheduleBlockList>
</TrackSchedule>
<CustomExtensionList>
<!--opt, array, custom extension list, subType:object-->
<CustomExtension>
    <!--opt, object, custom extension-->
    <CustomExtensionName>
        <!--req, string, custom extension name-->www.xxx.com/RaCM/trackExt/ver10
    </CustomExtensionName>
    <enableSchedule>
        <!--opt, bool, whether to enable recording schedule configuration-->true
    </enableSchedule>
    <SaveAudio>
        <!--opt, bool, whether to enable audio recording-->true
    </SaveAudio>
    <PreRecordTimeSeconds>
        <!--opt, int, pre-record, unit:s-->1
    </PreRecordTimeSeconds>
    <PostRecordTimeSeconds>
        <!--opt, int, post-record, unit:s-->1
    </PostRecordTimeSeconds>
    <HolidaySchedule>
        <!--opt, object, holiday schedule-->
    <ScheduleBlock>
        <!--opt, object, schedule module-->
        <ScheduleBlockGUID>
            <!--req, string, schedule module GUID-->{00000000-0000-0000-0000-000000000000}
        </ScheduleBlockGUID>
        <ScheduleBlockType>
            <!--opt, string, schedule module type-->www.xxx.com/racm/schedule/ver10
        </ScheduleBlockType>
    </ScheduleBlock>
    </HolidaySchedule>
    </CustomExtension>
</CustomExtensionList>
<IntelligentRecord>
    <!--opt, bool, whether to enable VCA recording function-->false
</IntelligentRecord>
<delayTime>
    <!--opt, int, capture delay time, unit:s-->1
</delayTime>
<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>
<redundancyRec>
    <!--opt, bool, whether it is redundant recording-->false
</redundancyRec>
<passbackRecord>
    <!--opt, bool, whether to copy back recording-->false
</passbackRecord>
<lockDuration>
    <!--opt, int, recording Locking duration, unit:h, 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, bool, whether to enable video archiving-->false
</recordBackup>
<SVCLevel>
    <!--opt, enum, SVC frame extracting type, subType:int, desc:0 (not extracted), 1 (half extracted), 2 (three quarters extracted), 3 (a quarter
extracted)-->0
</SVCLevel>

```

```

extracted)-->0
</SVCLevel>
<recordManage>
  <!--opt, bool, whether to enable video scheduling, desc:true (enabled scheduled recording), false (no scheduled recording but the recording schedule is still in use, e.g., motion detection and copy-back)-->false
</recordManage>
<extraSaveAudio>
  <!--opt, bool, whether to enable separate audio storage-->false
</extraSaveAudio>
</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 (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>

```

12.3.2 Video Search

12.3.2.1 Get the capability of searching for video and picture files

Request URL

GET /ISAPI/ContentMgmt/search/capabilities

Query Parameter

None

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<CMSearchDescription xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, search conditions, attr:version{opt, string, protocolVersion}-->
  <timeSpanList>
    <!--ro, opt, array, list of time periods, subType:object-->
    <timeSpan>
      <!--ro, opt, object, time period-->
      <laneNumber min="1" max="99">
        <!--ro, opt, int, Lane No., range:[1,99], attr:min{req, int},max{req, int}-->1
      </laneNumber>
      <carType opt="all,unknow,coach,truck,car,minibus,smalltruck,bigcar,smallcar,human,tumbrel,triike,suv,midcar">
        <!--ro, opt, enum, vehicle type, subType:string, attr:opt{req, string}, desc:"coach", "truck", "car", "minibus", "smalltruck", "bigcar", "smallcar", "human", "tumbrel", "triike"; "suv"; "midcar", "all", "unknown"-->coach
      </carType>
      <startTime>
        <!--ro, req, datetime, start time-->1970-01-01T00:00:00+08:00
      </startTime>
      <endTime>
        <!--ro, req, datetime, end time-->1970-01-01T00:00:00+08:00
      </endTime>
    </timeSpan>
  </timeSpanList>
</CMSearchDescription>

```

12.4 PTZ

12.4.1 PTZ Maintenance

12.4.1.1 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}-->
  <maxPTZChannelNum min="1" max="10">
    <!--ro, opt, int, number of PTZ channels, attr:min{req, int},max{req, int}-->1
  </maxPTZChannelNum>
  <isSupportPTZWithPreview opt="true,false">
    <!--ro, opt, bool, whether to support Live view, attr:opt{req, string}-->true
  </isSupportPTZWithPreview>
  <isSupportZoomLinkage opt="true,false">
    <!--ro, opt, bool, attr:opt{req, string}-->true
  </isSupportZoomLinkage>
  <isSupportManualWiper opt="true,false">
    <!--ro, opt, bool, whether to support wiper control interface manually, attr:opt{req, string}-->true
  </isSupportManualWiper>
  <isSupportTiltSync opt="true,false">
    <!--ro, opt, bool, whether to support tilt sync control of sound and bullet camera, attr:opt{req, string}-->true
  </isSupportTiltSync>
  <isSupportPTZProtocolList>
    <!--ro, opt, bool, whether to support getting PTZ protocol supported by device, desc:/ISAPI/PTZCtrl/PTZProtocolList?format=json-->true
  </isSupportPTZProtocolList>
  <isSupportRailwayRobot>
    <!--ro, opt, bool-->true
  </isSupportRailwayRobot>
  <isSupportPTZViewPreset>
    <!--ro, opt, bool-->true
  </isSupportPTZViewPreset>
  <isSupportPresetsProFocus>
    <!--ro, opt, bool-->true
  </isSupportPresetsProFocus>
  <isSupportGetCurrentPresetInfo>
    <!--ro, opt, bool-->true
  </isSupportGetCurrentPresetInfo>
</PTZCtrlCap>
```

12.4.1.2 Set a specified PTZ channel's parameters

Request URL

PUT /ISAPI/PTZCtrl/channels/<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

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

<PTZChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, PTZ channel's parameters, attr:version{req, string, protocolVersion}-->
  <id>
    <!--req, int, ID-->1
  </id>
  <enabled>
    <!--req, bool, whether to enable-->true
  </enabled>
  <serialNumber>
    <!--req, int, serial port No., desc:this field should be supported by both GET and PUT methods. If the value is 0, it indicates that no serial port is
    linked-->1
  </serialNumber>
  <videoInputID>
```

```

<!--req, int, video input ID-->
</videoInputID>
<panMaxSpeed>
    <!--opt, int, the maximum panning speed, unit: degrees/sec-->1
</panMaxSpeed>
<tiltMaxSpeed>
    <!--opt, int, the maximum titling speed, unit: degrees/sec-->1
</tiltMaxSpeed>
<presetSpeed>
    <!--opt, int, preset speed-->1
</presetSpeed>
<autoPatrolSpeed>
    <!--opt, int, automatic patrol speed-->1
</autoPatrolSpeed>
<keyBoardControlSpeed>
    <!--opt, int, keypad control speed, range:[0,100]-->1
</keyBoardControlSpeed>
<controlProtocol>
    <!--opt, enum, control protocol, subType:string, desc:"pelco-d", "modbus-RTU", "modbus-ASCII"-->pelco-d
</controlProtocol>
<controlAddress>
    <!--req, object, control address-->
<enabled>
    <!--req, bool, whether to enable-->true
</enabled>
<Address>
    <!--req, string, address, range:[1,255]-->test
</Address>
</controlAddress>
<defaultPresetID>
    <!--req, string, preset ID-->test
</defaultPresetID>
<PTZRs485Para>
    <!--req, object-->
<baudRate>
    <!--req, int, baud rate-->9600
</baudRate>
<dataBits>
    <!--req, int, data bit-->1
</dataBits>
<parityType>
    <!--req, enum, parity type, subType:string, desc:"none" (no parity bit), "even" (even parity), "odd" (odd parity), "mark" (total number of parity bits is 1), "space" (total number of parity bits is 0)-->space
</parityType>
<stopBits>
    <!--req, enum, stop bit, subType:string, desc:"1", "1.5", "2"-->1
</stopBits>
<flowCtrl>
    <!--req, enum, flow control type, subType:string, desc:"none", "software", "hardware"-->hardware
</flowCtrl>
</PTZRs485Para>
<manualControlSpeed>
    <!--opt, enum, manual control speed type, subType:string, desc:"pedestrian", "nonMotorVehicle" (non-motor vehicles), "motorVehicle" (vehicle), "selfadaptive" (self-adaptive), "compatible"-->pedestrian
</manualControlSpeed>
<panSupport>
    <!--opt, bool, whether it supports panning-->true
</panSupport>
<tiltSupport>
    <!--opt, bool, whether it supports tilting-->true
</tiltSupport>
<zoomSupport>
    <!--opt, bool, whether it supports zooming-->true
</zoomSupport>
<PTPositiveDirection>
    <!--opt, object, panning and tilting positive direction, desc:panning and tilting positive direction-->
<pan>
    <!--opt, enum, panning positive direction, subType:string, desc:"Left", "right"-->left
</pan>
<tile>
    <!--opt, enum, tilting positive direction, subType:string, desc:"up", "down"-->up
</tile>
</PTPositiveDirection>
<PTQueryFrequency>
    <!--opt, enum, subType:int-->30
</PTQueryFrequency>
</PTZChannel>

```

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>

```

12.4.1.3 Get parameters of a specified PTZ channel

Request URL

GET /ISAPI/PTZCtrl/channels/<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

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

<PTZChannel 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, ro-->true
  </enabled>
  <serialNumber>
    <!--ro, req, int-->1
  </serialNumber>
  <videoInputID>
    <!--ro, req, int-->1
  </videoInputID>
  <panMaxSpeed>
    <!--ro, opt, int, the maximum speed for panning-->1
  </panMaxSpeed>
  <tiltMaxSpeed>
    <!--ro, opt, int, the maximum speed for tilting-->1
  </tiltMaxSpeed>
  <presetSpeed>
    <!--ro, opt, int, preset speed-->1
  </presetSpeed>
  <autoPatrolSpeed>
    <!--ro, opt, int, automatic patrol speed-->1
  </autoPatrolSpeed>
  <keyBoardControlSpeed>
    <!--ro, opt, int, keyboard control speed, range:[0,100]-->1
  </keyBoardControlSpeed>
  <controlProtocol>
    <!--ro, opt, enum, "pelco-d,modbus-RTU,modbus-ASCII", subType:string, desc:"pelco-d,modbus-RTU,modbus-ASCII"-->pelco-d
  </controlProtocol>
  <controlAddress>
    <!--ro, opt, object-->
    <enabled>
      <!--ro, req, bool, ro-->true
    </enabled>
    <Address>
      <!--ro, opt, string, 1-255, range:[1,255]-->test
    </Address>
  </controlAddress>
  <defaultPresetID>
    <!--ro, opt, string, default preset ID-->test
  </defaultPresetID>
  <PTZRs485Para>

```

```

<!--ro, opt, object-->
<baudRate>
    <!--ro, req, int, baud rate-->9600
</baudRate>
<dataBits>
    <!--ro, req, int, data bit-->1
</dataBits>
<parityType>
    <!--ro, req, enum, "none,even,odd,mark,space", subType:string, desc:"none,even,odd,mark,space"-->space
</parityType>
<stopBits>
    <!--ro, req, enum, stop bit, subType:string, desc:stop bit-->1
</stopBits>
<flowCtrl>
    <!--ro, req, enum, flow control mode, subType:string, desc:flow control mode-->hardware
</flowCtrl>
</PTZRs485Para>
<manualControlSpeed>
    <!--ro, opt, enum, manual control speed, subType:string, desc:manual control speed-->pedestrian
</manualControlSpeed>
<panSupport>
    <!--ro, opt, bool, whether it supports panning-->true
</panSupport>
<tiltSupport>
    <!--ro, opt, bool, whether it supports tilting-->true
</tiltSupport>
<zoomSupport>
    <!--ro, opt, bool, whether it supports zooming-->true
</zoomSupport>
<PTPositiveDirection>
    <!--ro, opt, object, panning and tilting positive direction, desc:panning and tilting positive direction-->
<pan>
    <!--ro, opt, enum, panning positive direction (horizontal), subType:string, desc:"left", "right"-->left
</pan>
<tile>
    <!--ro, opt, enum, tilting positive direction (vertical), subType:string, desc:"up", "down"-->up
</tile>
</PTPositiveDirection>
<PTQueryFrequency>
    <!--ro, opt, enum, subType:int-->30
</PTQueryFrequency>
<tiltNegationEnabled>
    <!--ro, opt, bool-->false
</tiltNegationEnabled>
<presetAccuracy>
    <!--ro, opt, float, range:[0.00,0.50]-->0.01
</presetAccuracy>
<autoFocusCompletionWaitEnabled>
    <!--ro, opt, bool-->false
</autoFocusCompletionWaitEnabled>
</PTZChannel>

```

12.5 Vehicle Recognition

12.5.1 Motor Vehicle Recognition

12.5.1.1 Get the total capability of vehicle capture and recognition

Request URL

GET /ISAPI/ITC/capabilities

Query Parameter

None

Request Message

None

Response Message

```

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

<ITCCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
    <isSupportITC>
        <!--ro, opt, bool-->true
    </isSupportITC>
    <isSupportITCStatus>
        <!--ro, opt, bool-->true
    </isSupportITCStatus>
    <isSupportIllegalDictionary>
        <!--ro, opt, bool-->true
    </isSupportIllegalDictionary>
    <isSupportITCSetUp>
        <!--ro, opt, bool-->true
    </isSupportITCSetUp>

```

```
<!--ro, opt, bool-->true
</isSupportITCSetUp>
<isSupportTrafficParam>
    <!--ro, opt, bool-->true
</isSupportTrafficParam>
<isSupportManualCap>
    <!--ro, opt, bool-->true
</isSupportManualCap>
<isSupportIllegalUploadPic>
    <!--ro, opt, bool-->true
</isSupportIllegalUploadPic>
<isSupportContinueCap>
    <!--ro, opt, bool-->true
</isSupportContinueCap>
<isSupportWiper>
    <!--ro, opt, bool-->true
</isSupportWiper>
<isSupportEntranceCap>
    <!--ro, opt, bool-->true
</isSupportEntranceCap>
<isSupportPlateRecognitionParam>
    <!--ro, opt, bool-->true
</isSupportPlateRecognitionParam>
<isSupportSyncSignalOutput>
    <!--ro, opt, bool-->true
</isSupportSyncSignalOutput>
<isSupportImageMerge>
    <!--ro, opt, bool-->true
</isSupportImageMerge>
<isSupportCabinetParam>
    <!--ro, opt, bool-->true
</isSupportCabinetParam>
<isSupportCarFeatureParam>
    <!--ro, opt, bool-->true
</isSupportCarFeatureParam>
<isSupportIllegalCodeData>
    <!--ro, opt, bool-->true
</isSupportIllegalCodeData>
<isSupportSnapshot>
    <!--ro, opt, bool-->true
</isSupportSnapshot>
<isSupportSyncPower>
    <!--ro, opt, bool-->true
</isSupportSyncPower>
<isSupportNetStorage>
    <!--ro, opt, bool-->true
</isSupportNetStorage>
<isSupportLightCorrect>
    <!--ro, opt, bool-->true
</isSupportLightCorrect>
<isSupportEntranceShowCap>
    <!--ro, opt, bool-->true
</isSupportEntranceShowCap>
<isSupportRadarSetUp>
    <!--ro, opt, bool-->true
</isSupportRadarSetUp>
<isSupportAlgorithmsState>
    <!--ro, opt, bool-->true
</isSupportAlgorithmsState>
<isSupportAlgorithmsVersion>
    <!--ro, opt, bool-->true
</isSupportAlgorithmsVersion>
<isSupportPlateCorrection>
    <!--ro, opt, bool-->true
</isSupportPlateCorrection>
<isSupportVideoSpeedDetection>
    <!--ro, opt, bool-->true
</isSupportVideoSpeedDetection>
<isSupportRadarMeasurement>
    <!--ro, opt, bool-->true
</isSupportRadarMeasurement>
<isSupportEvidenceDictionary>
    <!--ro, opt, bool-->true
</isSupportEvidenceDictionary>
<isSupportVehicleDetection>
    <!--ro, opt, bool-->true
</isSupportVehicleDetection>
<isSupportHVTVehicleDetection>
    <!--ro, opt, bool-->true
</isSupportHVTVehicleDetection>
<isSupportLicencePlateAuditData>
    <!--ro, opt, bool-->true
</isSupportLicencePlateAuditData>
<isSupportSearchLPLListAudit>
    <!--ro, opt, bool-->true
</isSupportSearchLPLListAudit>
<isSupportVehicleDetectionManualTrigger opt="false,true">
    <!--ro, opt, bool, attr:opt(freq, string)-->true
</isSupportVehicleDetectionManualTrigger>
<isSupportWebReceiveAlarm>
    <!--ro, opt, bool-->true
</isSupportWebReceiveAlarm>
<isSupportRealtimeView>
    <!--ro, opt, bool-->true
```

```

</isSupportRealtimeView>
<isSupportDifferIllegalPic>
  <!--ro, opt, bool-->true
</isSupportDifferIllegalPic>
<isSupportEntranceShow>
  <!--ro, opt, bool-->true
</isSupportEntranceShow>
<isSupportErectParam>
  <!--ro, opt, bool-->true
</isSupportErectParam>
<LSTSCap>
  <!--ro, opt, object-->
<isSupportLiveReceivePicture>
  <!--ro, opt, bool-->true
</isSupportLiveReceivePicture>
<isSupportTrafficLightStatus>
  <!--ro, opt, bool-->true
</isSupportTrafficLightStatus>
<isSupportTrafficStatistics>
  <!--ro, opt, bool-->true
</isSupportTrafficStatistics>
</LSTSCap>
<isSupportIllegalPicImport>
  <!--ro, opt, bool-->true
</isSupportIllegalPicImport>
</ITCCap>

```

12.5.1.2 Get the vehicle feature parameters capability

Request URL

GET /ISAPI/ITC/carFeatureParam/capabilities?channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	optional, for specifying channel No. for multi-channel devices; when this node is not returned, it is channel 1 by default

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<CarFeatureParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, vehicle feature parameters, attr:version{req, string, protocolVersion}-->
  <isSupportCarColor>
    <!--ro, opt, bool, whether to support vehicle color recognition-->true
  </isSupportCarColor>
  <isSupportCarLogo>
    <!--ro, opt, bool, whether to support vehicle logo recognition-->true
  </isSupportCarLogo>
  <isSupportSafeBelt>
    <!--ro, opt, bool, whether to support safety belt detection, desc:whether to support safety belt detection-->true
  </isSupportSafeBelt>
  <isSupportSunVisor>
    <!--ro, opt, bool, whether to support sun visor detection-->true
  </isSupportSunVisor>
  <isSupportHighPollutVehicle>
    <!--ro, opt, bool, whether to support yellow-label vehicle detection-->true
  </isSupportHighPollutVehicle>
  <isSupportDangerousVehicle>
    <!--ro, opt, bool, whether to support dangerous goods transport vehicle detection 1-->true
  </isSupportDangerousVehicle>
  <isSupportVehicleSubBrand>
    <!--ro, opt, bool, whether to support vehicle sub-brand detection-->true
  </isSupportVehicleSubBrand>
  <isSupportEnvprosign>
    <!--ro, opt, bool, whether to support dangerous goods transport vehicle detection 2-->true
  </isSupportEnvprosign>
  <isSupportFaceDetect>
    <!--ro, req, bool, whether to enable face matting-->true
  </isSupportFaceDetect>
  <isSupportUphone>
    <!--ro, opt, bool, whether to support phone call detection-->true
  </isSupportUphone>
  <isSupportPendant>
    <!--ro, opt, bool, whether to support window hangings detection-->true
  </isSupportPendant>
  <isSupportPilotFace>
    <!--ro, req, bool, whether to support driver's face matting-->true
  </isSupportPilotFace>

```

```

</isSupportPilotFace>
<isSupportCopilotFace>
  <!--ro, req, bool, whether to support front passenger's face picture matting-->true
<isSupportCopilotFace>
<isSupportNonmotorface>
  <!--ro, opt, bool, whether to support non-motor vehicle driver's face matting-->true
<isSupportNonmotorface>
<faceImgScale min="1" max="10">
  <!--ro, req, int, close-up face picture zooming ratio, range:[1,10], attr:min{req, int},max{req, int}-->1
</faceImgScale>
<faceImgRatio min="1" max="3">
  <!--ro, req, int, close-up face picture matting ratio, range:[1,3], attr:min{req, int},max{req, int}, desc:close-up face picture matting ratio-->1
</faceImgRatio>
<FaceImgInfoEx>
  <!--ro, opt, object, extended information of face matting-->
  <enabled opt="true,false">
    <!--ro, opt, bool, whether to enable face matting types of close-up face picture, attr:opt{req, string}, desc:the node faceImgScale will be invalid when this function is enabled-->false
  </enabled>
<faceImgType opt="custom, IDCard">
  <!--ro, opt, enum, close-up face picture type, subType:string, attr:opt{req, string}, desc:"custom", "IDCard" (ID picture)-->custom
</faceImgType>
<faceCloseupRatio>
  <!--ro, opt, object, target capture size, desc:this node is valid when faceImgType is "custom"-->
  <upRatio min="0" max="100">
    <!--ro, req, int, enlarge from top, attr:min{req, int},max{req, int}-->20
  </upRatio>
  <downRatio min="0" max="100">
    <!--ro, req, int, enlarge from bottom, attr:min{req, int},max{req, int}-->20
  </downRatio>
  <leftRatio min="0" max="100">
    <!--ro, req, int, enlarge from left, attr:min{req, int},max{req, int}-->20
  </leftRatio>
  <rightRatio min="0" max="100">
    <!--ro, req, int, enlarge from right, attr:min{req, int},max{req, int}-->20
  </rightRatio>
</faceCloseupRatio>
</FaceImgInfoEx>
<faceImgOutput opt="none,overPic,alarmUpload,overPicAndAlarmUpload">
  <!--ro, opt, enum, close-up face picture output mode, subType:string, attr:opt{req, string}, desc:"none" (no output), "overPic" (overlay on scene picture), "alarmUpload" (upload arm), "overPicAndAlarmUpload" (overlay on scene picture and upload arm)-->none
</faceImgOutput>
<isSupportDrawFaceRect>
  <!--ro, opt, bool, whether to overlay frame on face matting-->true
</isSupportDrawFaceRect>
<isSupportFaceImgBrightenEnhance>
  <!--ro, opt, bool, whether to support close-up face picture contrast enhancement-->true
</isSupportFaceImgBrightenEnhance>
<faceImgBrightenEnhanceLevel min="0" max="100">
  <!--ro, req, int, contrast enhancement Level, range:[0,100], attr:min{req, int},max{req, int}-->0
</faceImgBrightenEnhanceLevel>
<tissueBoxEnable>
  <!--ro, opt, bool, whether to support enabling tissue box detection-->true
</tissueBoxEnable>
<frontChildEnable>
  <!--ro, opt, bool, whether to support enabling baby in arm detection-->true
</frontChildEnable>
<labelEnable>
  <!--ro, opt, bool, whether to support enabling label detection-->true
</labelEnable>
<decorationEnable>
  <!--ro, opt, bool, whether to support enabling decoration detection-->true
</decorationEnable>
<safeBeltSensitivity min="1" max="100">
  <!--ro, opt, int, safety belt detection sensitivity, range:[1,100], attr:min{req, int},max{req, int}, desc:this node is valid when safeBeltEnabled is "true"-->1
</safeBeltSensitivity>
<uphoneSensitivity min="1" max="100">
  <!--ro, opt, int, phone call detection sensitivity, range:[1,100], attr:min{req, int},max{req, int}, desc:this node is valid when uphoneEnabled is "true"-->1
</uphoneSensitivity>
<pendantSensitivity min="1" max="100">
  <!--ro, opt, int, window hangings detection sensitivity, range:[1,100], attr:min{req, int},max{req, int}, desc:this node is valid when pendantEnabled is "true"-->1
</pendantSensitivity>
<isSupportSmoking>
  <!--ro, opt, bool, whether to support smoking detection-->true
</isSupportSmoking>
<isSupportTheHelmet>
  <!--ro, opt, bool, whether to support helmet detection-->true
</isSupportTheHelmet>
<isSupportTwoWheelVehicle>
  <!--ro, opt, bool, whether to support manned two wheeler detection-->true
</isSupportTwoWheelVehicle>
<targetCutout>
  <!--ro, opt, object, target face matting-->
  <enabled opt="true,false">
    <!--ro, opt, bool, whether to enable target face matting, attr:opt{req, string}-->true
  </enabled>
  <size>
    <!--ro, req, enum, target matting ratio, subType:int, desc:1 (small), 2 (medium), 3 (Large)-->1
  </size>
  <zoom>
    <!--ro, req, enum, zooming in/out ratio, subType:int, desc:1 (small), 2 (medium), 3 (large)-->1
  </zoom>

```

```

    <!-- ro, req, object, vehicle feature parameters, attr:version{req, string, protocolVersion}-->
</zoom>
</targetCutout>
<playMobilePhone>
    <!--ro, opt, bool, whether to enable driver using mobile phone detection-->true
</playMobilePhone>
<isSupportShed>
    <!--ro, opt, bool, whether to enable non-motor vehicle umbrella tent detection-->true
</isSupportShed>
<separateData>
    <!--ro, opt, bool, upload license plate No. and picture information separately and for multiple times-->true
</separateData>
<helmetDaySensitivity min="1" max="100">
    <!--ro, req, int, Sensitivity for Not Wearing Helmet at Daytime, attr:min{req, int},max{req, int}-->0
</helmetDaySensitivity>
<helmetNightSensitivity min="1" max="100">
    <!--ro, req, int, Sensitivity for Not Wearing Helmet at Night, attr:min{req, int},max{req, int}-->0
</helmetNightSensitivity>
<isSupportFocedCopilotFace>
    <!--ro, opt, bool, whether to mandate face matting for front passenger-->true
</isSupportFocedCopilotFace>
<smokingSensitivity min="0" max="100">
    <!--ro, opt, int, smoking detection sensitivity, range:[0,100], attr:min{req, int},max{req, int}-->1
</smokingSensitivity>
<helmetCheckMode opt="checkOutFirst,checkAccurateFirst,custom">
    <!--ro, opt, enum, helmet detection mode, subType:string, dep:and, ${.CarFeatureParam.isSupportHelmet,eq,true}, attr:opt{req, string}, desc:1, "checkOutFirst", "checkAccurateFirst"; it is "checkAccurateFirst" by default when this node is not returned
2. during detection of not wearing helmet: "checkOutFirst" is for higher detection rate but comes with higher false alarm rate which requires manual verification; "checkAccurateFirst" is for high-quality detection but comes with higher under-detection rate-->checkOutFirst
</helmetCheckMode>
<reflectiveStripeEnabled opt="true,false">
    <!--ro, opt, bool, whether to enable reflective stripe detection, attr:opt{req, string}-->true
</reflectiveStripeEnabled>
</CarFeatureParam>

```

12.5.1.3 Get the vehicle feature configuration parameters

Request URL

GET /ISAPI/ITC/carFeatureParam?channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	optional, for specifying channel No. for multi-channel devices; when this node is not returned, it is channel 1 by default

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<CarFeatureParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, vehicle feature parameters, attr:version{req, string, protocolVersion}-->
    <colorEnabled>
        <!--ro, req, bool, vehicle color recognition-->true
    </colorEnabled>
    <carLogoEnabled>
        <!--ro, req, bool, vehicle log recognition-->true
    </carLogoEnabled>
    <sunVisorEnabled>
        <!--ro, req, bool, sun visor detection-->true
    </sunVisorEnabled>
    <faceDetectEnabled>
        <!--ro, req, bool, face picture matting-->true
    </faceDetectEnabled>
    <uphoneEnabled>
        <!--ro, opt, bool, phone call detection-->true
    </uphoneEnabled>
    <pendantEnabled>
        <!--ro, opt, bool, window hangings detection-->true
    </pendantEnabled>
    <tissueBoxEnable>
        <!--ro, opt, bool, tissue box detection-->true
    </tissueBoxEnable>
    <frontChildEnable>
        <!--ro, opt, bool, baby in arm detection-->true
    </frontChildEnable>
    <labelEnable>
        <!--ro, opt, bool, label detection-->true
    </labelEnable>
    <uphoneSensitivity>

```

```

<!--ro, opt, int, phone call detection sensitivity, range:[1,100]-->1
</uPhoneSensitivity>
<pendantSensitivity>
  <!--ro, opt, int, window hangings detection sensitivity, range:[1,100]-->1
</pendantSensitivity>
<decorationEnable>
  <!--ro, opt, bool, decoration detection-->true
</decorationEnable>
<OutputCtrl>
  <!--ro, opt, object, face matting-->
  <hostFaceEnabled>
    <!--ro, req, bool, driver's face matting-->true
  </hostFaceEnabled>
  <viceFaceEnabled>
    <!--ro, req, bool, front passenger's face matting-->true
  </viceFaceEnabled>
  <nonMotorFaceEnabled>
    <!--ro, req, bool, non-motor vehicle driver's face matting-->true
  </nonMotorFaceEnabled>
  <faceImgScale>
    <!--ro, req, int, close-up face picture zooming ratio, range:[1,10]-->1
  </faceImgScale>
  <faceImgRatio>
    <!--ro, req, enum, close-up face picture cutting ratio, subType:int, desc:1 (small), 2 (medium), 3 (Large)-->1
  </faceImgRatio>
  <FaceImgInfoEx>
    <!--ro, opt, object, extended information of face matting-->
    <enabled>
      <!--ro, req, bool, whether to enable face matting types of close-up face picture, desc:the node faceImgScale will be invalid when this function is
enabled-->false
    </enabled>
    <faceImgType>
      <!--ro, req, enum, close-up face picture type, subType:string, desc:"custom", "IDCard" (ID picture)-->IDCard
    </faceImgType>
    <faceCloseupRatio>
      <!--ro, opt, object, target capture size, desc:this node is valid when faceImgType is "custom"-->
      <upRatio>
        <!--ro, req, int, enlarge from top, range:[0,100]-->20
      </upRatio>
      <downRatio>
        <!--ro, req, int, enlarge from bottom, range:[0,100]-->20
      </downRatio>
      <leftRatio>
        <!--ro, req, int, enlarge from left, range:[0,100]-->20
      </leftRatio>
      <rightRatio>
        <!--ro, req, int, enlarge from right, range:[0,100]-->20
      </rightRatio>
    </faceCloseupRatio>
    </FaceImgInfoEx>
    <faceImgOutputMode>
      <!--ro, req, enum, close-up face picture output mode, subType:string, desc:"none" (no output), "overPic" (overlay on scene picture), "alarmUpload"
(upload arm), "overPicAndAlarmUpload" (overlay on scene picture and upload arm)-->none
    </faceImgOutputMode>
    <DrawFaceRectEnabled>
      <!--ro, req, bool, whether to overlay frame on face matting-->true
    </DrawFaceRectEnabled>
    <faceImgBrightenEnhanceEnabled>
      <!--ro, req, bool, close-up face picture contrast enhancement-->true
    </faceImgBrightenEnhanceEnabled>
    <faceImgBrightenEnhanceLevel>
      <!--ro, req, int, contrast enhancement Level, range:[0,100]-->0
    </faceImgBrightenEnhanceLevel>
    <faceCopyIllegal>
      <!--ro, opt, bool, whether need enable face matting for violation picture-->true
    </faceCopyIllegal>
    <forcedViceFaceEnabled>
      <!--ro, opt, bool, whether to mandate face matting for front passenger-->true
    </forcedViceFaceEnabled>
  </OutputCtrl>
  <smokingEnabled>
    <!--ro, req, bool, smoking detection-->true
  </smokingEnabled>
  <helmetEnabled>
    <!--ro, opt, bool, helmet detection-->true
  </helmetEnabled>
  <twoWheelVehicleEnabled>
    <!--ro, opt, bool, manned two wheeler detection-->true
  </twoWheelVehicleEnabled>
  <playMobilePhone>
    <!--ro, opt, bool, whether to enable driver using mobile phone detection-->true
  </playMobilePhone>
  <isSupportShed>
    <!--ro, opt, bool, whether to enable non-motor vehicle umbrella tent detection-->true
  </isSupportShed>
  <separatData>
    <!--ro, opt, bool, upload license plate No. and picture information separately and for multiple times-->true
  </separatData>
  <helmetDaySensitivity>
    <!--ro, req, int, sensitivity for detecting not wearing helmet at daytime-->0
  </helmetDaySensitivity>
  <helmetNightSensitivity>
    <!--ro, req, int, sensitivity for detecting not wearing helmet at night-->0
  </helmetNightSensitivity>

```

```

</nonmetNightSensitivity>
<shedSensitivity>
  <!--ro, opt, int, non-motor vehicle umbrella tent detection sensitivity, range:[1,100]-->0
</shedSensitivity>
<twoWheelVehicleSensitivity>
  <!--ro, opt, int, manned two wheeler detection sensitivity, range:[1,100]-->0
</twoWheelVehicleSensitivity>
<smokingSensitivity>
  <!--ro, opt, int, smoking detection sensitivity, range:[0,100]-->0
</smokingSensitivity>
<CarWindowFeature>
  <!--ro, opt, object, vehicle window feature detection-->
<tempPlateEnable>
  <!--ro, opt, bool, whether to enable detection of temporary license plate on the window, desc: detect whether there is temporary license plate on the window-->true
</tempPlateEnable>
<tempPlateSensitivity>
  <!--ro, opt, int, detection sensitivity of temporary license plate on the window, range:[0,100]-->1
</tempPlateSensitivity>
<passCardEnable>
  <!--ro, opt, bool, whether to enable detection of entry & exit pass on the window, desc: whether there is entry & exit pass on the window-->true
</passCardEnable>
<passCardSensitivity>
  <!--ro, opt, int, detection sensitivity of entry & exit pass on the window, range:[0,100]-->1
</passCardSensitivity>
<carCardEnable>
  <!--ro, opt, bool, whether to enable detection of cards attached to the window, desc: whether there is any card (business cards, flyers, etc) attached to the window-->true
</carCardEnable>
<carCardSensitivity>
  <!--ro, opt, int, detection sensitivity of cards attached to the window, range:[0,100]-->1
</carCardSensitivity>
<CarWindowFeature>
<CarBodyFeature>
  <!--ro, opt, object, bodywork feature detection-->
<enabled>
  <!--ro, opt, bool, whether to enable bodywork feature detection, desc: whether the vehicle is with spare tire, with sunroof, with roof rack, painted, people sticking out of the sunroof, covered (for dump truck)-->true
</enabled>
</CarBodyFeature>
<vehicleUseEnable>
  <!--ro, opt, bool, whether to enable vehicle purpose detection, desc: recognize bus, school bus, taxi, and ambulance-->true
</vehicleUseEnable>
<helmetCheckMode>
  <!--ro, opt, enum, helmet detection mode, subType:string, dep:and{$.CarFeatureParam.helmetEnabled, eq, true}, desc:1. "checkOutFirst", "checkAccurateFirst"; it is "checkAccurateFirst" by default when this node is not returned 2. during detection of not wearing helmet: "checkOutFirst" is for higher detection rate but comes with higher false alarm rate which requires manual verification; "checkAccurateFirst" is for high-quality detection but comes with higher under-detection rate-->checkOutFirst
</helmetCheckMode>
<reflectiveStripeEnabled>
  <!--ro, opt, bool, whether to enable reflective stripe detection-->true
</reflectiveStripeEnabled>
</CarFeatureParam>

```

12.5.1.4 Set the vehicle feature configuration parameters

Request URL

PUT /ISAPI/ITC/carFeatureParam?channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	optional, for specifying channel No. for multi-channel devices; when this node is not returned, it is channel 1 by default

Request Message

```

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

<CarFeatureParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, vehicle feature parameters, attr:version{req, string, protocolVersion}-->
  <colorEnabled>
    <!--req, bool, whether to enable vehicle color recognition-->true
  </colorEnabled>
  <carLogoEnabled>
    <!--req, bool, whether to enable vehicle log recognition-->true
  </carLogoEnabled>
  <pilotSafebeltEnabled>
    <!--req, bool, whether to enable safety belt detection-->true
  </pilotSafebeltEnabled>
  <sunVisorEnabled>
    <!--req, bool, Sun Visor Detection-->true
  </sunVisorEnabled>

```

```

<envprosignEnabled>
    <!--req, bool, Dangerous Goods Transport Vehicle Detection-->true
</envprosignEnabled>
<dangmarkEnabled>
    <!--req, bool, whether to enable dangerous goods transport vehicle detection-->true
</dangmarkEnabled>
<FaceDetectEnabled>
    <!--req, bool, Face Picture Matting-->true
</faceDetectEnabled>
<uphoneEnabled>
    <!--opt, bool, Phone Call Detection-->true
</uphoneEnabled>
<pendantEnabled>
    <!--opt, bool, whether to enable window hangings detection-->true
</pendantEnabled>
<tissueBoxEnable>
    <!--opt, bool, Tissue Box Detection-->true
</tissueBoxEnable>
<frontChildEnable>
    <!--opt, bool, Baby in Arm Detection-->true
</frontChildEnable>
<labelEnable>
    <!--opt, bool, Label Detection-->true
</labelEnable>
<safebeltSensitivity>
    <!--opt, int, safety belt detection sensitivity, range:[1,100]-->1
</safebeltSensitivity>
<uphoneSensitivity>
    <!--opt, int, Phone Call Detection, range:[1,100]-->1
</uphoneSensitivity>
<pendantSensitivity>
    <!--opt, int, window hangings detection sensitivity, range:[1,100]-->1
</pendantSensitivity>
<decorationEnable>
    <!--opt, bool, Decoration Detection-->true
</decorationEnable>
<OutputCtrl>
    <!--opt, object, Face Picture Matting-->
<hostFaceEnabled>
    <!--req, bool, Driver's Face Picture Matting-->true
</hostFaceEnabled>
<viceFaceEnabled>
    <!--req, bool, Passenger's Face Picture Matting-->true
</viceFaceEnabled>
<nonMotorFaceEnabled>
    <!--req, bool, Pedestrian's and Non-Motor Vehicle User's Face Picture Matting-->true
</nonMotorFaceEnabled>
<faceImgScale>
    <!--req, int, face close-up picture cutting ratio, range:[1,10]-->1
</faceImgScale>
<faceImgRatio>
    <!--req, enum, Face Close-up Picture Cutting Ratio, subType:int, desc:face close-up picture zooming ratio, which ranges from 1 to 3-->1
</faceImgRatio>
<FaceImgInfoEx>
    <!--ro, opt, object, extended information of face matting-->
<enabled>
    <!--ro, req, bool, whether to enable face matting types of close-up face picture, desc:the node faceImgScale will be invalid when this function is enabled-->false
</enabled>
<faceImgType>
    <!--ro, req, enum, close-up face picture type, subType:string, desc:"custom", "IDCard" (ID picture)-->IDCard
</faceImgType>
<faceCloseupRatio>
    <!--ro, opt, object, target capture size, desc:this node is valid when faceImgType is "custom"-->
<upRatio>
    <!--ro, req, int, enlarge from top, range:[0,100]-->20
</upRatio>
<downRatio>
    <!--ro, req, int, enlarge from bottom, range:[0,100]-->20
</downRatio>
<leftRatio>
    <!--ro, req, int, enlarge from left, range:[0,100]-->20
</leftRatio>
<rightRatio>
    <!--ro, req, int, enlarge from right, range:[0,100]-->20
</rightRatio>
</faceCloseupRatio>
</FaceImgInfoEx>
<faceImgOutputMode>
    <!--req, enum, close-up face picture output mode, subType:string, desc:"none" (no output), "overPic" (overlay on scene picture), "alarmUpload" (upload arm), "overPicAndAlarmUpload" (overlay on scene picture and upload arm)-->none
</faceImgOutputMode>
<DrawFaceRectEnabled>
    <!--req, bool, whether to overlay frame on face matting-->true
</DrawFaceRectEnabled>
<faceImgBrightenEnhanceEnabled>
    <!--req, bool, close-up face picture contrast enhancement-->true
</faceImgBrightenEnhanceEnabled>
<faceImgBrightenEnhanceLevel>
    <!--req, int, contrast enhancement level, range:[0,100]-->0
</faceImgBrightenEnhanceLevel>
<faceCopyIllegal>
    <!--opt, bool, whether need enable face matting for violation picture-->true
</faceCopyIllegal>

```

```

</traceCopyIllegal>
<forcedViceFaceEnabled>
    <!--opt, bool, whether to mandate face matting for front passenger-->true
</forcedViceFaceEnabled>
</OutputCtrl>
<smokingEnabled>
    <!--req, bool, smoking detection-->true
</smokingEnabled>
<perfumeBoxEnabled>
    <!--opt, bool, perfume box detection-->true
</perfumeBoxEnabled>
<helmetEnabled>
    <!--opt, bool, helmet detection-->true
</helmetEnabled>
<twoWheelVehicleEnabled>
    <!--opt, bool, manned two wheeler detection-->true
</twoWheelVehicleEnabled>
<threeWheelVehicleEnabled>
    <!--opt, bool, manned tricycle detection-->true
</threeWheelVehicleEnabled>
<targetCutout>
    <!--opt, object, target face matting-->
<enabled>
    <!--opt, bool, whether to enable target face matting-->true
</enabled>
<size>
    <!--req, enum, target matting ratio, subType:int, desc:1 (small), 2 (medium), 3 (Large)-->1
</size>
<zoom>
    <!--req, enum, zooming in/out ratio, subType:int, desc:1 (small), 2 (medium), 3 (Large)-->1
</zoom>
</targetCutout>
<playMobilePhone>
    <!--opt, bool, whether to enable driver using mobile phone detection-->true
</playMobilePhone>
<isSupportShed>
    <!--opt, bool, whether to enable non-motor vehicle umbrella tent detection-->true
</isSupportShed>
<separatData>
    <!--opt, bool, upload license plate No. and picture information separately and for multiple times-->true
</separatData>
<rudder>
    <!--opt, enum, Left/right rudder, subType:string, desc:"left", "right"-->left
</rudder>
<helmetSensitivity>
    <!--opt, int, helmet detection sensitivity, range:[1,100]-->0
</helmetSensitivity>
<helmetDaySensitivity>
    <!--ro, req, int, Sensitivity for Not Wearing Helmet at Daytime-->0
</helmetDaySensitivity>
<helmetNightSensitivity>
    <!--ro, req, int, Sensitivity for Not Wearing Helmet at Night-->0
</helmetNightSensitivity>
<shedSensitivity>
    <!--opt, int, non-motor vehicle umbrella tent detection sensitivity, range:[1,100]-->0
</shedSensitivity>
<twoWheelVehicleSensitivity>
    <!--opt, int, manned two wheeler detection sensitivity, range:[1,100]-->0
</twoWheelVehicleSensitivity>
<smokingSensitivity>
    <!--opt, int, smoking detection sensitivity, range:[0,100]-->0
</smokingSensitivity>
<CarWindowFeature>
    <!--opt, object, vehicle window feature detection-->
<tempPlateEnable>
    <!--opt, bool, whether to enable detection of temporary license plate on the window, desc:detect whether there is temporary license plate on the window-->true
</tempPlateEnable>
<tempPlateSensitivity>
    <!--opt, int, detection sensitivity of temporary license plate on the window, range:[0,100]-->1
</tempPlateSensitivity>
<passCardEnable>
    <!--opt, bool, whether to enable detection of entry & exit pass on the window, desc:whether there is entry & exit pass on the window-->true
</passCardEnable>
<passCardSensitivity>
    <!--opt, int, detection sensitivity of entry & exit pass on the window, range:[0,100]-->1
</passCardSensitivity>
<carCardEnable>
    <!--opt, bool, whether to enable detection of cards attached to the window, desc:whether there is any card (business cards, flyers, etc.) attached to the window-->true
</carCardEnable>
<carCardSensitivity>
    <!--opt, int, detection sensitivity of cards attached to the window, range:[0,100]-->1
</carCardSensitivity>
<CarWindowFeature>
<CardBodyFeature>
    <!--opt, object, bodywork feature detection-->
<enabled>
    <!--opt, bool, whether to enable bodywork feature detection, desc:whether the vehicle is with spare tire, with sunroof, with roof rack, painted, people sticking out of the sunroof, covered (for dump truck)-->true
</enabled>
</CardBodyFeature>
<vehicleUseEnable>
    <!--opt, bool, whether to enable vehicle purpose detection, desc:recognize bus, school bus, taxi, and ambulance-->true
</vehicleUseEnable>

```

```

</vehicleUseEnable>
<channelId>
    <!--ro, opt, int, channel No., range:[1,5]-->1
</channelId>
<helmetCheckMode>
    <!--opt, enum, helmet detection mode, subType:string, dep:and, ${.CarFeatureParam.helmetEnabled},eq,true,>
desc:1, "checkOutFirst", "checkAccurateFirst"; it is "checkAccurateFirst" by default when this node is not returned
2. during detection of not wearing helmet: "checkOutFirst" is for higher detection rate but comes with higher false alarm rate which requires manual
verification; "checkAccurateFirst" is for high-quality detection but comes with higher under-detection rate-->checkOutFirst
</helmetCheckMode>
<reflectiveStripeEnabled>
    <!--ro, opt, bool, whether to enable reflective stripe detection-->true
</reflectiveStripeEnabled>
<illegalMannedNonMotorVehicle>
    <!--opt, object-->
    <peopleNumThreshold>
        <!--opt, int, range:[2,3]-->2
    </peopleNumThreshold>
</illegalMannedNonMotorVehicle>
</CarFeatureParam>

```

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>

```

12.5.1.5 Get the parameters of traffic rule violation dictionary

Request URL

GET /ISAPI/ITC/illegalDictionary

Query Parameter

None

Request Message

None

Response Message

```

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

<IllegalDictionary xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, traffic rule violation dictionary, attr:version{req, string, protocolVersion}-->
  <IllegalCodeList>
    <!--ro, req, array, violation code List, subType:object-->
    <IllegalCodeItem>
      <!--ro, req, object, violation code-->
      <idx>
        <!--ro, req, int, ID, range:[0,65535]-->0
      </idx>
      <illegalCode>
        <!--ro, req, string, violation code, range:[0,64]-->test
      </illegalCode>
      <illegalName>
        <!--ro, req, string, violation name, range:[0,256]-->test
      </illegalName>
      <illegalStringCode>
        <!--ro, opt, string, violation code in string format, range:[0,128]-->test
      </illegalStringCode>
      <illegalDescription>
        <!--ro, opt, string, violation description, range:[0,128]-->test
      </illegalDescription>
    </IllegalCodeItem>
  </IllegalCodeList>
</IllegalDictionary>

```

12.5.1.6 Set parameters of traffic violation dictionary

Request URL

PUT /ISAPI/ITC/illegalDictionary

Query Parameter

None

Request Message

```

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

<IllegalDictionary xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, traffic rule violation dictionary, attr:version{req, string, protocolVersion}-->
  <IllegalCodeList>
    <!--req, array, violation code List, subType:object-->
    <IllegalCodeItem>
      <!--req, object, violation code-->
      <idx>
        <!--req, int, index, range:[0,65535]-->0
      </idx>
      <illegalCode>
        <!--req, string, illegal action code, range:[0,64]-->test
      </illegalCode>
      <illegalName>
        <!--req, string, illegal action name, range:[0,256]-->test
      </illegalName>
      <illegalStringCode>
        <!--opt, string, illegal action code (string), range:[0,128]-->test
      </illegalStringCode>
      <illegalDescription>
        <!--opt, string, illegal action description, range:[0,128]-->test
      </illegalDescription>
    </IllegalCodeItem>
  </IllegalCodeList>
</IllegalDictionary>

```

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, 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>

```

12.5.1.7 Get the capability of traffic rule violation dictionary

Request URL

GET /ISAPI/ITC/illegalDictionary/capabilities

Query Parameter

None

Request Message

None

Response Message

```

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

<IllegalDictionary xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, traffic rule violation dictionary, attr:version{req, string, protocolVersion}-->
  <IllegalCodeList>
    <!--ro, req, array, violation code list, subtype:object-->
    <IllegalCodeItem>
      <!--ro, req, object, violation code-->
      <idx min="0" max="64">
        <!--ro, req, int, index, attr:min{req, int},max{req, int}-->0
      </idx>
      <illegalCode min="0" max="64">
        <!--ro, req, string, violation code, attr:min{req, int},max{req, int}-->test
      </illegalCode>
      <illegalName min="0" max="256">
        <!--ro, req, string, violation name, attr:min{req, int},max{req, int}-->test
      </illegalName>
      <illegalStringCode min="0" max="10">
        <!--ro, opt, string, violation code (string), attr:min{req, int},max{req, int}-->test
      </illegalStringCode>
      <illegalCodeLetterNum min="0" max="10">
        <!--ro, opt, int, supported number of letters in violation code, attr:min{req, int},max{req, int}-->0
      </illegalCodeLetterNum>
      <illegalDescription size="256">
        <!--ro, opt, string, violation description, attr:size{req, int}-->test
      </illegalDescription>
    </IllegalCodeItem>
  </IllegalCodeList>
  <isSupportReset>
    <!--ro, opt, bool, whether it supports restoring to default setting, desc:/ISAPI/ITC/illegalDictionary/reset-->true
  </isSupportReset>
  <isSupportOverspeedAdvancedParam>
    <!--ro, opt, bool, whether it supports overspeed advanced configuration of traffic rule violation dictionary-->true
  </isSupportOverspeedAdvancedParam>
</IllegalDictionary>

```

12.5.1.8 Restore the advanced parameters of traffic violation dictionary to default settings

Request URL

PUT /ISAPI/ITC/illegalDictionary/overspeedAdvancedParam/reset?format=json

Query Parameter

None

Request Message

None

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 node is required when the value of statusCode is not 1*/  
}
```

12.5.1.9 Get the advanced configuration of traffic rule violation dictionary

Request URL

GET /ISAPI/ITC/illegalDictionary/overspeedAdvancedParam?format=json

Query Parameter

None

Request Message

None

Response Message

```
{  
    "OverspeedAdvancedParam": {  
        /*ro, opt, object, advanced configuration of traffic rule violation dictionary*/  
        "enabled": true,  
        /*ro, opt, bool, whether to enable this function*/  
        "ParamList": [  
            /*ro, opt, array, parameter list, subType:object*/  
            {  
                "Param": {  
                    /*ro, opt, object, parameters*/  
                    "id": 1,  
                    /*ro, req, int, serial No.*/  
                    "speedUpperLimit": 60,  
                    /*ro, opt, int, upper speed limit, range:[0,999], unit:km/h*/  
                    "speedLowerLimit": 25,  
                    /*ro, opt, int, lower speed limit, range:[0,999], unit:km/h*/  
                    "overspeedRatioUpperLimit": 1,  
                    /*ro, opt, int, overspeed ratio upper limit, range:[0,999]*/  
                    "overspeedRatioLowerLimit": 1,  
                    /*ro, opt, int, overspeed ratio lower limit, range:[0,999]*/  
                    "laneType": "highway",  
                    /*ro, opt, enum, lane type, subType:string, desc:"highway", "urbanFastWay" (urban expressway), "carriageWay" (roadway), "other",  
                    "unknown"*/  
                    "overspeedVehicleClass": "schoolBus",  
                    /*ro, opt, enum, vehicle type for overspeed detection, subType:string, desc:"schoolBus", "tankTruck" (tanker), "largeTruck", "largeBus",  
                    "slagTruck" (dump truck), "dangerousVehicle" (vehicle carrying hazardous material), "smallVehicle" (normal small-sized vehicle), "largeVehicle" (normal  
                    large-sized vehicle), "other"*/  
                    "illegalCode": "1301",  
                    /*ro, opt, string, violation code, range:[1,32]*/  
                    "illegalName": "逆行",  
                    /*ro, opt, string, violation name, range:[1,32]*/  
                    "illegalDescription": "逆行",  
                    /*ro, opt, string, violation description, range:[1,128]*/  
                }  
            }  
        ]  
    }  
}
```

12.5.1.10 Set the advanced parameters of traffic rule violation dictionary

Request URL

PUT /ISAPI/ITC/illegalDictionary/overspeedAdvancedParam?format=json

Query Parameter

None

Request Message

```
{  
    "OverspeedAdvancedParam": {  
        /*opt, object, advanced configuration of traffic rule violation dictionary*/  
        "enabled": true,  
        /*opt, bool, whether to enable this function*/  
        "ParamList": [  
            /*opt, array, parameter list, subType:object*/  
            {  
                "Param": {  
                    /*opt, object, parameters*/  
                    "id": 1,  
                    /*req, int, serial No.*/  
                    "speedUpperLimit": 60,  
                    /*opt, int, upper speed limit, range:[0,999], unit:km/h*/  
                    "speedLowerLimit": 25,  
                    /*opt, int, lower speed limit, range:[0,999], unit:km/h*/  
                    "overspeedRatioUpperLimit": 1,  
                    /*opt, int, overspeed ratio upper limit, range:[0,999]*/  
                    "overspeedRatioLowerLimit": 1,  
                    /*opt, int, overspeed ratio lower limit, range:[0,999]*/  
                    "laneType": "highway",  
                    /*opt, enum, lane type, subType:string, desc:"highway", "urbanFastWay" (urban expressway), "carriageway" (roadway), "other", "unknown"*/  
                    "overspeedVehicleClass": "schoolBus",  
                    /*opt, enum, vehicle type for overspeed detection, subType:string, desc:"schoolBus", "tankTruck" (tanker), "largeTruck", "largeBus",  
                    "slagTruck" (dump truck), "dangerousVehicle" (vehicle carrying hazardous material), "smallVehicle" (normal small-sized vehicle), "largeVehicle" (normal  
                    large-sized vehicle), "other"*/  
                    "illegalCode": "1301",  
                    /*opt, string, violation code, range:[1,32]*/  
                    "illegalName": "逆行",  
                    /*opt, string, violation name, range:[1,32]*/  
                    "illegalDescription": "逆行",  
                    /*opt, string, violation description, range:[1,128]*/  
                }  
            }  
        ]  
    }  
}
```

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 node is required when the value of statusCode is not 1*/  
}
```

12.5.1.11 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-->0
  </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>
```

12.5.1.12 Get the capability of arming parameters for traffic violations capture

Request URL

GET /ISAPI/ITC/illegalSchedules/capabilities?format=json

Query Parameter

None

Request Message

None

Response Message

```

{
    "eventPriorityEnable": {
        /*ro, opt, object, whether to enable violation priority*/
        "@opt": [true, false]
        /*ro, opt, array, options, subType:bool*/
    },
    "IllegalScheduleListCap": {
        /*ro, req, object, list of arming parameters for traffic violations capture, desc:this node is only available for ANPR violation information, related node is illegalInfo*/
        "@size": 1,
        /*ro, req, int, the maximum number supported in a single capture*/
    },
    "Schedule": {
        /*ro, opt, object, arming parameters*/
        "eventType": "post,driveLine",
        /*ro, req, string, event type, desc:"post" (checkpoint), "driveLine" (driving on the lane line), "reverse" (wrong-way driving), "banSign" (prohibition violation), "overSpeed" (driving in overspeed), "lowSpeed" (driving in low speed), "changeLane" (illegal lane change), "safeBelt", "uphone" (making a phone call when driving), "gasser" (queue jumping), "congestion", "helmet" (driver is wearing a helmet), "twoWheelPassenger" (manned non-motor vehicle), "nonMotorExist" (non-motor vehicle on motor vehicle lane), "intersectionCongest" (overstay at intersection), "nonDriveWay" (motor vehicle on non-motor vehicle lane), "redLight" (red light running), "intersectionStop" (stop vehicle at intersection), "direction" (driving against direction guidance), "greenLightStop (stop vehicle when green light), "illegalTurn", "accident", "runGreenLight" (running green light in congestion), "leftNotYieldStraight" (left turn not yield to straight), "rightNotYieldLeft" (right turn not yield to left turn), "uTurnNotYieldGoingStraight", "leftSideDrivingAfterLeftTurn", "failYieldPed" (not yield to pedestrian), "illegalParking" (illegal parking), "nonMotorRedLight" (non-motor vehicle red light running), "nonMotorCrossing" (non-motor vehicle crossing line), "nonMotorOccupiedZebraCrossing" (non-motor vehicle crossing zebra line), "roadBlock", "construction", "notKeepDistance" (not keeping a safe distance), "overtakeRightSide" (overtaking on the right), "dragRacing" (street racing), "changeLaneContinuously" (continuous lane change), "SSharpDriving" (slalom driving), "abandonedObject" (thrown object), "pedestrian" (pedestrian pre-alarm), "notSlowZebraCrossing", "overThermal", "turnRightStop" (large-sized vehicle not slowing down when turning right), "enterNotYieldToInRoundabout" (not yield at roundabout intersection), "rampNotYieldToMainRoad" (vehicle from ramp not yielding to vehicle on main road), "notYieldToRightRoad" (not yielding to vehicle from right side), "TruckOccupyOvertakingLane" (large truck occupying overtaking lane), "TruckTurnRightStop" (large truck not slowing down when turning right), "banStraight" (going straight prohibited), "emergencyLane" (occupy emergency lane), "smoke", "conflagration", "VehicleSpeedDrop", "animalsOccupy", "suonaDet" (whistle), "roarDet*/
    },
    "noPlateCaptureEnable": {
        /*ro, opt, object, whether to enable no plate vehicle capture*/
        "@opt": [true, false]
        /*ro, opt, array, options, subType:bool*/
    },
    "eventPriority": {
        /*ro, opt, object, event capture priority, desc:value range 0 to 10; the higher the value, the higher the priority; when the value is 0, this node will not have priority*/
        "@min": 0,
        /*ro, req, int, the minimum value*/
        "@max": 10
        /*ro, req, int, the maximum value*/
    },
    "TimeBlockList": {
        /*ro, opt, object, list of time periods*/
        "@size": 7,
        /*ro, req, int, the maximum number supported in a single capture*/
    },
    "TimeBlock": {
        /*ro, opt, object, time period*/
        "dayOfWeek": {
            /*ro, req, object, day of a week, desc:Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday*/
            "@min": 1,
            /*ro, req, int, the minimum value*/
            "@max": 7
            /*ro, req, int, the maximum value*/
        },
        "TimeRangeList": {
            /*ro, opt, object, list of upload intervals*/
            "@size": 4,
            /*ro, req, int, the maximum number supported in a single capture*/
            "TimeRange": {
                /*ro, opt, object, time interval*/
                "beginTime": "00:00:00+08:00",
                /*ro, req, time, start time*/
                "endTime": "00:00:00+08:00"
                /*ro, req, time, end time*/
            }
        }
    },
    "RelatedLanList": {
        /*ro, opt, object, list of linked lane information*/
        "@size": 6,
        /*ro, opt, int, size*/
    },
    "RelatedLan": {
        /*ro, opt, object, linked lane information*/
        "lanID": {
            /*ro, opt, object, whether to enable linked lane No.*/
            "@min": 1,
            /*ro, req, int, the minimum value*/
            "@max": 6
            /*ro, req, int, the maximum value*/
        }
    }
}
}

```

12.5.1.13 Get the arming parameters for traffic violation capture

Request URL

GET /ISAPI/ITC/illegalSchedules?format=json

Query Parameter

None

Request Message

None

Response Message

```
{  
    "eventPriorityEnable": true,  
    /*ro, req, bool, whether to enable violation priority*/  
    "IllegalScheduleList": [  
        /*ro, req, array, arming time of traffic violation capture, subType:object, desc:this node is only available for ANPR violation information, which is  
        related to the node illegalInfo*/  
        {  
            "Schedule": {  
                /*ro, opt, object, arming schedules*/  
                "eventType": "post",  
                /*ro, req, enum, event type, subType:string, desc:"post" (checkpoint), "driveLine" (driving on the lane line), "reverse" (wrong-way  
                driving), "banSign" (prohibition violation), "overSpeed" (driving in overspeed), "lowSpeed" (driving in low speed), "changeLane" (illegal lane change),  
                "safeBelt", "uphone" (making a phone call when driving), "gasser" (queue jumping), "congestion", "helmet" (driver is wearing a helmet), "twoWheelPassenger"  
                (manned non-motor vehicle), "nonMotorExist" (non-motor vehicle on motor vehicle lane), "intersectionCongest" (overstay at intersection), "nonDriveWay"  
                (motor vehicle on non-motor vehicle lane), "redLight" (red light running), "intersectionStop" (stop vehicle at intersection), "direction" (driving against  
                direction guidance), "greenLightStop (stop vehicle when green light), "illegalTurn", "accident", "runGreenLight" (running green light in congestion),  
                "leftNotYieldStraight" (left turn not yield to straight), "rightNotYieldLeft" (right turn not yield to left turn), "uTurnNotYieldGoingStraight",  
                "leftSideDrivingAfterLeftTurn", "failYieldPed" (not yield to pedestrian), "illegalParking" (illegal parking), "nonMotorRedLight" (non-motor vehicle red  
                light running), "nonMotorCrossing" (non-motor vehicle crossing line), "nonMotorOccupyZebraCrossing" (non-motor vehicle crossing zebra line), "roadBlock",  
                "construction", "notKeepDistance" (not keeping a safe distance), "overtakeRightSide" (overtaking on the right), "dragRacing" (street racing),  
                "changeLaneContinuously" (continuous lane change), "SSharpDriving" (slalom driving), "abandonedObject" (thrown object), "pedestrian" (pedestrian pre-alarm),  
                "notSlowZebraCrossing", "overThermal", "turnRightStop" (large-sized vehicle not slowing down when turning right), "enterNotYieldToInRoundabout" (not yield  
                at roundabout intersection), "rampNotYieldToMainRoad" (vehicle from ramp not yielding to vehicle on main road), "notYieldToRightRoad" (not yielding to  
                vehicle from right side), "TruckOccupyOvertakingLane" (large truck occupying overtaking lane), "TruckTurnRightStop" (large truck not slowing down when  
                turning right), "banStraight" (going straight prohibited), "emergencyLane" (occupy emergency lane), "smoke", "conflagration", "VehicleSpeedDrop",  
                "animalsOccupy", "suonaDet" (whistle), "roarDet"*/  
                "noPlateCaptureEnable": true,  
                /*ro, opt, bool, whether to enable no plate vehicle capture*/  
                "eventPriority": 0,  
                /*ro, opt, int, event capture priority, range:[0,10], desc:value ranges from 0 to 10; the higher the value, the higher the priority; when  
                the value is 0, this event will not have priority*/  
                "TimeBlockList": [  
                    /*ro, opt, array, list of time periods, subType:object, range:[0,7]*/  
                    {  
                        "TimeBlock": {  
                            /*ro, opt, object, time period*/  
                            "dayOfTheWeek": 1,  
                            /*ro, req, enum, day of a week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6  
                            (Saturday), 7 (Sunday)*/  
                            "TimeRangeList": [  
                                /*ro, opt, array, list of time intervals, subType:object, range:[0,4], desc:4 time periods can be configured for a day*/  
                                {  
                                    "TimeRange": {  
                                        /*ro, opt, object, time interval*/  
                                        "beginTime": "00:00:00+08:00",  
                                        /*ro, req, time, start time*/  
                                        "endTime": "00:00:00+08:00"  
                                        /*ro, req, time, end time*/  
                                    }  
                                }  
                            ]  
                        }  
                    }  
                ],  
                "RelatedLaneList": [  
                    /*ro, opt, array, list of linked lane information, subType:object, range:[0,6]*/  
                    {  
                        "RelatedLane": {  
                            /*ro, opt, object, linked lane information*/  
                            "lanID": 1  
                            /*ro, opt, int, whether to enable linked lane No.*/  
                        }  
                    }  
                ]  
            }  
        }  
    ]  
}
```

12.5.1.14 Set the arming parameters for traffic violations capture

Request URL

PUT /ISAPI/ITC/illegalSchedules?format=json

Query Parameter

None

Request Message

```
{
    "eventPriorityEnable": true,
    /*req, bool, whether to enable violation priority*/
    "IllegalScheduleList": [
        /*req, array, arming time of traffic violations capture, subType:object, desc:this node is only available for ANPR violation information, related node is illegalInfo*/
        {
            "Schedule": {
                /*opt, object, arming schedules*/
                "eventType": "post",
                /*req, enum, event type, subType:string, desc:"post" (checkpoint), "driveLine" (driving on the lane line), "reverse" (wrong-way driving),
                "banSign" (prohibition violation), "overSpeed" (driving in overspeed), "lowSpeed" (driving in low speed), "changeLane" (illegal lane change), "safeBelt",
                "uphone" (making a phone call when driving), "gasser" (queue jumping), "congestion", "helmet" (driver is wearing a helmet), "twoWheelPassenger" (manned non-motor vehicle), "nonMotorExist" (non-motor vehicle on motor vehicle lane), "intersectionCongest" (overstay at intersection), "nonDriveWay" (motor vehicle on non-motor vehicle lane), "redLight" (red light running), "intersectionStop" (stop vehicle at intersection), "direction" (driving against direction guidance), "greenlightStop" (stop vehicle when green light), "illegalTurn", "accident", "runGreenLight" (running green light in congestion),
                "leftNotYieldStraight" (left turn not yield to straight), "rightNotYieldLeft" (right turn not yield to left turn), "uTurnNotYieldGoingStraight",
                "leftSideDrivingAfterLeftTurn", "failYieldPed" (not yield to pedestrian), "illegalParking" (illegal parking), "nonMotorRedLight" (non-motor vehicle red light running), "nonMotorCrossing" (non-motor vehicle crossing line), "nonMotorOccupyZebraCrossing" (non-motor vehicle crossing zebra line), "roadBlock",
                "construction", "notKeepDistance" (not keeping a safe distance), "overtakeRightSide" (overtaking on the right), "dragRacing" (street racing),
                "changeLaneContinuously" (continuous lane change), "SSharpDriving" (slalom driving), "abandonedObject" (thrown object), "pedestrian" (pedestrian pre-alarm),
                "notSlowZebraCrossing", "overThermal", "turnRightStop" (large-sized vehicle not slowing down when turning right), "enterNotYieldToInRoundabout" (not yield at roundabout intersection), "rampNotYieldToMainRoad" (vehicle from ramp not yielding to vehicle on main road), "notYieldToRightRoad" (not yielding to vehicle from right side), "TruckOccupyOvertakingLane" (large truck occupying overtaking lane), "TruckTurnRightStop" (large truck not slowing down when turning right), "banStraight" (going straight prohibited), "emergencyLane" (occupy emergency lane), "smoke", "conflagration", "VehicleSpeedDrop",
                "animalsOccupy", "suonaDet" (whistle), "roarDet"*/
                "noPlateCaptureEnable": true,
                /*opt, bool, whether to enable no plate vehicle capture*/
                "eventPriority": 0,
                /*opt, int, event capture priority, range:[0,10], desc:value range 0 to 10; the higher the value, the higher the priority; when the value is 0, this node will not have priority*/
            }
            "TimeBlockList": [
                /*opt, array, list of time periods, subType:object, range:[0,7]*/
                {
                    "TimeBlock": {
                        /*opt, object, time period*/
                        "dayOfWeek": 1,
                        /*req, enum, day of a week, subType:int, desc:1 (Monday), 2 (Tuesday), 3 (Wednesday), 4 (Thursday), 5 (Friday), 6 (Saturday), 7
                        (Sunday)*/
                        "TimeRangeList": [
                            /*opt, array, list of time intervals, subType:object, range:[0,4], desc:4 time periods can be configured for a day*/
                            {
                                "TimeRange": {
                                    /*opt, object, time interval*/
                                    "beginTime": "00:00:00+08:00",
                                    /*req, time, start time*/
                                    "endTime": "00:00:00+08:00"
                                    /*req, time, end time*/
                                }
                            }
                        ]
                    }
                }
            ],
            "RelatedLaneList": [
                /*opt, array, list of linked lane information, subType:object, range:[0,6]*/
                {
                    "RelatedLane": {
                        /*opt, object, linked lane information*/
                        "laneID": 1
                        /*opt, int, whether to enable linked lane No.*/
                    }
                }
            ]
        }
    ]
}
```

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 node is required when the value of statusCode is not 1*/
}
}
```

12.5.1.15 Get ANPR configuration capability

Request URL

GET /ISAPI/ITC/plateRecognitionParam/capabilities?channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Camera ID (this optional parameter is supported when isSupportChannelPlateRecognitionParam is "true" returned from GET /ISAPI/ITC/capabilities).

Request Message

None

Response Message

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

<PlateRecognitionParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, ANPR parameters, attr:version{req, string, protocolVersion}-->
    <isSupportDefaultCHN>
        <!--ro, opt, string, province name-->test
    </isSupportDefaultCHN>
    <isSupportfrontPlateReco>
        <!--ro, opt, bool, whether to support front license plate recognition-->true
    </isSupportfrontPlateReco>
    <isSupportrearPlateReco>
        <!--ro, opt, bool, whether to support rear license plate recognition-->true
    </isSupportrearPlateReco>
    <isSupportsmallPlateReco>
        <!--ro, opt, bool, whether to support small license plate recognition-->true
    </isSupportsmallPlateReco>
    <isSupportlargePlateReco>
        <!--ro, opt, bool, whether to support large license plate recognition-->true
    </isSupportlargePlateReco>
    <isSupportfarmVehicle>
        <!--ro, opt, bool, whether to support agricultural vehicle recognition-->true
    </isSupportfarmVehicle>
    <isSupportmotor>
        <!--ro, opt, bool, whether to support motorcycle recognition-->true
    </isSupportmotor>
    <isSupportfuzzyDisc>
        <!--ro, opt, bool, whether to support fuzzy recognition-->true
    </isSupportfuzzyDisc>
    <isSupportmicroPlateReg>
        <!--ro, opt, bool, whether to support tiny license plate recognition-->true
    </isSupportmicroPlateReg>
    <isSupportcAPlateRegEnabled>
        <!--ro, opt, bool, whether to support civil aviation license plate recognition-->true
    </isSupportcAPlateRegEnabled>
    <isSupporttiltPlateRegEnabled>
        <!--ro, opt, bool, whether to support tilted license plate recognition-->true
    </isSupporttiltPlateRegEnabled>
    <isSupportsuperPlateRecoEnabled>
        <!--ro, opt, bool, whether to support oversized license plate recognition-->true
    </isSupportsuperPlateRecoEnabled>
    <isSupportEmbassyPlateRecoEnabled>
        <!--ro, opt, bool, whether to support embassy license plate recognition-->true
    </isSupportEmbassyPlateRecoEnabled>
    <isSupportfrontAndRearReg>
        <!--ro, opt, bool, whether to enable front license plate recognition and rear license plate recognition-->true
    </isSupportfrontAndRearReg>
    <nation opt="EU,ER,EUandCIS,ME,APAC,AFandAM,All">
        <!--ro, opt, enum, country/region, subType:string, attr:opt{req, string}, desc:"ER" (Russian-speaking region), "EU" (Europe), "ME" (Middle East), "APAC" (Asia-Pacific region), "AFandAM" (Africa and America), "All" (all regions), "EUandCIS" (Europe and Russia)-->ER
    </nation>

```

```

</nation>
<countryIndex opt="1,2,3,4,5,6,7,8,12,14,17,18,19,20,23,39,44,55">
  <!--ro, opt, int, country/region index, attr:opt{req, string}-->0
</countryIndex>
<nonMotorPlateCityRecoEnabled opt="true,false">
  <!--ro, opt, bool, whether to enable belonged city recognition of non-motor vehicle license plate, attr:opt{req, string}-->true
</nonMotorPlateCityRecoEnabled>
<city>
  <!--ro, opt, string, belonged city of license plate-->test
</city>
<fakePlateFilterEnabled opt="true,false">
  <!--ro, opt, bool, attr:opt{req, string}-->false
</fakePlateFilterEnabled>
<citySeparatorEnabled opt="true,false">
  <!--ro, opt, bool, attr:opt{req, string}-->true
</citySeparatorEnabled>
<fakePlateDetectionUploadEnabled opt="true,false">
  <!--ro, opt, bool, dep:and, ${.PlateRecognitionParam.fakePlateFilterEnabled,eq,true}, attr:opt{req, string}-->false
</fakePlateDetectionUploadEnabled>
<provinceThresholds min="0" max="100">
  <!--ro, opt, int, range:[0,100], attr:min{req, int},max{req, int}-->0
</provinceThresholds>
<secondLicensePlateRegEnabled opt="true,false">
  <!--ro, opt, bool, attr:opt{req, string}-->true
</secondLicensePlateRegEnabled>
<yellowGreenPlateRecoRule opt="yellow,green,yellowGreen,yellowGreenBicolor">
  <!--ro, opt, enum, subType:string, attr:opt{req, string}-->yellow
</yellowGreenPlateRecoRule>
<nonStandardPlateIllegalHandleRules opt="none,filter,wipePlate">
  <!--ro, opt, enum, subType:string, attr:opt{req, string}-->none
</nonStandardPlateIllegalHandleRules>
<nonStandardPlatePostHandleRules opt="none,filter,wipePlate">
  <!--ro, opt, enum, subType:string, attr:opt{req, string}-->none
</nonStandardPlatePostHandleRules>
<nonStandardPlateThresholds min="0" max="100">
  <!--ro, opt, int, range:[0,100], attr:min{req, int},max{req, int}-->0
</nonStandardPlateThresholds>
<nonStandardPlateMinCorrectFactor min="0" max="100">
  <!--ro, opt, int, range:[0,100], attr:min{req, int},max{req, int}-->0
</nonStandardPlateMinCorrectFactor>
<nonStandardPlateMaxCorrectFactor min="0" max="100">
  <!--ro, opt, int, range:[0,100], attr:min{req, int},max{req, int}-->0
</nonStandardPlateMaxCorrectFactor>
<nonStandardVehicleCorrectFactor min="0" max="100">
  <!--ro, opt, int, range:[0,100], attr:min{req, int},max{req, int}-->0
</nonStandardVehicleCorrectFactor>
<nonStandardPlateRegEnabled opt="true,false">
  <!--ro, opt, bool, attr:opt{req, string}-->true
</nonStandardPlateRegEnabled>
<threePlateRecoEnabled opt="true,false">
  <!--ro, opt, bool, attr:opt{req, string}-->true
</threePlateRecoEnabled>
<newEnergyGreenPlateRecoRule opt="green,newEnergyGreen,gradationGreen">
  <!--ro, opt, enum, subType:string, attr:opt{req, string}-->green
</newEnergyGreenPlateRecoRule>
<customPlateRule>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <customPlateColorEnabled opt="true,false">
    <!--ro, opt, bool, attr:opt{req, string}-->true
  </customPlateColorEnabled>
  <customPlateColor opt="black,blue,green,white,yellow">
    <!--ro, opt, enum, subType:string, attr:opt{req, string}-->black
  </customPlateColor>
  <wildcardRuleList size="10">
    <!--ro, opt, object, attr:size{req, int}-->
    <wildcardRule>
      <!--ro, opt, object-->
      <enabled opt="true,false">
        <!--ro, req, bool, attr:opt{req, string}-->true
      </enabled>
      <wildcard min="0" max="16">
        <!--ro, opt, string, range:[0,16], attr:min{req, int},max{req, int}-->蓝园A?!!*
      </wildcard>
    </wildcardRule>
  </wildcardRuleList>
</customPlateRule>
<plateMappingRule>
  <!--ro, opt, object-->
  <enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
  </enabled>
  <plateMappingList size="64">
    <!--ro, opt, array, subType:object, range:[0,64], attr:size{req, int}-->
    <plateMapping>
      <!--ro, opt, object-->
      <detectPlateNo min="0" max="16">
        <!--ro, opt, string, range:[0,16], attr:min{req, int},max{req, int}-->浙A12348
      </detectPlateNo>
      <replacePlateNo min="0" max="16">
        <!--ro, opt, string, range:[0,16], attr:min{req, int},max{req, int}-->浙A1234B
      </replacePlateNo>
    </plateMapping>
  </plateMappingList>
</plateMappingRule>

```

```

<!-->
</plateMapping>
</plateMappingList>
</plateMappingRule>
<primaryAndSecondaryPlate>
<!--ro, opt, array, subType:object-->
<enabled>
<!--ro, opt, array, subType:object-->
<mainlandPlateEnable opt="true,false">
<!--ro, opt, bool, attr:opt{req, string}-->true
</mainlandPlateEnable>
<hongKongPlateEnable opt="true,false">
<!--ro, opt, bool, attr:opt{req, string}-->true
</hongKongPlateEnable>
<maCaoPlateEnable opt="true,false">
<!--ro, opt, bool, attr:opt{req, string}-->true
</maCaoPlateEnable>
</enabled>
<plateRegionType>
<!--ro, opt, array, subType:object-->
<mainAreaOption opt="mainland,hongKong,maCao">
<!--ro, opt, enum, subType:string, attr:opt{req, string}-->mainland
</mainAreaOption>
<secondAreaOption opt="mainland,hongKong,maCao">
<!--ro, opt, enum, subType:string, attr:opt{req, string}-->hongKong
</secondAreaOption>
</plateRegionType>
</primaryAndSecondaryPlate>
</PlateRecognitionParam>

```

12.5.1.16 Set ANPR parameters

Request URL

PUT /ISAPI/ITC/plateRecognitionParam?channelID= <channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<PlateRecognitionParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, ANPR parameters, attr:version{req, string, protocolVersion}-->
  <defaultCHN>
    <!--req, string, province/state abbreviation-->test
  </defaultCHN>
  <frontPlateRecoEnabled>
    <!--opt, bool, front license plate recognition, desc:front license plate recognition-->true
  </frontPlateRecoEnabled>
  <rearPlateRecoEnabled>
    <!--opt, bool, rear license plate recognition-->true
  </rearPlateRecoEnabled>
  <smallPlateRecoEnabled>
    <!--opt, bool, small license plate recognition-->true
  </smallPlateRecoEnabled>
  <largePlateRecoEnabled>
    <!--opt, bool, large license plate recognition-->true
  </largePlateRecoEnabled>
  <farmVehicleEnabled>
    <!--opt, bool, agricultural vehicle recognition-->true
  </farmVehicleEnabled>
  <motorEnabled>
    <!--opt, bool, motorcycle recognition-->true
  </motorEnabled>
  <fuzzyDiscEnabled>
    <!--opt, bool, fuzzy recognition-->true
  </fuzzyDiscEnabled>
  <microPlateRegEnabled>
    <!--opt, bool, tiny license plate recognition-->true
  </microPlateRegEnabled>
  <cAPlateRegEnabled>
    <!--opt, bool, civil aviation license plate recognition-->true
  </cAPlateRegEnabled>
  <tiltPlateRegEnabled>
    <!--opt, bool, tilted license plate recognition-->true
  </tiltPlateRegEnabled>
  <superPlateRecoEnabled>
    <!--opt, bool, oversized license plate recognition-->true
  </superPlateRecoEnabled>
  <embassyPlateRecoEnabled>
    <!--opt, bool, embassy license plate recognition-->true
  </embassyPlateRecoEnabled>
  <nation>
    <!--opt, enum, country/region, subType:string, desc:"ER" (Russian-speaking region), "EU" (Europe), , "EUandCIS" (Europe and Russia), "ME" (Middle East), "APAC" (Asia-Pacific region), "AFandAM" (Africa and America), "ALL" (all regions)-->ER
  </nation>
  <countryIndex>
    <!--opt, int, country/region index, desc:refer to the <CRIndex> data field-->0
  </countryIndex>
  <nonMotorPlateCityRecoEnabled>
    <!--opt, bool, whether to enable belonged city recognition of non-motor vehicle license plate-->true
  </nonMotorPlateCityRecoEnabled>
  <city>
    <!--opt, string, License Plate Belonged City-->test
  </city>
  <fakePlateFilterEnabled>
    <!--opt, bool, filter fake license plate-->false
  </fakePlateFilterEnabled>
</PlateRecognitionParam>

```

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:detailed information of custom error returned by device applications, used for fast debugging-->badXmlFormat
  </description>
</ResponseStatus>

```

12.5.1.17 Get ANPR parameters

Request URL

GET /ISAPI/ITC/plateRecognitionParam?channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Camera ID (this optional parameter is supported when isSupportChannelPlateRecognitionParam is "true" returned from GET /ISAPI/ITC/capabilities).

Request Message

None

Response Message

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

<PlateRecognitionParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, ANPR parameters, attr:version{req, string, protocolVersion}-->
    <defaultCHN>
        <!--ro, opt, string, province name-->test
    </defaultCHN>
    <frontPlateRecoEnabled>
        <!--ro, opt, bool, front license plate recognition, desc:license plate recognition principles: when <isSupportfrontAndRearReg> is true and both front and rear license plate recognition are supported, "true" is returned and the front license plate recognition parameters will be modified by default; when <isSupportfrontAndRearReg> is false and either front or rear license plate recognition is supported, "false" is returned-->true
    </frontPlateRecoEnabled>
    <rearPlateRecoEnabled>
        <!--ro, opt, bool, rear license plate recognition-->true
    </rearPlateRecoEnabled>
    <smallPlateRecoEnabled>
        <!--ro, opt, bool, small license plate recognition-->true
    </smallPlateRecoEnabled>
    <largePlateRecoEnabled>
        <!--ro, opt, bool, large license plate recognition-->true
    </largePlateRecoEnabled>
    <farmVehicleEnabled>
        <!--ro, opt, bool, agricultural vehicle recognition-->true
    </farmVehicleEnabled>
    <motorEnabled>
        <!--ro, opt, bool, motorcycle recognition-->true
    </motorEnabled>
    <fuzzyDiscEnabled>
        <!--ro, opt, bool, fuzzy recognition-->true
    </fuzzyDiscEnabled>
    <microPlateRegEnabled>
        <!--ro, opt, bool, tiny license plate recognition-->true
    </microPlateRegEnabled>
    <cAPlateRegEnabled>
        <!--ro, opt, bool, civil aviation license plate recognition-->true
    </cAPlateRegEnabled>
    <tiltPlateRegEnabled>
        <!--ro, opt, bool, tilted license plate recognition-->true
    </tiltPlateRegEnabled>
    <superPlateRecoEnabled>
        <!--ro, opt, bool, large license plate recognition-->true
    </superPlateRecoEnabled>
    <embassyPlateRecoEnabled>
        <!--ro, opt, bool, embassy license plate recognition-->true
    </embassyPlateRecoEnabled>
    <nation>
        <!--ro, opt, enum, country/region, subType:string, desc:"ER" (Russian-speaking region), "EU" (Europe), , "EuandCIS" (Europe and Russia), "ME" (Middle East), "APAC" (Asia-Pacific region), "AFandAM" (Africa and America), "ALL" (all regions)-->ER
    </nation>
    <countryIndex>
        <!--ro, opt, int, country/region index, desc:refer to the <CRIndex> data field-->0
    </countryIndex>
    <nonMotorPlateCityRecoEnabled>
        <!--ro, opt, bool, whether to enable belonged city recognition of non-motor vehicle license plate-->true
    </nonMotorPlateCityRecoEnabled>
    <city>
        <!--ro, opt, string, license plate belonged city-->test
    </city>
    <fakePlateFilterEnabled>
        <!--ro, opt, bool, filter fake license plate, desc:filter fake license plate-->false
    </fakePlateFilterEnabled>
    <citySeparatorEnabled>
        <!--ro, opt, bool-->true
    </citySeparatorEnabled>
    <fakePlateDetectionUploadEnabled>
        <!--ro, opt, bool, den and ${PlateRecognitionParam.fakePlateFilterEnabled.en:true}-->false
    </fakePlateDetectionUploadEnabled>

```

```
.. ro, opt, bytes, deprecated, for license recognition or any other reason, license plate detection is disabled
```

```
</fakePlateDetectionUploadEnabled>
```

```
<provinceThresholds>
```

```
  <!--ro, opt, int, range:[0,100]-->0
```

```
</provinceThresholds>
```

```
<secondLicensePlateRegEnabled>
```

```
  <!--ro, opt, bool-->true
```

```
</secondLicensePlateRegEnabled>
```

```
<yellowGreenPlateRecoRule>
```

```
  <!--ro, opt, enum, subType:string-->yellow
```

```
</yellowGreenPlateRecoRule>
```

```
<nonStandardPlateIllegalHandleRules>
```

```
  <!--ro, opt, enum, subType:string-->none
```

```
</nonStandardPlateIllegalHandleRules>
```

```
<nonStandardPlatePostHandleRules>
```

```
  <!--ro, opt, enum, subType:string-->none
```

```
</nonStandardPlatePostHandleRules>
```

```
<nonStandardPlateThresholds>
```

```
  <!--ro, opt, int, range:[0,100]-->0
```

```
</nonStandardPlateThresholds>
```

```
<nonStandardPlateMinCorrectFactor>
```

```
  <!--ro, opt, int, range:[0,100]-->0
```

```
</nonStandardPlateMinCorrectFactor>
```

```
<nonStandardPlateMaxCorrectFactor>
```

```
  <!--ro, opt, int, range:[0,100]-->0
```

```
</nonStandardPlateMaxCorrectFactor>
```

```
<nonStandardVehicleCorrectFactor>
```

```
  <!--ro, opt, int, range:[0,100]-->0
```

```
</nonStandardVehicleCorrectFactor>
```

```
<nonStandardPlateRegEnabled>
```

```
  <!--ro, opt, bool-->true
```

```
</nonStandardPlateRegEnabled>
```

```
<threePlateRecoEnabled>
```

```
  <!--ro, opt, bool-->true
```

```
</threePlateRecoEnabled>
```

```
<newEnergyGreenPlateRecoRule>
```

```
  <!--ro, opt, enum, subType:string-->green
```

```
</newEnergyGreenPlateRecoRule>
```

```
<customPlateRule>
```

```
  <!--ro, opt, object-->
```

```
  <enabled>
```

```
    <!--ro, req, bool-->true
```

```
  </enabled>
```

```
  <customPlateColorEnabled>
```

```
    <!--ro, opt, bool-->true
```

```
  </customPlateColorEnabled>
```

```
  <customPlateColor>
```

```
    <!--ro, opt, enum, subType:string-->black
```

```
  </customPlateColor>
```

```
<wildcardRuleList>
```

```
  <!--ro, opt, array, subType:object, range:[0,10]-->
```

```
  <wildcardRule>
```

```
    <!--ro, opt, object-->
```

```
    <enabled>
```

```
      <!--ro, req, bool-->true
```

```
    </enabled>
```

```
    <wildcard>
```

```
      <!--ro, opt, string, range:[0,16]-->蓝圆A?!!**
```

```
    </wildcard>
```

```
  </wildcardRule>
```

```
</wildcardRuleList>
```

```
</customPlateRule>
```

```
<plateMappingRule>
```

```
  <!--ro, opt, object-->
```

```
  <enabled>
```

```
    <!--ro, req, bool-->true
```

```
  </enabled>
```

```
<plateMappingList>
```

```
  <!--ro, opt, array, subType:object, range:[0,64]-->
```

```
  <plateMapping>
```

```
    <!--ro, opt, object-->
```

```
    <detectPlateNo>
```

```
      <!--ro, opt, string, range:[0,16]-->浙A12348
```

```
    </detectPlateNo>
```

```
    <replacePlateNo>
```

```
      <!--ro, opt, string, range:[0,16]-->浙A12348
```

```
    </replacePlateNo>
```

```
  </plateMapping>
```

```
</plateMappingList>
```

```
</plateMappingRule>
```

```
<primaryAndSecondaryPlate>
```

```
  <!--ro, opt, array, subType:object-->
```

```
  <enabled>
```

```
    <!--ro, opt, array, subType:object-->
```

```
    <mainlandPlateEnable>
```

```
      <!--ro, opt, bool-->true
```

```
    </mainlandPlateEnable>
```

```
    <hongKongPlateEnable>
```

```
      <!--ro, opt, bool-->true
```

```
    </hongKongPlateEnable>
```

```
    <maCaoPlateEnable>
```

```
      <!--ro, opt, bool-->true
```

```
    </maCaoPlateEnable>
```

```
  </enabled>
```

```

<plateRegionType>
  <!--ro, opt, array, subType:object-->
  <mainAreaOption>
    <!--ro, opt, enum, subType:string-->mainland
  </mainAreaOption>
  <secondAreaOption>
    <!--ro, opt, enum, subType:string-->hongKong
  </secondAreaOption>
</plateRegionType>
</primaryAndSecondaryPlate>
</PlateRecognitionParam>

```

12.5.1.18 Get trigger mode capability of traffic cameras

Request URL

GET /ISAPI/ITC/TriggerMode/capabilities

Query Parameter

None

Request Message

None

Response Message

```

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

<TriggerMode xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, trigger mode, attr:version{req, string, protocolVersion}-->
  <usedTriggerMode>
    opt="postIOSpeed,postSingleIO,postRS485,postRadar,postVTCoil,postHVT,postMPR,epoliceRS485,postPoliceRS485,videoEpolice,TPS,videoMonitor,postNoComityPed,postRedLightPed,postSideRadarSpeed,parkingIncident">
      <!--ro, req, enum, trigger mode supported by the device, different devices support different trigger modes, and a device may support multiple trigger modes, subType:string, attr:opt{req, string}, desc:"postMobile", "postDetect", "TPS", "trafficIncident", "trafficIncidentTPS", "roadWarning", "postIOSpeed", "postSingleIO", "postRS485", "postRadar", "postVTCoil", "postHVT", "postMPR", "epoliceRS485", "postPoliceRS485", "videoEpolice", "videoMonitor", "postNoComityPed", "postRedlightPed", "postSideRadarSpeed", "parkingIncident", "vehicleDetect", "boxNumDetection", "parkSpeedDetection"-->postMobile
    </usedTriggerMode>
    <isSupportChannelConfig>
      <!--ro, opt, bool, whether it supports channel configuration, desc:related URL: /ISAPI/ITC/channels/<ID>/TriggerMode/capabilities-->true
    </isSupportChannelConfig>
    <isSupportChannelSwitchStatus>
      <!--ro, opt, bool, whether it supports switching trigger mode status by channel No., desc:true (the optional node <channelID> in /ISAPI/ITC/TriggerMode/switchStatus?format=json&channelID=<channelID> is supported)-->true
    </isSupportChannelSwitchStatus>
  </TriggerMode>

```

12.5.1.19 Get parameters of intersection violation system vehicle detector

Request URL

GET /ISAPI/ITC/TriggerMode/epoliceRS485

Query Parameter

None

Request Message

None

Response Message

```

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

<EpoliceRS485 xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, parameters of intersection violation system vehicle detector, attr:version{req, string, protocolVersion}-->
  <relatedLaneCount>
    <!--ro, req, int, number of supported Lanes, range:[1,4]-->0
  </relatedLaneCount>
  <trafficLightSource>
    <!--ro, req, enum, traffic Light source, subType:int, desc:0-vehicle detector, 1-traffic Light detector.-->0
  </trafficLightSource>
  <EpoliceRS485LaneList>
    <!--ro, opt, array, Lane List, subType:object-->
    <EpoliceRS485Lane>
      <!--ro, opt, object, Lane-->
      <laneId>
        <!--ro, req, int, Lane ID, range:[1,4]-->0
      </laneId>
      <laneDirectionType>
        <!--ro, req, enum, Lane direction, subType:int, desc:0 (unknown),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
      </laneDirectionType>
      <relatedDriveWay>
        <!--ro, req, int, Linked Lane No., range:[1,16]-->0
      </relatedDriveWay>
      <redLightDriveWay>
        <!--ro, req, int, red Light Lane No., range:[1,16]-->0
      </redLightDriveWay>
      <yellowLightDriveWay>
        <!--ro, req, int, yellow Light Lane No., range:[1,16]-->0
      </yellowLightDriveWay>
      <osDdriveWay>
        <!--ro, req, int, overlay Lane No., range:[1,99]-->0
      </osDdriveWay>
      <distance>
        <!--ro, req, int, coil distance, range:[0,2000], unit:cm-->0
      </distance>
      <recordEnable>
        <!--ro, req, bool, whether to enable recording-->true
      </recordEnable>
      <recordType>
        <!--ro, req, enum, recording type, subType:string, desc:"preRecord" (pre-record), "delayRecord" (post-record)-->preRecord
      </recordType>
      <preRecordTime>
        <!--ro, req, int, pre-record duration, range:[0,100]-->0
      </preRecordTime>
      <recordDelayTime>
        <!--ro, req, int, post-record duration, range:[0,100]-->0
      </recordDelayTime>
      <recordTimeOut>
        <!--ro, req, int, timeout duration, range:[0,100]-->0
      </recordTimeOut>
      <serialProtocol>
        <!--ro, req, enum, serial port protocol, subType:int, desc:0-vehicle detector, 1-OEM, 2-other vehicle detector-->0
      </serialProtocol>
      <runRedLightCaptureLogic>
        <!--ro, req, enum, red Light running capture Logic, subType:string, desc:"singelPro" (single_enter 1_exit 1_delay 1_1), "singlePro2" (singelPro), "doublePro1" (double_enter 1_exit 1_exit 2_1), "doublePro2" (double_enter 1_exit 2_delay 2_1), "doublePro3" (double_enter 2_exit 1_delay 2_1), "doublePro4" (double/three_enter 2_exit 2_delay 2_1), "doublePro5" (double/three_enter 2_exit 2_delay 2_2), "doublePro6" (double/three_enter 1_exit 2_delay 2_2), "doublePro7" (double_enter 1_exit 1_delay 2_1), "doublePro8" (double_enter 1_exit 1_delay 2_2), "other"-->singelPro
      </runRedLightCaptureLogic>
      <oppositeDirectionCaptureLogic>
        <!--ro, req, enum, Wrong-Way Driving Capture Logic, subType:string, desc:"noCap" (do not capture), "reversePro1" (double/three_enter_enter_delay), "reversePro2" (double/three_enter_delay)-->noCap
      </oppositeDirectionCaptureLogic>
      <intervalType>
        <!--ro, req, enum, Burst Interval Type, subType:string, desc:"time" (by time), "distance" (by distance)-->time
      </intervalType>
      <interval>
        <!--ro, req, int, interval, range:[0,2000]-->0
      </interval>
      <iOOutList>
        <!--ro, opt, array, List of I/O output ports, subType:object-->
        <iOOut>
          <!--ro, req, object, I/O output port-->
          <id>
            <!--ro, req, int, Linked I/O output No., range:[1,10]-->0
          </id>
          <enabled>
            <!--ro, req, bool, whether to Link the I/O output-->true
          </enabled>
        </iOOut>
      </iOOutList>
      <flashMode>
        <!--ro, req, enum, flashing mode, subType:string, desc:"together" (simultaneously), "alternation" (alternatively)-->together
      </flashMode>
    </EpoliceRS485Lane>
  </EpoliceRS485LaneList>
</EpoliceRS485>

```

12.5.1.20 Set parameters of intersection violation system vehicle detector

Request URL

PUT /ISAPI/ITC/TriggerMode/epoliceRS485

Query Parameter

None

Request Message

```

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

<EpolicerRS485 xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, parameters of intersection violation system vehicle detector, attr:version{req, string, protocolVersion}-->
  <relatedLaneCount>
    <!--req, int, number of supported Lanes, range:[1,4]-->0
  </relatedLaneCount>
  <trafficLightSource>
    <!--req, enum, traffic light source, subType:int, desc:0-vehicle detector, 1-traffic light detector.-->0
  </trafficLightSource>
  <EpolicerRS485LaneList>
    <!--opt, array, Lane List, subType:object-->
    <EpolicerRS485Lane>
      <!--opt, object, Lane-->
      <laneId>
        <!--req, int, Lane ID, range:[1,4]-->0
      </laneId>
      <laneDirectionType>
        <!--req, enum, Lane direction, subType:int, desc:0 (unknown),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
      </laneDirectionType>
      <relatedDriveWay>
        <!--req, int, Linked Lane No., range:[1,16]-->0
      </relatedDriveWay>
      <redLightDriveWay>
        <!--req, int, red Light Lane No., range:[1,16]-->0
      </redLightDriveWay>
      <yellowLightDriveWay>
        <!--req, int, yellow Light Lane No., range:[1,16]-->0
      </yellowLightDriveWay>
      <osDdriveWay>
        <!--req, int, Overlay Lane No., range:[1,99]-->0
      </osDdriveWay>
      <distance>
        <!--req, int, coil distance, range:[0,2000], unit:cm-->0
      </distance>
      <recordEnable>
        <!--req, bool, whether to enable recording-->true
      </recordEnable>
      <recordType>
        <!--req, enum, recording type, subType:string, desc:"preRecord" (pre-record), "delayRecord" (post-record)-->preRecord
      </recordType>
      <preRecordTime>
        <!--req, int, pre-record duration, range:[0,100]-->0
      </preRecordTime>
      <recordDelayTime>
        <!--req, int, post-record, range:[0,100]-->0
      </recordDelayTime>
      <recordTimeOut>
        <!--req, int, timeout duration, range:[0,100]-->0
      </recordTimeOut>
      <serialProtocol>
        <!--req, enum, serial port protocol, subType:int, desc:0-vehicle detector, 1-OEM, 2-other vehicle detector-->0
      </serialProtocol>
      <runRedLightCaptureLogic>
        <!--req, enum, Red Light Running Capture Logic, subType:string, desc:"singelPro" (single_enter 1_exit 1_delay 1_1), "singlePro2" (singelPro),
        "doublePro1" (double_enter 1_exit 2_1), "doublePro2" (double_enter 1_exit 2_delay 2_1), "doublePro3" (double_enter 2_exit 1_delay 2_1), "doublePro4"
        (double/three_enter 2_exit 2_delay 2_1), "doublePro5" (double/three_enter 2_exit 2_delay 2_2), "doublePro6" (double/three_enter 1_exit 2_delay 2_2),
        "doublePro7" (double_enter 1_exit 1_delay 2_1), "doublePro8" (double_enter 1_exit 1_delay 2_2), "other"-->singelPro
      </runRedLightCaptureLogic>
      <oppositeDirectionCaptureLogic>
        <!--req, enum, Wrong-Way Driving Capture Logic, subType:string, desc:"noCap" (do not capture), "reversePro1" (double/three_enter_enter_delay),
        "reversePro2" (double/three_enter_delay)-->noCap
      </oppositeDirectionCaptureLogic>
      <intervalType>
        <!--req, enum, Burst Interval Type, subType:string, desc:"time" (by time), "distance" (by distance)-->time
      </intervalType>
      <interval>
        <!--req, int, interval, range:[0,2000]-->0
      </interval>
      <iOOutList>
        <!--opt, array, List of I/O output ports, subType:object-->
        <iOOut>
          <!--req, object, I/O output port-->
          <id>
            <!--req, int, Linked I/O output No., range:[1,10]-->0
          </id>
          <enabled>
            <!--req, bool, whether to Link the I/O output-->true
          </enabled>
        </iOOut>
      </iOOutList>
      <flashMode>
        <!--req, enum, flashing mode, subType:string, desc:"together" (simultaneously), "alternation" (alternatively)-->together
      </flashMode>
    </EpolicerRS485Lane>
  </EpolicerRS485LaneList>
</EpolicerRS485>

```

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-->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>
```

12.5.1.21 Get the parameters capability of intersection violation system vehicle detector

Request URL

GET /ISAPI/ITC/TriggerMode/epoliceRS485/capabilities

Query Parameter

None

Request Message

None

Response Message

```

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

<EpoliceRS485 xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, parameters of intersection violation system vehicle detector, attr:version{req, string, protocolVersion}-->
  <relatedLaneCount min="1" max="6">
    <!--ro, req, int, number of supported Lanes, range:[1,6], attr:min{req, int},max{req, int}-->0
  </relatedLaneCount>
  <trafficLightSource opt="0,1">
    <!--ro, req, enum, traffic Light source, subType:int, attr:opt{req, string}, desc:0-vehicle detector, 1-traffic Light detector.-->0
  </trafficLightSource>
  <relatedDriveWay min="1" max="16">
    <!--ro, req, int, Linked Lane No., range:[1,16], attr:min{req, int},max{req, int}-->0
  </relatedDriveWay>
  <redLightDriveWay min="1" max="16">
    <!--ro, req, int, red Light Lane No., range:[1,16], attr:min{req, int},max{req, int}-->0
  </redLightDriveWay>
  <yellowLightDriveWay min="1" max="16">
    <!--ro, req, int, yellow Light Lane No., range:[1,16], attr:min{req, int},max{req, int}-->0
  </yellowLightDriveWay>
  <OSDDriveWay min="1" max="99">
    <!--ro, req, int, overlay Lane No., range:[1,99], attr:min{req, int},max{req, int}-->0
  </OSDDriveWay>
  <laneDirectionType opt="0,1,2,3,4,5,6,7,8">
    <!--ro, req, enum, Lane direction, subType:int, attr:opt{req, string}, desc:0 (unknown),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)-->0
  </laneDirectionType>
  <distance min="0" max="2000">
    <!--ro, req, int, coil distance, range:[0,200], attr:min{req, int},max{req, int}-->0
  </distance>
  <recordEnable>
    <!--ro, req, bool, whether to enable recording-->true
  </recordEnable>
  <recordType opt="delayRecord,preRecord">
    <!--ro, req, enum, recording type, subType:string, attr:opt{req, string}, desc:"preRecord" (pre-record), "delayRecord" (post-record)-->preRecord
  </recordType>
  <preRecordTime min="0" max="100">
    <!--ro, req, int, pre-record, range:[0,100], attr:min{req, int},max{req, int}-->0
  </preRecordTime>
  <recordDelayTime min="0" max="100">
    <!--ro, req, int, post-record, range:[0,100], attr:min{req, int},max{req, int}-->0
  </recordDelayTime>
  <recordTimeOut min="0" max="100">
    <!--ro, req, int, timeout duration, range:[0,100], attr:min{req, int},max{req, int}-->0
  </recordTimeOut>
  <serialProtocol opt="0,1,2">
    <!--ro, req, enum, serial port protocol, subType:int, attr:opt{req, string}, desc:0-vehicle detector, 1-OEM, 2-other vehicle detector-->0
  </serialProtocol>
  <runRedLightCaptureLogic opt="singelPro,doublePro1,doublePro2,doublePro3,doublePro4,doublePro5,doublePro6,doublePro7,doublePro8,singlePro2,other">
    <!--ro, req, enum, red Light running capture Logic, subType:string, attr:opt{req, string}, desc:"singelPro" (single_enter 1_exit 1_delay 1_1), "singlePro2" (singelPro), "doublePro1" (double_enter 1_exit 1_exit 2_1), "doublePro2" (double_enter 1_exit 2_delay 2_1), "doublePro3" (double_enter 2_exit 1_delay 2_1), "doublePro4" (double/three_enter 2_exit 2_delay 2_1), "doublePro5" (double/three_enter 2_exit 2_delay 2_2), "doublePro6" (double/three_enter 1_exit 1_delay 2_1), "doublePro7" (double_enter 1_exit 1_delay 2_1), "doublePro8" (double_enter 1_exit 1_delay 2_2), "other"-->singelPro
  </runRedlightCaptureLogic>
  <oppositeDirectionCaptureLogic opt="noCap,reversePro1,reversePro2">
    <!--ro, req, enum, Wrong-Way Driving Capture Logic, subType:string, attr:opt{req, string}, desc:"noCap" (do not capture), "reversePro1" (double/three_enter_enter_delay), "reversePro2" (double/three_enter_delay)-->noCap
  </oppositeDirectionCaptureLogic>
  <intervalType opt="time,distance">
    <!--ro, req, enum, Burst Interval Type, subType:string, attr:opt{req, string}, desc:"time" (by time), "distance" (by distance)-->time
  </intervalType>
  <interval>
    <!--ro, req, int, interval, range:[0,2000]-->0
  </interval>
  <IOOut min="1" max="7">
    <!--ro, req, int, number of I/O outputs, range:[1,7], attr:min{req, int},max{req, int}-->0
  </IOOut>
  <flashMode opt="together,alternation">
    <!--ro, req, enum, flashing mode, subType:string, attr:opt{req, string}, desc:"together" (simultaneously), "alternation" (alternatively)-->together
  </flashMode>
</EpoliceRS485>

```

12.5.1.22 Get the recommended configurations of intersection violation system vehicle detector

Request URL

GET /ISAPI/ITC/TriggerMode/epoliceRS485/recommendation

Query Parameter

None

Request Message

None

Response Message

```

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

<EpoliceRS485 xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, parameters of intersection violation system vehicle detector, attr:version{req, string, protocolVersion}-->
  <relatedLaneCount>
    <!--ro, req, int, number of supported Lanes, range:[1,4]-->0
  </relatedLaneCount>
  <trafficLightSource>
    <!--ro, req, enum, traffic Light source, subType:int, desc:0-vehicle detector, 1-traffic Light detector.-->0
  </trafficLightSource>
  <EpoliceRS485LaneList>
    <!--ro, opt, array, Lane List, subType:object-->
    <EpoliceRS485Lane>
      <!--ro, opt, object, Lane-->
      <laneId>
        <!--ro, req, int, Lane ID, range:[1,4]-->0
      </laneId>
      <laneDirectionType>
        <!--ro, req, enum, Lane direction, subType:int, desc:0 (unknown),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
      </laneDirectionType>
      <relatedDriveWay>
        <!--ro, req, int, Linked Lane No., range:[1,16]-->0
      </relatedDriveWay>
      <redLightDriveWay>
        <!--ro, req, int, red Light Lane No., range:[1,16]-->0
      </redLightDriveWay>
      <yellowLightDriveWay>
        <!--ro, req, int, yellow Light Lane No., range:[1,16]-->0
      </yellowLightDriveWay>
      <OSDDriveWay>
        <!--ro, req, int, overlay Lane No., range:[1,99]-->0
      </OSDDriveWay>
      <distance>
        <!--ro, req, int, coil distance, range:[0,2000], unit:cm-->0
      </distance>
      <recordEnable>
        <!--ro, req, bool, whether to enable recording-->true
      </recordEnable>
      <recordType>
        <!--ro, req, enum, recording type, subType:string, desc:"preRecord" (pre-record), "delayRecord" (post-record)-->preRecord
      </recordType>
      <preRecordTime>
        <!--ro, req, int, pre-record duration, range:[0,100]-->0
      </preRecordTime>
      <recordDelayTime>
        <!--ro, req, int, post-record, range:[0,100]-->0
      </recordDelayTime>
      <recordTimeOut>
        <!--ro, req, int, timeout duration, range:[0,100]-->0
      </recordTimeOut>
      <serialProtocol>
        <!--ro, req, enum, serial port protocol, subType:int, desc:0-vehicle detector, 1-OEM, 2-other vehicle detector-->0
      </serialProtocol>
      <runRedLightCaptureLogic>
        <!--ro, req, enum, Red Light Running Capture Logic, subType:string, desc:"singelPro" (single_enter 1_exit 1_delay 1_1), "singlePro2" (singelPro), "doublePro1" (double_enter 1_exit 1_exit 2_1), "doublePro2" (double_enter 1_exit 2_delay 2_1), "doublePro3" (double_enter 2_exit 1_delay 2_1), "doublePro4" (double/three_enter 2_exit 2_delay 2_1), "doublePro5" (double/three_enter 2_exit 2_delay 2_2), "doublePro6" (double/three_enter 1_exit 2_delay 2_2), "doublePro7" (double_enter 1_exit 1_delay 2_1), "doublePro8" (double_enter 1_exit 1_delay 2_2), "other"-->singelPro
      </runRedLightCaptureLogic>
      <oppositeDirectionCaptureLogic>
        <!--ro, req, enum, Wrong-Way Driving Capture Logic, subType:string, desc:"noCap" (do not capture), "reversePro1" (double/three_enter_enter_delay), "reversePro2" (double/three_enter_delay)-->noCap
      </oppositeDirectionCaptureLogic>
      <intervalType>
        <!--ro, req, enum, burst interval type, subType:string, desc:"time" (by time), "distance" (by distance)-->time
      </intervalType>
      <interval>
        <!--ro, req, int, interval, range:[0,2000]-->0
      </interval>
      <IOOutList>
        <!--ro, opt, array, List of I/O output ports, subType:object-->
        <IOOut>
          <!--ro, req, object, I/O output port-->
          <id>
            <!--ro, req, int, Linked I/O output No., range:[1,10]-->0
          </id>
          <enabled>
            <!--ro, req, bool, whether to Link the I/O output-->true
          </enabled>
        </IOOut>
      </IOOutList>
      <flashMode>
        <!--ro, req, enum, flashing mode, subType:string, desc:"together" (simultaneously), "alternation" (alternatively)-->together
      </flashMode>
    </EpoliceRS485Lane>
  </EpoliceRS485LaneList>
</EpoliceRS485>

```

12.5.1.23 Get the status of video detection traffic lights

Request URL

GET /ISAPI/ITC/TriggerMode/lightStatus?channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Camera ID (this optional parameter is supported when isSupportChannelLightStatus is "true" returned from /ISAPI/ITC/TriggerMode/capabilities).

Request Message

None

Response Message

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

<VepLightStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, status of video detection traffic lights, attr:version{req, string, protocolVersion}-->
  <lightStatus opt="normal,abnormal">
    <!--ro, req, enum, traffic light status, subType:string, attr:opt{req, string}, desc:"normal", "abnormal"-->normal
  </lightStatus>
</VepLightStatus>
```

12.5.1.24 Get the license plate recognition configuration capability of a specific trigger mode

Request URL

GET /ISAPI/ITC/TriggerMode/plateRecognition/<triggerMode>/capabilities?channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
triggerMode	enum	--
channelID	string	Camera ID (this optional parameter is supported when isSupportChannelPlateRecognition is "true" returned from /ISAPI/ITC/TriggerMode/capabilities).

Request Message

None

Response Message

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

<PlateRecognition xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, the license plate recognition parameters of a specific trigger mode, attr:version{req, string, protocolVersion}-->
  <triggerMode opt="postIOSpeed,postSingleIO,postRS485,postRadar,postVTCoil,postHT,postMpr,epoliceRS485,postEpoliceRS485,videoEpolice,postBOT">
    <!--ro, req, enum, trigger modes supported by device, subType:string, attr:opt{req, string}, desc:"postiospeed" (checkpoint i/o speed detection), "postsingleio" (checkpoint single i/o trigger), "postrs485" (checkpoint vehicle detector), "postradar" (checkpoint radar), "postvtcoil" (checkpoint virtual coil), "postht" (checkpoint multi-target-type detection), "postmpr" (entrance and exit video trigger), "epolicers485" (intersection violation system vehicle detector), "postepolicers485" (checkpoint intersection violation system vehicle detector), "videoepolice" (video intersection violation system), "postbot" (bus lane occupation)-->postIOSpeed
  </triggerMode>
  <backupType opt="0,1">
    <!--ro, req, enum, backup mode type, subType:int, attr:opt{req, string}, desc:0-backup mode, 1-normal configuration mode. When this node is 0, the current trigger mode is checkpoint vehicle detector; when this node is 1, the current trigger mode is checkpoint virtual coil-->0
  </backupType>
  <plateRecogEnabled opt="true,false">
    <!--ro, opt, bool, whether to enable license plate recognition function of this region, attr:opt{req, string}-->true
  </plateRecogEnabled>
  <regionMode opt="rectangle,polygon">
    <!--ro, req, enum, area type, subType:string, attr:opt{req, string}, desc:"rectangle", "polygon"-->rectangle
  </regionMode>
  <plateRegionCount min="1" max="6">
    <!--ro, req, int, total number of linked license plate recognition regions, range:[1,6], attr:min{req, int},max{req, int}-->0
  </plateRegionCount>
</PlateRecognition>
```

12.5.1.25 Get parameters of checkpoint intersection violation system vehicle detector

Request URL

GET /ISAPI/ITC/TriggerMode/postEpoliceRS485

Query Parameter

None

Request Message

None

Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<PostRS485 xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, parameters of checkpoint vehicle detector, attr:version{req, string, protocolVersion}-->
</PostRS485>
```

12.5.1.26 Set parameters of checkpoint intersection violation system vehicle detector

Request URL

PUT /ISAPI/ITC/TriggerMode/postEpoliceRS485

Query Parameter

None

Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<PostRS485 xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, object, parameters of checkpoint vehicle detector, attr:version{req, string, protocolVersion}-->
</PostRS485>
```

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 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 application, which is used for fast debugging-->badXmlFormat
    </description>
</ResponseStatus>
```

12.5.1.27 Get the parameters capability of checkpoint intersection violation system vehicle detector

Request URL

GET /ISAPI/ITC/TriggerMode/postEpoliceRS485/capabilities

Query Parameter

None

Request Message

None

Response Message

```

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

<PostRS485 xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, parameters of checkpoint vehicle detector, attr:version{req, string, protocolVersion}-->
  <relatedLaneCount min="1" max="6">
    <!--ro, req, int, number of supported lanes, attr:min{req, int},max{req, int}-->0
  </relatedLaneCount>
  <maxSwitchTime min="5" max="255">
    <!--ro, req, int, fault-tolerant time, attr:min{req, int},max{req, int}-->0
  </maxSwitchTime>
  <backupMode opt="0,1,2">
    <!--ro, req, enum, Backup Mode, subType:int, attr:opt{req, string}, desc:"0" (none), "1" (trigger of checkpoint virtual coil) Pattern-->0
  </backupMode>
  <relatedDriveWay min="1" max="99">
    <!--ro, req, int, Linked Lane No., attr:min{req, int},max{req, int}-->0
  </relatedDriveWay>
  <OSDDriveWay min="1" max="99">
    <!--ro, req, int, Lane No. to be linked, attr:min{req, int},max{req, int}-->0
  </OSDDriveWay>
  <laneType opt="none,highway,urbanFastWay,other">
    <!--ro, req, enum, Lane attribute, subType:string, attr:opt{req, string}, desc:"none"-not configured, "highway"-high way, "urbanFastWay"-urban expressway, "other"-->none
    </laneType>
    <laneUsage opt="carriageWay,emergency">
      <!--ro, req, enum, Lane purpose, subType:string, attr:opt{req, string}, desc:"carriageway" (normal Lane), "emergency" (emergency Lane)-->carriageway
    </laneUsage>
    <distance min="0" max="20000">
      <!--ro, req, int, coil distance, attr:min{req, int},max{req, int}-->0
    </distance>
    <delayTime min="0" max="2000">
      <!--ro, req, int, trigger delay time, attr:min{req, int},max{req, int}-->0
    </delayTime>
    <delayDistance min="0" max="255">
      <!--ro, req, int, trigger delay distance, attr:min{req, int},max{req, int}-->0
    </delayDistance>
    <speedCapEnabled>
      <!--ro, req, bool, overspeed capture-->true
    </speedCapEnabled>
    <lowSpeedCapEnable>
      <!--ro, req, bool, Low speed capture-->true
    </lowSpeedCapEnable>
    <emergencyCapEnable>
      <!--ro, req, bool, occupying emergency Lane capture-->true
    </emergencyCapEnable>
    <signSpeed min="0" max="255">
      <!--ro, req, int, marked speed limit, attr:min{req, int},max{req, int}-->0
    </signSpeed>
    <speedLimit min="0" max="255">
      <!--ro, req, int, speed limit, attr:min{req, int},max{req, int}-->0
    </speedLimit>
    <bigCarSignSpeed min="0" max="255">
      <!--ro, req, int, marked speed limit for large-sized vehicle, attr:min{req, int},max{req, int}-->0
    </bigCarSignSpeed>
    <bigCarLowSpeedLimit min="0" max="255">
      <!--ro, req, int, Low speed limit for large-sized vehicle, attr:min{req, int},max{req, int}-->0
    </bigCarLowSpeedLimit>
    <lowSpeedLimit min="0" max="60">
      <!--ro, req, int, Low speed limit for small-sized vehicle, attr:min{req, int},max{req, int}-->0
    </lowSpeedLimit>
    <bigCarSpeedLimit min="0" max="60">
      <!--ro, req, int, speed limit for large-sized vehicle, attr:min{req, int},max{req, int}-->0
    </bigCarSpeedLimit>
    <carHighSpeed min="0" max="255">
      <!--ro, req, int, abnormal overspeed, attr:min{req, int},max{req, int}-->0
    </carHighSpeed>
    <carLowSpeed min="0" max="255">
      <!--ro, req, int, abnormal low speed, attr:min{req, int},max{req, int}-->0
    </carLowSpeed>
    <bigCarHighSpeed min="0" max="255">
      <!--ro, req, int, abnormal overspeed of large-sized vehicle, attr:min{req, int},max{req, int}-->0
    </bigCarHighSpeed>
    <bigCarLowSpeed min="0" max="255">
      <!--ro, req, int, abnormal low speed of large-sized vehicle, attr:min{req, int},max{req, int}-->0
    </bigCarLowSpeed>
    <snapTimes min="0" max="5">
      <!--ro, req, int, continuous capture times, attr:min{req, int},max{req, int}-->0
    </snapTimes>
    <laneDirectionType opt="0,1,2,3,4,5,6,7,8">
      <!--ro, req, enum, types of lane directions, subType:int, attr:opt{req, string}, desc:0 (unknown),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)-->0
    </laneDirectionType>
    <intervalType opt="time,distance">
      <!--ro, req, enum, Burst Interval Type, subType:string, attr:opt{req, string}, desc:"time" (by time), "distance" (by distance)-->time
    </intervalType>
    <IOOut min="1" max="7">
      <!--ro, req, int, I/O outputs, attr:min{req, int},max{req, int}-->0
    </IOOut>
    <flashMode opt="together,alternation">
      <!--ro, req, enum, flashing mode, subType:string, attr:opt{req, string}, desc:"together" (simultaneously), "alternation" (alternatively)-->together
    </flashMode>
    <emergencyTimeSwitchList size="4">

```

```

<!--ro, req, array, time period for emergency lanes, subType:object, attr:size{req, int}-->
<emergencyTimeSwitch>
  <!--ro, req, object, emergency lane time-->
  <timeId min="1" max="4">
    <!--ro, req, int, index, attr:min{req, int},max{req, int}-->0
  </timeId>
  <startHour min="0" max="23">
    <!--ro, opt, int, start time (hour), attr:min{req, int},max{req, int}-->0
  </startHour>
  <startMinute min="0" max="59">
    <!--ro, opt, int, start time (minute), attr:min{req, int},max{req, int}-->0
  </startMinute>
  <endHour min="0" max="23">
    <!--ro, opt, int, end time (hour), attr:min{req, int},max{req, int}-->0
  </endHour>
  <endMinute min="0" max="59">
    <!--ro, opt, int, end time (minute), attr:min{req, int},max{req, int}-->0
  </endMinute>
</emergencyTimeSwitch>
</emergencyTimeSwitchList>
<noSupportPlateRegion>
  <!--ro, opt, bool, areas where ANPR is not supported-->true
</noSupportPlateRegion>
<LaneLine>
  <!--ro, opt, object, Lane Line-->
  <lineName min="0" max="64">
    <!--ro, req, string, Line name, attr:min{req, int},max{req, int}-->test
  </lineName>
  <RegionCoordinatesList num="2">
    <!--ro, opt, array, region coordinate list, subType:object, range:[0,2], attr:num{opt, object}-->
    <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]-->0
      </positionX>
      <positionY>
        <!--ro, req, int, y-coordinate, range:[0,1000]-->0
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</LaneLine>
<LaneRightBoundaryLine>
  <!--ro, req, object, right border line of Lane-->
  <lineName>
    <!--ro, req, enum, Line name, subType:string, desc:"LaneRightBoundaryLine" (right border line of Lane)-->laneRightBoundaryLine
  </lineName>
  <lineType>
    <!--ro, req, enum, Line type, subType:string, desc:"unknown" (white line), "white" (white solid line), "singleYellow" (single yellow line), "doubleYellow" (double yellow line), "guardRail" (Lane Line with guardrail), "noCross" (Lane Line prohibited to cross by vehicle)-->unknown
  </lineType>
  <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]-->0
    </positionX>
    <positionY>
      <!--ro, req, int, y-coordinate, range:[0,1000]-->0
    </positionY>
  </RegionCoordinates>
</LaneRightBoundaryLine>
<safeBelt>
  <!--ro, opt, bool, whether to enable seatbelt detection-->true
</safeBelt>
<safeBeltCapNo min="1" max="3">
  <!--ro, opt, int, number of captured pictures in which the driver does not wear the seatbelt, attr:min{req, int},max{req, int}-->0
</safeBeltCapNo>
<uphone>
  <!--ro, opt, bool, whether to enable phone call detection-->true
</uphone>
<uphoneCapNo min="1" max="3">
  <!--ro, opt, int, number of captured pictures in which the driver is making call, attr:min{req, int},max{req, int}-->0
</uphoneCapNo>
</PostRS485>

```

12.5.1.28 Get the recommended configurations of checkpoint intersection violation system vehicle detector

Request URL

GET /ISAPI/ITC/TriggerMode/postEpoliceRS485/recommendation

Query Parameter

None

Request Message

None

Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<PostRS485 xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, parameters of checkpoint vehicle detector, attr:version{req, string, protocolVersion}-->
</PostRS485>
```

12.5.1.29 Set video trigger mode parameters of entrances and exits

Request URL

PUT /ISAPI/ITC/TriggerMode/postMPR

Query Parameter

None

Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PostMPR xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, parameters of entrances and exits, attr:version{req, string, protocolVersion}-->
  <triggerType>
    <!--req, enum, condition for triggering alarm output, subType:int, desc:"0" (triggered by video), "1" (triggered by I/O), "2" (triggered by RS-485 serial port)-->0
  </triggerType>
  <uploadType>
    <!--req, enum, upload method, subType:int, desc:"0" (upload all), "1" (upload), "2" (reversely upload)-->0
  </uploadType>
  <relatedLaneCount>
    <!--req, int, number of supported lanes, range:[1,6]-->0
  </relatedLaneCount>
  <LaneParamList>
    <!--req, array, Lane parameter list, subType:object-->
    <LaneParam>
      <!--opt, object, Lane parameters-->
      <laneId>
        <!--req, int, Lane ID, range:[1,8]-->1
      </laneId>
      <laneDirectionType>
        <!--req, enum, Lane direction No., subType:int, desc:0 (unknown), 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)-->0
      </laneDirectionType>
      <relatedDriveWay>
        <!--req, int, Linked Lane No., range:[1,99]-->0
      </relatedDriveWay>
      <relate485Chan>
        <!--opt, int, Linked RS-485 port number, range:[1,16]-->0
      </relate485Chan>
      <relateIO>
        <!--opt, int, Linked IO port, range:[1,2]-->0
      </relateIO>
      <IODefaultStatus>
        <!--opt, enum, default I/O status, subType:string, desc:"rising, falling, both"-->falling
      </IODefaultStatus>
      <ViolationDetectLine>
        <!--req, object, draw violation detection Line on the single lane-->
        <lineName>
          <!--req, enum, Line name, subType:string, desc:detection line name,"LaneLine" (for single lane only)-->laneLine
        </lineName>
        <RegionCoordinates>
          <!--opt, object, coordinates, desc:the origin is the upper-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>
      </ViolationDetectLine>
    </LaneParam>
  </LaneParamList>
  <VirtualLane>
    <!--req, object, virtual lane-->
    <lineName>
      <!--req, enum, Line name, subType:string, desc:detection line name,"LaneLine" (for single lane only)-->laneRightBoundaryLine
    </lineName>
    <RegionCoordinates>
      <!--opt, object, area coordinates, desc:the origin is the upper-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>
  </VirtualLane>
</PostMPR>

```

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>
```

12.5.1.30 Get video trigger mode parameters of entrances and exits

Request URL

GET /ISAPI/ITC/TriggerMode/postMPR

Query Parameter

None

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PostMPR xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, parameters of entrances and exits, attr:version{req, string, protocolVersion}-->
  <triggerType>
    <!--ro, req, enum, condition for triggering alarm output, subType:int, desc:0 (triggered by video), 1 (triggered by I/O), 2 (triggered by RS-485 serial port)-->
  </triggerType>
  <uploadType>
    <!--ro, req, enum, upload method, subType:int, desc:0 (upload all), 1 (upload), 2 (reversely upload)-->0
  </uploadType>
  <relatedLaneCount>
    <!--ro, req, int, number of supported lanes, range:[1,6]-->0
  </relatedLaneCount>
  <LaneParamList>
    <!--ro, req, array, Lane parameter list, subType:object-->
    <LaneParam>
      <!--ro, opt, object, Lane parameters-->
      <laneId>
        <!--ro, req, int, Lane ID, range:[1,8]-->1
      </laneId>
      <laneDirectionType>
        <!--ro, req, enum, Lane direction No., subType:int, desc:0 (unknown), 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)-->0
      </laneDirectionType>
      <relatedDriveWay>
        <!--ro, req, int, Linked Lane No., range:[1,99]-->0
      </relatedDriveWay>
      <relate485Chan>
        <!--ro, opt, int, Linked RS-485 port number, range:[1,16]-->0
      </relate485Chan>
      <relateIO>
        <!--ro, opt, int, Linked IO port, range:[1,2]-->0
      </relateIO>
      <IODefaultStatus>
        <!--ro, opt, enum, default I/O status, subType:string, desc:"rising, falling, both"-->falling
      </IODefaultStatus>
      <ViolationDetectLine>
        <!--ro, req, object, draw violation detection line on the single lane-->
        <lineName>
          <!--ro, req, enum, Line name, subType:string, desc:detection line name,"LaneLine" (for single lane only)-->laneLine
        </lineName>
        <RegionCoordinates>
          <!--ro, opt, object, coordinates, desc:the origin is the upper-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>
      </ViolationDetectLine>
    </LaneParam>
  </LaneParamList>
  <VirtualLane>
    <!--ro, req, object, draw violation detection line on all lanes-->
    <lineName>
      <!--ro, req, enum, Line name, subType:string, desc:detection line name,"LaneLine" (for single lane only)-->laneRightBoundaryLine
    </lineName>
    <RegionCoordinates>
      <!--ro, opt, object, only two coordinates are required, desc:the origin is the upper-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>
  </VirtualLane>
</PostMPR>

```

12.5.1.31 Get video trigger mode capability of entrances and exits

Request URL

GET /ISAPI/ITC/TriggerMode/postMPR/capabilities

Query Parameter

None

Request Message

None

Response Message

```

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

<PostMPR xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, parameters of entrances and exits, attr:version{req, string, protocolVersion}-->
  <relatedLaneCount min="1" max="6">
    <!--ro, req, int, number of supported Lanes, attr:min{req, int},max{req, int}-->0
  </relatedLaneCount>
  <relatedDriveWay min="1" max="99">
    <!--ro, req, int, Linked Lane No., attr:min{req, int},max{req, int}-->0
  </relatedDriveWay>
  <triggerType opt="0,1,2">
    <!--ro, req, enum, 0-triggered by video,1-triggered by I/O,2-triggered by RS-485 serial port, subType:int, attr:opt{req, string}, desc:0 (triggered by video), 1 (triggered by I/O), 2 (triggered by RS-485 serial port)-->0
  </triggerType>
  <uploadType opt="0,1,2">
    <!--ro, req, enum, 0-upload all,1-upload,2-reversely upload, subType:int, attr:opt{req, string}, desc:0 (upload all), 1 (upload), 2 (reversely upload)-->0
  </uploadType>
  <relate485Chan min="1" max="16">
    <!--ro, req, int, Linked RS-485 port number, attr:min{req, int},max{req, int}-->0
  </relate485Chan>
  <relateIO min="1" max="4">
    <!--ro, req, int, Linked I/O port number, attr:min{req, int},max{req, int}-->0
  </relateIO>
  <laneDirectionType opt="0,1,2,3,4,5,6,7,8">
    <!--ro, req, enum, types of lane directions, subType:int, attr:opt{req, string}, desc:0 (unknown),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)-->0
  </laneDirectionType>
  <IODefaultStatus opt="falling,rising,both">
    <!--ro, req, enum, default I/O status, subType:string, attr:opt{req, string}, desc:default I/O status-->falling
  </IODefaultStatus>
</PostMPR>

```

12.5.1.32 Get the trigger mode parameters of not yielding to pedestrian

Request URL

GET /ISAPI/ITC/TriggerMode/postNoComityPed

Query Parameter

None

Request Message

None

Response Message

```

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

<PostNoComityPed xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, the trigger mode of not yielding to pedestrian, attr:version{req, string, protocolVersion}-->
  <enable>
    <!--ro, req, bool, whether to enable the function-->true
  </enable>
  <laneNum>
    <!--ro, opt, int, Lane amount, range:[1,3]-->0
  </laneNum>
  <RightTriggerLine>
    <!--ro, req, object, right Lane Line parameter-->
    <lineName>
      <!--ro, req, string, Line name, range:[0,64], desc:"rightTriggerLine" (right Lane Line)-->test
    </lineName>
    <lineName>
      <!--ro, req, enum, Line name, subType:string, desc:"rightTriggerLine" (right Lane Line)-->rightTriggerLine
    </lineName>
    <lineType>
      <!--ro, req, enum, Line type, subType:string, desc:"unknown", "white" (solid white line), "singleYellow" (single yellow line), "doubleYellow" (double yellow line), "guardrail" (Lane line with guardrail), "noCross" (Lane line that cannot be crossed)-->unknown
    </lineType>
    <TriggerLineCoordinatesList>
      <!--ro, opt, array, triggering Line coordinate List, subType:object, range:[0,2]-->
      <TriggerLineCoordinates>
        <!--ro, opt, object, triggering Line coordinates, desc:the origin is the upper-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>
      </TriggerLineCoordinates>
    </TriggerLineCoordinatesList>
  </RightTriggerLine>
  <Triggerline>
    <!--ro, req, object, triggering Line-->
  </Triggerline>
</PostNoComityPed>

```

```

<lineName>
  <!--ro, req, enum, Line name, subType:string, desc:"triggerLine" (triggering Line)-->triggerLine
</lineName>
<lineType>
  <!--ro, req, enum, Line type, subType:string, desc:"unknown", "white" (solid white line), "singleYellow" (single yellow line), "doubleYellow" (double yellow line), "guardRail" (Lane Line with guardrail), "noCross" (Lane Line that cannot be crossed)-->unknown
</lineType>
<TriggerLineCoordinatesList>
  <!--ro, opt, array, triggering Line coordinate List, subType:object, range:[0,2]-->
<TriggerLineCoordinates>
  <!--ro, opt, object, triggering Line coordinates, desc:the origin is the upper-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>
</TriggerLineCoordinates>
</TriggerLineCoordinatesList>
</TriggerLine>
<PedesDetRegionCoordinatesList>
  <!--ro, opt, array, pedestrian detection area List, subType:object, range:[3,10]-->
<RegionCoordinates>
  <!--ro, opt, object, coordinate point, desc:the origin is the upper-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>
</PedesDetRegionCoordinatesList>
<LaneParamList>
  <!--ro, opt, object, Lane parameter List-->
<LaneParam>
  <!--ro, req, object, Lane parameters-->
<relatedDriveWay>
  <!--ro, req, int, Linked Lane No., range:[1,3]-->0
</relatedDriveWay>
<laneDirectionType>
  <!--ro, req, enum, Lane direction, subType:int, desc:0 (unknown), 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)-->0
</laneDirectionType>
<pedestriansNum>
  <!--ro, opt, int, pedestrian amount, range:[1,100]-->0
</pedestriansNum>
<vehicleSpeed>
  <!--ro, opt, int, vehicle speed, range:[0,100]-->0
</vehicleSpeed>
<vehicleInterval>
  <!--ro, opt, int, distance between vehicles, range:[0,65536]-->0
</vehicleInterval>
<pedesDetRule>
  <!--ro, req, enum, pedestrian detection rules, subType:string, desc:"left", "right", "biDirectional"-->left
</pedesDetRule>
<laneline>
  <!--ro, req, object, Lane line parameters-->
<lineName>
  <!--ro, req, enum, Line name, subType:string, desc:"laneline" (Lane Line)-->laneline
</lineName>
<lineType>
  <!--ro, req, enum, Line type, subType:string, desc:"unknown", "white" (solid white line), "singleYellow" (single yellow line), "doubleYellow" (double yellow line), "guardRail" (Lane Line with guardrail), "noCross" (Lane Line that cannot be crossed)-->unknown
</lineType>
<TriggerLineCoordinatesList>
  <!--ro, opt, array, triggering Line coordinate List, subType:object, range:[0,2]-->
<TriggerLineCoordinates>
  <!--ro, opt, object, triggering Line coordinates, desc:the origin is the upper-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>
</TriggerLineCoordinates>
</TriggerLineCoordinatesList>
</laneline>
<stopLine>
  <!--ro, req, object, stop Line-->
<lineName>
  <!--ro, req, enum, Line name, subType:string, desc:"stopLine" (stop Line)-->stopline
</lineName>
<lineType>
  <!--ro, req, enum, Line type, subType:string, desc:"unknown", "white" (solid white line), "singleYellow" (single yellow line), "doubleYellow" (double yellow line), "guardRail" (Lane Line with guardrail), "noCross" (Lane Line that cannot be crossed)-->unknown
</lineType>
<TriggerLineCoordinatesList>
  <!--ro, opt, array, triggering Line coordinate List, subType:object, range:[0,2]-->
<TriggerLineCoordinates>
  <!--ro, opt, object, triggering Line coordinates, desc:the origin is the upper-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>

```

```

<!--req, int, Y-coordinate, range:[0,1000]-->
</positionY>
</TriggerLineCoordinates>
</TriggerLineCoordinatesList>
</stopLine>
</PlateRegionCoordinatesList>
<!--req, array, LPR area coordinate list, subType:object, range:[3,10]-->
<RegionCoordinates>
  <!--req, opt, object, area coordinates, desc:the origin is the upper-left corner of the screen-->
  <positionX>
    <!--req, int, X-coordinate, range:[0,1000]-->
  </positionX>
  <positionY>
    <!--req, int, Y-coordinate, range:[0,1000]-->
  </positionY>
</RegionCoordinates>
</PlateRegionCoordinatesList>
</LaneParam>
</LaneParamList>
</PostNoComityPed>

```

12.5.1.33 Set the trigger mode parameters of not yielding to pedestrian

Request URL

PUT /ISAPI/ITC/TriggerMode/postNoComityPed

Query Parameter

None

Request Message

```

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

<PostNoComityPed xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, the trigger mode of not yielding to pedestrian, attr:version[req, string, protocolVersion]-->
  <enable>
    <!--req, bool, whether to enable the function-->true
  </enable>
  <laneNum>
    <!--opt, int, Lane Amount, range:[1,3]-->0
  </laneNum>
  <RightTriggerLine>
    <!--req, object, right lane line parameters-->
    <lineName>
      <!--req, string, Line name, range:[0,64], desc:"rightTriggerLine" (right lane line)-->test
    </lineName>
    <lineName>
      <!--req, enum, Line name, subType:string, desc:"rightTriggerLine" (right lane line)-->rightTriggerLine
    </lineName>
    <lineType>
      <!--req, enum, Line type, subType:string, desc:"unknown", "white" (solid white line), "singleYellow" (single yellow line), "doubleYellow" (double yellow line), "guardrail" (Lane line with guardrail), "noCross" (Lane line that cannot be crossed)-->unknown
    </lineType>
    <TriggerLineCoordinatesList>
      <!--opt, array, triggering line coordinate list, subType:object, range:[0,2]-->
      <TriggerLineCoordinates>
        <!--opt, object, triggering line coordinates, desc:the origin is the upper-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>
      </TriggerLineCoordinates>
    </TriggerLineCoordinatesList>
  </RightTriggerLine>
  <TriggerLine>
    <!--req, object, Triggering Line-->
    <lineName>
      <!--req, enum, Line name, subType:string, desc:"triggerLine" (triggering line)-->triggerLine
    </lineName>
    <lineType>
      <!--req, enum, Line type, subType:string, desc:"unknown", "white" (solid white line), "singleYellow" (single yellow line), "doubleYellow" (double yellow line), "guardrail" (Lane line with guardrail), "noCross" (Lane line that cannot be crossed)-->unknown
    </lineType>
    <TriggerLineCoordinatesList>
      <!--opt, array, triggering line coordinate list, subType:object, range:[0,2]-->
      <TriggerLineCoordinates>
        <!--opt, object, triggering line coordinates, desc:the origin is the upper-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>
      </TriggerLineCoordinates>
    </TriggerLineCoordinatesList>
  </TriggerLine>
</PostNoComityPed>

```

```

<!--->
</TriggerLine>
<PedesDetRegionCoordinatesList>
<!---opt, array, pedestrian detection area list, subType:object, range:[3,10]-->
<RegionCoordinates>
<!---opt, object, coordinate point, desc:the origin is the upper-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>
</PedesDetRegionCoordinatesList>
<LaneParamList>
<!---opt, array, lane parameter list, subType:object-->
<LaneParam>
<!---req, object, lane parameters-->
<relatedDriveWay>
<!---req, int, linked lane no., range:[1,3]-->0
</relatedDriveWay>
<laneDirectionType>
<!---req, enum, lane direction, subType:int, desc:0 (unknown), 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)-->0
</laneDirectionType>
<pedestriansNum>
<!---opt, int, pedestrian amount, range:[1,100]-->0
</pedestriansNum>
<vehicleSpeed>
<!---opt, int, vehicle speed, range:[0,100]-->0
</vehicleSpeed>
<vehicleInterval>
<!---opt, int, distance between vehicles, range:[0,65536]-->0
</vehicleInterval>
<pedesDetRule>
<!---req, enum, pedestrian detection rules, subType:string, desc:"left", "right", "biDirectional"-->left
</pedesDetRule>
<laneline>
<!---req, object, lane line parameters-->
<lineName>
<!---req, enum, line name, subType:string, desc:"laneline" (lane line)-->laneline
</lineName>
<lineType>
<!---req, enum, line type, subType:string, desc:"unknown", "white" (solid white line), "singleYellow" (single yellow line), "doubleYellow" (double yellow line), "guardrail" (lane line with guardrail), "noCross" (lane line that cannot be crossed)-->unknown
</lineType>
<TriggerLineCoordinatesList>
<!---opt, array, triggering line coordinate list, subType:object, range:[0,2]-->
<TriggerLineCoordinates>
<!---opt, object, triggering line coordinates, desc:the origin is the upper-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>
</TriggerLineCoordinates>
</TriggerLineCoordinatesList>
</laneline>
<stopLine>
<!---req, object, stop line-->
<lineName>
<!---req, enum, line name, subType:string, desc:"stopLine" (stop line)-->stopLine
</lineName>
<lineType>
<!---req, enum, line type, subType:string, desc:"unknown", "white" (solid white line), "singleYellow" (single yellow line), "doubleYellow" (double yellow line), "guardrail" (lane line with guardrail), "noCross" (lane line that cannot be crossed)-->unknown
</lineType>
<TriggerLineCoordinatesList>
<!---opt, array, triggering line coordinate list, subType:object, range:[0,2]-->
<TriggerLineCoordinates>
<!---opt, object, triggering line coordinates, desc:the origin is the upper-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>
</TriggerLineCoordinates>
</TriggerLineCoordinatesList>
</stopLine>
<PlateRegionCoordinatesList>
<!---req, array, LPR area coordinate list, subType:object, range:[3,10]-->
<RegionCoordinates>
<!---opt, object, area point coordinates, desc:the origin is the upper-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>
</PlateRegionCoordinatesList>
</LaneParam>

```

```
</LaneParamList>
</PostNoComityPed>
```

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-->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>
```

12.5.1.34 Get the trigger mode capability of not yielding to pedestrian

Request URL

GET /ISAPI/ITC/TriggerMode/postNoComityPed/capabilities

Query Parameter

None

Request Message

None

Response Message

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

<PostNoComityPed xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, the trigger mode of not yielding to pedestrian, attr:version{req, string, protocolVersion}-->
    <enable>
        <!--ro, req, bool, whether to enable the function-->true
    </enable>
    <laneNum opt="1,2,3" def="3">
        <!--ro, opt, int, Lane Amount, attr:opt{req, string},def{req, string}-->0
    </laneNum>
    <RightTriggerLine>
        <!--ro, req, object, right Lane Line parameters-->
        <lineName>
            <!--ro, req, string, Line name, desc:"rightTriggerLine" (right Lane Line)-->test
        </lineName>
        <lineName>
            <!--ro, req, enum, Line name, subType:string, desc:"rightTriggerLine" (right Lane Line)-->rightTriggerLine
        </lineName>
        <lineType opt="unknown,white,singleYellow,doubleYellow,guardRail,noCross">
            <!--ro, req, enum, Line type, subType:string, attr:opt{req, string}, desc:"unknown", "white" (solid white Line), "singleYellow" (single yellow Line), "doubleYellow" (double yellow Line), "guardRail" (Lane Line with guardrail), "noCross" (Lane Line that cannot be crossed)-->unknown
        </lineType>
        <TriggerLineCoordinatesList size="2">
            <!--ro, opt, array, triggering Line coordinate List, subType:object, range:[0,2], attr:size{req, int}-->
            <TriggerLineCoordinates>
                <!--ro, opt, object, triggering Line coordinates, desc:the origin is the upper-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>
            </TriggerLineCoordinates>
        </TriggerLineCoordinatesList>
    </RightTriggerLine>
    <TriggerLine>
        <!--ro, req, object, Triggering Line-->
        <lineName>
            <!--ro, req, enum, Line name, subType:string, desc:"triggerLine" (triggering Line)-->triggerLine
        </lineName>
        <lineType opt="unknown,white,singleYellow,doubleYellow,guardRail,noCross">

```

```

<!--ro, req, enum, Line type, subType:string, attr:opt{req, string}, desc:"unknown", "white" (solid white line), "singleYellow" (single yellow line),
"doubleYellow" (double yellow line), "guardRail" (Lane line with guardrail), "noCross" (Lane line that cannot be crossed)-->unknown
</lineType>
<TriggerLineCoordinatesList size="2">
<!--ro, opt, array, triggering Line coordinate List, subType:object, range:[0,2], attr:size{req, int}-->
<TriggerLineCoordinates>
<!--ro, opt, object, triggering Line coordinates, desc:the origin is the upper-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>
</TriggerLineCoordinates>
</TriggerLineCoordinatesList>
</TriggerLine>
<PedesDetRegionCoordinatesList size="20">
<!--ro, opt, array, pedestrian detection area List, subType:object, range:[3,10], attr:size{req, int}-->
<RegionCoordinates>
<!--ro, opt, object, coordinate point, desc:the origin is the upper-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>
</PedesDetRegionCoordinatesList>
<LaneParamList size="6">
<!--ro, opt, object, Lane parameter List, attr:size{req, int}-->
<LaneParam>
<!--ro, req, object, Lane parameters-->
<relatedDriveWay opt="1,2,3" def="3">
<!--ro, req, int, Linked Lane No., range:[1,3], attr:opt{req, string},def{req, string}-->0
</relatedDriveWay>
<laneDirectionType opt="0,1,2,3,4,5,6,7,8">
<!--ro, req, enum, Lane direction, subType:int, attr:opt{req, string}, desc:0 (unknown),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)-->0
</laneDirectionType>
<pedestriansNum min="1" max="100" def="1">
<!--ro, opt, int, pedestrian amount, attr:min{req, int},max{req, int},def{req, int}-->0
</pedestriansNum>
<vehicleSpeed min="0" max="100" def="0">
<!--ro, opt, int, vehicle speed, attr:min{req, int},max{req, int},def{req, int}-->0
</vehicleSpeed>
<vehicleInterval min="0" max="65536">
<!--ro, opt, int, distance between vehicles, attr:min{req, int},max{req, int}-->0
</vehicleInterval>
<pedesDetRule opt="left,right,biDirectional">
<!--ro, req, enum, Pedestrian Detection Rules, subType:string, attr:opt{req, string}, desc:"left", "right", "biDirectional"-->left
</pedesDetRule>
<laneline>
<!--ro, req, object, Lane line parameters-->
<lineName>
<!--ro, req, enum, Line name, subType:string, desc:"laneline" (Lane line)-->laneline
</lineName>
<lineType opt="unknown,white,singleYellow,doubleYellow,guardRail,noCross">
<!--ro, req, enum, Line type, subType:string, attr:opt{req, string}, desc:"unknown", "white" (solid white line), "singleYellow" (single yellow
line), "doubleYellow" (double yellow line), "guardRail" (Lane line with guardrail), "noCross" (Lane line that cannot be crossed)-->unknown
</lineType>
<TriggerLineCoordinatesList size="2">
<!--ro, opt, array, triggering Line coordinate List, subType:object, range:[0,2], attr:size{req, int}-->
<TriggerLineCoordinates>
<!--ro, opt, object, triggering Line coordinates, desc:the origin is the upper-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>
</TriggerLineCoordinates>
</TriggerLineCoordinatesList>
</laneline>
<stopLine>
<!--ro, req, object, stop line-->
<lineName>
<!--ro, req, enum, Line name, subType:string, desc:"stopLine" (stop line)-->stopline
</lineName>
<lineType opt="unknown,white,singleYellow,doubleYellow,guardRail,noCross">
<!--ro, req, enum, Line type, subType:string, attr:opt{req, string}, desc:"unknown", "white" (solid white line), "singleYellow" (single yellow
line), "doubleYellow" (double yellow line), "guardRail" (Lane line with guardrail), "noCross" (Lane line that cannot be crossed)-->unknown
</lineType>
<TriggerLineCoordinatesList size="2">
<!--ro, opt, array, triggering Line coordinate List, subType:object, range:[0,2], attr:size{req, int}-->
<TriggerLineCoordinates>
<!--ro, opt, object, triggering Line coordinates, desc:the origin is the upper-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>
</TriggerLineCoordinates>
</TriggerLineCoordinatesList>
</stopLine>

```

```

</!triggerLineCoordinates>
</TriggerLineCoordinatesList>
</stopLine>
<PlateRegionCoordinatesList size="20">
    <!--ro, req, array, LPR area coordinate List, subType:object, range:[3,10], attr:size{req, int}-->
    <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]-->0
        </positionX>
        <positionY>
            <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
        </positionY>
    </RegionCoordinates>
</PlateRegionCoordinatesList>
</LaneParam>
</LaneParamList>
</PostNoComityPed>

```

12.5.1.35 Set the parameters of ANPR system

Request URL

PUT /ISAPI/ITC/TriggerMode/postPRS

Query Parameter

None

Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PostPRS xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--req, object, all detection Lines, attr:version{req, string, protocolVersion}-->
    <triggerType>
        <!--req, enum, condition for triggering alarm output, subType:string, desc:"videodetection", "IO", "RS485", "radarDetection"-->videoDetection
    </triggerType>
    <captureMode>
        <!--req, enum, capture mode, subType:string, dep:and, ${.PostPRS.triggerType,eq,videoDetection}, desc:"strobe", "flashing"-->strobe
    </captureMode>
    <totalLaneNum>
        <!--req, int, number of total lanes, range:[0,10]-->0
    </totalLaneNum>
    <LaneParamList size="5">
        <!--opt, array, Lane parameter List, subType:object, attr:size{req, int}-->
        <LaneParam>
            <!--opt, object, Lane parameters-->
            <relatedDriveWay>
                <!--req, int, Linked Lane No., range:[1,99]-->0
            </relatedDriveWay>
            <RS485No>
                <!--opt, int, 485 No., range:[0,10], dep:and, ${.PostPRS.triggerType,eq,RS485}-->0
            </RS485No>
            <IONo>
                <!--opt, int, IO port No., range:[0,10], dep:and, ${.PostPRS.triggerType,eq,IO}-->0
            </IONo>
            <IODefStatus>
                <!--opt, enum, IO default status, subType:string, dep:and, ${.PostPRS.triggerType,eq,IO}, desc:"risingEdge", "fallingEdge"-->risingEdge
            </IODefStatus>
            <ViolationDetectLine>
                <!--req, object, draw violation detection line on the single Lane-->
                <lineName>
                    <!--req, enum, Line name, subType:string, desc:"LaneLine"-->laneLine
                </lineName>
            <RegionCoordinatesList>
                <!--opt, array, area coordinates List, subType:object, range:[0,2]-->
                <RegionCoordinates>
                    <!--opt, object, area coordinates, desc:the origin is the upper-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>
        </ViolationDetectLine>
        <TriggerLine>
            <!--req, object, Triggering Line-->
            <lineName>
                <!--req, enum, Line name, subType:string, desc:"triggerLine" (triggering Line)-->triggerLine
            </lineName>
            <RegionCoordinatesList>
                <!--opt, array, area coordinates List, subType:object, range:[0,2]-->
                <RegionCoordinates>
                    <!--opt, object, area coordinates, desc:the origin is the upper-left corner of the screen-->
                    <positionX>
                        <!--req, int, X-coordinate, range:[0,1000]-->0
                    </positionX>
                </RegionCoordinates>
            </RegionCoordinatesList>
        </TriggerLine>
    </LaneParam>
</LaneParamList>

```

```

<positionY>
    <!--req, int, Y-coordinate, range:[0,1000]-->0
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</TriggerLine>
<forwardRadarIO opt="I01,I02,I03,I04">
    <!--opt, enum, Linked I/O output No. of forward radar, subType:string, dep:and,{$.PostPRS.triggerType,eq,HRT}, attr:opt{req, string}, desc:"I01",
    "I02", "I03", "I04"-->I01
</forwardRadarIO>
<backwardRadarIO opt="I01,I02,I03,I04">
    <!--opt, enum, Linked I/O output No. of backward radar, subType:string, dep:and,{$.PostPRS.triggerType,eq,HRT}, attr:opt{req, string}, desc:"I01",
    "I02", "I03", "I04"-->I01
</backwardRadarIO>
</LaneParam>
</LaneParamList>
<VirtualLane>
    <!--req, object, virtual line-->
<lineName>
    <!--req, enum, Line name, subType:string, desc:"LaneRightBoundaryLine" (right border Line of Lane)-->lineRightBoundaryLine
</lineName>
</VirtualLane>
<sceneMode>
    <!--opt, enum, scene mode, subType:string, desc:"commonEntrance" (common entrance and exit scene), "tollGate" (toll station scene (vehicles will stay
for longer time)), "ParkingEntrance" (entrance and exit of underground parking lot scene (there will be dark day and night))-->commonEntrance
</sceneMode>
<capPicMode>
    <!--opt, enum, capture mode, subType:string, desc:captured picture type: "scene"-scene picture,"sceneCloseup"-scene picture and close-up picture-->scene
</capPicMode>
<humanNonMotorCaptureEnable>
    <!--ro, opt, bool, whether to enable capturing vehicles, non-motor vehicles, and pedestrian-->true
</humanNonMotorCaptureEnable>
</PostPRS>

```

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:the detailed information of custom error returned by device
applications, which is used for fast debugging-->badXmlFormat
    </description>
</ResponseStatus>

```

12.5.1.36 Get the parameters of ANPR system

Request URL

GET /ISAPI/ITC/TriggerMode/postPRS

Query Parameter

None

Request Message

None

Response Message

```

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

<PostPRS xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, all detection lines, attr:version{req, string, protocolVersion}-->
    <triggerType>
        <!--ro, req, enum, triggering mode, subType:string, desc:"videoDetection", "IO", "RS485", "radarDetection", "HRT" (double radars mixed traffic)-->
        videoDetection
    </triggerType>
    <captureMode>
        <!--ro, req, enum, capture mode, subType:string, dep:and,{$.PostPRS.triggerType,eq,videoDetection}, desc:"strobe", "flashing"-->strobe
    </captureMode>

```

```

</captureMode>
<totalLaneNum>
  <!--ro, req, int, number of total lanes, range:[0,10]-->0
</totalLaneNum>
<LaneParamList size="5">
  <!--ro, opt, array, Lane parameter list, subType:object, attr:size{req, int}-->
  <LaneParam>
    <!--ro, opt, object, Lane parameters-->
    <id>
      <!--ro, req, int, ID-->0
    </id>
    <laneNo>
      <!--ro, req, int, Lane No., range:[0,10]-->0
    </laneNo>
    <relatedDriveWay>
      <!--ro, req, int, Linked Lane No., range:[1,99]-->0
    </relatedDriveWay>
    <enterExitDirectionType>
      <!--ro, req, enum, direction type of entrance and exit, subType:int, desc:this node is valid when enterExitDetectEnable is "true"; 0 (unknown), 1
      (enter up to down), 2 (exit up to down)-->0
    </enterExitDirectionType>
    <captureMode>
      <!--ro, req, enum, capture mode, subType:string, dep:and,{$.PostPRS.triggerType,eq,videoDetection}, desc:"strobe", "flashing"-->strobe
    </captureMode>
    <RS485No>
      <!--ro, opt, int, 485 No., range:[0,10], dep:and,{$.PostPRS.triggerType,eq,RS485}-->0
    </RS485No>
    <iOno>
      <!--ro, opt, int, IO port No., range:[0,10], dep:and,{$.PostPRS.triggerType,eq,IO}-->0
    </iOno>
    <iodefStatus>
      <!--ro, opt, enum, IO default status, subType:string, dep:and,{$.PostPRS.triggerType,eq,IO}, desc:"risingEdge", "fallingEdge"-->risingEdge
    </iodefStatus>
    <ViolationDetectLine>
      <!--ro, req, object, draw violation detection line on the single Lane-->
      <lineName>
        <!--ro, req, enum, Line name, subType:string, desc:"laneLine"-->laneLine
      </lineName>
      <normalizedScreenSize>
        <!--ro, req, object, normalized coordinates-->
        <normalizedScreenWidth>
          <!--ro, req, int, normalized width, range:[0,1000]-->1
        </normalizedScreenWidth>
        <normalizedScreenHeight>
          <!--ro, req, int, normalized height, range:[0,1000]-->1
        </normalizedScreenHeight>
      </normalizedScreenSize>
      <RegionCoordinatesList>
        <!--ro, opt, array, area coordinates list, subType:object, range:[0,2]-->
        <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]-->0
          </positionX>
          <positionY>
            <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </ViolationDetectLine>
    <TriggerLine>
      <!--ro, req, object, Triggering Line-->
      <lineName>
        <!--ro, req, enum, Line name, subType:string, desc:"triggerLine" (triggering line)-->triggerLine
      </lineName>
      <normalizedScreenSize>
        <!--ro, req, object, normalized coordinates-->
        <normalizedScreenWidth>
          <!--ro, req, int, normalized width, range:[0,1000]-->1
        </normalizedScreenWidth>
        <normalizedScreenHeight>
          <!--ro, req, int, normalized height, range:[0,1000]-->1
        </normalizedScreenHeight>
      </normalizedScreenSize>
      <RegionCoordinatesList>
        <!--ro, opt, array, area coordinates list, subType:object, range:[0,2]-->
        <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]-->0
          </positionX>
          <positionY>
            <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </TriggerLine>
    <forwardRadarIO opt="I01,I02,I03,I04">
      <!--ro, opt, enum, Linked I/O output No. of forward radar, subType:string, dep:and,{$.PostPRS.triggerType,eq,HRT}, attr:opt{req, string},
      desc:"I01", "I02", "I03", "I04"-->I01
    </forwardRadarIO>
    <backwardRadarIO opt="I01,I02,I03,I04">
      <!--ro, opt, enum, Linked I/O output No. of backward radar, subType:string, dep:and,{$.PostPRS.triggerType,eq,HRT}, attr:opt{req, string},
      desc:"I01", "I02", "I03", "I04"-->I01
    </backwardRadarIO>
  </LaneParamList>

```

```

        <!-- ro , req , sub , type -->
        </backwardRadarIO>
    </LaneParam>
</LaneParamList>
<VirtualLane>
    <!-- ro, req, object, virtual Line-->
    <lineName>
        <!-- ro, req, enum, line name, subType:string, desc:"LaneRightBoundaryLine" (right border line of Lane)-->laneRightBoundaryLine
    </lineName>
    <NormalizedScreenSize>
        <!-- ro, req, object, normalized coordinates-->
        <normalizedScreenWidth>
            <!-- ro, req, int, normalized width, range:[0,1000]-->1
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!-- ro, req, int, normalized height, range:[0,1000]-->1
        </normalizedScreenHeight>
    </NormalizedScreenSize>
    <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]-->0
        </positionX>
        <positionY>
            <!-- ro, req, int, Y-coordinate, range:[0,1000]-->0
        </positionY>
    </RegionCoordinates>
</VirtualLane>
<noPlatCarCap>
    <!-- ro, opt, bool, no plate vehicle capture-->true
</noPlatCarCap>
<sceneMode>
    <!-- ro, opt, enum, scene mode, subType:string, desc:"commonEntrance" (common entrance and exit scene), "tollGate" (toll station scene),
"ParkingEntrance" (entrance and exit of parking lot scene)-->commonEntrance
    <sceneMode>
    <capPicMode>
        <!-- ro, opt, enum, capture mode, subType:string, desc:captured picture type: "scene"-scene picture, "sceneCloseup"-scene picture and close-up picture-->
    <scene>
        <capPicMode>
        <radarDetection>
            <!-- ro, opt, object, radar detection mode-->
            <detectDistance1>
                <!-- ro, req, int, radar detection distance 1, range:[0,2000], unit:cm-->0
            </detectDistance1>
            <detectDistance2>
                <!-- ro, req, int, radar detection distance 2, range:[0,2000], unit:cm-->0
            </detectDistance2>
            <alarmDistance>
                <!-- ro, req, int, radar alarm distance, range:[0,600], unit:cm-->0
            </alarmDistance>
        </radarDetection>
        <detectPosition>
            <!-- ro, opt, enum, detection position, subType:string, desc:"bodyStock" (vehicle body), "headStock" (vehicle head)-->headStock
        </detectPosition>
        <recordEnable>
            <!-- ro, opt, bool, whether to enable recording-->true
        </recordEnable>
        <recordType>
            <!-- ro, opt, enum, recording type, subType:string, desc:"preRecord", "delayRecord"-->preRecord
        </recordType>
        <beginTime>
            <!-- ro, opt, enum, start time of pre-record when capturing pictures, subType:int, desc:0 (default: the 2nd picture), 1 (the 1st picture), 2 (the 2nd
picture), 3 (the 3rd picture)-->0
        </beginTime>
        <preRecordTime>
            <!-- ro, opt, int, pre-record, range:[0,100], unit:s-->0
        </preRecordTime>
        <recordDelayTime>
            <!-- ro, opt, int, post-record, range:[0,100], unit:s-->0
        </recordDelayTime>
        <recordTimeOut>
            <!-- ro, opt, int, timeout threshold, range:[0,100], unit:s-->0
        </recordTimeOut>
        <enterExitDetectEnable>
            <!-- ro, opt, bool, whether to enable entrance and exit detection-->true
        </enterExitDetectEnable>
        <twoWheeICaptureEnable>
            <!-- ro, opt, bool, whether to enable capturing two wheelers-->true
        </twoWheeICaptureEnable>
        <humanNonMotorCaptureEnable>
            <!-- ro, opt, bool, whether to enable capturing vehicles, non-motor vehicles, and pedestrian-->true
        </humanNonMotorCaptureEnable>
    </PostPRS>

```

12.5.1.37 Get the trigger mode configuration capability of ANPR system

Request URL

GET /ISAPI/ITC/TriggerMode/postPRS/capabilities

Query Parameter

None

Request Message

None

Response Message

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

<PostPRS xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, road parameters, attr:version{req, string, protocolVersion}-->
  <triggerType opt="videoDetection,IO,RS485,radarDetection,HRT">
    <!--ro, req, enum, condition for triggering alarm output, subType:string, attr:opt{req, string}, desc:"videoDetection", "IO", "RS485", "radarDetection", "HRT" (double radars mixed traffic)-->videoDetection
  </triggerType>
  <captureMode opt="strobe,flashing">
    <!--ro, req, enum, capture mode, subType:string, attr:opt{req, string}, desc:"strobe", "flashing", it is valid when triggerType is "videoDetection"-->
  >strobe
  </captureMode>
  <totalLaneNum min="0" max="10">
    <!--ro, req, int, number of total lanes, range:[0,10], attr:min{req, int},max{req, int}-->0
  </totalLaneNum>
  <LaneParamList size="5">
    <!--ro, opt, object, List of Lane parameters, attr:size{req, int}-->
    <LaneParam>
      <!--ro, opt, object, Lane parameters-->
      <id>
        <!--ro, req, int, ID, range:[1,5]-->0
      </id>
      <laneNo min="0" max="10">
        <!--ro, req, int, Lane No., range:[0,10], attr:min{req, int},max{req, int}-->0
      </laneNo>
      <laneDirectionType opt="0,1,2,3,4,5,6,7,8">
        <!--ro, req, enum, Lane direction type, subType:int, attr:opt{req, string}, desc:0 (unknown),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)-->0
      </laneDirectionType>
      <enterExitDirectionType opt="0,1,2">
        <!--ro, req, enum, direction type of entrance and exit, subType:int, attr:opt{req, string}, desc:this node is valid when enterExitDetectEnable is "true"; 0 (unknown), 1 (enter up to down), 2 (exit up to down)-->0
      </enterExitDirectionType>
      <relatedDriveWay min="1" max="99">
        <!--ro, req, int, Linked Lane No., range:[1,99], attr:min{req, int},max{req, int}-->0
      </relatedDriveWay>
      <captureMode opt="strobe,flashing">
        <!--ro, req, enum, Capture Mode, subType:string, attr:opt{req, string}, desc:"strobe,flashing", it is valid when triggerType is "videoDetection"-->
      >strobe
      </captureMode>
      <RS485No min="0" max="10">
        <!--ro, opt, int, 485 No., range:[0,10], attr:min{req, int},max{req, int}, desc:it is valid when triggerType is "RS485"-->0
      </RS485No>
      <IONo min="0" max="10">
        <!--ro, opt, int, IO port No., range:[0,10], attr:min{req, int},max{req, int}, desc:it is valid when triggerType is "IO"-->0
      </IONo>
      <IODefStatus opt="risingEdge,fallingEdge">
        <!--ro, opt, enum, IO default status, subType:string, attr:opt{req, string}, desc:it is valid when triggerType is "IO"; "risingEdge", "fallingEdge"-->
      >risingEdge
      </IODefStatus>
      <ViolationDetectLine>
        <!--ro, req, object, draw violation detection line on the single Lane-->
        <lineName min="0" max="64">
          <!--ro, req, enum, "LaneLine", subType:string, attr:min{req, int},max{req, int}, desc:"LaneLine"-->laneLine
        </lineName>
        <NormalizedScreenSize>
          <!--ro, req, object, normalized coordinates-->
          <normalizedScreenWidth>
            <!--ro, req, int, normalized width, range:[0,1000]-->1
          </normalizedScreenWidth>
          <normalizedScreenHeight>
            <!--ro, req, int, normalized height, range:[0,1000]-->1
          </normalizedScreenHeight>
        </NormalizedScreenSize>
        <RegionCoordinatesList size="5">
          <!--ro, opt, array, area coordinates List, subType:object, range:[0,2], attr:size{req, int}-->
          <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]-->0
            </positionX>
            <positionY>
              <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
            </positionY>
          </RegionCoordinates>
        </RegionCoordinatesList>
      </ViolationDetectLine>
      <TriggerLine>
        <!--ro, req, object, Triggering Line-->
        <lineName min="0" max="64">
          <!--ro, req, enum, "LaneLine", subType:string, attr:min{req, int},max{req, int}, desc:"LaneLine"-->triggerLine
        </lineName>
      </TriggerLine>
    </LaneParam>
  </LaneParamList>
</PostPRS>
```

```

<NormalizedScreenSize>
    <!--ro, req, object, normalized coordinates-->
    <normalizedScreenWidth>
        <!--ro, req, int, normalized width, range:[0,1000]-->1
    </normalizedScreenWidth>
    <normalizedScreenHeight>
        <!--ro, req, int, normalized height, range:[0,1000]-->1
    </normalizedScreenHeight>
</NormalizedScreenSize>
<RegionCoordinatesList size="5">
    <!--ro, opt, array, area coordinates list, subType:object, range:[0,2], attr:size{req, int}-->
    <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]-->0
        </positionX>
        <positionY>
            <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
        </positionY>
    </RegionCoordinates>
</RegionCoordinatesList>
</TriggerLine>
<forwardRadarIO opt="I01,I02,I03,I04">
    <!--ro, opt, enum, Linked I/O output No. of forward radar, subType:string, dep:and,{$.PostPRS.triggerType,eq,HRT}, attr:opt{req, string}, desc:"I01", "I02", "I03", "I04"-->I01
</forwardRadarIO>
<backwardRadarIO opt="I01,I02,I03,I04">
    <!--ro, opt, enum, Linked I/O output No. of backward radar, subType:string, dep:and,{$.PostPRS.triggerType,eq,HRT}, attr:opt{req, string}, desc:"I01", "I02", "I03", "I04"-->I01
</backwardRadarIO>
</LaneParam>
</LaneParamList>
<VirtualLane>
    <!--ro, req, object, virtual Line-->
    <lineName min="0" max="64">
        <!--ro, req, enum, "LaneLine", subType:string, attr:min{req, int},max{req, int}, desc:"LaneRightBoundaryLine" (right border Line of Lane)-->laneRightBoundaryLine
    </lineName>
    <NormalizedScreenSize>
        <!--ro, req, object, normalized coordinates-->
        <normalizedScreenWidth>
            <!--ro, req, int, normalized width, range:[0,1000]-->1
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!--ro, req, int, normalized height, range:[0,1000]-->1
        </normalizedScreenHeight>
    </NormalizedScreenSize>
    <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]-->0
        </positionX>
        <positionY>
            <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
        </positionY>
    </RegionCoordinates>
</VirtualLane>
<noPlatCarCap>
    <!--ro, opt, bool, No Plate Vehicle Capture-->true
</noPlatCarCap>
<sceneMode opt="commonEntrance,tollGate,ParkingEntrance">
    <!--ro, opt, enum, scene mode, subType:string, attr:opt{req, string}, desc:"commonEntrance" (common entrance and exit), "tollGate" (toll station (vehicles will stay for longer time)), "ParkingEntrance"->entrance of underground parking lot-->commonEntrance
</sceneMode>
<capPicMode opt="scene,sceneCloseup">
    <!--ro, opt, enum, Capture Mode, subType:string, attr:opt{req, string}, desc:"scene" (scene picture), "sceneCloseup" (close-up picture)-->scene
</capPicMode>
<radarDetection>
    <!--ro, opt, object, radar detection mode-->
    <detectDistance1 min="0" max="2000">
        <!--ro, req, int, radar detection distance, range:[0,2000], unit:cm, attr:min{req, int},max{req, int}-->0
    </detectDistance1>
    <detectDistance2 min="0" max="2000">
        <!--ro, req, int, radar detection distance, range:[0,2000], unit:cm, attr:min{req, int},max{req, int}-->0
    </detectDistance2>
    <alarmDistance min="0" max="600" def="300">
        <!--ro, req, int, radar alarm distance, range:[0,600], unit:cm, attr:min{req, int},max{req, int},def{req, int}-->0
    </alarmDistance>
</radarDetection>
<detectPosition opt="tailStock,headStock">
    <!--ro, opt, enum, detection position, subType:string, attr:opt{req, string}, desc:"bodyStock" (vehicle body), "headStock" (vehicle head)-->headStock
</detectPosition>
<recordEnable opt="false,true">
    <!--ro, opt, bool, whether to enable recording, attr:opt{req, string}-->true
</recordEnable>
<recordType opt="delayRecord,preRecord">
    <!--ro, opt, enum, recording type, subType:string, attr:opt{req, string}, desc:"preRecord", "delayRecord"-->preRecord
</recordType>
<preRecordTime min="0" max="100">
    <!--ro, opt, int, pre-record, range:[0,100], unit:s, attr:min{req, int},max{req, int}-->0
</preRecordTime>
<recordDelayTime min="0" max="100">
    <!--ro, opt, int, post-record, range:[0,100], unit:s, attr:min{req, int},max{req, int}-->0
</recordDelayTime>

```

```

</recordDelayTime>
<recordTimeOut min="0" max="100">
    <!--ro, opt, int, timeout threshold, range:[0,100], unit:s, attr:min{req, int},max{req, int}-->0
</recordTimeOut>
<enterExitDetectEnable opt="false,true">
    <!--ro, opt, bool, whether to enable entrance and exit detection, attr:opt{req, string}-->true
</enterExitDetectEnable>
<twoWheelCaptureEnable opt="false,true">
    <!--ro, opt, bool, whether to enable capturing two wheelers, attr:opt{req, string}-->true
</twoWheelCaptureEnable>
<humanNonMotorCaptureEnable opt="false,true">
    <!--ro, opt, bool, whether to enable capturing vehicles, non-motor vehicles, and pedestrian, attr:opt{req, string}-->true
</humanNonMotorCaptureEnable>
</PostPRS>

```

12.5.1.38 Get statuses of switching trigger mode

Request URL

GET /ISAPI/ITC/TriggerMode/switchStatus?format=json&channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	channel No. (this optional parameter is supported when isSupportChannelSwitchStatus is "true" returned from GET /ISAPI/ITC/TriggerMode/capabilities)

Request Message

None

Response Message

```
{
    "switchStatus": "completed"
    /*ro, req, enum, status, subType:string, desc:"completed", "executing"*/
}
```

12.5.1.39 Get the trigger mode configuration capability of traffic incidents

Request URL

GET /ISAPI/ITC/TriggerMode/trafficIncident/capabilities?format=json

Query Parameter

None

Request Message

None

Response Message

```
{
    "TrafficIncidentCap": {
        /*ro, opt, object, configuration capability of traffic incident detection*/
        "enabled": [true, false],
        /*ro, req, array, whether to enable traffic incident detection, subType:bool*/
        "sceneMode": "highway,urban,tunnel",
        /*ro, opt, string, scene mode, desc:"highway", "urban" (urban road), "tunnel"*/
        "isSupportChannelTrafficIncident": false
        /*ro, opt, bool, whether it support configuring parameters of traffic incident detection, desc:if this node is true, it indicates that channelID can be set in /ISAPI/ITC/TriggerMode/trafficIncident?format=json&channelID=<channelID> and /ISAPI/ITC/TriggerMode/trafficIncident/defaultParam?format=json&channelID=<channelID>*/}
    }
}
```

12.5.1.40 Set the trigger mode parameters of traffic incidents

Request URL

PUT /ISAPI/ITC/TriggerMode/trafficIncident?format=json&channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	channel No. (this optional parameter is supported when isSupportChannelTrafficIncident is "true" returned from GET /ISAPI/ITC/TriggerMode/trafficIncident/capabilities?format=json)

Request Message

```
{
  "TrafficIncident": {
    /*opt, object, parameters of traffic incident detection*/
    "enabled": true,
    /*req, bool, whether to enable the traffic incident trigger mode*/
    "sceneMode": "highway"
    /*opt, enum, scene mode, subType:string, desc:"highway", "urban" (urban road), "tunnel"*/
  }
}
```

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:error code*/
  "errorMsg": "ok"
  /*ro, opt, string, error information, desc:this node is required when the value of statusCode is not 1*/
}
```

12.5.1.41 Get the trigger mode parameters of traffic incidents

Request URL

GET /ISAPI/ITC/TriggerMode/trafficIncident?format=json&channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	channel No. (this optional parameter is supported when isSupportChannelTrafficIncident is "true" returned from GET /ISAPI/ITC/TriggerMode/trafficIncident/capabilities?format=json)

Request Message

None

Response Message

```
{
  "TrafficIncident": {
    /*ro, opt, object, parameters of traffic incident detection*/
    "enabled": true,
    /*ro, req, bool, whether to enable the function*/
    "sceneMode": "highway"
    /*ro, opt, enum, scene mode, subType:string, desc:"highway", "urban" (urban road), "tunnel"*/
  }
}
```

12.5.1.42 Get the capability of video intersection violation parameters

Request URL

GET /ISAPI/ITC/TriggerMode/videoPolice/capabilities?channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Camera ID (this optional parameter is supported when isSupportChannelVideoEplice is "true" returned from /ISAPI/ITC/TriggerMode/capabilities).

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<VideoPolice xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, video intersection violation parameters, attr:version{req, string, protocolVersion}-->
  <intervalType opt="time,distance">
    <!--ro, req, enum, burst interval type, subType:string, attr:opt{req, string}, desc:"time" (by time), "distance" (by distance)-->
  </intervalType>
  <interval min="1" max="1000">
    <!--ro, req, int, time interval, range:[1,1000], attr:min{req, int},max{req, int}-->
  </interval>
  <post>
    <!--ro, req, bool, whether to enable checkpoint detection-->true
  </post>
  <postCapNo opt="1,3">
    <!--ro, req, int, captured pictures of checkpoint, range:[1,3], attr:opt{req, string}-->1
  </postCapNo>
  <driveLine>
    <!--ro, req, bool, whether to enable driving on Lane Line detection-->true
  </driveLine>
  <driveLineCapNo opt="2,3">
    <!--ro, req, int, capture times, range:[2,3], attr:opt{req, string}-->0
  </driveLineCapNo>
  <driveLineSensitivity min="0" max="100">
    <!--ro, req, int, over Line sensitivity, range:[0,100], attr:min{req, int},max{req, int}-->0
  </driveLineSensitivity>
  <reverse>
    <!--ro, req, bool, whether to enable wrong-way driving detection-->true
  </reverse>
  <reverseCapNo opt="2,3">
    <!--ro, req, int, number of captured pictures, range:[2,3], attr:opt{req, string}-->0
  </reverseCapNo>
  <redLight>
    <!--ro, req, bool, whether to enable red Light running detection-->true
  </redLight>
  <direction>
    <!--ro, req, bool, whether to enable detection of driving in wrong Lane at intersection-->true
  </direction>
  <intersectionCongest>
    <!--ro, req, bool, whether to enable intersection congestion detection-->true
  </intersectionCongest>
  <nonDriveWay>
    <!--ro, opt, bool, whether to enable detection of motor vehicle on non-motor vehicle Lane-->true
  </nonDriveWay>
  <nonDriveWayCapNo opt="2,3">
    <!--ro, opt, int, capture times, range:[2,3], attr:opt{req, string}-->0
  </nonDriveWayCapNo>
  <changeLane>
    <!--ro, req, bool, whether to enable illegal Lane change detection-->true
  </changeLane>
  <changeLaneCapNo opt="2,3">
    <!--ro, req, int, number of captured pictures, range:[2,3], attr:opt{req, string}-->0
  </changeLaneCapNo>
  <banSign>
    <!--ro, opt, bool, whether to enable ban breaking detection-->true
  </banSign>
  <banSignCapNo opt="2,3">
    <!--ro, opt, int, number of captured pictures, range:[2,3], attr:opt{req, string}-->0
  </banSignCapNo>
  <intersectionStop>
    <!--ro, opt, object, stop vehicle on intersection-->
    <enable opt="true,false">
      <!--ro, req, bool, whether to enable the detection, attr:opt{req, string}-->true
    </enable>
    <lastTime min="10" max="180">
      <!--ro, req, int, congestion duration, unit: second, range:[0,180], attr:min{req, int},max{req, int}-->0
    </lastTime>
    <sensitivity min="1" max="100">
      <!--ro, opt, int, triggering sensitivity, which is between 0 to 100, range:[1,100], attr:min{req, int},max{req, int}-->0
    </sensitivity>
    <capNo opt="3">
      <!--ro, req, int, number of captures, attr:opt{req, string}-->3
    </capNo>
  </intersectionStop>
  <greenLightStop>
    <!--ro, opt, bool, whether to enable detection of parking at green Light-->true
  </greenLightStop>
  <overSpeed>
  </overSpeed>

```

```

<!--ro, req, bool, whether to enable overspeed detection-->true
</overSpeed>
<overSpeedCapNo opt="2,3">
    <!--ro, req, int, number of captured pictures in which the overspeed is detected, range:[2,3], attr:opt{req, string}-->0
</overSpeedCapNo>
<detectMode>
    <!--ro, opt, enum, detection mode, subType:string, desc:"video", "coilLoop" (video and coil)-->video
</detectMode>
<illegalTurn>
    <!--ro, req, bool, whether to enable illegal U-turning detection-->true
</illegalTurn>
<TrafficLight opt="IO,RS485,videoDetect,network,autoDetectTrafficLight">
    <!--ro, req, object, traffic lights, attr:opt{req, string}-->
    <source>
        <!--ro, req, enum, source, subType:string, desc:"IO", "RS485", "videoDetect" (video detection), "network"-->IO
    </source>
</TrafficLight>
<relatedLaneCount min="1" max="6">
    <!--ro, req, int, number of supported lanes, range:[1,6], attr:min{req, int},max{req, int}-->0
</relatedLaneCount>
<relatedDriveWay min="1" max="99">
    <!--ro, req, int, linked lane No., range:[1,99], attr:min{req, int},max{req, int}-->0
</relatedDriveWay>
<recordEnable>
    <!--ro, req, bool, whether to enable recording when traffic violation happened-->true
</recordEnable>
<recordType opt="delayRecord,preRecord">
    <!--ro, req, enum, recording type, subType:string, attr:opt{req, string}, desc:"preRecord" (pre-record), "delayRecord" (post-record)-->preRecord
</recordType>
<preRecordTime min="0" max="100">
    <!--ro, req, int, pre-record time, range:[0,100], unit:s, attr:min{req, int},max{req, int}-->0
</preRecordTime>
<recordDelayTime min="0" max="100">
    <!--ro, req, int, post-record time, range:[0,100], unit:s, attr:min{req, int},max{req, int}-->0
</recordDelayTime>
<recordTimeOut min="0" max="100">
    <!--ro, req, int, recording timeout, range:[0,100], unit:s, attr:min{req, int},max{req, int}-->0
</recordTimeOut>
<signSpeed min="0" max="255">
    <!--ro, opt, int, marked speed limit, range:[0,255], unit:km/h, attr:min{req, int},max{req, int}-->0
</signSpeed>
<speedLimit min="0" max="255">
    <!--ro, opt, int, speed limit, range:[0,255], unit:km/h, attr:min{req, int},max{req, int}-->0
</speedLimit>
<usageType opt="carriageWay,bus,slow,nonmotor,reverse,emergency,banNonMotor,ramp,fast,special">
    <!--ro, req, enum, lane usage, subType:string, attr:opt{req, string}, desc:"carriageway" (roadway), "bus" (bus lane), "slow" (slow lane), "nonmotor" (non-motor lane), "reverse" (contrafow lane), "emergency" (emergency lane), "banNonMotor" (non-motor vehicle vehicles prohibited lane), "ramp"-->
</usageType>
<LaneUsageList size="5">
    <!--ro, opt, object, list of lane purposes, attr:size{req, int}, desc:list of lane purposes-->
    <laneUsage opt="banTrucks,bus,banLeft,banRight,motor,banStraight,variableLane">
        <!--ro, req, enum, lane purpose, subType:string, attr:opt{req, string}, desc:lane usage type: "bus"-bus lane,"banTrucks"-truck prohibited lane,"banLeft"-left turn prohibited lane,"banRight"-right turn prohibited lane,"motor"-motorcycle lane-->banTrucks
    </laneUsage>
</LaneUsageList>
<variableLaneTimeSwitchList size="4">
    <!--ro, opt, object, time period for changeable lanes, attr:size{req, int}-->
    <variableLaneTimeSwitch>
        <!--ro, req, object, time for changeable lanes-->
        <timeId min="1" max="4">
            <!--ro, req, int, index, attr:min{req, int},max{req, int}-->0
        </timeId>
        <startHour min="0" max="23">
            <!--ro, opt, int, start time (hour), attr:min{req, int},max{req, int}-->0
        </startHour>
        <startMinute min="0" max="59">
            <!--ro, opt, int, start time (minute), attr:min{req, int},max{req, int}-->0
        </startMinute>
        <endHour min="0" max="23">
            <!--ro, opt, int, end time (hour), attr:min{req, int},max{req, int}-->0
        </endHour>
        <endMinute min="0" max="59">
            <!--ro, opt, int, end time (minute), attr:min{req, int},max{req, int}-->0
        </endMinute>
    </variableLaneTimeSwitch>
</variableLaneTimeSwitchList>
<laneDirection opt="unkown,left,straight,leftStraight,right,leftRight,rightStraight,leftRightStraight,leftWait,straightWait,strightWaitRight">
    <!--ro, req, enum, driving direction, subType:string, attr:opt{req, string}, desc:"unkown", "Left", "straight", "leftStraight", "right", "leftRight", "rightStraight", "leftRightStraight", "LeftWait", "straightWait", "straightWaitRight"-->leftRightStraight
    </laneDirection>
    </variableLaneTimeSwitch>
</variableLaneTimeSwitchList>
<directionType opt="unkown,left,straight,leftStraight,right,leftRight,rightStraight,leftRightStraight,leftWait,straightWait,strightWaitRight">
    <!--ro, req, enum, driving direction, subType:string, attr:opt{req, string}, desc:direction type,multiple options can be supported at one time-->leftRightStraight
    </directionType>
<laneDirectionType opt="0,1,2,3,4,5,6,7,8">
    <!--ro, req, enum, lane direction type, subType:int, attr:opt{req, string}, desc:0 (unknown),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)-->0
    </laneDirectionType>
<stopLineDistance min="0" max="300">
    <!--ro, req, int, distance between the vehicle and the stop line in the second captured picture of red light running violation, range:[0,300], unit:m, attr:min{req, int},max{req, int}, desc:distance between the vehicle and the stop line in the second captured picture of red light running violation-->0
</stopLineDistance>

```

```

</stopLineInstance>
<loopSensitivity min="1" max="100">
    <!--ro, opt, int, Loop sensitivity, range:[0,100], attr:min{req, int},max{req, int}-->0
</loopSensitivity>
<capPositionOffset min="0" max="300">
    <!--ro, req, int, second captured picture deviation, range:[0,300], attr:min{req, int},max{req, int}, desc:unit: pixel-->50
</capPositionOffset>
<videoDetectLightId min="1" max="12">
    <!--ro, req, int, area ID of video detection traffic Light, range:[1,12], attr:min{req, int},max{req, int}-->1
</videoDetectLightId>
<lightNum min="0" max="12">
    <!--ro, req, int, number of lights, range:[1,3], attr:min{req, int},max{req, int}-->1
</lightNum>
<straightLight>
    <!--ro, req, bool, straight signal-->true
</straightLight>
<leftLight>
    <!--ro, req, bool, left turn signal-->true
</leftLight>
<rightLight>
    <!--ro, req, bool, right turn signal-->true
</rightLight>
<greenLight>
    <!--ro, req, bool, green Light-->true
</greenLight>
<yellowLight>
    <!--ro, req, bool, yellow Light-->true
</yellowLight>
<yellowLightTime min="0" max="10000">
    <!--ro, req, int, yellow time, range:[0,10000], unit:ms, attr:min{req, int},max{req, int}-->0
</yellowLightTime>
<beginTime opt="1, 2, 3">
    <!--ro, req, enum, recording start time, subType:int, attr:opt{req, string}, desc:recording start time-->1
</beginTime>
<deferredFrmNum min="0" max="25">
    <!--ro, opt, int, delay a few seconds to trigger the mode, range:[0,25], attr:min{req, int},max{req, int}-->0
</deferredFrmNum>
<redLightCapNo opt="3,4">
    <!--ro, opt, int, captured picture amount in red Light, range:[3,4], attr:opt{req, string}-->3
</redLightCapNo>
<directionCapNo opt="3,4">
    <!--ro, opt, int, number of captured pictures of driving against direction guidance, range:[3,4], attr:opt{req, string}-->3
</directionCapNo>
<firstShowTailVep opt="0,1,2,3">
    <!--ro, opt, enum, place of the first capture time, subType:int, attr:opt{req, string}, desc:0 (none), 1 (expose vehicle tail), 2 (expose vehicle tail and LPR success), 3(parking over stop Line)-->0
</firstShowTailVep>
<nonMotorPassenger>
    <!--ro, opt, object, Manned Non-Motor Vehicle-->
<enabled opt="true,false">
    <!--ro, req, bool, whether to enable the detection, attr:opt{req, string}-->true
</enabled>
<capNo opt="1,2">
    <!--ro, req, int, the number of captured pictures, range:[1,2], attr:opt{req, string}-->0
</capNo>
<refineMannedIllegalCode opt="true/false">
    <!--ro, opt, bool, whether to enable detailed manned violation code, attr:opt{req, string}-->true
</refineMannedIllegalCode>
</nonMotorPassenger>
<speedDetectMode opt="none,radar">
    <!--ro, opt, enum, speed detection mode, subType:string, attr:opt{req, string}, desc:"none", "radar"-->none
</speedDetectMode>
<Radar>
    <!--ro, opt, object, radar parameters-->
<radarType opt="none,MultiLane,csr,IOBox,SSTK,custom" def="MultiLane">
    <!--ro, opt, string, radar type, attr:opt{req, string},def{req, string}, desc:"none", "MultiLane" (multi-Lane radar), "csr" (TransMicrowave), "IOBox" (radar connecting I/O expansion box), "SSTK" (SensorTech),"custom" (custom type), the default value is MultiLane-->MultiLane
</radarType>
<validRadarSpeedTime min="1" max="9000" def="2000">
    <!--ro, opt, int, radar speed valid time, range:[0,9000], unit:ms, attr:min{req, int},max{req, int},def{req, int}, desc:if this node does not exist, it indicates the function is supported. The default value is 2,000-->0
</validRadarSpeedTime>
<radarLinearCorrection min="0.0" max="2.0" def="1.0">
    <!--ro, opt, float, linear correction parameters, range:[0.0,2.0], attr:min{req, float},max{req, float},def{req, float}, desc:if this node does not exist, it indicates the function is supported. The default value is 1.0, and the value should be accurate to 3 decimal places-->1.0
</radarLinearCorrection>
<radarConstantCorrection min="-100" max="100" def="0">
    <!--ro, opt, int, constant correction parameter,, range:[-100,100], attr:min{req, int},max{req, int},def{req, int}, desc:if this node does not exist, it indicates the function is supported. The default value is 0-->0
</radarConstantCorrection>
<radarRS485 min="1" max="5" def="2">
    <!--ro, opt, int, 485 radar No., range:[1,5], attr:min{req, int},max{req, int},def{req, int}, desc:if this node does not exist, it indicates the function is supported. The default value is 2-->1
</radarRS485>
<Radar>
<lineType opt="unknown,white,singleYellow,doubleYellow,guardRail,noCross,fullDottedLine,dottedFullLine">
    <!--ro, opt, enum, line type, subType:string, attr:opt{req, string}, desc:"unknown", "white" (white solid Line), "singleYellow" (single yellow Line), "doubleYellow" (double yellow Line), "guardRail" (Lane Line with guardrail), "noCross" (Lane Line prohibited to cross by vehicle), "fullDottedLine" (double yellow Line (left: dotted, right: solid)), "dottedFullLine" (double yellow Line (left: solid, right: dotted))-->unknown
</lineType>
<bigCarSignSpeed min="0" max="255">
    <!--ro, opt, int, marked speed limit for large-sized vehicle, attr:min{req, int},max{req, int}-->0
</bigCarSignSpeed>
<bigCarSpeedLimit min="0" max="60">

```

```

<!--ro, opt, int, maximum speed limit for large-sized vehicle, attr:min{req, int},max{req, int}-->0
</bigCarSpeedLimit>
<carHighSpeed min="0" max="255">
    <!--ro, opt, int, abnormal speeding of small-sized vehicle, attr:min{req, int},max{req, int}-->0
</carHighSpeed>
<bigCarHighSpeed min="0" max="255">
    <!--ro, opt, int, abnormal speeding of large-sized vehicle, attr:min{req, int},max{req, int}-->0
</bigCarHighSpeed>
<greenLightStopCapNo opt="3">
    <!--ro, opt, int, captured picture amount in green light, attr:opt{req, string}-->3
</greenLightStopCapNo>
<illegalTurnCapNo opt="3">
    <!--ro, req, int, number of captured violation turing picture(s), attr:opt{req, string}-->3
</illegalTurnCapNo>
</VideoPolice>

```

12.5.1.43 Get the trigger mode parameters of video intersection violation system

Request URL

GET /ISAPI/ITC/TriggerMode/videoPolice?channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Camera ID (this optional parameter is supported when isSupportChannelVideoPolice is "true" returned from /ISAPI/ITC/TriggerMode/capabilities).

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<VideoPolice xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, the trigger mode parameters of video intersection violation system, attr:version{req, string, protocolVersion}-->
    <enabled>
        <!--ro, req, bool, whether to enable intersection violation system-->true
    </enabled>
    <relatedLaneCount>
        <!--ro, req, int, number of linked lanes, range:[1,4]-->0
    </relatedLaneCount>
    <TrafficLight>
        <!--ro, req, object, traffic lights-->
        <source>
            <!--ro, req, enum, signal source, subType:string, desc:"IO, RS485, videoDetect, network"-->IO
        </source>
        <IOLightList>
            <!--ro, opt, array, list, subType:object-->
            <IOLight>
                <!--ro, opt, object, I/O access-->
                <lightId>
                    <!--ro, req, int, traffic light No., range:[1,6]-->1
                </lightId>
                <lightType>
                    <!--ro, req, enum, guiding direction of traffic light, subType:string, desc:"left, straight, right"-->straight
                </lightType>
                <relatedIO>
                    <!--ro, req, enum, linked I/O input, subType:string, desc:"NON_IO, I01, I02, I03, I04, I05, I06, I07, I08"-->I07
                </relatedIO>
                <redLightStatus>
                    <!--ro, req, enum, red light level status, subType:string, desc:"low" (Low-Level red light), "high" (high-Level red light)-->low
                </redLightStatus>
            </IOLight>
        </IOLightList>
        <RS485LightList>
            <!--ro, opt, array, list, subType:object-->
            <RS485Light>
                <!--ro, opt, object, RS-485 access-->
                <lightId>
                    <!--ro, req, int, traffic light No., range:[1,6]-->0
                </lightId>
                <lightType>
                    <!--ro, req, enum, guiding direction of traffic light, subType:string, desc:"left, right, straight"-->straight
                </lightType>
                <relatedLightChan>
                    <!--ro, req, int, detector channel No. linked to red light, range:[0,16]-->0
                </relatedLightChan>
                <yellowLightChan>
                    <!--ro, req, int, detector channel No. linked to yellow light, range:[0,16]-->0
                </yellowLightChan>
            </RS485Light>
        </RS485LightList>
    </TrafficLight>
</VideoPolice>

```

```

</RS485LightList>
<NetworkLightList>
  <!--ro, opt, array, List, subType:object-->
  <NetworkLight>
    <!--ro, opt, object, network indicator light-->
    <lightId>
      <!--ro, req, int, traffic Light No., range:[1,6]-->0
    </lightId>
    <lightType>
      <!--ro, req, enum, guiding direction of traffic Light, subType:string, desc:"left", "right", "straight"-->straight
    </lightType>
    <relatedLightChan>
      <!--ro, req, int, detector channel No. Linked to red Light, range:[0,16]-->0
    </relatedLightChan>
    <yellowLightChan>
      <!--ro, req, int, detector channel No. Linked to yellow Light, range:[0,16]-->0
    </yellowLightChan>
  </NetworkLight>
</NetworkLightList>
<videoDetectLightList>
  <!--ro, opt, array, List, subType:object-->
  <videoDetectLight>
    <!--ro, opt, object, video detection access-->
    <videoDetectLightId>
      <!--ro, req, int, video detection signal No., range:[1,12]-->0
    </videoDetectLightId>
    <lightNum>
      <!--ro, req, int, number of traffic lights, range:[0,3]-->0
    </lightNum>
    <straightLight>
      <!--ro, req, bool, whether the straight driving Light is on-->true
    </straightLight>
    <leftLight>
      <!--ro, req, bool, whether the left-turn Light is on-->true
    </leftLight>
    <rightLight>
      <!--ro, req, bool, whether the right-turn Light is on-->true
    </rightLight>
    <redLight>
      <!--ro, req, bool, whether the red Light is on-->true
    </redLight>
    <greenLight>
      <!--ro, req, bool, whether the green Light is on-->true
    </greenLight>
    <yellowLight>
      <!--ro, req, bool, whether the yellow Light is on-->true
    </yellowLight>
    <yellowLightTime>
      <!--ro, req, int, yellow Light duration, range:[0,10], unit:s-->0
    </yellowLightTime>
    <rectRegion>
      <!--ro, req, object, traffic light region, desc:the origin is the upper-left corner of the screen-->
      <positionLeft>
        <!--ro, req, int, Left boundary, range:[0,65535]-->0
      </positionLeft>
      <positionTop>
        <!--ro, req, int, top boundary, range:[0,65535]-->0
      </positionTop>
      <positionRight>
        <!--ro, req, int, right boundary, range:[0,65535]-->0
      </positionRight>
      <positionBottom>
        <!--ro, req, int, bottom boundary, range:[0,65535]-->0
      </positionBottom>
    </rectRegion>
  </videoDetectLight>
</videoDetectLightList>
</TrafficLight>
<LaneParamList>
  <!--ro, req, array, Lane parameter List, subType:object-->
</LaneParamList>
<VirtualLaneList>
  <!--ro, req, array, List, subType:object-->
  <VirtualLane>
    <!--ro, req, object, virtual Lane Line-->
    <lineName>
      <!--ro, req, enum, detection Line name, subType:string, desc:"laneLine, stopLine, redlightLine, cancelline, waitLine"-->laneRightBoundaryLine
    </lineName>
    <lineType>
      <!--ro, opt, enum, detection Line type, subType:string, desc:"unknown", "white" (white solid Line), "singleYellow" (single yellow Line), "doubleYellow" (double yellow Line), "guardRail" (Lane Line with guardrail), "noCross" (Lane Line prohibited to cross by vehicle)-->unknown
    </lineType>
    <RegionCoordinatesList>
      <!--ro, opt, array, region coordinate List, subType:object, range:[0,2]-->
      <RegionCoordinates>
        <!--ro, opt, object, region coordinates, desc:the origin is the upper-left corner of the screen-->
        <positionX>
          <!--ro, req, int, X-coordinate, range:[0,1000]-->0
        </positionX>
        <positionY>
          <!--ro, req, int, Y-coordinate-->0
        </positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
  </VirtualLane>
</VirtualLaneList>

```

```

<!--ro, object, capture linkage, desc:pedestrian linkage-->
</VirtualLane>
</VirtualLaneList>
<Posting>
  <!--ro, opt, object, positioning frame-->
  <postingEnable>
    <!--ro, req, bool, whether to enable positioning frame-->true
  </postingEnable>
  <postingSize>
    <!--ro, req, enum, positioning frame size, subType:string, desc:"small" (300 pixels), "medium" (400 pixels), "large" (500 pixels)-->small
  </postingSize>
</Posting>
<leftNotYieldStraight>
  <!--ro, opt, object, left turn not yield to straight-->
  <enable>
    <!--ro, req, bool, whether to enable detection of left-turn without straight driving-->true
  </enable>
  <capNo>
    <!--ro, opt, int, range:[3,3]-->3
  </capNo>
</leftNotYieldStraight>
<rightNotYieldLeft>
  <!--ro, opt, object, right turn not yield to left turn-->
  <enable>
    <!--ro, req, bool, whether to enable detection of left-turn without straight driving-->true
  </enable>
  <capNo>
    <!--ro, opt, int, range:[3,3]-->3
  </capNo>
</rightNotYieldLeft>
<uTurnNotYieldGoingStraight>
  <!--ro, opt, object, U-Turn not Yield to Straight-->
  <enable>
    <!--ro, req, bool, whether to enable detection of left-turn without straight driving-->true
  </enable>
  <capNo>
    <!--ro, opt, int, range:[3,3]-->3
  </capNo>
</uTurnNotYieldGoingStraight>
<leftSideDrivingAfterLeftTurn>
  <!--ro, opt, object, turn left in short turn radius-->
  <enable>
    <!--ro, req, bool, whether to enable detection of left-turn without straight driving-->true
  </enable>
  <capNo>
    <!--ro, opt, int, range:[3,3]-->3
  </capNo>
</leftSideDrivingAfterLeftTurn>
<failYieldPed>
  <!--ro, opt, object, failing to yield for pedestrians-->
  <enable>
    <!--ro, req, bool, whether to enable detection of left-turn without straight driving-->true
  </enable>
<vehicleFromDirection>
  <!--ro, req, enum, driving direction of the vehicle which does not yield to pedestrian, subType:string, desc:"Front, Back"-->Front
</vehicleFromDirection>
<noDisplacementPedEnabled>
  <!--ro, req, bool, whether to enable no displacement capture of pedestrian-->true
</noDisplacementPedEnabled>
<failYieldPedLineList>
  <!--ro, opt, array, List, subType:object-->
  <failYieldPedLine>
    <!--ro, opt, object, failing to yield for pedestrians line-->
    <lineId>
      <!--ro, req, int, Line ID, range:[1,2]-->0
    </lineId>
    <lineName>
      <!--ro, req, string, detection line name, range:[0,64]-->test
    </lineName>
  <RegionCoordinatesList>
    <!--ro, opt, array, region coordinate list, subType:object, range:[0,2]-->
    <RegionCoordinates>
      <!--ro, opt, object, region coordinates, desc:the origin is the upper-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>
  </failYieldPedLine>
</failYieldPedLineList>
<notification>
  <!--ro, opt, object, capture linkage, desc:pedestrian linkage-->
  <enable>
    <!--ro, req, bool, whether to enable detection of left-turn without straight driving-->true
  </enable>
  <ip>
    <!--ro, opt, string, IP address of traffic camera, range:[0,32]-->test
  </ip>
  <port>
    <!--ro, opt, int, port No. of traffic camera, range:[0,65535]-->1
  </port>

```

```

<username>
    <!--ro, opt, string, user name of traffic camera, range:[0,32]-->test
</username>
<password>
    <!--ro, opt, string, password of traffic camera, range:[0,16]-->test
</password>
<sensitivity>
    <!--ro, req, int, detection sensitivity, range:[0,100]-->0
</sensitivity>
<RS485Port>
    <!--ro, req, int, RS-485 port No. for the traffic Light to access the camera, range:[1,10]-->1
</RS485Port>
</notification>
</failieldPed>
<illegalParking>
    <!--ro, opt, object, illegal parking-->
<enable>
    <!--ro, req, bool, whether to enable detection of left-turn without straight driving-->true
</enable>
<sensitivity>
    <!--ro, req, int, detection sensitivity, range:[0,100]-->0
</sensitivity>
<lastTime>
    <!--ro, req, int, congestion duration, range:[1,180], unit:s-->0
</lastTime>
<illegalParkingLineList>
    <!--ro, opt, array, List of illegal parking Lines, subType:object, range:[0,3]-->
<illegalParkingLine>
    <!--ro, opt, object, illegal parking lines-->
<lineId>
    <!--ro, req, int, Line ID, range:[1,3]-->0
</lineId>
<lineName>
    <!--ro, req, string, detection line name, range:[0,64]-->test
</lineName>
<RegionCoordinatesList>
    <!--ro, opt, array, List, subType:object, range:[0,2]-->
<RegionCoordinates>
    <!--ro, opt, object, region coordinates, desc:the origin is the upper-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>
</illegalParkingLine>
</illegalParkingLineList>
<capNo>
    <!--ro, opt, int, range:[3,3]-->3
</capNo>
</illegalParking>
<detectMode>
    <!--ro, opt, enum, detection mode, subType:string, desc:"video", "coilLoop" (video and coil)-->video
</detectMode>
<helmet>
    <!--ro, opt, bool, Without Helmet-->true
</helmet>
<helmetCapNo>
    <!--ro, opt, int, number of captured pictures in which the driver does not wear the helmet, range:[1,3]-->1
</helmetCapNo>
<nonMotorExist>
    <!--ro, opt, bool, Non-Motor Vehicle on Motor Vehicle Lane-->true
</nonMotorExist>
<nonMotorExistCapNo>
    <!--ro, opt, int, number of captured pictures in which the motor Lane is occupied by non-motor vehicle, range:[2,3]-->2
</nonMotorExistCapNo>
<capType>
    <!--ro, opt, enum, "vehicle,nonMotor,human,face", subType:string, desc:"vehicle, nonMotor, human, face"-->mixObject
</capType>
<deferredFrmNum>
    <!--ro, req, int, delay a few seconds to trigger the mode, range:[0,25]-->0
</deferredFrmNum>
<speedDetectMode>
    <!--ro, opt, enum, speed detection mode, subType:string, desc:"none", "radar"-->none
</speedDetectMode>
<uphone>
    <!--ro, opt, object, making a phone call-->
<enable>
    <!--ro, req, bool-->true
</enable>
<capNo>
    <!--ro, req, int, number of captured pictures, range:[1,3]-->2
</capNo>
</uphone>
<greenLightStopCapNo>
    <!--ro, opt, int, range:[3,3]-->3
</greenLightStopCapNo>
<illegalTurnCapNo>
    <!--ro, opt, int, range:[3,3]-->3
</illegalTurnCapNo>
<nonMotorShed>
    <!--ro, opt, object-->

```

```

<!--ro, opt, object-->
<enabled>
  <!--ro, req, bool-->true
</enabled>
<capNo>
  <!--ro, req, int, range:[1,2]-->1
</capNo>
</nonMotorShed>
<NonMotorCrossing>
  <!--ro, opt, object-->
  <enabled>
    <!--ro, opt, bool-->true
  </enabled>
  <lastTime>
    <!--ro, req, int, range:[10,180], unit:s-->0
  </lastTime>
  <sensitivity>
    <!--ro, opt, int, range:[1,100]-->0
  </sensitivity>
</nonMotorStopLine>
<RegionCoordinatesList>
  <!--ro, opt, array, subType:object, range:[0,2]-->
  <RegionCoordinates>
    <!--ro, opt, object-->
    <positionX>
      <!--ro, req, int, range:[0,1000]-->0
    </positionX>
    <positionY>
      <!--ro, req, int, range:[0,1000]-->0
    </positionY>
  </RegionCoordinates>
</RegionCoordinatesList>
</nonMotorCrossing>
</VideoEpolice>

```

12.5.1.44 Set the trigger mode parameters of video intersection violation system

Request URL

PUT /ISAPI/ITC/TriggerMode/videoEpolice?channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<VideoPolice xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, trigger mode parameters of video intersection violation system, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool, whether to enable intersection violation system-->true
  </enabled>
  <relatedLaneCount>
    <!--req, int, number of supported Lanes, range:[1,4]-->0
  </relatedLaneCount>
  <TrafficLight>
    <!--req, object, traffic light-->
    <source>
      <!--req, enum, signal source, subType:string, desc:"IO", "RS485", "videoDetect" (video detection), "network"-->IO
    </source>
    <source>
      <!--req, enum, signal source, subType:string, desc:"IO", "RS485", "videoDetect" (video detection), "network"-->RS485
    </source>
    <IOLightList>
      <!--opt, array, list, subType:object-->
      <IOLight>
        <!--opt, object, I/O access-->
        <lightId>
          <!--req, int, traffic light No., range:[1,3]-->1
        </lightId>
        <lightType>
          <!--req, enum, guiding direction of traffic light, subType:string, desc:"left", "right", "straight"-->straight
        </lightType>
        <relatedIO>
          <!--req, enum, linked I/O input, subType:string, desc:"I01", "I02", "I03", "I04", "I05", "I06", "I07", "I08"-->IO
        </relatedIO>
        <redLightStatus>
          <!--req, enum, red light level status, subType:string, desc:"Low" (Low-Level red Light), "high" (High-Level red Light)-->low
        </redLightStatus>
      </IOLight>
    </IOLightList>
    <RS485LightList>
      <!--opt, array, list, subType:object-->
      <RS485Light>
        <!--opt, object, RS-485 access-->
        <lightId>

```

```

<!--req, int, traffic light No., range:[1,3]-->0
</lightId>
<lightType>
    <!--req, enum, guiding direction of traffic light, subType:string, desc:"left", "right", "straight"-->straight
</lightType>
<relatedLightChan>
    <!--req, int, detector channel No. Linked to red light, range:[0,16]-->0
</relatedLightChan>
<yellowLightChan>
    <!--req, int, detector channel No. Linked to yellow light, range:[0,16]-->0
</yellowLightChan>
</RS485Light>
</RS485LightList>
<videoDetectLightList>
    <!--opt, array, list of video detection traffic lights, subType:object-->
    <videoDetectLight>
        <!--opt, object, video detection traffic light-->
        <videoDetectLightId>
            <!--req, int, video detection signal No., range:[1,2]-->0
        </videoDetectLightId>
        <lightNum>
            <!--req, int, number of traffic lights, range:[1,3]-->0
        </lightNum>
        <straightLight>
            <!--req, bool, whether the straight driving light is on-->true
        </straightLight>
        <leftLight>
            <!--req, bool, whether the Left-turn Light is on-->true
        </leftLight>
        <rightLight>
            <!--req, bool, whether the right-turn light is on-->true
        </rightLight>
        <redLight>
            <!--req, bool, whether the red Light is on-->true
        </redLight>
        <greenLight>
            <!--req, bool, whether the green Light is on-->true
        </greenLight>
        <yellowLight>
            <!--req, bool, whether the yellow Light is on-->true
        </yellowLight>
        <yellowLightTime>
            <!--req, int, yellow light duration, range:[0,10], unit:s-->0
        </yellowLightTime>
        <rectRegion>
            <!--req, object, traffic light region, desc:the origin is the upper-left corner of the screen-->
            <positionLeft>
                <!--req, int, x-coordinate of upper left corner, range:[0,65535]-->0
            </positionLeft>
            <positionTop>
                <!--req, int, y-coordinate of upper left corner, range:[0,65535]-->0
            </positionTop>
            <positionRight>
                <!--req, int, x-coordinate of lower right corner, range:[0,65535]-->0
            </positionRight>
            <positionBottom>
                <!--req, int, y-coordinate of lower right corner, range:[0,65535]-->0
            </positionBottom>
        </rectRegion>
    </videoDetectLight>
</videoDetectLightList>
</TrafficLight>
<LaneParamList>
    <!--req, array, Lane parameter list, subType:object-->
</LaneParamList>
<VirtualLaneList>
    <!--req, array, list of virtual line, subType:object-->
    <VirtualLane>
        <!--req, object, virtual lane line-->
        <lineName>
            <!--req, enum, line name, subType:string, desc:detection line name: "laneLine,stopLine,redLightLine,cancelLine,waitLine"-->laneRightBoundaryLine
        </lineName>
        <lineType>
            <!--opt, enum, detection line type, subType:string, desc:"unknown", "white" (white solid line), "singleYellow" (single yellow line), "doubleYellow" (double yellow line), "guardRail" (Lane Line with guardrail), "noCross" (Lane Line prohibited to cross by vehicle)-->unknown
        </lineType>
        <RegionCoordinatesList>
            <!--opt, array, area coordinate list, subType:object, range:[0,2]-->
            <RegionCoordinates>
                <!--opt, object, area coordinates, desc:the origin is the upper-left corner of the screen-->
                <positionX>
                    <!--req, int, x-coordinate, range:[0,1000]-->0
                </positionX>
                <positionY>
                    <!--req, int, y-coordinate-->0
                </positionY>
            </RegionCoordinates>
        </RegionCoordinatesList>
    </VirtualLane>
</VirtualLaneList>
<Posting>
    <!--opt, object, positioning frame-->
    <postingEnable>

```

```

<!--req, bool, whether to enable positioning frame-->true
</postingEnable>
<postingSize>
    <!--req, enum, positioning frame size, subType:string, desc:"small,middle,large"-->small
</postingSize>
</Posting>
<leftNotYieldStraight>
    <!--opt, object, left turn not yield to straight-->
    <enable>
        <!--req, bool, whether to enable detection of left-turn without straight driving-->true
    </enable>
</leftNotYieldStraight>
<rightNotYieldLeft>
    <!--opt, object, right turn not yield to left turn-->
    <enable>
        <!--req, bool, whether to enable detection of left-turn without straight driving-->true
    </enable>
</rightNotYieldLeft>
<uTurnNotYieldGoingStraight>
    <!--opt, object, U-turn not yield to straight-->
    <enable>
        <!--req, bool, whether to enable detection of left-turn without straight driving-->true
    </enable>
</uTurnNotYieldGoingStraight>
<leftSideDrivingAfterLeftTurn>
    <!--opt, object, turn Left in short turn radius-->
    <enable>
        <!--req, bool, whether to enable detection of left-turn without straight driving-->true
    </enable>
</leftSideDrivingAfterLeftTurn>
<failYieldPed>
    <!--opt, object, failing to yield for pedestrians-->
    <enable>
        <!--req, bool, whether to enable detection of left-turn without straight driving-->true
    </enable>
<vehicleFromDirection>
    <!--req, enum, driving direction of the vehicle which does not yield to pedestrian, subType:string, desc:"Front", "Back"-->Front
</vehicleFromDirection>
<noDisplacementPedEnabled>
    <!--req, bool, whether to enable no displacement capture of pedestrian-->true
</noDisplacementPedEnabled>
<failYieldPedLineList>
    <!--opt, array, List, subType:object-->
    <failYieldPedLine>
        <!--opt, object, failing to yield for pedestrians Line-->
        <lineId>
            <!--req, int, Line ID, range:[1,2]-->0
        </lineId>
        <lineName>
            <!--req, string, detection line name, range:[0,64]-->test
        </lineName>
        <RegionCoordinatesList>
            <!--opt, array, area coordinate list, subType:object, range:[0,2]-->
            <RegionCoordinates>
                <!--opt, object, area coordinates, desc:the origin is the upper-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>
    </failYieldPedLine>
</failYieldPedLineList>
<notification>
    <!--req, object, capture Linkage, desc:capture Linkage via HCNetSDK-->
    <enable>
        <!--req, bool, whether to enable detection of left-turn without straight driving-->true
    </enable>
    <ip>
        <!--opt, string, IPv4 address, range:[0,32]-->test
    </ip>
    <port>
        <!--opt, int, port No. of traffic camera, range:[0,65535]-->1
    </port>
    <username>
        <!--opt, string, user name of traffic camera, range:[0,32]-->test
    </username>
    <password>
        <!--opt, string, password of traffic camera, range:[0,16]-->test
    </password>
    <sensitivity>
        <!--req, int, detection sensitivity, range:[0,100]-->0
    </sensitivity>
    <RS485Port>
        <!--req, int, RS-485 port No. for the traffic light to access the camera, range:[1,10]-->1
    </RS485Port>
</notification>
</failYieldPed>
</VideoPolice>

```

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>
  <!--ro, req, string, sub status code, desc:sub status code-->OK
  </subStatusCode>
</ResponseStatus>
```

12.5.1.45 Get the multiple-frame recognition mode parameters of smart video security

Request URL

GET /ISAPI/ITC/TriggerMode/videoMonitor

Query Parameter

None

Request Message

None

Response Message

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

<videoMonitor xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, trigger mode parameters of smart video security, attr:version{req, string, protocolVersion}-->
  <relatedLaneCount>
    <!--ro, req, int, total number of Linked Lanes, range:[1,3]-->0
  </relatedLaneCount>
  <capType>
    <!--ro, req, enum, captured target type, subType:string, desc:"Vehicle" (motor vehicle), "vehWithHum" (non-motor vehicle)-->vehicle
  </capType>
  <ViolationDetectLineList>
    <!--ro, req, array, List of violation detection Line on the single Lane, subType:object-->
    <ViolationDetectLine>
      <!--ro, opt, object, violation detection Line on the single Lane-->
      <lineName>
        <!--ro, req, enum, line name, subType:string, desc:"LaneLine" (Lane Line)-->laneLine
      </lineName>
      <!--ro, req, int, Linked Lane No., range:[1,99]-->1
    </ViolationDetectLine>
  </ViolationDetectLineList>
  <!--ro, req, enum, Lane direction type No., subType:int, desc:0 (unknown), 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
  </laneDirectionType>
  <carDriveDirect>
    <!--ro, opt, enum, subType:string-->up2down
  </carDriveDirect>
  <RegionCoordinatesList>
    <!--ro, opt, array, area coordinates List, subType:object, range:[0,2]-->
    <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]-->0
      </positionX>
      <positionY>
        <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
  <signSpeed>
    <!--ro, opt, int, marked speed limit, range:[0,255]-->0
  </signSpeed>
  <speedLimit>
    <!--ro, opt, int, speed limit, range:[0,255]-->0
  </speedLimit>
  <lowSpeedLimit>
    <!--ro, opt, int, low speed limit for small-sized vehicle, range:[0,255]-->0
  </lowSpeedLimit>
</videoMonitor>
```

```

<bigCarSignSpeed>
    <!--ro, opt, int, marked speed Limit for Large-sized vehicle, range:[0,255]-->0
</bigCarSignSpeed>
<bigCarSpeedLimit>
    <!--ro, opt, int, speed limit for large-sized vehicle, range:[0,255]-->0
</bigCarSpeedLimit>
<bigCarLowSpeedLimit>
    <!--ro, opt, int, low speed limit for large-sized vehicle, range:[0,255]-->0
</bigCarLowSpeedLimit>
</ViolationDetectLine>
</ViolationDetectLineList>
<VirtualLane>
    <!--ro, req, object, draw violation detection Line on all lanes-->
<lineName>
    <!--ro, req, enum, Line name, subType:string, desc:line name-->laneRightBoundaryLine
</lineName>
<RegionCoordinatesList>
    <!--ro, opt, array, area coordinates list, subType:object, range:[0,2]-->
<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]-->0
</positionX>
<positionY>
    <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</VirtualLane>
<enabled>
    <!--ro, opt, bool, whether to enable smart video security, desc:if this node is not returned, smart video security of the device is enabled by default-->true
</enabled>
<TriggerLine>
    <!--ro, opt, object, triggering Line of smart video security capture-->
<RegionCoordinatesList>
    <!--ro, opt, array, area coordinates list, subType:object, range:[0,2]-->
<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]-->0
</positionX>
<positionY>
    <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</TriggerLine>
<capNo>
    <!--ro, opt, int, captured pictures of smart video security Linked cameras, range:[1,2]-->1
</capNo>
<interval>
    <!--ro, opt, int, capture interval of smart video security, range:[1,3000], unit:ms-->1
</interval>
<carDriveDirect>
    <!--ro, opt, enum, vehicle detection type, subType:string, desc:"up2down" (up to down), "down2up" (down to up)-->up2down
</carDriveDirect>
<sceneMode>
    <!--ro, opt, enum, scene mode, subType:string, desc:"standardScene" (standard scene), "LowPoleScene" (low pole scene), "entrance" (entrance and exit scene), "other" (other scenes)-->standardScene
</sceneMode>
<overSpeed>
    <!--ro, opt, bool, whether to enable overspeed trigger mode, desc:whether to enable overspeed trigger mode-->true
</overSpeed>
<overSpeedCapNo>
    <!--ro, opt, int, capture times of overspeed, range:[1,3]-->1
</overSpeedCapNo>
<lowSpeed>
    <!--ro, opt, bool, whether to enable low speed trigger mode-->true
</lowSpeed>
<lowSpeedCapNo>
    <!--ro, opt, int, capture times of low speed, range:[1,3]-->1
</lowSpeedCapNo>
<carHighSpeed>
    <!--ro, opt, int, abnormal overspeed, range:[0,255]-->0
</carHighSpeed>
<carLowSpeed>
    <!--ro, opt, int, abnormal low speed, range:[0,255]-->0
</carLowSpeed>
<bigCarHighSpeed>
    <!--ro, opt, int, abnormal overspeed of large-sized vehicle, range:[0,255]-->0
</bigCarHighSpeed>
<bigCarLowSpeed>
    <!--ro, opt, int, abnormal low speed of large-sized vehicle, range:[0,255]-->0
</bigCarLowSpeed>
<nonmotorOverSpeed>
    <!--ro, opt, bool, whether to enable capturing non-motor vehicle-->true
</nonmotorOverSpeed>
<nonmotorHighSpeed>
    <!--ro, opt, int, speed limit for non-motor vehicle (unit: km/h), range:[0,255], unit:km/h-->0
</nonmotorHighSpeed>
</VideoMonitor>

```

12.5.1.46 Set the multiple-frame recognition mode parameters of smart video security

Request URL

PUT /ISAPI/ITC/TriggerMode/videoMonitor

Query Parameter

None

Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<VideoMonitor xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, object, trigger mode parameters of smart video security, attr:version{req, string, protocolVersion}-->
    <relatedLaneCount>
        <!--req, int, number of supported Lanes, range:[1,3]-->0
    </relatedLaneCount>
    <capType>
        <!--req, enum, captured target type, subType:string, desc:"vehicle" (motor vehicle), "vehWithHum" (non-motor vehicle)-->vehicle
    </capType>
    <ViolationDetectLineList>
        <!--req, array, List of violation detection line on the single lane, subType:object-->
        <ViolationDetectLine>
            <!--opt, object, violation detection line on the single Lane-->
            <lineName>
                <!--req, enum, detection Line name, subType:string, desc:"LaneLine" (for a single lane only)-->laneLine
            </lineName>
            <relatedDriveWay>
                <!--req, int, Linked Lane No., range:[1,99]-->1
            </relatedDriveWay>
            <laneDirectionType>
                <!--req, enum, Lane direction type No., subType:int, desc:0 (unknown),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
            </laneDirectionType>
            <RegionCoordinatesList>
                <!--opt, array, area coordinate List, subType:object, range:[0,2]-->
                <RegionCoordinates>
                    <!--opt, object, only two coordinates are required, desc:the origin is the upper-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>
        </ViolationDetectLine>
    </ViolationDetectLineList>
    <VirtualLane>
        <!--req, object, draw violation detection line on all Lanes-->
        <lineName>
            <!--req, enum, detection Line name, subType:string, desc:"LaneRightBoundaryLine" (parking space boundary Line)-->laneRightBoundaryLine
        </lineName>
        <RegionCoordinatesList>
            <!--opt, array, area coordinate list, subType:object, range:[0,2]-->
            <RegionCoordinates>
                <!--opt, object, area coordinates, desc:the origin is the upper-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>
    </VirtualLane>
</VideoMonitor>
```

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, 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>

```

12.5.1.47 Get the multiple-frame recognition mode capability of smart video security

Request URL

GET /ISAPI/ITC/TriggerMode/videoMonitor/capabilities

Query Parameter

None

Request Message

None

Response Message

```

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

<VideoMonitor xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, multiple-frame recognition mode capability of smart video security, attr:version{req, string, protocolVersion}-->
  <relatedLaneCount min="1" max="16">
    <!--ro, req, int, total number of linked lanes, attr:min{req, int},max{req, int}-->0
  </relatedLaneCount>
  <relatedDriveWay min="1" max="99">
    <!--ro, req, int, linked lane No., attr:min{req, int},max{req, int}-->0
  </relatedDriveWay>
  <laneDirectionType opt="0,1,2,3,4,5,6,7,8">
    <!--ro, req, enum, Lane direction, subType:int, attr:opt{req, string}, desc:0-unknown, 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-->0
  </laneDirectionType>
  <capType opt="vehWithHum,vehicle,nonmotor,human,motorWithHum,nonmotorWithHum,motorWithNonmotor">
    <!--ro, req, enum, captured target type, subType:string, attr:opt{req, string}, desc:"vehWithHum" (motor vehicle, non-motor vehicle, and pedestrian), "vehicle" (motor vehicle), "nonmotor" (non-motor vehicle), "human" (pedestrian), "motorWithHum" (motor vehicle and pedestrian), "nonmotorWithHum" (non-motor vehicle and pedestrian), "motorNonmotor" (motor vehicle and non-motor vehicle)-->vehWithHum
  </capType>
  <enabled opt="true,false">
    <!--ro, opt, bool, whether to enable smart video security, attr:opt{req, string}-->true
  </enabled>
  <TriggerLine>
    <!--ro, opt, object, triggering line of smart video security capture-->
    <RegionCoordinatesList>
      <!--ro, opt, array, area coordinate list, subType:object, range:[0,2]-->
      <RegionCoordinates>
        <!--ro, opt, object, region coordinates, desc:the origin is the upper-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>
  </TriggerLine>
  <capNo opt="1,2">
    <!--ro, opt, int, number of captured pictures, attr:opt{req, string}-->0
  </capNo>
  <interval min="1" max="3000">
    <!--ro, opt, int, capture time, unit:ms, attr:min{req, int},max{req, int}-->0
  </interval>
  <carDriveDirect opt="up2down,down2up">
    <!--ro, req, enum, vehicle detection type, subType:string, attr:opt{req, string}, desc:"up2down" (up to down), "down2up" (down to up)-->up2down
  </carDriveDirect>
  <sceneMode opt="standardScene ,lowPoleScene">

```

```

<!--ro, opt, enum, scene mode, subType:string, attr:opt{req, string}, desc:"standardScene" (standard scene), "LowPoleScene" (low pole scene), "entrance"
(entrance and exit scene), "other" (other scenes)-->standardScene
</sceneMode>
<overSpeed>
  <!--ro, opt, bool, whether to enable overspeed trigger mode-->true
</overSpeed>
<overSpeedCapNo opt="1,2,3">
  <!--ro, opt, int, capture times of overspeed, attr:opt{req, string}-->0
</overSpeedCapNo>
<lowSpeed>
  <!--ro, opt, bool, whether to enable low speed trigger mode-->true
</lowSpeed>
<lowSpeedCapNo opt="1,2,3">
  <!--ro, opt, int, capture times of low speed, attr:opt{req, string}-->0
</lowSpeedCapNo>
<signSpeed min="0" max="255">
  <!--ro, opt, int, marked speed limit, attr:min{req, int},max{req, int}-->0
</signSpeed>
<speedLimit min="0" max="255">
  <!--ro, opt, int, speed limit, attr:min{req, int},max{req, int}-->0
</speedLimit>
<bigCarSignSpeed min="0" max="255">
  <!--ro, opt, int, marked speed limit for large-sized vehicle, attr:min{req, int},max{req, int}-->0
</bigCarSignSpeed>
<bigCarHighSpeedLimit min="0" max="255">
  <!--ro, opt, int, maximum speed limit for large-sized vehicle, attr:min{req, int},max{req, int}-->0
</bigCarHighSpeedLimit>
<bigCarLowSpeedLimit min="0" max="255">
  <!--ro, opt, int, low speed limit for large-sized vehicle, attr:min{req, int},max{req, int}-->0
</bigCarLowSpeedLimit>
<lowSpeedLimit min="0" max="60">
  <!--ro, opt, int, low speed limit for small-sized vehicle, attr:min{req, int},max{req, int}-->0
</lowSpeedLimit>
<bigCarSpeedLimit min="0" max="60">
  <!--ro, opt, int, speed limit for large-sized vehicle, attr:min{req, int},max{req, int}-->0
</bigCarSpeedLimit>
<carHighSpeed min="0" max="255">
  <!--ro, opt, int, abnormal overspeed, attr:min{req, int},max{req, int}-->0
</carHighSpeed>
<carLowSpeed min="0" max="255">
  <!--ro, opt, int, abnormal low speed, attr:min{req, int},max{req, int}-->0
</carLowSpeed>
<bigCarHighSpeed min="0" max="255">
  <!--ro, opt, int, abnormal overspeed of large-sized vehicle, attr:min{req, int},max{req, int}-->0
</bigCarHighSpeed>
<bigCarLowSpeed min="0" max="255">
  <!--ro, opt, int, abnormal low speed of large-sized vehicle, attr:min{req, int},max{req, int}-->0
</bigCarLowSpeed>
<nonmotorHighSpeedLimit min="0" max="255">
  <!--ro, opt, int, speed limit for non-motor vehicle, attr:min{req, int},max{req, int}-->0
</nonmotorHighSpeedLimit>
</VideoMonitor>

```

12.5.1.48 Get the recommended value of smart video security multi-frame recognition parameter

Request URL

GET /ISAPI/ITC/TriggerMode/videoMonitor/recommendation?channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

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

<VideoMonitor xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, smart video security multi-frame recognition parameter, attr:version{req, string, protocolVersion}-->
  <relatedLaneCount>
    <!--ro, req, int, total number of linked lanes, range:[1,3]-->0
  </relatedLaneCount>
  <capType>
    <!--ro, req, enum, detection type, subType:string, desc:detection type-->vehicle
  </capType>
  <ViolationDetectLineList>
    <!--ro, req, array, subType:object-->
    <ViolationDetectLine>
      <!--ro, opt, object-->
      <lineName>
        <!--ro, req, enum, line name, subType:string, desc:line name-->laneLine
    </ViolationDetectLine>
  </ViolationDetectLineList>
</VideoMonitor>

```

```

</lineName>
<relatedDriveWay>
  <!--ro, req, int, Lane No., range:[1,99]-->1
</relatedDriveWay>
<laneDirectionType>
  <!--ro, req, enum, Lane direction, subType:int, desc:0-unknown, 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
</laneDirectionType>
<carDriveDirect>
  <!--ro, opt, enum, subType:string-->up2down
</carDriveDirect>
<RegionCoordinatesList>
  <!--ro, opt, array, region coordinate List, subType:object, range:[0,2]-->
<RegionCoordinates>
  <!--ro, opt, object, region coordinates, desc:the origin is the upper-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>
<signSpeed>
  <!--ro, opt, int, range:[0,255]-->0
</signSpeed>
<speedLimit>
  <!--ro, opt, int, range:[0,255]-->0
</speedLimit>
<lowSpeedLimit>
  <!--ro, opt, int, range:[0,255]-->0
</lowSpeedLimit>
<bigCarSignSpeed>
  <!--ro, opt, int, range:[0,255]-->0
</bigCarSignSpeed>
<bigCarSpeedLimit>
  <!--ro, opt, int, range:[0,255]-->0
</bigCarSpeedLimit>
<bigCarLowSpeedLimit>
  <!--ro, opt, int, range:[0,255]-->0
</bigCarLowSpeedLimit>
</ViolationDetectLine>
</ViolationDetectLineList>
<VirtualLane>
  <!--ro, req, object-->
<lineName>
  <!--ro, req, enum, Line name, subType:string, desc:Line name-->laneRightBoundaryLine
</lineName>
<RegionCoordinatesList>
  <!--ro, opt, array, region coordinate List, subType:object, range:[0,2]-->
<RegionCoordinates>
  <!--ro, opt, object, region coordinates, desc:the origin is the upper-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>
</VirtualLane>
<enabled>
  <!--ro, opt, bool-->true
</enabled>
<TriggerLine>
  <!--ro, opt, object-->
<RegionCoordinatesList>
  <!--ro, opt, array, region coordinate List, subType:object, range:[0,2]-->
<RegionCoordinates>
  <!--ro, opt, object, region coordinates, desc:the origin is the upper-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>
</TriggerLine>
<capNo>
  <!--ro, opt, int, range:[1,2]-->1
</capNo>
<interval>
  <!--ro, opt, int, range:[1,3000], unit:ms-->1
</interval>
<sceneMode>
  <!--ro, opt, enum, subType:string-->standardScene
</sceneMode>
<overSpeed>
  <!--ro, opt, bool-->true
</overSpeed>
<overSpeedCapNo>
  <!--ro, opt, int, range:[1,3]-->1
</overSpeedCapNo>

```

```

</overSpeedCapNo>
<lowSpeed>
  <!--ro, opt, bool-->true
</lowSpeed>
<lowSpeedCapNo>
  <!--ro, opt, int, range:[1,3]-->1
</lowSpeedCapNo>
<carHighSpeed>
  <!--ro, opt, int, range:[0,255]-->0
</carHighSpeed>
<carLowSpeed>
  <!--ro, opt, int, range:[0,255]-->0
</carLowSpeed>
<bigCarHighSpeed>
  <!--ro, opt, int, range:[0,255]-->0
</bigCarHighSpeed>
<bigCarLowSpeed>
  <!--ro, opt, int, range:[0,255]-->0
</bigCarLowSpeed>
<nonmotorOverSpeed>
  <!--ro, opt, bool-->true
</nonmotorOverSpeed>
<nonmotorHighSpeed>
  <!--ro, opt, int, range:[0,255], unit:km/h-->0
</nonmotorHighSpeed>
<channelId>
  <!--ro, opt, int, range:[1,5]-->1
</channelId>
<radarTrack>
  <!--ro, opt, bool-->true
</radarTrack>
</VideoMonitor>

```

12.5.1.49 Set the video mark parameters

Request URL

PUT /ISAPI/ITC/videoMark

Query Parameter

None

Request Message

```

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

<VideoMarkInfo xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, video mark information, attr:version{req, string, protocolVersion}-->
  <horizontalField>
    <!--req, int, horizontal FOV (field of view), range:[0,360], desc:the default value is 0-->1
  </horizontalField>
  <verticalField>
    <!--req, int, vertical FOV (field of view), range:[0,360], desc:the default value is 0-->1
  </verticalField>
  <autoGetGps>
    <!--req, bool, whether to get GPS information automatically, desc:the value of this node is false by default-->true
  </autoGetGps>
  <longitude>
    <!--req, string, Longitude, range:[-180,180], desc:Longitude, which contains 11 digits-->W040D30M28S
  </longitude>
  <latitude>
    <!--req, string, Latitude, range:[-90,90], desc:Latitude, which contains 11 digits-->N040D30M28S
  </latitude>
</VideoMarkInfo>

```

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>

```

12.5.1.50 Get the video mark parameters

Request URL

GET /ISAPI/ITC/videoMark

Query Parameter

None

Request Message

None

Response Message

```

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

<VideoMarkInfo xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, video mark information, attr:version{req, string, protocolVersion}-->
  <horizontalField>
    <!--ro, req, int, horizontal FOV (field of view), range:[0,360], desc:the default value is 0-->1
  </horizontalField>
  <verticalField>
    <!--ro, req, int, vertical FOV (field of view), range:[0,360], desc:the default value is 0-->1
  </verticalField>
  <autoGetGps>
    <!--ro, req, bool, whether to get GPS information automatically: "true"-yes, "false"-no (default), desc:the value of this node is false by default-->true
  </autoGetGps>
  <longitude>
    <!--ro, req, string, longitude, range:[-180,180], desc:longitude, which contains 11 digits-->W040D30M28S
  </longitude>
  <latitude>
    <!--ro, req, string, latitude, range:[-90,90], desc:latitude, which contains 11 digits-->N040D30M28S
  </latitude>
</VideoMarkInfo>

```

12.5.1.51 Get the video tag capability

Request URL

GET /ISAPI/ITC/videoMark/capabilities

Query Parameter

None

Request Message

None

Response Message

```

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

<VideoMarkInfo xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, video mark information, attr:version{req, string, protocolVersion}-->
  <horizontalField min="0" max="360">
    <!--ro, req, int, horizontal FOV (field of view), range:[0,360], attr:min{req, int},max{req, int}, desc:the default value is 0-->1
  </horizontalField>
  <verticalField min="0" max="360">
    <!--ro, req, int, vertical FOV (field of view), range:[0,360], attr:min{req, int},max{req, int}, desc:the default value is 0-->1
  </verticalField>
  <autoGetGps>
    <!--ro, req, bool, whether to get GPS information automatically: "true"-yes, "false"-no (default), desc:the value of this node is false by default-->true
  </autoGetGps>
  <longitude min="1" max="11">
    <!--ro, req, string, longitude, range:[1,11], attr:min{req, int},max{req, int}, desc:longitude, which contains 11 digits-->W040D30M28S
  </longitude>
  <latitude min="1" max="11">
    <!--ro, req, string, latitude, range:[1,11], attr:min{req, int},max{req, int}, desc:latitude, which contains 11 digits-->N040D30M28S
  </latitude>
</VideoMarkInfo>

```

12.5.1.52 Set ANR parameters

Request URL

PUT /ISAPI/Traffic/ANR

Query Parameter

None

Request Message

```

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

<ANRControl xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, ANR parameters, attr:version{req, string, protocolVersion}-->
  <uploadInterval>
    <!--req, int, upload interval, unit:ms-->0
  </uploadInterval>
  <uploadTimeout>
    <!--opt, int, uploading timed out, unit:ms-->0
  </uploadTimeout>
  <DeviceIpAddress>
    <!--opt, object, IP address of ANR arming server-->
    <addressingFormatType>
      <!--req, enum, address type, subType:string, desc:"ipaddress", "hostname"-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--opt, string, host server name, range:[0,64], dep:and,{$.ANRControl.DeviceIpAddress.addressingFormatType,eq,hostname}-->test
    </hostName>
    <ipAddress>
      <!--opt, string, IPv4 address, range:[0,32], dep:and,{$.ANRControl.DeviceIpAddress.addressingFormatType,eq,ipaddress}-->test
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPv6 address, range:[0,128], dep:and,{$.ANRControl.DeviceIpAddress.addressingFormatType,eq,ipaddress}-->test
    </ipv6Address>
  </DeviceIpAddress>
</ANRControl>

```

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>

```

12.5.1.53 Get the ANR parameters

Request URL

GET /ISAPI/Traffic/ANR

Query Parameter

None

Request Message

None

Response Message

```

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

<ANRControl xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, ANR (Automatic Network Replenishment) parameters, attr:version{req, string, protocolVersion}-->
  <uploadInterval>
    <!--ro, req, int, uploading interval, unit:ms-->0
  </uploadInterval>
  <uploadTimeout>
    <!--ro, opt, int, unit: ms, unit:ms-->0
  </uploadTimeout>
  <DeviceIpAddress>
    <!--ro, opt, object, IP address of ANR arming server-->
    <addressingFormatType>
      <!--ro, req, enum, address type, subType:string, desc:"ipaddress", "hostname"-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, host name, range:[0,64], dep:and,{$.ANRControl.DeviceIpAddress.addressingFormatType,eq,hostname}-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, this node is valid only when addressingFormatType is "ipaddress", range:[0,32], dep:and,{$.ANRControl.DeviceIpAddress.addressingFormatType,eq,ipaddress}-->test
      <ipAddress>
        <!--ro, opt, string, this node is valid only when addressingFormatType is "ipaddress", range:[0,128], dep:and,{$.ANRControl.DeviceIpAddress.addressingFormatType,eq,ipaddress}-->test
        <ipv6Address>
          <!--ro, opt, string, this node is valid only when addressingFormatType is "ipaddress", range:[0,128], dep:and,{$.ANRControl.DeviceIpAddress.addressingFormatType,eq,ipaddress}-->test
        </ipv6Address>
      </DeviceIpAddress>
    </ipAddress>
  </DeviceIpAddress>
</ANRControl>

```

12.5.1.54 Set parameters of traffic picture composition

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/imageMerge

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageMerge xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, attr:version{req, string, protocolVersion}-->
  <isMerge>
    <!--req, int, whether to enable composition, range:[0,1]-->0
  </isMerge>
  <oneMergeType>
    <!--req, enum, subType:int-->101
  </oneMergeType>
  <twoMergeType>
    <!--req, enum, subType:int-->201
  </twoMergeType>
  <threeMergeType>
    <!--req, enum, subType:int-->301
  </threeMergeType>
  <fourMergeType>
    <!--req, enum, subType:int-->401
  </fourMergeType>
  <closeupIndex>
    <!--opt, int, range:[0,5]-->0
  </closeupIndex>
  <closeupScale>
    <!--opt, int, range:[0,100]-->0
  </closeupScale>
  <positionOffset>
    <!--opt, int, range:[50,2047]-->50
  </positionOffset>
</ImageMerge>
```

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>
```

12.5.1.55 Get the picture composition parameters

Request URL

GET /ISAPI/Traffic/channels/<channelID>/imageMerge

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Channel No.

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageMerge xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <isMerge>
    <!--ro, req, int, range:[0,1]-->0
  </isMerge>
  <oneMergeType>
    <!--ro, req, enum, subType:string-->101
  </oneMergeType>
  <twoMergeType>
    <!--ro, req, enum, subType:string-->201
  </twoMergeType>
  <threeMergeType>
    <!--ro, req, enum, subType:string-->301
  </threeMergeType>
  <fourMergeType>
    <!--ro, req, enum, subType:string-->401
  </fourMergeType>
  <closeupIndex>
    <!--ro, opt, int, range:[0,5]-->0
  </closeupIndex>
  <closeupScale>
    <!--ro, opt, int, range:[0,100]-->0
  </closeupScale>
  <positionOffset>
    <!--ro, opt, int, range:[0,2159]-->50
  </positionOffset>
</ImageMerge>

```

12.5.1.56 Set the parameters of vehicle detection captured pictures

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/picParam

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	channel No.

Request Message

```

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

<PicParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, object, parameters of vehicle detection captured pictures, attr:version{opt, string, protocolVersion}-->
    <PictureCfg>
        <!--opt, object, picture parameters-->
        <mode>
            <!--req, enum, capture mode, subType:string, desc:"quality" (by quality), "size" (by size)-->quality
        </mode>
        <pictureQuality>
            <!--opt, int, picture quality, range:[1,100]-->50
        </pictureQuality>
        <pictureSize>
            <!--opt, int, picture size, unit:KB-->100
        </pictureSize>
    </PictureCfg>
    <Overlap>
        <!--opt, object, picture overlay-->
        <enabled>
            <!--req, bool, whether to enable text overlay-->true
        </enabled>
        <OverlapItemList>
            <!--opt, array, overlay item list, subType:object-->
            <OverlapItem>
                <!--opt, object, overlay items-->
                <id>
                    <!--req, int, index-->1
                </id>
                <item opt="positionNo">
                    <!--req, string, overlay content, attr:opt{opt, enum, subType:string}-->test
                </item>
            </OverlapItem>
        </OverlapItemList>
        <fontColor>
            <!--opt, string, font color, desc:the previous version of protocol; hexBinary-->test
        </fontColor>
        <backColor>
            <!--opt, string, background color, desc:the previous version of protocol; hexBinary-->test
        </backColor>
    </Overlap>
    <CapturePicOverlays>
        <!--opt, object, Capture Overlay, desc:professional OSD configuration for picture capture, which is only supported by capture cameras now-->
    </CapturePicOverlays>
    <MergePicOverlays>
        <!--opt, object, Composite Picture Overlay, desc:professional OSD configuration for picture composite, which is only supported by capture cameras now-->
    </MergePicOverlays>
    <PlateEnhancement>
        <!--opt, object, License Plate Enhancement-->
        <enabled>
            <!--opt, bool, whether to enable license plate enhancement on captured picture-->true
        </enabled>
        <level>
            <!--opt, int, set the level of license plate enhancement from 0 to 100, range:[0,100]-->1
        </level>
    </PlateEnhancement>
    <CapturePicTypeList>
        <!--opt, array, the list of the types of captured pictures, subType:object, range:[1,3], desc:all types of picture shall be captured if this node is not returned-->
        <capturePicType>
            <!--opt, enum, types of captured pictures, subType:string, desc:"licensePlatePicture" (license plate picture), "vehiclePicture" (vehicle picture), "detectionPicture" (detection picture (background picture))-->licensePlatePicture
        </capturePicType>
    </CapturePicTypeList>
    <resolution>
        <!--opt, string, picture resolution, desc:subject to the returned range-->1920*1080
    </resolution>
    <capturePicIntervalEnabled>
        <!--ro, opt, bool, whether to enable picture capture interval-->true
    </capturePicIntervalEnabled>
    <capturePicIntervalTime>
        <!--ro, opt, int, capture interval, range:[1,30], unit:s, dep:and, {$PicParam.capturePicIntervalEnabled, eq, true}-->1
    </capturePicIntervalTime>
    <syncCaptureBackPanoramaEnabled>
        <!--opt, bool, whether to enable background picture of panoramic channel, desc:a scene can be monitored by both channels of the front-end dual-channel device. The alarm from the detailed-image channel should be linked with the capture of panoramic background pictures by the panoramic-image channel-->true
    </syncCaptureBackPanoramaEnabled>
</PicParam>

```

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:detailed information of custom error returned by device applications, used for fast debugging-->badXmlFormat
      </description>
    </statusCode>
  </statusCode>
</ResponseStatus>

```

12.5.1.57 Get parameters of vehicle detection captured pictures

Request URL

GET /ISAPI/Traffic/channels/<channelID>/picParam

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	channel No.

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<PicParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, parameters of vehicle detection captured pictures, attr:version{opt, string, protocolVersion}-->
    <PictureCfg>
        <!--ro, opt, object, picture parameters-->
        <mode>
            <!--ro, req, enum, capture mode, subType:string, desc:"quality" (by quality), "size" (by size)-->quality
        </mode>
        <pictureQuality>
            <!--ro, opt, int, picture quality, range:[1,100]-->50
        </pictureQuality>
        <pictureSize>
            <!--ro, opt, int, picture size, unit:KB-->100
        </pictureSize>
    </PictureCfg>
    <Overlap>
        <!--ro, opt, object, picture overlay-->
        <enabled>
            <!--ro, req, bool, whether to enable text overlay-->true
        </enabled>
        <OverlapItemList>
            <!--ro, opt, array, overlay item list, subType:object-->
            <OverlapItem>
                <!--ro, opt, object, overlay items-->
                <id>
                    <!--ro, req, int, ID-->1
                </id>
                <item opt="positionNo">
                    <!--ro, req, string, overlay content, attr:opt{opt, enum, subType:string}-->test
                </item>
            </OverlapItem>
        </OverlapItemList>
        <fontColor>
            <!--ro, opt, string, font color, desc:the previous version of protocol; hexBinary-->test
        </fontColor>
        <backColor>
            <!--ro, opt, string, background color, desc:the previous version of protocol; hexBinary-->test
        </backColor>
    </Overlap>
    <CapturePicOverlays>
        <!--ro, opt, object, capture overlay, desc:professional OSD configuration for picture capture, which is only supported by capture cameras now-->
    </CapturePicOverlays>
    <MergePicOverlays>
        <!--ro, opt, object, composite picture overlay, desc:professional OSD configuration for picture composite, which is only supported by capture cameras now-->
    </MergePicOverlays>
    <PlateEnhancement>
        <!--ro, opt, object, license plate enhancement-->
        <enabled>
            <!--ro, opt, bool, whether to enable license plate enhancement on captured picture-->true
        </enabled>
        <level>
            <!--ro, opt, int, set the level of license plate enhancement from 0 to 100, range:[0,100]-->1
        </level>
    </PlateEnhancement>
    <CapturePicTypeList>
        <!--ro, opt, array, the list of the types of captured pictures, subType:object, range:[1,3], desc:all types of picture shall be captured if this node is not returned-->
        <capturePicType>
            <!--ro, opt, enum, types of captured pictures, subType:string, desc:"licensePlatePicture" (license plate picture), "vehiclePicture" (vehicle picture), "detectionPicture" (detection picture (background picture))-->licensePlatePicture
        </capturePicType>
    </CapturePicTypeList>
    <resolution>
        <!--ro, opt, string, picture resolution, desc:subject to the returned range-->1920*1080
    </resolution>
    <capturePicIntervalEnabled>
        <!--ro, opt, bool, whether to enable picture capture interval-->true
    </capturePicIntervalEnabled>
    <capturePicIntervalTime>
        <!--ro, opt, int, capture interval, range:[1,30], unit:s, dep:and,{$.PicParam.capturePicIntervalEnabled,eq,true}-->1
    </capturePicIntervalTime>
    <syncCaptureBackPanoramaEnabled>
        <!--ro, opt, bool, whether to enable background picture of panoramic channel, desc:a scene can be monitored by both channels of the front-end dual-channel device. The alarm from the detailed-image channel should be linked with the capture of panoramic background pictures by the panoramic-image channel-->true
    </syncCaptureBackPanoramaEnabled>
</PicParam>

```

12.5.1.58 Get captured picture configuration capability

Request URL

GET /ISAPI/Traffic/channels/<channelID>/picParam/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	channel No.

Request Message

None

Response Message

```

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

<PicParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, picture capture configuration capability, attr:version{opt, string, protocolVersion}-->
  <PictureCfg>
    <!--ro, req, object, picture parameters-->
    <mode opt="quality,size">
      <!--ro, req, enum, capture mode, subType:string, attr:opt{opt, enum, subType:string}, desc:"quality" (by quality), "size" (by size)-->quality
    </mode>
    <pictureQuality min="1" max="100">
      <!--ro, opt, int, picture quality, range:[1,100], attr:min{opt, string},max{opt, string}-->1
    </pictureQuality>
    <pictureSize min="1" max="6553500000">
      <!--ro, opt, int, picture size, unit:KB, attr:min{req, int},max{req, int}-->1
    </pictureSize>
  </PictureCfg>
  <Overlap>
    <!--ro, opt, object, picture overlay-->
    <enabled opt="true,false">
      <!--ro, req, bool, whether to enable text overlay, attr:opt{req, string}-->true
    </enabled>
    <OverlapItem opt="positionNo,positionInfo,cameraNo,captureTime,plateNo,vehicleColor,sceneName,carType,vehicleLogo,sceneNo, GPS, vehicleSpeed">
      <!--ro, req, enum, overlay items, subType:string, attr:opt{opt, enum}, desc:"positionNo" (detection point No.), "positionInfo" (detection point), "cameraNo" (camera No.), "captureTime" (capture time), "plateNo" (license plate No.), "vehicleColor" (vehicle color), "sceneName" (scene name), "carType" (vehicle type), "vehicleLogo" (main vehicle brand) , "sceneNo" (scene No.), "GPS" (GPS device)-->positionNo
    </OverlapItem>
    <fontColor>
      <!--ro, opt, string, font color, desc:the previous version of protocol, hexBinary-->test
    </fontColor>
    <backColor>
      <!--ro, opt, string, background color, desc:the previous version of protocol; hexBinary-->test
    </backColor>
  </Overlap>
  <CapturePicOverlays>
    <!--ro, opt, object, professional OSD configuration for captured pictures, currently this configuration is only supported by capture cameras, desc:professional OSD configuration for captured pictures, currently this configuration is only supported by capture cameras-->
  </CapturePicOverlays>
  <MergePicOverlays>
    <!--ro, opt, object, professional OSD configuration for combined pictures, currently this configuration is only supported by capture cameras, desc:professional OSD configuration for combined pictures, currently this configuration is only supported by capture cameras-->
  </MergePicOverlays>
  <PlateEnhancement>
    <!--ro, opt, object, license plate enhancement on captured picture-->
    <enabled opt="true,false" def="false">
      <!--ro, opt, bool, whether to enable license plate enhancement on captured picture, attr:opt{opt, string},def{opt, string}-->true
    </enabled>
    <level min="0" max="100" def="50">
      <!--ro, opt, int, license plate enhancement Level range, range:[0,100], attr:min{opt, string},max{opt, string},def{opt, string}-->1
    </level>
  </PlateEnhancement>
  <CapturePicTypeList size="3">
    <!--ro, opt, array, the List of the types of captured pictures, subType:object, attr:size{req, int}, desc:all types of picture shall be captured if this node is not returned-->
    <capturePicType opt="licensePlatePicture,vehiclePicture,detectionPicture">
      <!--ro, opt, string, types of captured pictures, attr:opt{req, string}, desc:"licensePlatePicture" (license plate picture), "vehiclePicture" (vehicle picture), "detectionPicture" (detection picture (background picture))-->licensePlatePicture
    </capturePicType>
  </CapturePicTypeList>
  <resolution opt="1920*1080,2688*1520,1280*720,704*576,352*288">
    <!--ro, opt, string, picture resolution, attr:opt{req, string}-->1920*1080
  </resolution>
  <capturePicIntervalEnabled opt="true,false">
    <!--ro, opt, bool, whether to enable picture capture interval, attr:opt{req, string}-->true
  </capturePicIntervalEnabled>
  <capturePicIntervalTime min="1" max="30" def="1">
    <!--ro, opt, int, capture interval, range:[1,30], unit:s, dep:and,{$.PicParam.capturePicIntervalEnabled,eq,true}, attr:min{req, int},max{req, int},def{req, int}-->1
  </capturePicIntervalTime>
  <syncCaptureBackPanoramaEnabled opt="true,false">
    <!--ro, opt, bool, whether to enable background picture of panoramic channel, attr:opt{req, string}, desc:a scene can be monitored by both channels of the front-end dual-channel device. The alarm from the detailed-image channel should be linked with the capture of panoramic background pictures by the panoramic-image channel-->true
  </syncCaptureBackPanoramaEnabled>
</PicParam>

```

12.5.1.59 Get overlay parameters of captured traffic pictures by channel

Request URL

GET /ISAPI/Traffic/channels/<channelID>/picParam/capturePicOverlays

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<CapturePicOverlays xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, see details in the message of XML_OverlayInfoList, attr:version{opt, string, protocolVersion}-->
  <overlayInfoEnabled>
    <!--ro, req, bool, whether to enable text overlay, desc:whether to enable text overlay-->true
  </overlayInfoEnabled>
  <OverlayInfoList>
    <!--ro, opt, array, subType:object-->
    <OverlayInfo>
      <!--ro, opt, object-->
      <overlayInfoID>
        <!--ro, req, int-->1
      </overlayInfoID>
      <itemType>
        <!--ro, req, enum, overlay mode, subType:int, desc:overlay mode-->70
      </itemType>
      <itemOverlayEnabled>
        <!--ro, req, bool-->true
      </itemOverlayEnabled>
      <customName>
        <!--ro, req, string, self-defined name of the overlay information-->test
      </customName>
      <changeLineNum>
        <!--ro, req, int, range:[0,100]-->2
      </changeLineNum>
      <spaceNum>
        <!--ro, req, int, space number, range:[0,255]-->1
      </spaceNum>
      <startPosEnable>
        <!--ro, opt, bool, whether to enable coordinate configuration-->true
      </startPosEnable>
      <startPosTop>
        <!--ro, req, int, top coordinate of the start point-->0
      </startPosTop>
      <startPosLeft>
        <!--ro, req, int, left coordinate of the start point-->0
      </startPosLeft>
      <overlayInfoText>
        <!--ro, opt, string, overlay character string, range:[0,128]-->test
      </overlayInfoText>
      <overlayInfoText2>
        <!--ro, opt, string, overlay character string, range:[0,32]-->test
      </overlayInfoText2>
    </OverlayInfo>
  </OverlayInfoList>
  <linePercent>
    <!--ro, req, int, percentage of overlaying lines, range:[0,100]-->1
  </linePercent>
  <itemsStyle>
    <!--ro, req, enum, overlay mode, subType:string, desc:"horizontal", "vertical"-->horizontal
  </itemsStyle>
  <charStyle>
    <!--ro, req, enum, font type, subType:string, desc:font type: 0-SimSun (default), 1-Wei-->song_type
  </charStyle>
  <charSize>
    <!--ro, req, enum, character size, subType:int, desc:0 (8*16), 1 (16*32), 2 (48*48), 3 (32*64), 4 (64*128)-->0
  </charSize>
  <charPosition>
    <!--ro, req, enum, text position overlaid on the picture, subType:int, desc:0 (in the picture), 1 (outside the top edge of the picture), 2 (outside the bottom edge of the picture)-->1
  </charPosition>
  <charInterval>
    <!--ro, req, int, character separation distance, range:[0,16]-->0
  </charInterval>
  <foreColor>
    <!--ro, req, int, foreground color, range:[0,16777215]-->1
  </foreColor>
  <backColorEnabled>
    <!--ro, opt, bool-->true
  </backColorEnabled>
  <backColor>
    <!--ro, opt, int, background color, range:[0,16777215], desc:background color-->1
  </backColor>
```

```

<colorAdapt>
  <!--ro, opt, bool, whether to enable color self-adaption-->true
</colorAdapt>
<zeroizeEnable>
  <!--ro, opt, bool, whether to enable zero filling for OSD overlay-->true
</zeroizeEnable>
<platePicOverlay>
  <!--ro, opt, bool, whether to enable overlaying license plate thumbnail-->true
</platePicOverlay>
<platePicOverlayMultiplier>
  <!--ro, opt, int-->1
</platePicOverlayMultiplier>
<platePicPosTop>
  <!--ro, req, int, start top coordinate-->0
</platePicPosTop>
<platePicPosLeft>
  <!--ro, req, int, start left coordinate-->0
</platePicPosLeft>
<overlaidTextTransparency>
  <!--ro, opt, int, range:[0,100], unit:%-->1
</overlaidTextTransparency>
<featurePicOSDEnable>
  <!--ro, opt, bool-->true
</featurePicOSDEnable>
<dateStyle>
  <!--ro, opt, enum, subType:string-->YYYY-MM-DD
</dateStyle>
<timeStyle>
  <!--ro, opt, enum, subType:string-->12hour
</timeStyle>
<PictureInfoList>
  <!--ro, opt, array, subType:object-->
</PictureInfoList>
</CapturePicOverlays>

```

12.5.1.60 Set overlay parameters of captured traffic pictures by channel

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/picParam/capturePicOverlays

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<CapturePicOverlays xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, see details in the message of XML_OverlayInfoList, attr:version{opt, string, protocolVersion}-->
  <overlayInfoEnabled>
    <!--req, bool, whether to enable text overlay, desc:whether to enable text overlay-->true
  </overlayInfoEnabled>
  <OverlayInfoList>
    <!--opt, array, subType:object-->
    <OverlayInfo>
      <!--opt, object-->
      <overlayInfoID>
        <!--req, int-->1
      </overlayInfoID>
      <itemType>
        <!--req, enum, subType:int-->70
      </itemType>
      <itemOverlayEnabled>
        <!--req, bool-->true
      </itemOverlayEnabled>
      <customName>
        <!--req, string, self-defined name of the overlay information-->test
      </customName>
      <changeLineNum>
        <!--req, int, range:[0,100]-->2
      </changeLineNum>
      <spaceNum>
        <!--req, int, range:[0,255]-->1
      </spaceNum>
      <startPosEnable>
        <!--opt, bool, whether to enable coordinate configuration-->true
      </startPosEnable>
      <startPosTop>
        <!--req, int, top coordinate of the start point-->0
      </startPosTop>
      <startPosLeft>
        <!--req, int, left coordinate of the start point-->0
      </startPosLeft>
    </OverlayInfo>
  </OverlayInfoList>
</CapturePicOverlays>

```

```

<overlayInfoText>
  <!--opt, string, overlay character string-->test
</overlayInfoText>
<overlayInfoText2>
  <!--opt, string, overlay character string-->test
</overlayInfoText2>
</OverlayInfo>
</OverlayInfoList>
<linePercent>
  <!--req, int, percentage of overlaying lines, range:[0,100]-->1
</linePercent>
<itemsStyle>
  <!--req, enum, overlay mode, subType:string, desc:"horizontal" (default), "vertical"-->horizontal
</itemsStyle>
<charStyle>
  <!--req, enum, font type, subType:string, desc:0-SimSun (default), 1-Wei-->song_type
</charStyle>
<charSize>
  <!--req, enum, character size, subType:int, desc:0-16*16(Chinese)/8*16(English), 1-32*32(Chinese)/16*32(English), 2-48*48, 3-64*64(Chinese)/32*64(English)-->0
</charSize>
<charPosition>
  <!--req, enum, text position overlayed on the picture, subType:int, desc:0 (in the picture), 1 (outside the top edge of the picture), 2 (outside the bottom edge of the picture)-->1
</charPosition>
<charInterval>
  <!--req, int, character separation distance, range:[0,16]-->0
</charInterval>
<foreColor>
  <!--req, int, foreground color, which is the RGB value directly obtained by the palette, the value is between 0 and 0xffffffff and the default value is 0xffffffff (white)-->1
</foreColor>
<backColor>
  <!--req, int, background color, which is the RGB value directly obtained by the palette, the value is between 0 and 0xffffffff and the default value is 0x0 (black)-->1
</backColor>
<colorAdapt>
  <!--opt, bool, whether to enable color self-adaption: 0-no, 1-yes-->true
</colorAdapt>
<zeroizeEnable>
  <!--opt, bool, whether to enable zero filling for OSD overlay, which is used to enable zero filling for vehicle speed, speed limit, overspeed ratio and Lane No. Zero filling is enabled by default-->true
</zeroizeEnable>
<platePicOverlay>
  <!--opt, bool, whether to enable overlaying license plate thumbnail-->true
</platePicOverlay>
<platePicPosTop>
  <!--req, int, start top coordinate, which is only valid for overlaying within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
</platePicPosTop>
<platePicPosLeft>
  <!--req, int, start left coordinate, which is only valid for overlaying within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
</platePicPosLeft>
<posPercentMode>
  <!--opt, bool, whether to enable percentage mode instead of pixel value mode: "true"-yes, "false"-no-->true
</posPercentMode>
<startPosTopPercent>
  <!--opt, int, this node is valid when <posPercentMode> is "true"-->0
</startPosTopPercent>
<startPosLeftPercent>
  <!--opt, int, this node is valid when <posPercentMode> is "true"-->0
</startPosLeftPercent>
<itemsSpacesNumMode>
  <!--opt, bool-->true
</itemsSpacesNumMode>
<itemsSpacesNum>
  <!--opt, int, number of spaces-->1
</itemsSpacesNum>
<overlaidTextTransparency>
  <!--opt, int, range:[0,100], unit:%-->1
</overlaidTextTransparency>
<PictureInfoList>
  <!--opt, array, subType:object-->
<PictureInfo>
  <!--opt, object, picture information-->
<pictureType>
  <!--req, enum, picture type, subType:string-->illegalPicture
</pictureType>
<overlayInfoEnabled>
  <!--req, bool-->true
</overlayInfoEnabled>
<OverlayInfoList>
  <!--opt, array, list of overlay information, subType:object-->
<OverlayInfo>
  <!--opt, object-->
<overlayInfoID>
  <!--req, int-->1
</overlayInfoID>
<itemType>
  <!--req, enum, subType:int-->70
</itemType>
<itemOverlayEnabled>
  <!--req, bool-->true
</itemOverlayEnabled>

```

```

<!--req, bool-->true
</itemOverlayEnabled>
<customName>
  <!--req, string, self-defined name of the overlay information-->test
</customName>
<changeLineNum>
  <!--req, int, range:[0,100]-->2
</changeLineNum>
<spaceNum>
  <!--req, int, range:[0,255]-->1
</spaceNum>
<startPosEnable>
  <!--opt, bool, whether to enable coordinate configuration-->true
</startPosEnable>
<startPosTop>
  <!--req, int, top coordinate of the start point-->0
</startPosTop>
<startPosLeft>
  <!--req, int, left coordinate of the start point-->0
</startPosLeft>
<overlayInfoText>
  <!--opt, string, overlay character string-->test
</overlayInfoText>
<overlayInfoText2>
  <!--opt, string, overlay character string-->test
</overlayInfoText2>
</OverlayInfo>
</OverlayInfoList>
<linePercent>
  <!--req, int, range:[0,100]-->1
</linePercent>
<itemsStyle>
  <!--req, enum, subType:string-->horizontal
</itemsStyle>
<charStyle>
  <!--req, enum, font type, subType:string-->song_type
</charStyle>
<charSize>
  <!--req, enum, subType:int-->0
</charSize>
<charPosition>
  <!--req, enum, position of the text overlaid in the picture, subType:int, desc:0 (in the picture),1 (outside the top edge of the picture), 2 (outside the bottom edge of the picture)-->1
</charPosition>
<charInterval>
  <!--req, int, character pitch, range:[0,16]-->0
</charInterval>
<foreColor>
  <!--req, int, foreground color-->1
</foreColor>
<backColor>
  <!--req, int, background color-->1
</backColor>
<colorAdapt>
  <!--opt, bool, whether to enable color self-adaption-->true
</colorAdapt>
<zeroizeEnable>
  <!--opt, bool, whether to enable zero filling for OSD overlay-->true
</zeroizeEnable>
<platePicOverlay>
  <!--opt, bool, whether to enable overlaying license plate thumbnail-->true
</platePicOverlay>
<platePicPosTop>
  <!--req, int, top coordinate of the start point-->0
</platePicPosTop>
<platePicPosLeft>
  <!--req, int, left coordinate of the start point-->0
</platePicPosLeft>
<posPercentMode>
  <!--opt, bool-->true
</posPercentMode>
<startPosTopPercent>
  <!--opt, int-->0
</startPosTopPercent>
<startPosLeftPercent>
  <!--opt, int-->0
</startPosLeftPercent>
<itemsSpacesNumMode>
  <!--opt, bool-->true
</itemsSpacesNumMode>
<itemsSpacesNum>
  <!--opt, int-->1
</itemsSpacesNum>
<overlaidTextTransparency>
  <!--opt, int, range:[0,100], unit:%-->1
</overlaidTextTransparency>
</PictureInfo>
</PictureInfoList>
</CapturePicOverlays>

```

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, error reason description in detail, desc:error reason description in detail-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, error reason description in detail, desc:error reason description in detail-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

12.5.1.61 Get the capability of OSD overlay on a captured picture

Request URL

GET /ISAPI/Traffic/channels/<channelID>/picParam/capturePicOverlays/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<CapturePicOverlays xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{opt, string, protocolVersion}-->
  <overlayInfoEnabled opt="true,false">
    <!--ro, req, bool, whether to enable text overlay, attr:opt{req, string}, desc:whether to enable text overlay-->true
  </overlayInfoEnabled>
  <OverlayInfoList size="50">
    <!--ro, req, array, subType:object, attr:size{opt, string}-->
    <OverlayInfo>
      <!--ro, req, object-->
      <itemType>
        <!--ro, req, enum, subType:int, attr:opt{opt, string},min{req, int},max{req, int}-->70
      </itemType>
      <editAble opt="true,false">
        <!--ro, req, bool, whether the overlaid information is editable, attr:opt{opt, string}-->true
      </editAble>
      <itemOverlayEnabled opt="true,false">
        <!--ro, req, bool, attr:opt{opt, string}-->true
      </itemOverlayEnabled>
      <customName min="0" max="32">
        <!--ro, req, string, self-defined name of the overlay information, attr:min{opt, string},max{opt, string}-->test
      </customName>
      <changeLineNum min="0" max="100">
        <!--ro, req, int, range:[0,100], attr:min{opt, string},max{opt, string}-->50
      </changeLineNum>
      <spaceNum min="0" max="255">
        <!--ro, req, int, range:[0,255], attr:min{opt, string},max{opt, string}-->1
      </spaceNum>
      <startPosnable opt="true,false">
        <!--ro, opt, bool, whether to enable coordinate configuration, attr:opt{opt, string}-->true
      </startPosnable>
      <startPosTop>
        <!--ro, req, int, top coordinate of the start point-->0
      </startPosTop>
      <startPosLeft>
        <!--ro, req, int, left coordinate of the start point-->0
      </startPosLeft>
    </OverlayInfo>
  </OverlayInfoList>
  <linePercent min="0" max="100">
    <!--ro, req, int, percentage of overlaying lines, range:[0,100], attr:min{opt, string},max{opt, string}-->100
  </linePercent>
  <itemsStyle opt="horizontal,vertical">
    <!--ro, req, enum, overlay mode, subType:string, attr:opt{opt, string}, desc:"horizontal" (default), "vertical"-->vertical
  <!-- -->

```

```

</itemsStyle>
<charStyle opt="song_type,wei_type">
    <!--ro, req, enum, font type, subType:string, attr:opt{opt, string}, desc:font type-->song_type
</charStyle>
<charSize min="0" max="5">
    <!--ro, req, enum, character size, subType:int, attr:min{opt, string},max{opt, string}, desc:0-16*16(Chinese)/8*16(English),1-32*32(Chinese)/16*32(English),2-48*48,3-64*64(Chinese)/32*64(English)-->1
</charSize>
<charPosition min="0" max="2">
    <!--ro, req, enum, text position overlaid on the picture, subType:int, attr:min{opt, string},max{opt, string}, desc:0 (in the picture),1 (outside the top edge of the picture), 2 (outside the bottom edge of the picture)-->1
</charPosition>
<charInterval min="0" max="16">
    <!--ro, req, int, character separation distance, range:[0,16], unit:px, attr:min{opt, string},max{opt, string}-->1
</charInterval>
<foreColor min="0" max=" 0xffffffff">
    <!--ro, req, int, foreground color, which is the RGB value directly obtained by the palette, the value is between 0 and 0xffffffff and the default value is 0xffffffff (white), attr:min{opt, string},max{opt, string}-->1
</foreColor>
<backColor min="0" max=" 0xffffffff">
    <!--ro, req, int, background color, which is the RGB value directly obtained by the palette, the value is between 0 and 0xffffffff and the default value is 0x0 (black), attr:min{opt, string},max{opt, string}-->1
</backColor>
<colorAdapt>
    <!--ro, opt, bool, whether to enable color self-adaption: 0-no,1-yes-->true
</colorAdapt>
<platePicOverlay>
    <!--ro, opt, bool, whether to enable overlaying license plate thumbnail-->true
</platePicOverlay>
<platePicOverlayMultiplier min="1" max="3">
    <!--ro, opt, int, attr:min{req, int},max{req, int}-->1
</platePicOverlayMultiplier>
<platePicPosTop>
    <!--ro, req, int, start top coordinate, which is only valid for overlaying within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
</platePicPosTop>
<platePicPosLeft>
    <!--ro, req, int, start left coordinate, which is only valid for overlaying within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
</platePicPosLeft>
<overlaidTextTransparency min="0" max="100">
    <!--ro, opt, int, range:[0,100], unit:%, attr:min{req, int},max{req, int}-->1
</overlaidTextTransparency>
<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, attr:opt{req, string}-->12hour
</timeStyle>
<PictureInfoList>
    <!--ro, opt, array, subType:object-->
</PictureInfoList>
</CapturePicOverlays>

```

12.5.1.62 Set the configuration parameters for overlaying all OSD text information on a captured picture

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/picParam/capturePicOverlays/info

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<OverlayInfoList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--req, array, see details in the message of XML_OverlayInfo, subType:object, attr:version{opt, string, protocolVersion}-->
    <OverlayInfo>
        <!--opt, object, overlay information-->
        <overlayInfoID>
            <!--req, int, No., range:[1,50]-->1
        </overlayInfoID>
        <itemType>
            <!--req, enum, the type of overlaid information on a picture, subType:int, desc:the type of overlaid information on a picture-->70
        </itemType>
        <itemOverlayEnabled>
            <!--req, bool-->true
        </itemOverlayEnabled>
        <customName>
            <!--req, string, self-defined name of the overlay information, range:[1,32]-->test
        </customName>
        <changeLineNum>
            <!--req, int, range:[0,100]-->2
        </changeLineNum>
        <spaceNum>
            <!--req, int, range:[0,255]-->1
        </spaceNum>
        <startPosEnable>
            <!--opt, bool, whether to enable coordinate configuration-->true
        </startPosEnable>
        <startPosTop>
            <!--req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
        </startPosTop>
        <startPosLeft>
            <!--req, int, Left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
        </startPosLeft>
        <overlayInfoText>
            <!--opt, string, overlay character string, range:[0,128], desc:the maximum string Length of place is 128 bytes, the maximum string Length of intersection No., device No., direction No., direction description and Lane information is 32 bytes., the maximum string Length of camera 1 is 44 bytes, and the maximum string Length of verification unit, verification certificate No., calibration expiration date is 128 bytes-->test
        </overlayInfoText>
        <overlayInfoText2>
            <!--opt, string, overlay character string, range:[1,32], desc:overlay character string-->test
        </overlayInfoText2>
    </OverlayInfo>
</OverlayInfoList>

```

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>

```

12.5.1.63 Get the configuration parameters of overlaying all OSD text information on a captured picture

Request URL

GET /ISAPI/Traffic/channels/<channelID>/picParam/capturePicOverlays/info

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

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

<OverlayInfoList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, array, List of overlay information, subType:object, attr:version{opt, string, protocolVersion}-->
    <OverlayInfo>
        <!--ro, opt, object, overlay information-->
        <OverlayInfoID>
            <!--ro, opt, int, range:[1,50]-->1
        </OverlayInfoID>
        <itemType>
            <!--ro, req, enum, subType:int-->70
        </itemType>
        <itemOverlayEnabled>
            <!--ro, req, bool-->true
        </itemOverlayEnabled>
        <customName>
            <!--ro, req, string, self-defined name of the overlay information, range:[1,32]-->test
        </customName>
        <changeLineNum>
            <!--ro, req, int, range:[0,100]-->2
        </changeLineNum>
        <spaceNum>
            <!--ro, req, int, range:[0,255]-->1
        </spaceNum>
        <startPosEnable>
            <!--ro, opt, bool, whether to enable coordinate configuration-->true
        </startPosEnable>
        <startPosTop>
            <!--ro, opt, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
        </startPosTop>
        <startPosLeft>
            <!--ro, opt, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
        </startPosLeft>
        <OverlayInfoText>
            <!--ro, opt, string, overlay character string, range:[0,128], desc:the maximum string length of place is 128 bytes, the maximum string length of intersection No., device No., direction No., direction description and lane information is 32 bytes., the maximum string length of camera 1 is 44 bytes, and the maximum string length of verification unit, verification certificate No., calibration expiration date is 128 bytes-->test
        </OverlayInfoText>
        <OverlayInfoText2>
            <!--ro, opt, string, overlay character string, range:[1,32], desc:overlay character string-->test
        </OverlayInfoText2>
    </OverlayInfo>
</OverlayInfoList>
```

12.5.1.64 Get the configuration parameters of overlaying a piece of OSD text information on a captured picture

Request URL

GET /ISAPI/Traffic/channels/<channelID>/picParam/capturePicOverlays/info/<textID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
textID	string	--

Request Message

None

Response Message

```

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

<OverlayInfo xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, overlay information, attr:version{opt, string, protocolVersion}-->
    <overlayInfoID>
        <!--ro, opt, int, No., range:[1,50]-->1
    </overlayInfoID>
    <itemType>
        <!--ro, req, enum, overlaid information type, subType:int, desc:overlaid information type-->70
    </itemType>
    <itemOverlayEnabled>
        <!--ro, req, bool, whether to overlay the item-->true
    </itemOverlayEnabled>
    <customName>
        <!--ro, req, string, self-defined name of the overlay information, range:[1,32]-->test
    </customName>
    <changeLineNum>
        <!--ro, req, int, number of Line feeds, range:[0,100]-->2
    </changeLineNum>
    <spaceNum>
        <!--ro, req, int, number of spaces, range:[0,255]-->1
    </spaceNum>
    <startPosEnable>
        <!--ro, opt, bool, whether to enable coordinate configuration-->true
    </startPosEnable>
    <startPosTop>
        <!--ro, opt, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
    </startPosTop>
    <startPosLeft>
        <!--ro, opt, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
    </startPosLeft>
    <overlayInfoText>
        <!--ro, opt, string, overlay character string, range:[0,128], desc:the maximum string length of place is 128 bytes, the maximum string length of intersection No., device No., direction No., direction description and Lane information is 32 bytes., the maximum string length of camera 1 is 44 bytes, and the maximum string length of verification unit, verification certificate No., calibration expiration date is 128 bytes-->test
    </overlayInfoText>
    <overlayInfoText2>
        <!--ro, opt, string, overlay character string, range:[1,32], desc:overlay character string-->test
    </overlayInfoText2>
</OverlayInfo>

```

12.5.1.65 Set the configuration parameters for overlaying a

piece of OSD text information on a captured picture **Request URL**

PUT /ISAPI/Traffic/channels/<channelID>/picParam/capturePicOverlays/info/<textID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
textID	string	--

Request Message

```

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

<OverlayInfo xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, overlay information, attr:version{opt, string, protocolVersion}-->
  <overlayInfoID>
    <!--req, int, overlayed information ID, range:[1,50]-->
  </overlayInfoID>
  <itemType>
    <!--req, enum, overlayed information type, subType:int, desc:0-unknown, 1-place, 2-intersection No., 3-device No., 4-direction, 6-Lane No., 7-lane, 8-capture time (without millisecond), 9-capture time (with millisecond), 10-license plate number, 11-vehicle color, 12-vehicle type, 13-vehicle brand, 14-vehicle speed, 15-speed limit sign, 16-vehicle length (between 1 and 99 meters), 17-violation code (traffic violation information is more useful than code, e.g., normal, low speed, overspeed, reverse driving, running the red light, occupying lane, driving over yellow lane line, etc.), 18-camera information, 19-traffic violation, 20-overspeed ratio, 21-red light start time, 22-red light end time, 23-red light time, 24-security code, 25-capture No., 26-seatbelt, 27-reserved, 28-sun visor, 29-lane direction, 30/31/32-reserved, 33-yellow label vehicle detection, 34-dangerous goods transport vehicle detection, 35-vehicle sub-brand detection, 36-vehicle direction, 37-window hangings, 38-making a call, 39-confidence, 40-verification unit, 41-verification certificate No., 42-calibration expiration date, 43-longitude and latitude, 44-tissue box detection, 45-baby in arm detection, 46-label detection, 47-decoration detection, 48-face score, 49-face No., 50-violation description, 51-marked speed limit, 52-segment speed, 53-segment distance, 54-segment overspeed ratio, 55-segment name, 56-segment ID, 57-traffic accident detection, 58-smoking, 59-wearing helmet, 60-manned, 61-congestion-->70
  </itemType>
  <itemOverlayEnabled>
    <!--req, bool, whether to overlay the item-->true
  </itemOverlayEnabled>
  <customName>
    <!--req, string, custom overlaying name, range:[1,32]-->test
  </customName>
  <changeLineNum>
    <!--req, int, number of line feeds, range:[0,100]-->2
  </changeLineNum>
  <spaceNum>
    <!--req, int, number of spaces, range:[0,255]-->1
  </spaceNum>
  <startPosEnable>
    <!--opt, bool, whether to enable coordinate configuration-->true
  </startPosEnable>
  <startPosTop>
    <!--req, int, start top coordinate, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
  </startPosTop>
  <startPosLeft>
    <!--req, int, start left coordinate, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
  </startPosLeft>
  <overlayInfoText>
    <!--opt, string, overlay character string, range:[0,128], desc:the maximum string length of place is 128 bytes, the maximum string length of intersection No., device No., direction No., direction description and lane information is 32 bytes., the maximum string length of camera 1 is 44 bytes, and the maximum string length of verification unit, verification certificate No., calibration expiration date is 128 bytes-->test
  </overlayInfoText>
  <overlayInfoText2>
    <!--opt, string, overlay character string for camera 2, range:[1,32], desc:overlay character string for camera 2-->test
  </overlayInfoText2>
</OverlayInfo>

```

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>

```

12.5.1.66 Set the overlay parameters in the composite captured pictures

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/picParam/mergePicOverlays

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```

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

<MergePicOverlays xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, object, see details in the message of XML_OverlayInfoList, attr:version{req, string, protocolVersion}-->
    <overlayInfoEnabled>
        <!--req, bool, whether to enable the function-->true
    </overlayInfoEnabled>
    <OverlayInfoList>
        <!--req, array, list of overlay information, subType:object-->
        <OverlayInfo>
            <!--req, object, overlay information-->
            <overlayInfoID>
                <!--req, int, overlay ID-->0
            </overlayInfoID>
            <itemType>
                <!--req, enum, overlay information type, subType:string, desc:0-unknown, 1-place, 2-intersection No., 3-device No., 4-direction No., 5-direction, 6-Lane No., 7-Lane, 8-capture time (without millisecond), 9-capture time (with millisecond), 10-license plate number, 11-vehicle color, 12-vehicle type, 13-vehicle brand, 14-vehicle speed, 15-speed limit sign, 16-vehicle length (between 1 and 99 meters), 17-violation code (traffic violation information is more useful than code, e.g., normal, low speed, overspeed, reverse driving, running the red light, occupying lane, driving over yellow lane line, etc.), 18-camera information, 19-traffic violation, 20-overspeed ratio, 21-red light start time, 22-red light end time, 23-red light time, 24-security code, 25-capture No., 26-seatbelt, 27-reserved, 28-sun visor, 29-Lane direction, 30-license plate color, 31-scene No., 32-scene name, 33-yellow label vehicle detection, 34-dangerous goods transport vehicle detection, 35-vehicle sub-brand detection, 36-vehicle direction, 37-window hangings, 38-making a call, 39-confidence, 40-verification unit, 41-verification certificate No., 42-calibration expiration date, 43-longitude and latitude, 44-tissue box detection, 45-baby in arm detection, 46-label detection, 47-decoration detection, 48-face score, 49-face No., 50-violation description, 51-marked speed limit, 52-segment speed, 53-segment distance, 54-segment overspeed ratio, 55-segment name, 56-segment ID, 57-traffic accident detection, 58-smoking, 59-wearing helmet, 60-manned, 61-person features, 62-using mobile phone, 63-umbrella tent, 66-vehicle head/tail, 67-vehicle door status, 68-vehicle loading rate, 69-number of operating personnel, 70-parking space patrol, 71-intelligent tracing (scheduled capture for Parking Space Patrol), 72-Ringelmann emittance (for black smoke detection), 73-vehicle purpose (bus, school bus, coach, taxi, ambulance, etc.), 74-vehicle bodywork features, 75-vehicle entry & exit status, 76-parking space No., 77-parking space status, 78-total parking spaces, 79-occupied parking spaces, 80-vacant parking spaces, 81-fake license plate, 82-whether wearing mask, 83-vehicle noise decible detection by device (in inverse proportion to the distance between device and vehicle), 85-reflective stripe-->0
            </itemType>
            <itemOverlayEnabled>
                <!--req, bool, whether to overlay the information-->true
            </itemOverlayEnabled>
            <customName>
                <!--req, string, custom overlaying name, range:[0,32], desc:max. 32 bytes; if this node is not returned or empty, the default name will be overlaid-->test
            </customName>
            <changeLineNum>
                <!--req, int, number of line feeds, range:[0,100], desc:the default value is 0-->0
            </changeLineNum>
            <spaceNum>
                <!--req, int, number of spaces, range:[0,255], desc:the default value is 0-->0
            </spaceNum>
            <startPosEnable>
                <!--opt, bool, whether to enable coordinate configuration, desc:it is valid for overlay within the picture; line breaking and space function will be unavailable after it is enabled; it is recommended to be disabled-->true
            </startPosEnable>
            <startPostop>
                <!--req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
            </startPostop>
            <startPosLeft>
                <!--req, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
            </startPosLeft>
            <overlayInfoText>
                <!--opt, string, it is valid when itemType is set to place, device No., direction No., direction description, Lane information, and camera, range:[1,128], desc:the maximum string length of place is 128 bytes, the maximum string length of intersection No., device No., direction No., direction description and Lane information is 32 bytes., the maximum string length of camera 1 is 44 bytes, and the maximum string length of verification unit, verification certificate No., calibration expiration date is 128 bytes-->test
            </overlayInfoText>
            <overlayInfoText2>
                <!--opt, string, overlay character string, range:[1,32], desc:for camera 2, which is valid when <itemType> is camera information; the maximum string length of camera 2 is 32 bytes-->test
            </overlayInfoText2>
        </OverlayInfo>
    </OverlayInfoList>
    <linePercent>
        <!--req, int, percentage of overlaying lines, range:[0,100], desc:the default value is 100-->1
    </linePercent>
    <itemsStyle>
        <!--req, enum, overlay mode, subType:string, desc:"horizontal" (default), "vertical"-->horizontal
    </itemsStyle>
    <charStyle>
        <!--req, enum, font type, subType:string, desc:font type: 0-SimSun (default), 1-Wei-->song_type
    </charStyle>
    <charSize>
        <!--req, enum, character size, subType:string, desc:0 (8*16), 1 (16*32), 2 (48*48), 3 (32*64), 4 (64*128)-->0
    </charSize>
    <charPosition>
        <!--req, enum, position of the text overlaid in the picture, subType:int, desc:0 (in the picture), 1 (outside the top edge of the picture), 2 (outside the bottom edge of the picture)-->0
    </charPosition>

```

```

<charInterval>
    <!--req, int, character pitch, range:[0,16], desc:the default value is 0, unit: pixel-->0
</charInterval>
<fontColor>
    <!--req, int, foreground color, desc:it is the RGB value directly obtained via the palette and its default value is 0xffffffff (white)-->0
</fontColor>
<backColor>
    <!--req, int, background color, desc:it is the RGB value directly obtained via the palette and its default value is 0x0 (black)-->0
</backColor>
<colorAdapt>
    <!--opt, bool, whether to enable color self-adaption-->true
</colorAdapt>
<zeroizeEnable>
    <!--opt, bool, whether to enable zero filling for OSD overlay, desc:it is used to enable zero filling for vehicle speed, speed limit, overspeed ratio, and Lane No. Zero filling is enabled by default-->true
</zeroizeEnable>
<platePicOverlay>
    <!--opt, bool, whether to enable overlaying license plate thumbnail-->true
</platePicOverlay>
<platePicPosTop>
    <!--req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
</platePicPosTop>
<platePicPosLeft>
    <!--req, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
</platePicPosLeft>
<PictureInfoList>
    <!--opt, array, overlay information list of other captured picture types, subType:object, desc:non-detection picture (background picture)-->
<PictureInfo>
    <!--opt, object, picture information-->
<pictureType>
    <!--req, enum, picture type, subType:string, desc:"illegalPicture" (violation capture, which corresponds to "compositePicture" returned by ANPR for violation occurrence); whether violation occurred can be indicated by whether "illegalInfo" is returned by ANPR-->illegalPicture
</pictureType>
<overlayInfoEnabled>
    <!--req, bool, whether to enable the function-->true
</overlayInfoEnabled>
<OverlayInfoList>
    <!--req, array, list of overlay information, subType:object-->
<OverlayInfo>
    <!--req, object, overlay information-->
<overlayInfoID>
    <!--req, int, overlay ID-->0
</overlayInfoID>
<itemType>
    <!--req, enum, overlaid information type, subType:string, desc:0-unknown, 1-place, 2-intersection No., 3-device No., 4-direction No., 5-direction, 6-Lane No., 7-Lane, 8-capture time (without millisecond), 9-capture time (with millisecond), 10-license plate number, 11-vehicle color, 12-vehicle type, 13-vehicle brand, 14-vehicle speed, 15-speed limit sign, 16-vehicle length (between 1 and 99 meters), 17-violation code (traffic violation information is more useful than code, e.g., normal, low speed, overspeed, reverse driving, running the red light, occupying lane, driving over yellow lane line, etc.), 18-camera information, 19-traffic violation, 20-overspeed ratio, 21-red light start time, 22-red light end time, 23-red light time, 24-security code, 25-capture No., 26-seatbelt, 27-reserved, 28-sun visor, 29-lane direction, 30-license plate color, 31-scene No., 32-scene name, 33-yellow label vehicle detection, 34-dangerous goods transport vehicle detection, 35-vehicle sub-brand detection, 36-vehicle direction, 37-window hangings, 38-making a call, 39-confidence, 40-verification unit, 41-verification certificate No., 42-calibration expiration date, 43-longitude and latitude, 44-tissue box detection, 45-baby in arm detection, 46-label detection, 47-decoration detection, 48-face score, 49-face No., 50-violation description, 51-marked speed limit, 52-segment speed, 53-segment distance, 54-segment overspeed ratio, 55-segment name, 56-segment ID, 57-traffic accident detection, 58-smoking, 59-wearing helmet, 60-manned, 61-person features, 62-using mobile phone, 63-umbrella tent, 66-vehicle head/tail, 67-vehicle door status, 68-vehicle loading rate, 69-number of operating personnel, 70-parking space patrol, 71-intelligent tracing (scheduled capture for Parking Space Patrol), 72-Ringemann emittance (for black smoke detection), 73-vehicle purpose (bus, school bus, coach, taxi, ambulance, etc.), 74-vehicle bodywork features, 75-vehicle entry & exit status, 76-parking space No., 77-parking space status, 78-total parking spaces, 79-occupied parking spaces, 80-vacant parking spaces, 81-fake license plate, 82-whether wearing mask, 83-vehicle noise decible detection by device (in inverse proportion to the distance between device and vehicle), 85-reflective stripe-->0
</itemType>
<itemOverlayEnabled>
    <!--req, bool, whether to overlay the information-->true
</itemOverlayEnabled>
<customName>
    <!--req, string, custom overlaying name, range:[0,32], desc:max. 32 bytes; if this node is not returned or empty, the default name will be overlaid-->test
</customName>
<changeLineNum>
    <!--req, int, number of line feeds, range:[0,100], desc:the default value is 0-->0
</changeLineNum>
<spaceNum>
    <!--req, int, number of spaces, range:[0,255], desc:the default value is 0-->0
</spaceNum>
<startPosEnable>
    <!--opt, bool, whether to enable coordinate configuration, desc:it is valid for overlay within the picture; Line breaking and space function will be unavailable after it is enabled; it is recommended to be disabled-->true
</startPosEnable>
<startPosTop>
    <!--req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
</startPosTop>
<startPosLeft>
    <!--req, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
</startPosLeft>
<overlayInfoText>
    <!--opt, string, it is valid when itemType is set to place, device No., direction No., direction description, Lane information, and camera, range:[1,128], desc:the maximum string length of place is 128 bytes, the maximum string length of intersection No., device No., direction No., direction description and Lane information is 32 bytes., the maximum string length of camera 1 is 44 bytes, and the maximum string length of verification unit, verification certificate No., calibration expiration date is 128 bytes-->test

```

```

        </overlayInfoText>
        <overlayInfoText2>
            <!--opt, string, overlay character string, range:[1,32], desc:for camera 2, which is valid when <itemType> is camera information; the maximum
string length of camera 2 is 32 bytes-->test
        </overlayInfoText2>
        </OverlayInfo>
        </OverlayInfoList>
        <linePercent>
            <!--req, int, Percentage, range:[0,100], desc:100 by default-->1
        </linePercent>
        <itemsStyle>
            <!--req, enum, overlay mode, subType:string, desc:"horizontal" (default), "vertical"-->horizontal
        </itemsStyle>
        <charStyle>
            <!--req, enum, font type, subType:string, desc:"song_type"-SimSun (default), "wei_type"-Wei-->song_type
        </charStyle>
        <charSize>
            <!--req, enum, character size, subType:string, desc:0 (8*16), 1 (16*32), 2 (48*48), 3 (32*64), 4 (64*128)-->0
        </charSize>
        <charPosition>
            <!--req, enum, position of the text overlaid in the picture, subType:int, desc:0 (in the picture),1 (outside the top edge of the picture), 2
(outside the bottom edge of the picture)-->0
        </charPosition>
        <charInterval>
            <!--req, int, character pitch, range:[0,16], desc:the default value is 0, unit: pixel-->0
        </charInterval>
        <foreColor>
            <!--req, int, foreground color, desc:it is the RGB value directly obtained via the palette and its default value is 0xffffffff (white)-->0
        </foreColor>
        <backColor>
            <!--req, int, background color, desc:it is the RGB value directly obtained via the palette and its default value is 0x0 (black)-->0
        </backColor>
        <colorAdapt>
            <!--opt, bool, whether to enable color self-adaption-->true
        </colorAdapt>
        <zeroizeEnable>
            <!--opt, bool, whether to enable zero filling for OSD overlay, desc:it is used to enable zero filling for vehicle speed, speed limit, overspeed
ratio, and Lane No. Zero filling is enabled by default-->true
        </zeroizeEnable>
        <platePicOverlay>
            <!--opt, bool, whether to enable overlaying license plate thumbnail-->true
        </platePicOverlay>
        <platePicPosTop>
            <!--req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture
height, and the default value is 0-->0
        </platePicPosTop>
        <platePicPosLeft>
            <!--req, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture
width, and the default value is 0-->0
        </platePicPosLeft>
    </PictureInfo>
    <PictureInfoList>
</MergePicOverlays>
```

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>
```

12.5.1.67 Get the parameters of OSD overlay on the composite captured picture

Request URL

GET /ISAPI/Traffic/channels/<channelID>/picParam/mergePicOverlays

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<MergePicOverlays xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, see details in the message of XML_OverlayInfoList, attr:version{req, string, protocolVersion}-->
  <overlayInfoEnabled>
    <!--ro, req, bool, whether to enable the function-->true
  </overlayInfoEnabled>
  <OverlayInfoList>
    <!--ro, req, array, list of overlay information, subType:object-->
    <OverlayInfo>
      <!--ro, req, object, overlay information-->
      <overlayInfoID>
        <!--ro, req, int, overlay ID-->0
      </overlayInfoID>
      <itemType>
        <!--ro, req, enum, overlay information type, subType:string, desc:0-unknown, 1-place, 2-intersection No., 3-device No., 4-direction No., 5-direction, 6-Lane No., 7-lane, 8-capture time (without millisecond), 9-capture time (with millisecond), 10-license plate number, 11-vehicle color, 12-vehicle type, 13-vehicle brand, 14-vehicle speed, 15-speed limit sign, 16-vehicle length (between 1 and 99 meters), 17-violation code (traffic violation information is more useful than code, e.g., normal, Low speed, overspeed, reverse driving, running the red light, occupying lane, driving over yellow lane line, etc.), 18-camera information, 19-traffic violation, 20-overspeed ratio, 21-red light start time, 22-red light end time, 23-red light time, 24-security code, 25-capture No., 26-seatbelt, 27-reserved, 28-sun visor, 29-lane direction, 30-license plate color, 31-scene No., 32-scene name, 33-yellow label vehicle detection, 34-dangerous goods transport vehicle detection, 35-vehicle sub-brand detection, 36-vehicle direction, 37-window hangings, 38-making a call, 39-confidence, 40-verification unit, 41-verification certificate No., 42-calibration expiration date, 43-longitude and latitude, 44-tissue box detection, 45-baby in arm detection, 46-label detection, 47-decoration detection, 48-face score, 49-face No., 50-violation description, 51-marked speed limit, 52-segment speed, 53-segment distance, 54-segment overspeed ratio, 55-segment name, 56-segment ID, 57-traffic accident detection, 58-smoking, 59-wearing helmet, 60-manned, 61-person features, 62-using mobile phone, 63-umbrella tent, 66-vehicle head/tail, 67-vehicle door status, 68-vehicle loading rate, 69-number of operating personnel, 70-parking space patrol, 71-intelligent tracing (scheduled capture for Parking Space Patrol), 72-Ringelmann emittance (for black smoke detection), 73-vehicle purpose (bus, school bus, coach, taxi, ambulance, etc.), 74-vehicle bodywork features, 75-vehicle entry & exit status, 76-parking space No., 77-parking space status, 78-total parking spaces, 79-occupied parking spaces, 80-vacant parking spaces, 81-fake license plate, 82-whether wearing mask, 83-vehicle noise decible detection by device (in inverse proportion to the distance between device and vehicle), 85-reflective stripe-->0
      </itemType>
      <itemOverlayEnabled>
        <!--ro, req, bool, whether to overlay the information-->true
      </itemOverlayEnabled>
      <customName>
        <!--ro, req, string, custom overlaying name, range:[0,32], desc:max. 32 bytes; if this node is not returned or empty, the default name will be overlaid-->test
      </customName>
      <changeLineNum>
        <!--ro, req, int, number of line feeds, range:[0,100], desc:the default value is 0-->0
      </changeLineNum>
      <spaceNum>
        <!--ro, req, int, number of spaces, range:[0,255], desc:the default value is 0-->0
      </spaceNum>
      <startPosEnable>
        <!--ro, opt, bool, whether to enable coordinate configuration, desc:it is valid for overlay within the picture; Line breaking and space function will be unavailable after it is enabled; it is recommended to be disabled-->true
      </startPosEnable>
      <startPosTop>
        <!--ro, req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
      </startPosTop>
      <startPosLeft>
        <!--ro, req, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
      </startPosLeft>
      <overlayInfoText>
        <!--ro, opt, string, it is valid when itemType is set to place, device No., direction No., direction description, lane information, and camera, range:[1,128], desc:the maximum string length of place is 128 bytes, the maximum string length of intersection No., device No., direction No., direction description and lane information is 32 bytes., the maximum string length of camera 1 is 44 bytes, and the maximum string length of verification unit, verification certificate No., calibration expiration date is 128 bytes-->test
      </overlayInfoText>
      <overlayInfoText2>
        <!--ro, opt, string, overlay character string, range:[1,32], desc:for camera 2, which is valid when <itemType> is camera information; the maximum string length of camera 2 is 32 bytes-->test
      </overlayInfoText2>
    </OverlayInfo>
  </OverlayInfoList>
  <linePercent>
    <!--ro, req, int, percentage of overlaying lines, range:[0,100], desc:the default value is 100-->1
  </linePercent>
  <itemsStyle>
    <!--ro, req, enum, overlay mode, subType:string, desc:"horizontal" (default), "vertical"-->horizontal
  </itemsStyle>
  <charStyle>
    <!--ro, req, enum, font type, subType:string, desc:"song_type"-SimSun (default), "wei_type"-Wei-->song_type
  </charStyle>
  <charSize>
    <!--ro, req, enum, font size, subType:string, desc:"0"-16*16(Chinese)/8*16(English), "1"-32*32(Chinese)/16*32(English), "2"-48*48, "3"-64*64(Chinese)/32*32(English) -->0
  </charSize>

```

```

04:04(CRUISE)/>2:04(ENGLISH)-->
</charSize>
<charPosition>
    <!--ro, req, enum, position of the text overlaid in the picture, subType:int, desc:"0"-overlaid on the picture, "1"-overlaid outside the top edge of the picture, "2"-overlaid outside the bottom edge of the picture-->0
</charPosition>
<charInterval>
    <!--ro, req, int, character pitch, range:[0,16], desc:the default value is 0, unit: pixel-->0
</charInterval>
<foreColor>
    <!--ro, req, int, foreground color, desc:it is the RGB value directly obtained by the palette, the value is between 0 and 0xffffffff and the default value is 0xffffffff (white)-->0
</foreColor>
<backColor>
    <!--ro, req, int, background color, desc:it is the RGB value directly obtained via the palette and its default value is 0x0 (black)-->0
</backColor>
<colorAdapt>
    <!--ro, opt, bool, whether to enable color self-adaption-->true
</colorAdapt>
<zeroizeEnable>
    <!--ro, opt, bool, whether to enable zero filling for OSD overlay, desc:it is used to enable zero filling for vehicle speed, speed limit, overspeed ratio, and Lane No. Zero filling is enabled by default-->true
</zeroizeEnable>
<platePicOverlay>
    <!--ro, opt, bool, whether to enable overlaying license plate thumbnail-->true
</platePicOverlay>
<platePicPosTop>
    <!--ro, req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
</platePicPosTop>
<platePicPosLeft>
    <!--ro, req, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
</platePicPosLeft>
<PictureInfoList>
    <!--ro, opt, array, overlay information list of other captured picture types, subType:object, desc:non-detection picture (background picture)-->
<PictureInfo>
    <!--ro, opt, object, picture information-->
</pictureType>
    <!--ro, req, enum, picture type, subType:string, desc:"illegalPicture" (violation capture, which corresponds to "compositePicture" returned by ANPR for violation occurrence); whether violation occurred can be indicated by whether "illegalInfo" is returned by ANPR-->illegalPicture
</pictureType>
<overlayInfoEnabled>
    <!--ro, req, bool, whether to enable the function-->true
</overlayInfoEnabled>
<OverlayInfoList>
    <!--ro, req, array, List of overlay information, subType:object-->
<OverlayInfo>
    <!--ro, req, object, overlay information-->
<overlayInfoID>
    <!--ro, req, int, overlay ID-->0
</overlayInfoID>
<itemType>
    <!--ro, req, enum, overlaid information type, subType:string, desc:0-unknown, 1-place, 2-intersection No., 3-device No., 4-direction No., 5-direction, 6-Lane No., 7-Lane, 8-capture time (without millisecond) 9-capture time (with millisecond), 10-license plate number, 11-vehicle color, 12-vehicle type, 13-vehicle brand, 14-vehicle speed, 15-speed Limit sign, 16-vehicle Length (between 1 and 99 meters), 17-violation code (traffic violation information is more useful than code, e.g., normal, low speed, overspeed, reverse driving, running the red light, occupying Lane, driving over yellow Lane Line, etc.), 18-camera information, 19-traffic violation, 20-overspeed ratio, 21-red Light start time, 22-red Light end time, 23-red Light time, 24-security code, 25-camera No., 26-seatbelt, 27-reserved, 28-sun visor, 29-Lane direction, 30-license plate color, 31-scene No., 32-scene name, 33-yellow Label vehicle detection, 34-dangerous goods transport vehicle detection, 35-vehicle sub-brand detection, 36-vehicle direction, 37-window hangings, 38-making a call, 39-confidence, 40-verification unit, 41-verification certificate No., 42-calibration expiration date, 43-Longitude and Latitude, 44-tissue box detection, 45-baby in arm detection, 46-label detection, 47-decoration detection, 48-face score, 49-face No., 50-violation description, 51-marked speed limit, 52-segment speed, 53-segment distance, 54-segment overspeed ratio, 55-segment name, 56-segment ID, 57-traffic accident detection, 58-smoking, 59-wearing helmet, 60-manned, 61-person features, 62-using mobile phone, 63-umbrella tent, 66-vehicle head/tail, 67-vehicle door status, 68-vehicle Loading rate, 69-number of operating personnel, 70-parking space patrol, 71-intelligent tracing (scheduled capture for Parking Space Patrol), 72-Ringelmann emittance (for black smoke detection), 73-vehicle purpose (bus, school bus, coach, taxi, ambulance, etc.), 74-vehicle bodywork features, 75-vehicle entry & exit status, 76-parking space No., 77-parking space status, 78-total parking spaces, 79-occupied parking spaces, 80-vacant parking spaces, 81-fake license plate, 82-whether wearing mask, 83-vehicle noise decible detection by device (in inverse proportion to the distance between device and vehicle), 85-reflective stripe-->0
</itemType>
<itemOverlayEnabled>
    <!--ro, req, bool, whether to overlay the information-->true
</itemOverlayEnabled>
<customName>
    <!--ro, req, string, custom overlaying name, range:[0,32], desc:max. 32 bytes; if this node is not returned or empty, the default name will be overlaid-->test
</customName>
<changeLineNum>
    <!--ro, req, int, number of line feeds, range:[0,100], desc:the default value is 0-->0
</changeLineNum>
<spaceNum>
    <!--ro, req, int, number of spaces, range:[0,255], desc:the default value is 0-->0
</spaceNum>
<startPosEnable>
    <!--ro, opt, bool, whether to enable coordinate configuration, desc:it is valid for overlay within the picture; Line breaking and space function will be unavailable after it is enabled; it is recommended to be disabled-->true
</startPosEnable>
<startPosTop>
    <!--ro, req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
</startPosTop>
<startPosLeft>
    <!--ro, req, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
</startPosLeft>

```

```

</startPosLeft>
<overlayInfoText>
  <!--ro, opt, string, it is valid when itemType is set to place, device No., direction No., direction description, Lane information, and camera, range:[1,128], desc:the maximum string length of place is 128 bytes, the maximum string length of intersection No., device No., direction No., direction description and Lane information is 32 bytes., the maximum string length of camera 1 is 44 bytes, and the maximum string length of verification unit, verification certificate No., calibration expiration date is 128 bytes-->test
</overlayInfoText>
<overlayInfoText2>
  <!--ro, opt, string, overlay character string, range:[1,32], desc:for handling camera 2 when the overlay information type is camera information; the maximum length of camera 2 is 32 bytes-->test
</overlayInfoText2>
</OverlayInfo>
</OverlayInfoList>
<linePercent>
  <!--ro, req, int, overlay line percentage, range:[0,100], desc:the default value is 100-->1
</linePercent>
<itemsStyle>
  <!--ro, req, enum, overlay mode, subType:string, desc:"horizontal" (default), "vertical"-->horizontal
</itemsStyle>
<charStyle>
  <!--ro, req, enum, font type, subType:string, desc:"song_type"-SimSun (default), "wei_type"-Wei-->song_type
</charStyle>
<charSize>
  <!--ro, req, enum, font size, subType:string, desc:"0"-16*16(Chinese)/8*16(English), "1"-32*32(Chinese)/16*32(English), "2"-48*48, "3"-64*64(Chinese)/32*64(English)-->0
</charSize>
<charPosition>
  <!--ro, req, enum, position of the text overlaid in the picture, subType:int, desc:"0"-overlaid on the picture, "1"-overlaid outside the top edge of the picture, "2"-overlaid outside the bottom edge of the picture-->0
</charPosition>
<charInterval>
  <!--ro, req, int, character pitch, range:[0,16], desc:the default value is 0, unit: pixel-->0
</charInterval>
<foreColor>
  <!--ro, req, int, foreground color, desc:it is the RGB value directly obtained by the palette, the value is between 0 and 0xffffffff and the default value is 0xffffffff (white)-->0
</foreColor>
<backColor>
  <!--ro, req, int, background color, desc:it is the RGB value directly obtained via the palette and its default value is 0x0 (black)-->0
</backColor>
<colorAdapt>
  <!--ro, opt, bool, whether to enable color self-adaption-->true
</colorAdapt>
<zeroizeEnable>
  <!--ro, opt, bool, whether to enable zero filling for OSD overlay, desc:it is used to enable zero filling for vehicle speed, speed limit, overspeed ratio, and Lane No. Zero filling is enabled by default-->true
</zeroizeEnable>
<platePicOverlay>
  <!--ro, opt, bool, whether to enable overlaying license plate thumbnail-->true
</platePicOverlay>
<platePicPosTop>
  <!--ro, req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
</platePicPosTop>
<platePicPosLeft>
  <!--ro, req, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
</platePicPosLeft>
</PictureInfo>
</PictureInfoList>
</MergePicOverlays>

```

12.5.1.68 Get the overlay capability in the composite captured pictures

Request URL

GET /ISAPI/Traffic/channels/<channelID>/picParam/mergePicOverlays/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

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

```

12.5.1.69 Set the configuration parameters for overlaying all OSD text information on the composite captured picture

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/picParam/mergePicOverlays/info

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

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

<OverlayInfoList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--req, array, List of overlay information, subType:object, attr:version{req, string, protocolVersion}-->
    <OverlayInfo>
        <!--req, object, overlay information-->
        <OverlayInfoID>
            <!--req, int, overlay information ID-->0
        </OverlayInfoID>
        <itemType>
            <!--req, enum, overlay information type, subType:string, desc:0-unknown, 1-place, 2-intersection No., 3-device No., 4-direction No., 5-direction, 6-Lane No., 7-Lane, 8-capture time (without millisecond), 9-capture time (with millisecond), 10-license plate number, 11-vehicle color, 12-vehicle type, 13-vehicle brand, 14-vehicle speed, 15-speed limit sign, 16-vehicle length (between 1 and 99 meters), 17-violation code (traffic violation information is more useful than code, e.g., normal, low speed, overspeed, reverse driving, running the red light, occupying lane, driving over yellow Lane Line, etc.), 18-camera information, 19-traffic violation, 20-overspeed ratio, 21-red light start time, 22-red light end time, 23-red light time, 24-security code, 25-capture No., 26-seatbelt, 27-reserved, 28-sun visor, 29-Lane direction, 30-license plate color, 31-scene No., 32-scene name, 33-yellow label vehicle detection, 34-dangerous goods transport vehicle detection, 35-vehicle sub-brand detection, 36-vehicle direction, 37-window hangings, 38-making a call, 39-confidence, 40-verification unit, 41-verification certificate No., 42-calibration expiration date, 43-longitude and latitude, 44-tissue box detection, 45-baby in arm detection, 46-label detection, 47-decoration detection, 48-face score, 49-face No., 50-violation description, 51-marked speed limit, 52-segment speed, 53-segment distance, 54-segment overspeed ratio, 55-segment name, 56-segment ID, 57-traffic accident detection, 58-smoking, 59-wearing helmet, 60-manned, 61-person features, 62-using mobile phone, 63-umbrella tent, 66-vehicle head/tail, 67-vehicle door status, 68-vehicle loading rate, 69-number of operating personnel, 70-parking space patrol, 71-intelligent tracing (scheduled capture for parking space patrol), 72-Ringelmann emittance (for black smoke detection), 73-vehicle purpose (bus, school bus, coach, taxi, ambulance, etc.), 74-vehicle bodywork features, 75-vehicle entry & exit status, 76-parking space No., 77-parking space status, 78-total parking spaces, 79-occupied parking spaces, 80-vacant parking spaces, 81-fake license plate, 82-whether wearing mask-->0
            </itemType>
            <itemOverlayEnabled>
                <!--req, bool, whether to overlay the information-->true
            </itemOverlayEnabled>
            <customName>
                <!--req, string, custom overlaying name, range:[0,32], desc:max. 32 bytes; if this node is not returned or empty, the default name will be overlaid-->
            </customName>
            <changeLineNum>
                <!--req, int, number of line feeds, range:[0,100], desc:the default value is 0-->0
            </changeLineNum>
            <spaceNum>
                <!--req, int, number of spaces, range:[0,255], desc:the default value is 0-->0
            </spaceNum>
            <startPosEnable>
                <!--opt, bool, whether to enable coordinate configuration, desc:it is valid for overlay within the picture; line breaking and space function will be unavailable after it is enabled; it is recommended to be disabled-->true
            </startPosEnable>
            <startPosTop>
                <!--req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
            </startPosTop>
            <startPosLeft>
                <!--req, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
            </startPosLeft>
            <OverlayInfoText>
                <!--opt, string, it is valid when itemType is set to place, device No., direction No., direction description, Lane information, and camera, range:[1,128], desc:the maximum string length of place is 128 bytes, the maximum string length of intersection No., device No., direction No., direction description and Lane information is 32 bytes., the maximum string length of camera 1 is 44 bytes, and the maximum string length of verification unit, verification certificate No., calibration expiration date is 128 bytes-->test
            </OverlayInfoText>
            <OverlayInfoText2>
                <!--opt, string, overlay character string, range:[1,32], desc:for camera 2, which is valid when <itemType> is camera information; the maximum string length of camera 2 is 32 bytes-->test
            </OverlayInfoText2>
        </OverlayInfo>
    </OverlayInfoList>
```

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>
  <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>

```

12.5.1.70 Get parameters of overlaying characters on composite pictures

Request URL

GET /ISAPI/Traffic/channels/<channelID>/picParam/mergePicOverlays/info

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

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

<OverlayInfoList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, array, overlay information list, subType:object, attr:version{req, string, protocolVersion}-->
    <OverlayInfo>
        <!--ro, req, object, overlay information-->
        <OverlayInfoID>
            <!--ro, req, int, overlay ID-->0
        </OverlayInfoID>
        <itemType>
            <!--ro, req, enum, overlaid information type, subtype:string, desc:0-unknown, 1-place, 2-intersection No., 3-device No., 4-direction No., 5-direction, 6-lane No., 7-lane, 8-capture time (without millisecond), 9-capture time (with millisecond), 10-license plate number, 11-vehicle color, 12-vehicle type, 13-vehicle brand, 14-vehicle speed, 15-speed limit sign, 16-vehicle length (between 1 and 99 meters), 17-violation code (traffic violation information is more useful than code, e.g., normal, low speed, overspeed, reverse driving, running the red light, occupying lane, driving over yellow lane line, etc.), 18-camera information, 19-traffic violation, 20-overspeed ratio, 21-red light start time, 22-red light end time, 23-red light time, 24-security code, 25-capture No., 26-seatbelt, 27-reserved, 28-sun visor, 29-lane direction, 30-license plate color, 31-scene name, 33-yellow label vehicle detection, 34-dangerous goods transport vehicle detection, 35-vehicle sub-brand detection, 36-vehicle direction, 37-window hangings, 38-making a call, 39-confidence, 40-verification unit, 41-verification certificate No., 42-calibration expiration date, 43-longitude and latitude, 44-tissue box detection, 45-baby in arm detection, 46-label detection, 47-decoration detection, 48-face score, 49-face No., 50-violation description, 51-marked speed limit, 52-segment speed, 53-segment distance, 54-segment overspeed ratio, 55-segment name, 56-segment ID, 57-traffic accident detection, 58-smoking, 59-wearing helmet, 60-manned, 61-person features, 62-using mobile phone, 63-umbrella tent, 66-vehicle head/tail, 67-vehicle door status, 68-vehicle loading rate, 69-number of operating personnel, 70-parking space patrol, 71-intelligent tracing (scheduled capture for parking space patrol), 72-Ringelmann emittance (for black smoke detection), 73-vehicle purpose (bus, school bus, coach, taxi, ambulance, etc.), 74-vehicle bodywork features, 75-vehicle entry & exit status, 76-parking space No., 77-parking space status, 78-total parking spaces, 79-occupied parking spaces, 80-vacant parking spaces, 81-fake license plate, 82-whether wearing mask-->0
            </itemType>
            <itemOverlayEnabled>
                <!--ro, req, bool, whether to overlay this item-->true
            </itemOverlayEnabled>
            <customName>
                <!--ro, req, string, custom overlaying name, range:[0,32], desc:customize name of the overlay information-->test
            </customName>
            <changeLineNum>
                <!--ro, req, int, the default value is 0, range:[0,100], desc:the default value is 0-->0
            </changeLineNum>
            <spaceNum>
                <!--ro, req, int, the number of spaces, range:[0,255], desc:the default value is 0-->0
            </spaceNum>
            <startPosEnable>
                <!--ro, opt, bool, whether to enable coordinate configuration, desc:whether to enable coordinate configuration-->true
            </startPosEnable>
            <startPosTop>
                <!--ro, req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
            </startPosTop>
            <startPosLeft>
                <!--ro, req, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
            </startPosLeft>
            <OverlayInfoText>
                <!--ro, opt, string, it is valid when itemType is set to place, device No., direction No., direction description, lane information, and camera, range:[1,128], desc:the maximum string length of place is 128 bytes, the maximum string length of intersection No., device No., direction No., direction description and lane information is 32 bytes., the maximum string length of camera 1 is 44 bytes, and the maximum string length of verification unit, verification certificate No., calibration expiration date is 128 bytes-->test
            </OverlayInfoText>
            <OverlayInfoText2>
                <!--ro, opt, string, overlay characters, range:[1,32], desc:for camera 2, which is valid when <itemType> is camera information; the maximum string length of camera 2 is 32 bytes-->test
            </OverlayInfoText2>
        </OverlayInfo>
    </OverlayInfoList>

```

12.5.1.71 Set the configuration parameters for overlaying a piece of OSD text information on the composite captured picture

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/picParam/mergePicOverlays/info/<textID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
textID	string	--

Request Message

```

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

<OverlayInfo xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, overlay information, attr:version{req, string, protocolVersion}-->
  <overlayInfoID>
    <!--req, int, overlayed information ID-->
  </overlayInfoID>
  <itemType>
    <!--req, enum, overlay information type, subType:string, desc:0-unknown,1-place,2-intersection No.,3-device No.,4-direction,6-Lane No.,7-lane,8-capture time (without millisecond),9-capture time (with millisecond),10-license plate number,11-vehicle color,12-vehicle type,13-vehicle brand,14-vehicle speed,15-speed limit sign,16-vehicle length (between 1 and 99 meters),17-violation code (traffic violation information is more useful than code, e.g., normal, low speed, overspeed, reverse driving, running the red light, occupying Lane, driving over yellow Lane Line, etc.),18-camera information,19-traffic violation,20-overspeed ratio,21-red Light start time,22-red Light end time,23-red Light time,24-security code,25-capture No.,26-seatbelt,27-reserved,28-sun visor,29-Lane direction,30/31/32-reserved,33-yellow Label vehicle detection,34-dangerous goods transport vehicle detection,35-vehicle sub-brand detection,36-vehicle direction,37-window hangings,38-making a call,39-confidence,40-verification unit,41-verification certificate No.,42-calibration expiration date,43-longitude and latitude,44-tissue box detection,45-baby in arm detection,46-label detection,47-decoration detection,48-face score,49-face No.,50-violation description,51-marked speed limit,52-segment speed,53-segment distance,54-segment overspeed ratio,55-segment name,56-segment ID,57-traffic accident detection,58-smoking,59-wearing helmet,60-manned,61-congestion-->0
  </itemType>
  <itemOverlayEnabled>
    <!--req, bool, whether to overlay the item-->true
  </itemOverlayEnabled>
  <customName>
    <!--req, string, custom overlaying name, range:[0,32], desc:self-defined name of the overlay information-->test
  </customName>
  <changeLineNum>
    <!--req, int, number of line feeds, range:[0,100], desc:the default value is 0-->0
  </changeLineNum>
  <spaceNum>
    <!--req, int, number of spaces, range:[0,255], desc:the default value is 0-->0
  </spaceNum>
  <startPosEnable>
    <!--opt, bool, whether to enable coordinate configuration, desc:whether to enable coordinate configuration-->true
  </startPosEnable>
  <startPosTop>
    <!--req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
  </startPosTop>
  <startPosLeft>
    <!--req, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
  </startPosLeft>
  <overlayInfoText>
    <!--opt, string, it is valid when itemType is set to place, device No., direction No., direction description, lane information, and camera, range:[1,128], desc:the maximum string length of place is 128 bytes, the maximum string length of intersection No., device No., direction No., direction description and lane information is 32 bytes., the maximum string length of camera 1 is 44 bytes, and the maximum string length of verification unit, verification certificate No., calibration expiration date is 128 bytes-->test
  </overlayInfoText>
  <overlayInfoText2>
    <!--opt, string, overlay character string, range:[1,32], desc:for camera 2, which is valid when <itemType> is camera information; the maximum string length of camera 2 is 32 bytes-->test
  </overlayInfoText2>
</OverlayInfo>

```

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>
  <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>

```

12.5.1.72 Get the configuration parameters of overlaying a piece of OSD text information on the composite captured picture

Request URL

GET /ISAPI/Traffic/channels/<channelID>/picParam/mergePicOverlays/info/<textID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
textID	string	--

Request Message

None

Response Message

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

<OverlayInfo xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, overlay information, attr:version{req, string, protocolVersion}-->
  <overlayInfoID>
    <!--ro, req, int, overlayed information ID, which is between 1 and 50-->0
  </overlayInfoID>
  <itemType>
    <!--ro, req, enum, overlayed information type, subType:string, desc:@unknown, 1-place, 2-intersection No., 3-device No., 4-direction No., 5-direction, 6-lane No., 7-lane, 8-capture time (without millisecond), 9-capture time (with millisecond), 10-license plate number, 11-vehicle color, 12-vehicle type, 13-vehicle brand, 14-vehicle speed, 15-speed limit sign, 16-vehicle length (between 1 and 99 meters), 17-violation code (traffic violation information is more useful than code, e.g., normal, low speed, overspeed, reverse driving, running the red light, occupying lane, driving over yellow lane line, etc.), 18-camera information, 19-traffic violation, 20-overspeed ratio, 21-red light start time, 22-red light end time, 23-red light time, 24-security code, 25-capture No., 26-seatbelt, 27-reserved, 28-sun visor, 29-lane direction, 30-license plate color, 31-scene No., 32-scene name, 33-yellow label vehicle detection, 34-dangerous goods transport vehicle detection, 35-vehicle sub-brand detection, 36-vehicle direction, 37-window hangings, 38-making a call, 39-confidence, 40-verification unit, 41-verification certificate No., 42-calibration expiration date, 43-longitude and latitude, 44-tissue box detection, 45-baby in arm detection, 46-label detection, 47-decoration detection, 48-face score, 49-face No., 50-violation description, 51-marked speed limit, 52-segment speed, 53-segment distance, 54-segment overspeed ratio, 55-segment name, 56-segment ID, 57-traffic accident detection, 58-smoking, 59-wearing helmet, 60-manned, 61-person features, 62-using mobile phone, 63-umbrella tent, 66-vehicle head/tail, 67-vehicle door status, 68-vehicle loading rate, 69-number of operating personnel, 70-parking space patrol, 71-intelligent tracing (scheduled capture for parking space patrol), 72-ringelmann emittance (for black smoke detection), 73-vehicle purpose (bus, school bus, coach, taxi, ambulance, etc.), 74-vehicle bodywork features, 75-vehicle entry & exit status, 76-parking space No., 77-parking space status, 78-total parking spaces, 79-occupied parking spaces, 80-vacant parking spaces, 81-fake license plate, 82-whether wearing mask-->0
  </itemType>
  <itemOverlayEnabled>
    <!--ro, req, bool, whether to overlay the item-->true
  </itemOverlayEnabled>
  <customName>
    <!--ro, req, string, custom overlaying name, range:[0,32], desc:self-defined name of the overlay information-->test
  </customName>
  <changeLineNum>
    <!--ro, req, int, number of line feeds, and the default value is 0, range:[0,100], desc:the default value is 0-->0
  </changeLineNum>
  <spaceNum>
    <!--ro, req, int, number of spaces, and the default value is 0, range:[0,255], desc:the default value is 0-->0
  </spaceNum>
  <startPosEnable>
    <!--ro, opt, bool, whether to enable coordinate configuration, desc:whether to enable coordinate configuration-->true
  </startPosEnable>
  <startPosTop>
    <!--ro, req, int, top coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture height, and the default value is 0-->0
  </startPosTop>
  <startPosLeft>
    <!--ro, req, int, left coordinate of the start point, desc:it is valid for overlay within the picture, the value is between 0 and the actual picture width, and the default value is 0-->0
  </startPosLeft>
  <overlayInfoText>
    <!--ro, opt, string, it is valid when itemType is set to place, device No., direction description, lane information, and camera, range:[1,128], desc:the maximum string length of place is 128 bytes, the maximum string length of intersection No., device No., direction No., direction description and lane information is 32 bytes., the maximum string length of camera 1 is 44 bytes, and the maximum string length of verification certificate No., calibration expiration date is 128 bytes-->test
  </overlayInfoText>
  <overlayInfoText2>
    <!--ro, opt, string, overlay character string, range:[1,32], desc:for camera 2, which is valid when <itemType> is camera information; the maximum string length of camera 2 is 32 bytes-->test
  </overlayInfoText2>
</OverlayInfo>
```

12.5.1.73 Set vehicle detection parameters

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/vehicleDetect

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<VehicleDetectCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, vehicle detection parameters, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--req, bool, whether to enable the function-->true
  </enabled>
</VehicleDetectCfg>
```

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>
```

12.5.1.74 Get vehicle detection parameters

Request URL

GET /ISAPI/Traffic/channels/<channelID>/vehicleDetect

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<VehicleDetectCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, vehicle detection parameters, attr:version{req, string, protocolVersion}-->
  <enabled>
    <!--ro, req, bool, whether to enable the function-->true
  </enabled>
</VehicleDetectCfg>
```

12.5.1.75 Get the capability of vehicle detection parameters

Request URL

GET /ISAPI/Traffic/channels/<channelID>/vehicleDetect/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<VehicleDetectCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{req, string, protocolVersion}-->
  <enabled opt="true,false">
    <!--ro, req, bool, whether to enable the function, attr:opt{req, string}-->true
  </enabled>
</VehicleDetectCfg>
```

12.5.1.76 Get the recognition result of the license plate captured manually

Request URL

GET /ISAPI/Traffic/MNPR/channels/<channelID>?laneNo=<laneNo>&OSD=<OSDType>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
laneNo	string	lane No.
OSDType	string	whether to enable OSD overlay for manual capture: enabled by default; not returned and not enabled when is 1

Request Message

None

Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, vehicle information, 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:transport protocol type: HTTP,HTTPS,EHome-->HTTP
  </protocol>
  <macAddress>
    <!--ro, opt, string, MAC address-->01:17:24:45:D9:F4
  </macAddress>
  <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:"ANPR" (license plate recognition)-->ANPR
  </eventType>
  <eventState>
    <!--ro, 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:[1,64]-->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>
  <ANPR>
```

```

<!--ro, opt, object, information about license plate recognition alarm-->
<region>
  <!--ro, req, enum, region, subType:string, desc:region-->ER
</region>
<country>
  <!--ro, req, enum, country or region, subType:int, desc:country or region-->253
</country>
<area>
  <!--ro, req, enum, province or state, subType:string, desc:province or state-->FJR
</area>
<licensePlate>
  <!--ro, req, enum, license plate number, subType:string, desc:license plate number,e.g.,"123456"-->noPlate
</licensePlate>
<line>
  <!--ro, req, int, recognized Lane No.-->1
</line>
<direction>
  <!--ro, req, enum, target direction, subType:string, desc:"reverse", "forward", "unknown"-->reverse
</direction>
<confidenceLevel>
  <!--ro, req, int, confidence Level, range:[0,100]-->50
</confidenceLevel>
<plateType>
  <!--ro, req, enum, license plate type, subType:string, desc:license plate type: "unknown", "arm"-police vehicle, "embassy"-embassy vehicle, "motorola"-motorcycle, "coach"-driver-training vehicle, "tempTravl"-vehicle with temporary license plate, "trailer"-trailer, "consulate"-consular vehicle, "tempEntry"-temporary vehicle, "newEnergy"-new energy vehicle-->unknown
</plateType>
<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>
<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:coordinates of the license plate thumbnail in the matched picture-->
    <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:whether the driver is wearing safety belt: "unknown, yes, no"-->unknown
</pilotsafebelt>
<vicepilotsafebelt>
  <!--ro, opt, string, whether the co-driver is wearing a safety belt-->unknown
</vicepilotsafebelt>
<pilotsunvisor>
  <!--ro, opt, string, whether the driver room's sun visor is open-->unknown
</pilotsunvisor>
<vicepilotunvisor>
  <!--ro, opt, string, whether the co-driver room's sun visor is open-->unknown
</vicepilotunvisor>
<envprosign>
  <!--ro, req, enum, whether it is a yellow-label vehicle, subType:string, desc:"unknown" (unknown), "green" (green Label), "yellow" (yellow Label)-->unknown
</envprosign>
<dangmark>
  <!--ro, req, enum, whether it is a dangerous goods vehicle, subType:string, desc:"unknown", "yes", "no"-->unknown
</dangmark>
<uphone>
  <!--ro, req, enum, whether the driver is making a phone call, subType:string, desc:"unknown", "yes", "no"-->unknown
</uphone>
<pendant>
  <!--ro, req, enum, whether there are window hangings detected, subType:string, desc:"unknown", "yes", "no"-->unknown
</pendant>
<tissueBox>
  <!--ro, req, enum, whether there is a tissue box detected, subType:string, desc:"unknown", "yes", "no"-->unknown
</tissueBox>
<frontChild>
  <!--ro, req, enum, whether the co-driver is with a baby in arm, subType:string, desc:"unknown", "yes", "no"-->unknown
</frontChild>
<label>
  <!--ro, req, enum, whether there are stickers detected, subType:string, desc:"unknown", "yes", "no"-->unknown
</label>
<decoration>
  <!--ro, req, enum, whether there are decorations detected, subType:string, desc:"unknown", "yes", "no"-->unknown
</decoration>
<smoking>
  <!--ro, req, enum, whether anyone is smoking, subType:string, desc:whether there is smoking detected: "unknown, yes, no"-->yes
</smoking>
<perfumeBox>
  <!--ro, opt, enum, whether there is perfume box detected, subType:string, desc:whether there is perfume box detected: "unknown, yes, no"-->unknown
</perfumeBox>

```

```

</> <!-- vehicle type -->
<pdvs>
    <!---ro, req, enum, whether there is a person sticking out of sunroof, subType:string, desc:"unknown", "yes", "no"-->unknown
</pdvs>
<helmet>
    <!---ro, req, enum, whether the non-motor vehicle driver wears a helmet, subType:string, desc:"unknown", "yes", "no"-->no
</helmet>
<twoWheelVehicle>
    <!---ro, opt, enum, unknown, subType:string, desc:whether there is two-wheel detected: "unknown, yes, no"-->unknown
</twoWheelVehicle>
<threeWheelVehicle>
    <!---ro, opt, enum, whether there is three-wheel detected, subType:string, desc:whether there is three-wheel detected: "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:confidence of each character in the recognized license plate-->

```

>test

```

</plateCharBelieve>
<speedLimit>
    <!---ro, opt, int, upper speed limit, desc:it is valid only when overspeeding occurred-->50
</speedLimit>
<illegalInfo>
    <!---ro, opt, object, violation information-->
<illegalCode>
    <!---ro, req, string, violation code, range:[0,64]-->1101
</illegalCode>
<illegalName>
    <!---ro, req, string, violation name, range:[0,128]-->test
</illegalName>
<illegalDescription>
    <!---ro, opt, string, violation description, range:[0,256]-->test
</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:name of the picture selected as the checkpoint picture when illegal action occurs, "none" refers to not selecting any picture-->test
</postPicFileName>
<featurePicFileName>
    <!---ro, opt, string, red light running capture, desc:name of the picture selected as the close-up picture when running the red light in the intersection violation system is detected, "none" refers to not selecting any picture-->test
</featurePicFileName>
<detectDir>
    <!---ro, req, enum, detection direction, subType:int, desc:1 (upward), 2 (downward), 3 (bidirection), 4 (from east to west), 5 (from south to north), 6 (from west to east), 7 (from north to south), 8 (other)-->1
</detectDir>
<detectType>
    <!---ro, req, enum, detection type, subType:int, desc:1 (inductive Loop trigger), 2 (video trigger), 3 (multiple-frame recognition), 4 (radar trigger)-->

```

->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:1 (history data), 0 (real-time data)-->0
</alarmDataType>
<dIllegalTime>
    <!---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
</dIllegalTime>
<vehicleInfo>
    <!---ro, opt, object, vehicle information-->
<index>
    <!---ro, req, int, vehicle No.-->1
</index>
<vehicleType>
    <!---ro, req, 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:"green", "brown", "pink", "purple", "deepgray" (dark gray), "cyan", "orange", "white", "silver" (silvery), "gray", "black", "red", "deepblue" (dark blue), "blue", "yellow"-->green
</color>
<speed>
    <!---ro, req, int, target 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 one vehicle (buses with seven or less seats,trucks with capacity of 2 tons or less), 2-class two vehicle (buses with 8 to 19 seats,trucks with capacity of 2 to 5 (included) tons), 3-class three vehicle (buses with 20 to 39 seats,trucks with capacity of 5 to 10 (included) tons), 4-class four vehicle (buses with 40 or more seats,trucks with capacity of 10 to 15 (included) tons), 5-class five vehicle (trucks with capacity of more than 15 tons)-->4
</vehicleTypeByWeight>
</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, req, enum, target direction, subType:string, desc:"enter", "Leave"-->enter
    </direction>
    <cardNo>
        <!--ro, opt, string, card No.-->test
    </cardNo>
    <parkType>
        <!--ro, req, enum, parking type, subType:string, desc:"temporary", "permanent"-->temporary
    </parkType>
</EntranceInfo>
<pictureInfoList>
    <!--ro, req, array, picture list, subType:object-->
    <pictureInfo>
        <!--ro, req, object, picture information-->
        <fileName>
            <!--ro, req, enum, picture name, subType:string, desc:it must correspond to the picture name transmitted with the alarm message.
            "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).-->detectionPicture.jpg
        </fileName>
        <type>
            <!--ro, req, enum, parking 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>
        <picRecognMode>
            <!--ro, opt, enum, license plate recognition mode, subType:int, desc:0 (front license plate recognition), 1 (rear license plate recognition)-->0
        </picRecognMode>
        <redLightTime>
            <!--ro, opt, int, red light time elapsed, unit:s-->0
        </redLightTime>
        <vehicleHead>
            <!--ro, req, 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 number is time in millisecond-->
            >20090810235959999
        </absTime>
        <plateRect>
            <!--ro, opt, object, license plate area coordinates, desc:this node is valid only when type is "detectionPicture"-->
            <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, vehicle area coordinates, desc:the normalized value is the current image size in percentage multiplying 1000. This node is
            valid only when type is "detectionPicture"-->
            <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>

```

```

</vehicleImage>
<pictureURL>
    <!--ro, opt, string, picture URL, desc:it is valid only when dataType is "URL"-->test
</pictureURL>
<pId>
    <!--ro, opt, string, strlen.max=32, range:[1,32], desc:the recommended generation rule is the device's serial number + the time elapsed since the
device was started + a random number-->test
</pId>
</pictureInfo>
</pictureInfoList>
<hasMoreData>
    <!--ro, opt, bool, whether there is more data, desc:this node is used to report the license plate information first, and then report XML message with
picture data; the XM message with picture data and license plate information are linked by UUID-->true
</hasMoreData>
<listType>
    <!--ro, opt, enum, name List attribute, subType:string, desc:"white" (allowList), "black" (blocklist), "temporary" (temporary List)-->white
</listType>
<originalLicensePlate>
    <!--ro, opt, string, original license plate number, desc:when the license plate number is a minor language, return the original license plate number-->test
</originalLicensePlate>
<CRIndex>
    <!--ro, req, enum, country/region, subType:int, desc:country or region index-->258
</CRIndex>
<VehicleGPSInfo>
    <!--ro, opt, object, GPS information of the vehicle-->
<longitudeType>
    <!--ro, req, enum, W, subType:string, desc:longitude, "E,W"-->E
</longitudeType>
<latitudeType>
    <!--ro, req, enum, N, subType:string, desc:latitude, "S,N"-->S
</latitudeType>
<Longitude>
    <!--ro, req, object, longitude-->
<degree>
    <!--ro, req, int, degree-->60
</degree>
<minute>
    <!--ro, req, int, minute, range:[0,59]-->59
</minute>
<sec>
    <!--ro, req, float, second, range:[0,59.999999]-->59.000000
</sec>
</Longitude>
<Latitude>
    <!--ro, req, object, latitude-->
<degree>
    <!--ro, req, int, degree-->60
</degree>
<minute>
    <!--ro, req, int, minute, range:[0,59]-->59
</minute>
<sec>
    <!--ro, req, float, second, range:[0,59.999999], desc:the value is accurate to six decimal places-->59.000000
</sec>
</Latitude>
</VehicleGPSInfo>
<vehiclePositionControl>
    <!--ro, opt, enum, vehicle arming, subType:string, desc:arming type: "vehicleMonitor"-intelligent arming of vehicle (PUT
/ISAPI/Traffic/channels/<ID>/vehicleMonitor/<taskID>/startTask),"manualVehicleMonitor"-manual arming of vehicle (PUT
/ISAPI/Traffic/channels/<ID>/manualVehicleMonitor),"dailyVehicleMonitor"-daily arming of vehicle (you can check whether this arming type is supported via
the node isSupportDailyVehicleMonitor in the capability message returned by /ISAPI/Traffic/channels/<ID>/vehicleDetect/capabilities; when daily arming of
vehicle is enabled,both alarm of ANPR and intelligent arming of vehicle will be uploaded; if this node is not returned,it is normal vehicle detection-->test
</vehiclePositionControl>
<vehicleMonitorTaskID>
    <!--ro, opt, string, task ID of intelligent arming of vehicle, range:[1,64], desc:this node is returned when the value of vehiclePositionControl is
"vehicleMonitor"-->test
</vehicleMonitorTaskID>
<vehicleListName>
    <!--ro, opt, string, vehicle list name, 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>
</ANPR>
<UUID>
    <!--ro, opt, string, UUID, 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 number-->test
</monitoringSiteID>
<ePlateUUID>
    <!--ro, opt, string, electronic license plate ID, desc:If this node is configured with a value, it indicates that an electronic license plate is linked-->test

```

```

</ePlateUUID>
<isDataRetransmission>
    <!--ro, opt, bool, data retransmission mark-->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:[0,32]-->test
</sceneName>
<PTZPos>
    <!--ro, opt, object, PTZ position-->
<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]-->test
</monitorDescription>
<DeviceGPSInfo>
    <!--ro, opt, object, device GPS information-->
<longitudeType>
    <!--ro, req, enum, W, subType:string, desc:longitude, "E,W"-->E
</longitudeType>
<latitudeType>
    <!--ro, req, enum, N, subType:string, desc:latitude, "S,N"-->S
</latitudeType>
<Longitude>
    <!--ro, req, object, Longitude-->
<degree>
    <!--ro, req, int, degree-->60
</degree>
<minute>
    <!--ro, req, int, minute, range:[0,59]-->59
</minute>
<sec>
    <!--ro, req, float, second, range:[0,59.99999]-->59.000000
</sec>
</Longitude>
<Latitude>
    <!--ro, req, object, Latitude-->
<degree>
    <!--ro, req, int, degree-->60
</degree>
<minute>
    <!--ro, req, int, minute, range:[0,59]-->59
</minute>
<sec>
    <!--ro, req, float, second, range:[0,59.99999], desc:the value is accurate to six decimal places-->59.000000
</sec>
</Latitude>
</DeviceGPSInfo>
</EventNotificationAlert>

```

Parameter Name	Parameter Value	Parameter Type(Content-Type)	Content-ID	File Name	Description
EventNotificationAlert	[Message content]	application/xml	--	--	--

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.

12.5.1.77 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, 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), "ElandCIS" (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-Collectivit 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-Cura?ao, 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, 90-Iraq, 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-Guiana, 213-Suriname, 214-Guiane Francaise, 215-Ecuador, 216-Peru, 217-Bolivia, 218-Parauau, 219-Chile, 220-
```

Brazil, 221-Uruguay, 222-Venezuela, 223-Paraguay, 224-Bolivia, 225-Chile, 226-Colombia, 227-Ecuador, 228-France, 229-Greece, 229-Iraq, 229-Italy, 230-Japan, 231-Korea, 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", "UWW" (Umm al-Qaiwain), "other", "AM" (Ajman), "RAK" (Ras al-Khaimah), "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>

<vicePilotSunvisor>

<!--ro, opt, enum, whether the co-driver room's sun visor is open, subType:string, desc:"unknown", "yes", "no"-->unknown

</vicePilotSunvisor>

<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>
<towheelVehicle>
  <!--ro, opt, enum, whether there is a two-wheel vehicle detected, subType:string, desc:"unknown", "yes", "no"-->unknown
</towheelVehicle>
<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>
  <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,SUMPV,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>
<dwlIllegalTime>
  <!--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
</dwlIllegalTime>
<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,SUMPV,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>

```

```

<!--ro, opt, int, vehicle speed, unit:mile-->
</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 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>
<slagTruckCoverPlate>
  <!--ro, opt, enum, whether the back cover of the dump truck is covered, subType:string, desc:"unknown", "yes", "no"-->unknown
</slagTruckCoverPlate>
<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/HOVLaneDetection?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-->
        </plateRect>
    </pictureInfo>

```

```

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>
<vehicelRect>
  <!--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>
</vehicelRect>
<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-->0
</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>
</extraPlateRect>

```

```

        </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>
<vehicelRectName>
    <!--ro, opt, string, country/region where the vehicle triggering events belongs, range:[0,64]-->11
</vehicelRectName>
<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
</nodeType>
<Custom>
  <!--ro, opt, object, custom parameters, dep:and,{$.EventNotificationAlert.ANPR.nodeType,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, 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 <dwMatchNo> 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:t-->4.502
            </overWeight>
            <weight>
                <!--ro, opt, float, vehicle weight (unit: ton), range:[0.000,100.000], unit:t-->4.502
            </weight>
            <limitWeight>
                <!--ro, opt, float, weight limit (unit: ton), range:[0.000,100.000], unit:t-->4.502
            </limitWeight>
            <axleLen>
                <!--ro, opt, float, distance between tyres, range:[0.00,100.00], unit:m-->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, Hongkong, and Maokao are all 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"
(small-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]-->E855000011F9AA5
    </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--)-->E855000011F9AA5
    </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, 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 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>
    <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>
<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]-->0
</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.

12.6 Traffic Control

12.6.1 Motor Vehicle Traffic Enforcement

12.6.1.1 Get the capability of configuring column speaker parameters

Request URL

GET /ISAPI/System/columnSpeakers/capabilities?format=json

Query Parameter

None

Request Message

None

Response Message

```

{
    "ColumnSpeakersCap": {
        /*ro, req, object, capability of configuring column speaker parameters*/
        "@size": 2,
        /*ro, req, int, supported number of column speakers*/
        "enabled": {
            /*ro, req, object, whether to enable the function*/
            "@opt": [true, false]
            /*ro, req, array, subType:bool*/
        },
        "connectionMode": {
            /*ro, opt, object*/
            "@opt": ["network", "rs485"]
            /*ro, req, array, subType:string*/
        },
        "RS485Port": {
            /*ro, opt, object, range of the RS-485 port No.*/
            "@min": 1,
            /*ro, req, int, the minimum value*/
            "@max": 4,
            /*ro, req, int, the maximum value*/
            "@def": 1
            /*ro, req, int, the default value*/
        },
        "ipv4": {
            /*ro, opt, object*/
            "@min": 1,
            /*ro, req, int*/
            "@max": 17
            /*ro, req, int*/
        },
        "port": {
            /*ro, opt, object*/
            "@min": 0,
            /*ro, req, int*/
            "@max": 65535,
            /*ro, req, int*/
            "@def": 34508
            /*ro, req, int*/
        },
        "MessageList": {
            /*ro, req, object, information list*/
            "@size": 2,
            /*ro, req, int, supported number of column speakers*/
            "message": {
                /*ro, req, object, text information, desc:object, text information, the maximum string size is 60 bytes*/
                "@min": 0,
                /*ro, req, int, the minimum value*/
                "@max": 128
                /*ro, req, int, the maximum value*/
            },
            "volume": {
                /*ro, req, object, volume range*/
                "@min": 0,
                /*ro, req, int, the minimum value*/
                "@max": 9
                /*ro, req, int, the maximum value*/
            },
            "speaker": {
                /*ro, req, object, voice type*/
                "@opt": [1, 2, 3, 4, 5]
                /*ro, req, array, speaker, subType:int, desc:1 (male voice 1), 2 (male voice 2), 3 (female voice 1), 4 (female voice 2), 5 (child voice)*/
            },
            "DataTypes": [
                /*ro, req, array, data type, subType:object*/
                {
                    "code": 1,
                    /*ro, req, int, violation code*/
                    "text": "test"
                    /*ro, req, string, violation code description*/
                }
            ],
            "channelID": {
                /*ro, opt, object, list of linked channels*/
                "@opt": [1, 2, 3]
                /*ro, opt, array, options, subType:int, range:[1,64]*/
            }
        },
        "speakerID": {
            /*ro, opt, object*/
            "@opt": [1, 2]
            /*ro, opt, array, options, subType:int, range:[1,2]*/
        }
    }
}

```

12.6.1.2 Get the column speaker parameters

Request URL

GET /ISAPI/System/columnSpeakers?format=json

Query Parameter

None

Request Message

None

Response Message

```
{
    "ColumnSpeakers": [
        /*ro, req, array, column speaker parameters, subType:object*/
        {
            "enabled": true,
            /*ro, req, bool, whether to enable*/
            "connectionMode": "network",
            /*ro, opt, enum, subType:string*/
            "485Port": 1,
            /*ro, opt, int, RS-485 port No., range:[1,4], desc:RS-485 port No.*/
            "ipv4": "10.0.0.1",
            /*ro, opt, string, range:[1,17]*/
            "port": 34508,
            /*ro, opt, int, range:[1,65535]*/
            "authenticationEnabled": false,
            /*ro, opt, bool*/
            "userName": "test",
            /*ro, opt, string, range:[0,32], dep:and,{$.ColumnSpeakers[*].authenticationEnabled,eq,true}*/
            "485VoiceSpeed": 1,
            /*ro, opt, int, range:[1,9]*/
            "playFileList": ["1.mp3", "2.mp3", "3.wav"],
            /*ro, opt, array, subType:string, range:[0,16]*/
            "MessageList": [
                /*ro, opt, array, information list, subType:object, desc:information list*/
                {
                    "id": 1,
                    /*ro, req, int, ID, desc:it starts from 1*/
                    "message": "test",
                    /*ro, req, string, text information, range:[0,128], desc:text information*/
                    "volume": 0,
                    /*ro, req, int, volume, range:[0,9]*/
                    "speaker": 1,
                    /*ro, req, enum, voice type, subType:int, desc:1 (male voice 1), 2 (male voice 2), 3 (female voice 1), 4 (female voice 2), 5 (child voice)*/
                    "voiceSpeed": 1,
                    /*ro, opt, int, range:[0,9]*/
                    "repeatNum": 1,
                    /*ro, opt, int, range:[1,100]*/
                    "playType": 0,
                    /*ro, opt, enum, subType:int*/
                    "codes": [1, 2],
                    /*ro, req, array, violation code, subType:int, range:[0,2]*/
                    "channelID": [1, 2, 3]
                    /*ro, opt, array, subType:int, range:[1,64]*/
                }
            ],
            "speakerID": 1,
            /*ro, opt, int, column speaker ID, range:[1,6], desc:column speaker ID*/
            "schedule": {
                /*ro, opt, object*/
                "enabled": true,
                /*ro, opt, bool*/
                "timeRange": [
                    /*ro, opt, array, subType:object, range:[0,2]*/
                    {
                        "startTime": "00:00",
                        /*ro, opt, string, range:[0,64]*/
                        "endTime": "24:00"
                        /*ro, opt, string, range:[0,64]*/
                    }
                ]
            },
            "broadcastTimes": 5
            /*ro, opt, int, range:[0,15]*/
        }
    ]
}
```

Parameter Name	Parameter Value	Parameter Type(Content-Type)	Content-ID	File Name	Description
	[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.

12.6.1.3 Set the column speaker parameters

Request URL

PUT /ISAPI/System/columnSpeakers?format=json

Query Parameter

None

Request Message

```
{
    "ColumnSpeakers": [
        /*req, array, column speaker parameters, subType:object*/
        {
            "enabled": true,
            /*req, bool, whether to enable the function*/
            "connectionMode": "network",
            /*opt, enum, subType:string*/
            "485Port": 1,
            /*opt, int, RS-485 port No., value range: [1,4], range:[1,4]*/
            "ipv4": "10.0.0.1",
            /*opt, string, range:[1,17]*/
            "port": 34508,
            /*opt, int, range:[1,65535]*/
            "MessageList": [
                /*req, array, information list, subType:object*/
                {
                    "id": 1,
                    /*req, int, No., desc:it starts from 1*/
                    "message": "test",
                    /*req, string, text information, the maximum string size is 60 bytes, range:[0,128], desc:text information, the maximum string size is
60 bytes*/
                    "volume": 0,
                    /*req, int, volume, range:[0,9]*/
                    "speaker": 1,
                    /*req, enum, speaker, subType:int, desc:speaker*/
                    "codes": [1, 2],
                    /*req, array, violation code, subType:int, range:[1,2]*/
                    "channelID": [1, 2, 3]
                    /*opt, array, subType:int, range:[1,64]*/
                }
            ],
            "speakerID": 1
            /*opt, int, column speaker ID, which can be 1 or 2, range:[1,2]*/
        }
    ]
}
```

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*/
}
```

12.6.1.4 Get the traffic service capability

Request URL

GET /ISAPI/Traffic/capabilities

Query Parameter

None

Request Message

None

Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<TrafficCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version(opt, string, protocolVersion)-->
  <plateCap>
    <!--ro, opt, object, capability of license plate recognition-->
    <plateListNum>
      <!--ro, opt, int, the capacity of license plate blocklists, desc:maximum number of imported blocklists and allowlists supported by the device-->64
    </plateListNum>
    <plateMaskLen>
      <!--ro, opt, int, the length of license plate characters, desc:maximum length of license plate characters supported by the device-->64
    </plateMaskLen>
    <isNotSupportLicenseImport>
      <!--ro, opt, bool-->true
    </isNotSupportLicenseImport>
    <isSupportLicenseModify>
      <!--ro, opt, bool-->true
    </isSupportLicenseModify>
    <isSupportLPAuditDataDelete>
      <!--ro, opt, bool-->true
    </isSupportLPAuditDataDelete>
  </plateCap>
</TrafficCap>
```

12.6.2 Motor Vehicle Traffic Flow Statistics

12.6.2.1 Get the configuration capability of trigger mode for traffic data statistics

Request URL

GET /ISAPI/ITC/TriggerMode/TPS/capabilities?channelID=<channelID>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<TPS xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, traffic statistics parameters, attr:version(req, string, protocolVersion)-->
  <relatedLaneCount min="1" max="10">
    <!--ro, req, int, number of supported lanes, range:[1,10], attr:min{req, int},max{req, int}-->1
  </relatedLaneCount>
  <relatedDriveWay min="1" max="99">
    <!--ro, req, int, linked lane No., range:[1,99], attr:min{req, int},max{req, int}-->8
  </relatedDriveWay>
  <laneDirectionType opt="0,1,2,3,4,5,6,7,8">
    <!--ro, req, enum, types of lane directions, subType:int, attr:opt{req, string}, desc:@ (unknown),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)-->0
  </laneDirectionType>
  <enRealtimeDataUpload>
    <!--ro, req, bool, whether it supports uploading real-time data-->true
  </enRealtimeDataUpload>
  <enStatisticalDataUpload>
    <!--ro, req, bool, whether it supports uploading statistics-->true
  </enStatisticalDataUpload>
  <statisticsTime min="0" max="999">
    <!--ro, req, int, statistics interval, range:[0,999], unit:min, attr:min{req, int},max{req, int}-->0
  </statisticsTime>
  <posEnable opt="true,false">
    <!--ro, req, bool, whether it supports enabling position-->false
  </posEnable>
</TPS>
```

```

<!--ro, req, bool, attr:opt{req, string}-->true
</posEnable>
<CalibrationEnable opt="true,false">
  <!--ro, req, bool, attr:opt{req, string}-->true
</CalibrationEnable>
<laneVolumeEnable>
  <!--ro, req, bool-->true
</laneVolumeEnable>
<laneAverageSpeedEnable>
  <!--ro, req, bool-->true
</laneAverageSpeedEnable>
<timeHeadwayEnable>
  <!--ro, req, bool-->true
</timeHeadwayEnable>
<spaceHeadwayEnable>
  <!--ro, req, bool-->true
</spaceHeadwayEnable>
<timeOccupancyRationEnable>
  <!--ro, req, bool-->true
</timeOccupancyRationEnable>
<spaceOccupancyRationEnable>
  <!--ro, req, bool-->true
</spaceOccupancyRationEnable>
<queueLengthEnable>
  <!--ro, req, bool-->true
</queueLengthEnable>
<vehicleTypeEnable>
  <!--ro, req, bool-->true
</vehicleTypeEnable>
<trafficStatusEnable>
  <!--ro, req, bool-->true
</trafficStatusEnable>
<directionLineEnable>
  <!--ro, opt, object-->
  <lineName opt="DirectionLine">
    <!--ro, req, enum, Line name, subType:string, attr:opt{req, string}, desc:"DirectionLine"-->DirectionLine
  </lineName>
</RegionCoordinatesList>
  <!--ro, opt, array, area coordinate List, subType:object, range:[0,2]-->
<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]-->0
</positionX>
<positionY>
  <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
<laneDirection opt="upToDown,downToUp">
  <!--ro, opt, enum, subType:string, attr:opt{req, string}-->upToDown
</laneDirection>
<DirectionLineEnable>
<virtualRegionMode min="1" max="3">
  <!--ro, req, enum, protocol type, subType:int, attr:min{req, int},max{req, int}, desc:protocol type-->1
</virtualRegionMode>
<VirtualRegionList min="0" max="3">
  <!--ro, req, array, virtual coil list, subType:object, attr:min{req, int},max{req, int}-->
<VirtualRegion>
  <!--ro, opt, object, virtual coil-->
</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 upper-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>
</VirtualRegion>
</VirtualRegionList>
<LaneLine>
  <!--ro, opt, object, Lane Line-->
  <lineName min="0" max="64">
    <!--ro, req, string, Line name, range:[1,64], attr:min{req, int},max{req, int}-->test
  </lineName>
<RegionCoordinatesList num="2">
  <!--ro, opt, array, area coordinate List, subType:object, range:[0,2], attr:num{opt, object}-->
<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]-->0
</positionX>
<positionY>
  <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</LaneLine>
<CheckoutLineEnable opt="true,false">
  <!--ro, opt, object, Lane Line-->

```

```

<!--ro, opt, bool, attr:opt{req, string}-->true
</CheckoutLineEnable>
<LaneRightBoundaryLine>
<!--ro, opt, object-->
<lineName min="0" max="64">
<!--ro, req, string, Line name, range:[0,64], attr:min{req, int},max{req, int}-->test
</lineName>
<RegionCoordinatesList num="2">
<!--ro, opt, array, area coordinate list, subType:object, range:[0,2], attr:num{opt, object}-->
<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]-->0
</positionX>
<positionY>
<!--ro, req, int, Y-coordinate, range:[0,1000]-->0
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</LaneRightBoundaryLine>
<isSupportFlowReset>
<!--ro, opt, bool, whether it supports clearing traffic flow by one touch-->true
</isSupportFlowReset>
<statisticsTime>
<!--ro, opt, int, statistics time interval T2, range:[0,999], unit:min-->0
</statisticsTime2>
<enStatistical2DataUpload>
<!--ro, opt, bool, whether it supports uploading traffic statistics every T2 time interval, desc:whether it supports uploading traffic statistics every T2 time interval-->true
</enStatistical2DataUpload>
</TPS>

```

Parameter Name	Parameter Value	Parameter Type(Content-Type)	Content-ID	File Name	Description
TPS	[Message content]	application/xml	--	--	--

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.

12.6.2.2 Traffic statistics (passing vehicle statistics)

EventType:TPS

```

{
  "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, transmission communication 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*/
  "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, 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*/
  "dateTime": "2004-05-03T17:30:08+08:00",
  /*ro, req, datetime, alarm trigger time*/
  "activePostCount": 1,
  /*ro, opt, int, times that the same alarm has been uploaded, desc:times that the same alarm has been uploaded*/
  "eventType": "TPS",
  /*ro, req, string, event type, desc:"TPS" (passing vehicle statistics uploading)*/
  "eventState": "active"
}

```

```

eventState : active ,
/*ro, req, enum, event status, subType:string, desc:for durative event: active (valid event or event starts), inactive (invalid event or the event ends). For the heartbeat, the field value indicates the time headway,unit: secondheartbeat data, and it is uploaded every 10 seconds*/
"eventDescription": "traffic pass statistics",
/*ro, req, string, event description*/
"deviceID": "test0123",
/*ro, opt, string, device No., desc:device No.*/
"Target": [
/*ro, opt, array, subType:object*/
{
    "recognitionType": "vehicle",
    /*ro, req, enum, recognition type in traffic statistics, subType:string, desc:"vehicle"*/
    "TargetInfo": {
        /*ro, req, object, target information*/
        "recognition": "TPS",
        /*ro, req, string, specific recognition target*/
        "startTime": "2004-05-03T17:30:08.000+080:00",
        /*ro, req, date, statistics start time*/
        "stopTime": "2004-05-03T17:30:08.000+080:00",
        /*ro, req, date, statistics end time*/
        "samplePeriod": 3600,
        /*ro, req, int, statistics period, unit:s*/
        "totalLaneNum": 8,
        /*ro, req, int, number of valid lanes, range:[1,8]*/
        "LaneInfo": [
            /*ro, req, array, lane information, subType:object*/
            {
                "laneNo": 8,
                /*ro, req, int, lane No., range:[1,8]*/
                "averageSpeed": 60,
                /*ro, req, int, average speed of vehicles traveling on the lane, unit:km/h*/
                "smallCarNum": 20,
                /*ro, req, int, number of small-sized vehicles*/
                "midsizeCarNum": 20,
                /*ro, req, int, number of medium-sized vehicles*/
                "heavyVehicleNum": 20,
                /*ro, req, int, number of heavy vehicles*/
                "headTimeInterval": 20,
                /*ro, req, int, time headway, unit:s*/
                "headInterval": 20,
                /*ro, req, int, space headway, unit:m*/
                "spaceOccupyRation": 20.000,
                /*ro, req, float, lane space occupancy, range:[0.000,100.000]*/
                "timeOccupyRation": 20.000,
                /*ro, req, float, lane time occupancy, range:[0.000,100.000]*/
                "channelizationLaneNo": 2,
                /*ro, opt, int, channelized lane No.*/
                "averageParkingTime": 2,
                /*ro, opt, int, average parking times*/
                "averageDelay": 2,
                /*ro, opt, int, average delay time, unit:s*/
                "averageQueueLen": 2,
                /*ro, opt, int, average queuing length, unit:m*/
                "arrivalFlow": 20,
                /*ro, opt, int*/
                "vehicleNum": 20,
                /*ro, opt, int, number of vehicles in area*/
                "nonmotorVehicleNum": 20,
                /*ro, opt, int*/
                "oversizeVehicleNum": 20,
                /*ro, opt, int, large-sized vehicle quantity, range:[1,10000]*/
                "laneName": "车道1",
                /*ro, opt, string, lane name, range:[1,32]*/
                "laneType": 1,
                /*ro, opt, enum, lane type, subType:int, desc:lane type*/
                "laneState": 1,
                /*ro, opt, enum, lane status, subType:int, desc:lane status*/
                "varyType": 0,
                /*ro, opt, enum, subType:int*/
                "tpsType": 0,
                /*ro, opt, enum, subType:int*/
                "vehicleNum": 20,
                /*ro, opt, int, range:[0,10000]*/
                "QueueInfo": {
                    /*ro, opt, object*/
                    "HeadPosition": {
                        /*ro, opt, object*/
                        "x": 0.120,
                        /*ro, req, float, X-coordinate, range:[0.000,1.000]*/
                        "y": 0.120
                        /*ro, req, float, Y-coordinate, range:[0.000,1.000]*/
                    },
                    "TailPosition": {
                        /*ro, opt, object*/
                        "x": 0.120,
                        /*ro, req, float, X-coordinate, range:[0.000,1.000]*/
                        "y": 0.120
                        /*ro, req, float, Y-coordinate, range:[0.000,1.000]*/
                    },
                    "queueLen": 2
                    /*ro, opt, int, range:[0,500], unit:m*/
                },
                "CoilPosition": {
                    /*ro, opt, object*/
                }
            }
        ]
    }
}

```

```

        "x": 0.120,
        /*ro, req, float, X-coordinate, range:[0.000,1.000]*/
        "y": 0.120
        /*ro, req, float, Y-coordinate, range:[0.000,1.000]*/
    }
},
],
"totalCoilNum": 40,
/*ro, opt, int, number of valid coils, range:[1,40]*/
"CoilInfo": [
/*ro, opt, array, coil information, subType:object*/
{
    "coilNo": 18,
    /*ro, req, int, coil No.*/
    "laneNo": 18,
    /*ro, req, int, lane No., range:[1,18]*/
    "averageSpeed": 60,
    /*ro, req, int, average speed of vehicles traveling on the lane, unit:km/h*/
    "smallCarNum": 20,
    /*ro, req, int, number of small-sized vehicles*/
    "midsizedCarNum": 20,
    /*ro, req, int, number of medium-sized vehicles*/
    "heavyVehicleNum": 20,
    /*ro, req, int, number of heavy vehicles*/
    "headTimeInterval": 20,
    /*ro, req, int, time headway, unit:s*/
    "headInterval": 20,
    /*ro, req, int, space headway, unit:m*/
    "channelizationLaneNo": 2,
    /*ro, opt, int, channelized lane No.*/
    "nonmotorVehicleNum": 20
    /*ro, opt, int, number of non-motor vehicles*/
}
]
}
},
"Location": {
/*ro, opt, object, device location*/
    "lng": 116.46,
    /*ro, req, float, longitude*/
    "lat": 39.92
    /*ro, opt, float, latitude*/
},
"algorithmInfo": {
/*ro, opt, object, algorithm version information, this field is required for secondary analysis*/
    "algoId": "test",
    /*ro, opt, string, algorithm package ID*/
    "algoVendorId": "test",
    /*ro, opt, string, manufacturer ID*/
    "algoVersionId": "test",
    /*ro, opt, string, algorithm version No.*/
    "algoChipType": "test",
    /*ro, opt, string, chip type*/
    "algoModelVersion": "test"
    /*ro, opt, string, model version*/
},
"detectDir": 1,
/*ro, opt, enum, subType:int*/
"detectType": 1,
/*ro, opt, enum, detection mode, subType:int, desc:detection mode*/
"monitoringSiteID": "test",
/*ro, opt, string, camera No., range:[0,48]*/
"startCode": "0xfe",
/*ro, opt, string, range:[1,8]*/
"CMD": "01",
/*ro, opt, enum, Command, subType:string*/
"deviceUUID": "12345"
/*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*/
}

```

12.6.3 Motor Vehicle Traffic Incident

12.6.3.1 Get the parameters of fallen object detection.

Request URL

GET /ISAPI/Traffic/channels/<channelID>/AID/<SID>/abandonedObject

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Channel No.
SID	string	Scene No., by calling GET /ISAPI/Traffic/channels//sceneinfo/capabilities protocol, the min. and max. attributes of SceneInfo->scenesID in the response message indicate the value range of scene No. If the device does not support this protocol, the default value is 1.

Request Message

None

Response Message

```

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

<AbandonedObjectEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, fallen object detection, attr:version(req, string, protocolVersion)-->
  <scenesID>
    <!--ro, req, string, scene ID which is between 1 and 16-->1
  </scenesID>
  <enabled>
    <!--ro, req, bool, whether to enable-->true
  </enabled>
  <amplifyDetectionEnabled>
    <!--ro, opt, bool-->true
  </amplifyDetectionEnabled>
  <durationTime>
    <!--ro, req, int, duration of fallen object detection, range:[1,60], unit:s-->1
  </durationTime>
  <SnapImageParam>
    <!--ro, opt, object, picture capturing parameters-->
    <imgNum>
      <!--ro, req, int, each rule corresponds to a group of picture capturing parameters-->1
    </imgNum>
    <snapSensitivity>
      <!--ro, req, int, capture sensitivity, range:[1,100]-->1
    </snapSensitivity>
  </SnapImageParam>
  <ImgGroupInfoList>
    <!--ro, req, array, List of the picture group information, subType:object-->
    <ImgGroupInfo>
      <!--ro, opt, object, picture group information-->
      <imgInterval>
        <!--ro, opt, int, capture interval, unit:s-->1
      </imgInterval>
      <imgType>
        <!--ro, req, enum, picture type, subType:string, desc:far (distant view)-->far
      </imgType>
      <times>
        <!--ro, opt, int, capture times-->1
      </times>
    </ImgGroupInfo>
  </ImgGroupInfoList>
  </SnapImageParam>
  <RelatedLaneParam>
    <!--ro, opt, object, Linked Lane parameters, desc:Only one between RelatedLaneParam and RelatedAreaParam can be configured at the same time. That is, you cannot configure the Linked area if the Linked Lane has been configured, and vice versa.-->
    <RelatedLaneNoList>
      <!--ro, req, array, List of linked lane numbers, subType:object-->
      <relatedLaneNo>
        <!--ro, opt, int, linked lane number-->1
      </relatedLaneNo>
    <RelatedParkingNoList>
      <!--ro, opt, array, List of linked parking space numbers, subType:object-->
      <relatedParkingNo>
        <!--ro, opt, int, linked parking space number-->1
      </relatedParkingNo>
    </RelatedParkingNoList>
  </RelatedLaneParam>
  <NormalizedScreenSize>
    <!--ro, opt, object, normalized coordinates-->
    <normalizedScreenWidth>
      <!--ro, req, int, normalized width-->1000
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--ro, req, int, normalized height-->1000
    </normalizedScreenHeight>
  </NormalizedScreenSize>
  <RelatedAreaParam>
    <!--ro, opt, object, parameters of the linked target areas, desc:Only one between RelatedLaneParam and RelatedAreaParam can be configured at the same time. That is, you cannot configure the Linked area if the Linked Lane has been configured, and vice versa.-->
    <relatedAreaNo>
      <!--ro, opt, string, linked area numbers, desc:Multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
    </relatedAreaNo>
  </RelatedAreaParam>
  <relatedShieldRegionNo>
    <!--ro, opt, string, numbers of linked shielded areas, desc:Multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
  </relatedShieldRegionNo>
  <targetSizeRect>
    <!--ro, opt, object-->
    <displayEnabled>
      <!--ro, req, bool-->true
    </displayEnabled>
  </targetSizeRect>
  <relatedType>
    <!--ro, opt, enum, subType:string-->area
  </relatedType>
</AbandonedObjectEvent>

```

12.6.3.2 Configure the parameters of fallen object detection.

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/AID/<SID>/abandonedObject

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

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

<AbandonedObjectEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, fallen object detection, attr:version{req, string, protocolVersion}-->
  <scenesID>
    <!--req, string, scene ID which is between 1 and 16-->1
  </scenesID>
  <enabled>
    <!--req, bool, whether to enable-->true
  </enabled>
  <durationTime>
    <!--req, int, duration of fallen object detection, range:[1,60], unit:s-->1
  </durationTime>
  <SnapImageParam>
    <!--opt, object, picture capturing parameters-->
    <imgNum>
      <!--req, int, each rule corresponds to a group of picture capturing parameters-->1
    </imgNum>
    <snapSensitivity>
      <!--req, int, capture sensitivity, range:[1,100]-->1
    </snapSensitivity>
    <ImgGroupInfoList>
      <!--req, array, List of the picture group information, subType:object-->
      <ImgGroupInfo>
        <!--opt, object, picture group information-->
        <imgInterval>
          <!--req, int, capture interval-->1
        </imgInterval>
        <imgType>
          <!--req, enum, picture type, subType:string, desc:far (distant view)-->far
        </imgType>
        <times>
          <!--opt, int, capture times-->1
        </times>
      </ImgGroupInfo>
    </ImgGroupInfoList>
  </SnapImageParam>
  <RelatedLaneParam>
    <!--opt, object, Linked Lane parameters, desc:Only one between RelatedLaneParam and RelatedAreaParam can be configured at the same time. That is, you cannot configure the Linked area if the Linked Lane has been configured, and vice versa.-->
    <RelatedLaneNoList>
      <!--req, array, List of Linked Lane numbers, subType:object-->
      <relatedLaneNo>
        <!--opt, int, Linked Lane number-->1
      </relatedLaneNo>
    </RelatedLaneNoList>
    <RelatedParkingNoList>
      <!--req, array, List of Linked parking space numbers, subType:object-->
      <relatedParkingNo>
        <!--opt, int, Linked parking space number-->1
      </relatedParkingNo>
    </RelatedParkingNoList>
  </RelatedLaneParam>
  <NormalizedScreenSize>
    <!--opt, object, normalized coordinates-->
    <normalizedScreenWidth>
      <!--req, int, normalized width-->1000
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--req, int, normalized height-->1000
    </normalizedScreenHeight>
  </NormalizedScreenSize>
  <RelatedAreaParam>
    <!--opt, object, parameters of the Linked target areas, desc:Only one between RelatedLaneParam and RelatedAreaParam can be configured at the same time. That is, you cannot configure the linked area if the linked lane has been configured, and vice versa.-->
    <relatedAreaNo>
      <!--opt, string, Linked area numbers, desc:Multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
    </relatedAreaNo>
  </RelatedAreaParam>
  <relatedShieldRegionNo>
    <!--opt, string, numbers of linked shielded areas, desc:Multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
  </relatedShieldRegionNo>
</AbandonedObjectEvent>
```

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 description, subType:string, desc:OK, Device Busy, Device Error, Invalid Operation, Invalid XML Format, Invalid XML Content, Reboot-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, detailed description of the error code, desc:detailed description of the error code-->OK
    </subStatusCode>
  </ResponseStatus>

```

12.6.3.3 Get the capability of configuring parameters of fallen object detection.

Request URL

GET /ISAPI/Traffic/channels/<channelID>/AID/<SID>/abandonedObject/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Channel No.
SID	string	Scene No., by calling GET /ISAPI/Traffic/channels//sceneinfo/capabilities protocol, the min. and max. attributes of SceneInfo->scenesID in the response message indicate the value range of scene No. If the device does not support this protocol, the default value is 1.

Request Message

None

Response Message

```

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

<AbandonedObjectEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, parameters of fallen object detection, attr:version{req, int, protocolVersion}-->
  <scenesID min="1" max="16">
    <!--ro, req, string, scene ID, range:[1,16], attr:min{req, int},max{req, int}-->1
  </scenesID>
  <enabled opt="false, true">
    <!--ro, req, bool, whether to enable, attr:opt{req, string}-->true
  </enabled>
  <amplifyDetectionEnabled opt="false,true" def="false">
    <!--ro, opt, bool, attr:opt{req, string},def{req, bool}-->true
  </amplifyDetectionEnabled>
  <durationTime min="1" max="60" def="10">
    <!--ro, req, int, duration of fallen object detection, unit:s, attr:min{req, int},max{req, int},def{req, int}-->1
  </durationTime>
  <SnapImageParam>
    <!--ro, opt, object, picture capturing parameters-->
    <imgNum min="1" max="1">
      <!--ro, req, int, each rule corresponds to a group of picture capturing parameters, attr:min{req, int},max{req, int}-->1
    </imgNum>
    <snapSensitivity min="1" max="100" def="50">
      <!--ro, req, int, capture sensitivity, attr:min{req, int},max{req, int},def{req, int}-->1
    </snapSensitivity>
    <ImgGroupInfolist size="1">
      <!--ro, req, array, List of the picture group information, subType:object, attr:size{req, int}-->
      <ImgGroupInfo>
        <!--ro, opt, object, picture group information-->
        <imgInterval min="1" max="1" def="1">
          <!--ro, opt, int, capture interval, unit:s, attr:min{req, int},max{req, int},def{req, int}-->1
        </imgInterval>
        <imgType opt="far">
          <!--ro, req, enum, picture type, subType:string, attr:opt{req, string}, desc:far (distant view)-->far
        </imgType>
        <times min="3" max="3" def="3">
          <!--ro, opt, int, capture times, attr:min{req, int},max{req, int},def{req, int}-->3
        </times>
      </ImgGroupInfo>
    </ImgGroupInfolist>
  </SnapImageParam>
  <RelatedLaneParam>
    <!--ro, req, object, Linked Lane parameters-->
    <RelatedLaneNoList size="0">
      <!--ro, req, array, Linked Lane number, subType:object, attr:size{req, int}-->
      <relatedLaneNo min="1" max="6">
        <!--ro, req, int, Linked Lane number, attr:min{req, int},max{req, int}-->1
      </relatedLaneNo>
    </RelatedLaneNoList>
    <RelatedParkingNoList size="8">
      <!--ro, opt, array, List of Linked parking space numbers, subType:object, attr:size{req, int}-->
      <relatedParkingNo min="1" max="8">
        <!--ro, opt, int, linked parking space number, attr:min{req, int},max{req, int}-->1
      </relatedParkingNo>
    </RelatedParkingNoList>
  </RelatedLaneParam>
  <NormalizedScreenSize>
    <!--ro, opt, object, normalized coordinates-->
    <normalizedScreenWidth>
      <!--ro, req, int, normalized width-->1000
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--ro, req, int, normalized height-->1000
    </normalizedScreenHeight>
  </NormalizedScreenSize>
  <RelatedAreaParam>
    <!--ro, opt, object, Linked areas-->
    <relatedAreaNo opt="1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16">
      <!--ro, opt, string, Linked area numbers, attr:opt{req, string}, desc:Multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
    </relatedAreaNo>
  </RelatedAreaParam>
  <relatedShieldRegionNo min="1" max="6">
    <!--ro, opt, string, numbers of Linked shielded areas, attr:min{req, int},max{req, int}, desc:Multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
  </relatedShieldRegionNo>
  <targetSizeRect>
    <!--ro, opt, object-->
    <displayEnabled opt="true,false">
      <!--ro, req, bool, attr:opt{req, string}-->true
    </displayEnabled>
    <high>
      <!--ro, req, int, range:[0,1000]-->1
    </high>
    <width>
      <!--ro, req, int, range:[0,1000]-->1
    </width>
  </targetSizeRect>
  <relatedType opt="area,lane">
    <!--ro, opt, string, attr:opt{opt, string}-->area
  </relatedType>
</AbandonedObjectEvent>

```

12.6.3.4 Configure the event rule parameters of congestion detection.

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/AID/<SID>/congestion

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

```

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

<CongestionEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="1.0">
  <!--opt, object, congestion detection, attr:version{opt, string, protocolVersion}-->
  <scenesID>
    <!--req, string, scene ID, range:[1,16]-->1
  </scenesID>
  <enabled>
    <!--req, bool, whether to enable-->true
  </enabled>
  <durationTime>
    <!--req, int, detection duration, range:[1,600], unit:s-->1
  </durationTime>
  <filteringTime>
    <!--req, int, filtering time, range:[1,120], unit:min-->1
  </filteringTime>
  <SnapImageParam>
    <!--opt, object, picture capturing parameters-->
    <imgNum>
      <!--req, int, each rule corresponds to a group of picture capturing parameters-->1
    </imgNum>
    <snapSensitivity>
      <!--req, int, capture sensitivity, range:[1,100]-->1
    </snapSensitivity>
    <ImgGroupInfoList>
      <!--req, array, List of the picture group information, subType:object-->
      <ImgGroupInfo>
        <!--opt, object, picture group information-->
        <imgInterval>
          <!--req, int, capture interval, range:[1,3600], unit:s-->1
        </imgInterval>
        <imgType>
          <!--req, enum, picture type, subType:string, desc:far (distant view)-->far
        </imgType>
        <times>
          <!--opt, int, capture times-->1
        </times>
      </ImgGroupInfo>
    </ImgGroupInfoList>
  </SnapImageParam>
  <RelatedLaneParam>
    <!--req, object, Linked Lane parameters-->
    <RelatedLaneNoList>
      <!--req, array, List of Linked Lane numbers, subType:object-->
      <relatedLaneNo>
        <!--opt, int, Linked Lane number-->1
      </relatedLaneNo>
    </RelatedLaneNoList>
    <RelatedParkingNoList>
      <!--opt, array, List of Linked parking space numbers, subType:object-->
      <relatedParkingNo>
        <!--opt, int, Linked parking space number-->1
      </relatedParkingNo>
    </RelatedParkingNoList>
  </RelatedLaneParam>
  <NormalizedScreenSize>
    <!--opt, object, normalized coordinates-->
    <normalizedScreenWidth>
      <!--req, int, normalized width-->1000
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--req, int, normalized height-->1000
    </normalizedScreenHeight>
  </NormalizedScreenSize>
  <roadSmoothWaterLine>
    <!--opt, int, road smoothness threshold, range:[1,80], desc:This node specifies the road smoothness threshold. For example, if the value of this node is 30, the road will be considered smooth if the degree of road congestion is lower than 30.-->30
  </roadSmoothWaterLine>
  <roadCongestionWaterLine>
    <!--opt, int, road congestion threshold, range:[50,100], desc:1. This node specifies the road congestion threshold. For example, if the value of this node is 80, the road will be considered congested if the degree of road congestion is higher than 80.
2. If the degree of road congestion is between the threshold of road smoothness and that of road congestion, vehicles can move slowly on the road.-->70
  </roadCongestionWaterLine>
  <lowSpeedVehicleCongestionDistance>
    <!--opt, int, traffic congestion length of low speed vehicles, range:[50,100], unit:m, desc:This node specifies the traffic congestion length of low speed vehicles. For example, if the value of this node is 20, when the traffic congestion length of low speed vehicles is shorter than 20 meters, the road will not be considered congested and the congestion alarm will not be uploaded.-->70
  </lowSpeedVehicleCongestionDistance>
  <repeatAlarmEnable>
    <!--req, bool, whether to enable uploading alarms many times before the road is not congested, desc:This node specifies whether to upload the congestion alarms many times before the road is not congested. For example, if the congestion event occurred and it lasts for 10 minutes, the value of this node will determine whether to upload the congestion alarms many times during congestion.-->false
  </repeatAlarmEnable>
</CongestionEvent>

```

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, Device Busy, Device Error, Invalid Operation, Invalid XML Format, Invalid XML Content, Reboot-->OK
      </statusString>
      <subStatusCode>
        <!--ro, req, string, detailed description of the error code, desc:detailed description of the error code-->OK
      </subStatusCode>
    </statusString>
  </ResponseStatus>

```

12.6.3.5 Get the event rule parameters of congestion detection.

Request URL

GET /ISAPI/Traffic/channels/<channelID>/AID/<SID>/congestion

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

None

Response Message

```

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

<CongestionEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="1.0">
    <!--ro, opt, object, congestion detection, attr:version{opt, string, protocolVersion}-->
    <scenesID>
        <!--ro, req, string, scene ID, range:[1,16]-->1
    </scenesID>
    <enabled>
        <!--ro, req, bool, whether to enable-->true
    </enabled>
    <durationTime>
        <!--ro, req, int, detection duration, range:[1,600], unit:s-->1
    </durationTime>
    <filteringTime>
        <!--ro, req, int, filtering time, range:[1,120], unit:min-->1
    </filteringTime>
    <SnapImageParam>
        <!--ro, opt, object, picture capturing parameters-->
        <imgNum>
            <!--ro, req, int, each rule corresponds to a group of picture capturing parameters-->1
        </imgNum>
        <snapSensitivity>
            <!--ro, req, int, capture sensitivity, range:[1,100]-->1
        </snapSensitivity>
    </SnapImageParam>
    <ImgGroupInfoList>
        <!--ro, req, array, List of the picture group information, subType:object-->
        <ImgGroupInfo>
            <!--ro, opt, object, picture group information-->
            <imgInterval>
                <!--ro, req, int, capture interval, range:[1,3600], unit:s-->1
            </imgInterval>
            <imgType>
                <!--ro, req, enum, picture type, subType:string, desc:far (distant view)-->far
            </imgType>
            <times>
                <!--ro, opt, int, capture times-->1
            </times>
        </ImgGroupInfo>
    </ImgGroupInfoList>
    <RelatedLaneParam>
        <!--ro, opt, object, Linked Lane parameters-->
        <RelatedLaneNoList>
            <!--ro, req, array, List of linked lane numbers, subType:object-->
            <relatedLaneNo>
                <!--ro, opt, int, linked lane number-->1
            </relatedLaneNo>
        </RelatedLaneNoList>
        <RelatedParkingNoList>
            <!--ro, opt, array, List of linked parking space numbers, subType:object-->
            <relatedParkingNo>
                <!--ro, opt, int, linked parking space number-->1
            </relatedParkingNo>
        </RelatedParkingNoList>
    </RelatedLaneParam>
    <NormalizedScreenSize>
        <!--ro, opt, object, normalized coordinates-->
        <normalizedScreenWidth>
            <!--ro, req, int, normalized width-->1000
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!--ro, req, int, normalized height-->1000
        </normalizedScreenHeight>
    </NormalizedScreenSize>
    <roadSmoothWaterLine>
        <!--ro, opt, int, road smoothness threshold, range:[1,80], desc:This node specifies the road smoothness threshold. For example, if the value of this node is 30, the road will be considered smooth if the degree of road congestion is lower than 30.-->30
    </roadSmoothWaterLine>
    <roadCongestionWaterLine>
        <!--ro, opt, int, road congestion threshold, range:[50,100], desc:1. This node specifies the road congestion threshold. For example, if the value of this node is 80, the road will be considered congested if the degree of road congestion is higher than 80.
        2. If the degree of road congestion is between the threshold of road smoothness and that of road congestion, vehicles can move slowly on the road.-->70
    </roadCongestionWaterLine>
    <lowSpeedVehicleCongestionDistance>
        <!--ro, opt, int, traffic congestion length of low speed vehicles, range:[50,100], unit:m, desc:This node specifies the traffic congestion length of low speed vehicles. For example, if the value of this node is 20, when the traffic congestion length of low speed vehicles is shorter than 20 meters, the road will not be considered congested and the congestion alarm will not be uploaded.-->70
    </lowSpeedVehicleCongestionDistance>
    <repeatAlarmEnable>
        <!--ro, opt, bool, whether to enable uploading alarms many times before the road is not congested, desc:This node specifies whether to upload the congestion alarms many times before the road is not congested. For example, if the congestion event occurred and it lasts for 10 minutes, the value of this node will determine whether to upload the congestion alarms many times during congestion.-->false
    </repeatAlarmEnable>
</CongestionEvent>

```

12.6.3.6 Get the capability of configuring the event rule parameters of congestion detection.

Request URL

GET /ISAPI/Traffic/channels/<channelID>/AID/<SID>/congestion/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

None

Response Message

```

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

<CongestionEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="1.0">
    <!--ro, opt, object, congestion detection event, attr:version{opt, string, protocolVersion}-->
    <scenesID min="1" max="16">
        <!--ro, req, string, scene ID which is between 1 and 16, attr:min{req, int},max{req, int}-->1
    </scenesID>
    <enabled opt="false,true">
        <!--ro, req, bool, whether to enable, attr:opt{req, string}-->true
    </enabled>
    <durationTime min="1" max="600" def="30">
        <!--ro, req, int, detection duration, unit:s, attr:min{req, int},max{req, int},def{req, int}-->1
    </durationTime>
    <filteringTime min="1" max="10" def="2">
        <!--ro, req, int, Filtering time, unit:min, attr:min{req, int},max{req, int},def{req, int}-->1
    </filteringTime>
    <SnapImageParam>
        <!--ro, opt, object, picture capturing parameters-->
        <imgNum min="1" max="1">
            <!--ro, req, int, each rule corresponds to a group of picture capturing parameters, attr:min{opt, int},max{opt, int}-->1
        </imgNum>
        <snapSensitivity min="1" max="100" def="1">
            <!--ro, req, int, capture sensitivity, attr:min{req, int},max{req, int},def{req, int}-->1
        </snapSensitivity>
        <ImgGroupInfoList size="1">
            <!--ro, req, array, List of the picture group information, subType:object, attr:size{req, int}-->
            <ImgGroupInfo>
                <!--ro, opt, object, picture group information-->
                <imgInterval min="1" max="3600" def="600">
                    <!--ro, req, int, capture interval, unit:s, attr:min{req, int},max{req, int},def{req, int}-->1
                </imgInterval>
                <imgType opt="far">
                    <!--ro, req, enum, picture type, subType:string, attr:opt{req, string}, desc:far (distant view)-->far
                </imgType>
                <times min="1" max="1" def="1">
                    <!--ro, opt, int, capture times, attr:min{req, int},max{req, int},def{req, int}-->1
                </times>
            </ImgGroupInfo>
        </ImgGroupInfoList>
    </SnapImageParam>
    <RelatedLaneParam>
        <!--ro, opt, object, Linked Lane parameters-->
        <RelatedLaneNoList size="6">
            <!--ro, req, array, List of linked lane numbers, subType:object, attr:size{req, int}-->
            <relatedLaneNo min="1" max="6">
                <!--ro, opt, int, linked lane number, attr:min{req, int},max{req, int}-->1
            </relatedLaneNo>
        </RelatedLaneNoList>
        <RelatedParkingNoList size="8">
            <!--ro, opt, array, List of linked parking space numbers, subType:object, attr:size{opt, string}-->
            <relatedParkingNo min="1" max="8">
                <!--ro, opt, int, linked parking space number, attr:min{req, int},max{req, int}-->1
            </relatedParkingNo>
        </RelatedParkingNoList>
    </RelatedLaneParam>
    <NormalizedScreenSize>
        <!--ro, opt, object, normalized coordinates-->
        <normalizedScreenWidth>
            <!--ro, req, int, normalized width-->1000
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!--ro, req, int, normalized height-->1000
        </normalizedScreenHeight>
    </NormalizedScreenSize>
    <roadSmoothWaterLine min="1" max="60" def="30">
        <!--ro, opt, int, road smoothness threshold, unit:s, attr:min{req, int},max{req, int},def{req, int}, desc:This node specifies the road smoothness threshold. For example, if the value of this node is 30, the road will be considered smooth if the degree of road congestion is lower than 30.-->1
    </roadSmoothWaterLine>
    <roadCongestionWaterLine min="30" max="100" def="70">
        <!--ro, opt, int, road congestion threshold, unit:s, attr:min{req, int},max{req, int},def{req, int}, desc:This node specifies the road congestion threshold. For example, if the value of this node is 80, the road will be considered congested if the degree of road congestion is higher than 80.
    <2. If the degree of road congestion is between the threshold of road smoothness and that of road congestion, vehicles can move slowly on the road.-->70
    </roadCongestionWaterLine>
    <lowSpeedVehicleCongestionDistance min="1" max="200" def="50">
        <!--ro, opt, int, traffic congestion length of low speed vehicles, unit:m, attr:min{req, int},max{req, int},def{req, int}, desc:This node specifies the traffic congestion length of low speed vehicles. For example, if the value of this node is 20, when the traffic congestion length of low speed vehicles is shorter than 20 meters, the road will not be considered congested and the congestion alarm will not be uploaded.-->50
    </lowSpeedVehicleCongestionDistance>
    <repeatAlarmEnable opt="false,true" def="false">
        <!--ro, opt, bool, whether to enable uploading alarms many times before the road is not congested, attr:opt{req, string},def{req, string}, desc:This node specifies whether to upload the congestion alarms many times before the road is not congested. For example, if the congestion event occurred and it lasts for 10 minutes, the value of this node will determine whether to upload the congestion alarms many times during congestion.-->false
    </repeatAlarmEnable>
</CongestionEvent>

```

12.6.3.7 Get the parameters of construction detection.

Request URL

GET /ISAPI/Traffic/channels/<channelID>/AID/<SID>/construction

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

None

Response Message

```

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

<ConstructionEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, construction detection, attr:version{opt, string, protocolVersion}-->
  <scenesID>
    <!--ro, req, string, scene ID, range:[1,16]-->1
  </scenesID>
  <enabled>
    <!--ro, req, bool, whether to enable-->true
  </enabled>
  <durationTime>
    <!--ro, req, int, detection duration, range:[1,30], unit:s-->1
  </durationTime>
  <filteringTime>
    <!--ro, req, int, filtering time, range:[1,120], unit:min-->1
  </filteringTime>
  <SnapImageParam>
    <!--ro, opt, object, picture capturing parameters-->
    <imgNum>
      <!--ro, req, int, each rule corresponds to a group of picture capturing parameters-->1
    </imgNum>
    <snapSensitivity>
      <!--ro, req, int, capture sensitivity, range:[1,100]-->1
    </snapSensitivity>
    <ImgGroupInfolist>
      <!--ro, req, array, List of the picture group information, subType:object-->
      <ImgGroupInfo>
        <!--ro, opt, object, picture group information-->
        <imgInterval>
          <!--ro, opt, int, capture interval, range:[1,3600], unit:s-->1
        </imgInterval>
        <imgType>
          <!--ro, req, enum, picture type, subType:string, desc:far (distant view)-->far
        </imgType>
        <times>
          <!--ro, opt, int, capture times-->1
        </times>
      </ImgGroupInfo>
    </ImgGroupInfolist>
  </SnapImageParam>
  <RelatedLaneParam>
    <!--ro, opt, object, Linked Lane parameters-->
    <RelatedLaneNoList>
      <!--ro, req, array, List of linked lane numbers, subType:object-->
      <relatedLaneNo>
        <!--ro, opt, int, linked lane number-->1
      </relatedLaneNo>
    </RelatedLaneNoList>
    <RelatedParkingNoList>
      <!--ro, opt, array, List of linked parking space numbers, subType:object-->
      <relatedParkingNo>
        <!--ro, opt, int, linked parking space number-->1
      </relatedParkingNo>
    </RelatedParkingNoList>
  </RelatedLaneParam>
  <NormalizedScreenSize>
    <!--ro, opt, object, normalized coordinates-->
    <normalizedScreenWidth>
      <!--ro, req, int, normalized width-->1000
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--ro, req, int, normalized height-->1000
    </normalizedScreenHeight>
  </NormalizedScreenSize>
  <RelatedAreaParam>
    <!--ro, opt, object, parameters of the linked target areas-->
    <relatedAreaNo>
      <!--ro, opt, string, linked area numbers, desc:Multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
    </relatedAreaNo>
  </RelatedAreaParam>
  <relatedShieldRegionNo>
    <!--ro, opt, string, numbers of linked shielded areas, desc:Multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
  </relatedShieldRegionNo>
  <onlyReportOnceBeforeRevocationEnable>
    <!--ro, opt, bool-->true
  </onlyReportOnceBeforeRevocationEnable>
  <filterOtherEventsEnable>
    <!--ro, opt, bool-->true
  </filterOtherEventsEnable>
  <relatedType>
    <!--ro, opt, enum, subType:string-->area
  </relatedType>
</ConstructionEvent>

```

12.6.3.8 Get the capability of configuring parameters of construction detection.

Request URL

GET /ISAPI/Traffic/channels/<channelID>/AID/<SID>/construction/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

None

Response Message

```

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

<ConstructionEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, Get the capability of configuring parameters of construction detection., attr:version{req, string, protocolVersion}-->
    <scenesID min="1" max="16">
        <!--ro, req, string, scene ID which is between 1 and 16, attr:min{req, int},max{req, int}-->1
    </scenesID>
    <enabled opt="false,true">
        <!--ro, req, bool, whether to enable, attr:opt{req, string}-->true
    </enabled>
    <durationTime min="1" max="3600" def="600">
        <!--ro, req, int, detection duration, unit:s, attr:min{req, int},max{req, int},def{req, int}-->1
    </durationTime>
    <filteringTime min="1" max="120" def="30">
        <!--ro, req, int, filtering time, unit:min, attr:min{req, int},max{req, int},def{req, int}-->1
    </filteringTime>
    <SnapImageParam>
        <!--ro, opt, object, picture capturing parameters-->
        <imgNum min="1" max="1">
            <!--ro, req, int, each rule corresponds to a group of picture capturing parameters, attr:min{opt, int},max{opt, int}-->1
        </imgNum>
        <snapSensitivity min="1" max="100" def="1">
            <!--ro, req, int, capture sensitivity, attr:min{req, int},max{req, int},def{req, int}-->1
        </snapSensitivity>
        <ImgGroupInfoList size="1">
            <!--ro, req, array, List of the picture group information, subType:object, attr:size{req, int}-->
            <ImgGroupInfo>
                <!--ro, opt, object, picture group information-->
                <imgInterval min="1" max="3600" def="600">
                    <!--ro, opt, int, capture interval, unit:s, attr:min{req, int},max{req, int},def{req, int}-->1
                </imgInterval>
                <imgType opt="far">
                    <!--ro, req, enum, picture type, subType:string, attr:opt{req, string}, desc:far (distant view)-->far
                </imgType>
                <times min="1" max="1" def="1">
                    <!--ro, opt, int, capture times, attr:min{req, int},max{req, int},def{req, int}-->1
                </times>
            </ImgGroupInfo>
        </ImgGroupInfoList>
    </SnapImageParam>
    <RelatedLaneParam>
        <!--ro, opt, object, Linked Lane parameters-->
        <RelatedLaneNoList size="6">
            <!--ro, req, array, List of linked lane numbers, subType:object, attr:size{req, int}-->
            <relatedLaneNo min="1" max="6">
                <!--ro, opt, int, linked lane number, attr:min{req, int},max{req, int}-->1
            </relatedLaneNo>
        </RelatedLaneNoList>
        <RelatedParkingNoList size="8">
            <!--ro, opt, array, List of linked parking space numbers, subType:object, attr:size{opt, string}-->
            <relatedParkingNo min="1" max="8">
                <!--ro, opt, int, linked parking space number, attr:min{req, int},max{req, int}-->1
            </relatedParkingNo>
        </RelatedParkingNoList>
    </RelatedLaneParam>
    <NormalizedScreenSize>
        <!--ro, opt, object, normalized coordinates-->
        <normalizedScreenWidth>
            <!--ro, req, int, normalized width-->1000
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!--ro, req, int, normalized height-->1000
        </normalizedScreenHeight>
    </NormalizedScreenSize>
    <RelatedAreaParam>
        <!--ro, opt, object, Linked areas-->
        <relatedAreaNo opt="1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16">
            <!--ro, opt, string, linked area numbers, attr:opt{req, string}, desc:Multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
        </relatedAreaNo>
    </RelatedAreaParam>
    <relatedShieldRegionNo min="1" max="6">
        <!--ro, opt, string, numbers of linked shielded areas, attr:min{req, int},max{req, int}, desc:Multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
    </relatedShieldRegionNo>
    <onlyReportOnceBeforeRevocationEnable opt="false,true" def="true">
        <!--ro, opt, bool, attr:opt{req, string},def{req, int}-->true
    </onlyReportOnceBeforeRevocationEnable>
    <filterOtherEventsEnable opt="false,true" def="true">
        <!--ro, opt, bool, attr:opt{req, string},def{req, int}-->true
    </filterOtherEventsEnable>
    <relatedType opt="area,lane">
        <!--ro, opt, string, attr:opt{opt, string}-->area
    </relatedType>
</ConstructionEvent>

```

12.6.3.9 Get the rule parameters of pedestrian detection.

Request URL

GET /ISAPI/Traffic/channels/<channelID>/AID/<SID>/pedestrian

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

None

Response Message

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

<PedestrianEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--req, object, rule configuration of pedestrian detection, attr:version{req, string, protocolVersion}-->
    <scenesID>
        <!--req, string, scene ID which is between 1 and 16-->
    </scenesID>
    <enabled>
        <!--req, bool, whether to enable-->true
    </enabled>
    <amplifyDetectionEnabled>
        <!--opt, bool-->true
    </amplifyDetectionEnabled>
    <durationTime>
        <!--req, int, detection duration, range:[1,60], unit:s-->1
    </durationTime>
    <SnapImageParam>
        <!--opt, object, picture capturing parameters-->
        <imgNum>
            <!--req, int, each rule corresponds to a group of picture capturing parameters-->1
        </imgNum>
        <snapSensitivity>
            <!--req, int, capture sensitivity, range:[1,100]-->1
        </snapSensitivity>
        <ImgGroupInfoList>
            <!--req, array, List of the picture group information, subType:object-->
            <ImgGroupInfo>
                <!--opt, object, picture group information-->
                <imgInterval>
                    <!--opt, int, capture interval-->1
                </imgInterval>
                <imgType>
                    <!--req, enum, picture type, subType:string, desc:far (distant view)-->far
                </imgType>
                <times>
                    <!--opt, int, capture times-->1
                </times>
            </ImgGroupInfo>
        </ImgGroupInfoList>
    </SnapImageParam>
    <RelatedLaneParam>
        <!--opt, object, linked lane parameters-->
        <RelatedLaneNoList>
            <!--req, array, list of linked lane numbers, subType:object-->
            <relatedLaneNo>
                <!--opt, int, linked lane number-->1
            </relatedLaneNo>
            <RelatedParkingNoList>
                <!--opt, array, list of linked parking space numbers, subType:object-->
                <relatedParkingNo>
                    <!--opt, int, linked parking space number-->1
                </relatedParkingNo>
            </RelatedParkingNoList>
        </RelatedLaneNoList>
    </RelatedLaneParam>
    <NormalizedScreenSize>
        <!--opt, object, normalized coordinates-->
        <normalizedScreenWidth>
            <!--req, int, normalized width-->1000
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!--req, int, normalized height-->1000
        </normalizedScreenHeight>
    </NormalizedScreenSize>
    <RelatedAreaParam>
        <!--opt, object, parameters of the linked target areas-->
        <relatedAreaNo>
            <!--opt, string, linked area numbers, desc:Multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
        </relatedAreaNo>
    </RelatedAreaParam>
    <targetType>
        <!--opt, enum, detection target, subType:string, desc:pedestrian, TwoWheeler (two wheeler), Tricycle (tricycle), police (police officer), constructionStaff (construction staff)-->pedestrian
    </targetType>

```

```

<constructionStaff (construction staff)-->pedestrian
</targetType>
<TargetTypeList>
  <!--opt, array, list of target types, subType:string, desc:Multiple target types are supported.-->
  <targetType>
    <!--opt, enum, detection target, subType:string, desc:The default value is pedestrian. If the device does not support this node, the default value is also pedestrian. pedestrian, TwoWheeler (two wheeler), Tricycle (tricycle), police (police officer), constructionStaff (construction staff)-->pedestrian
    </targetType>
  </TargetTypeList>
  <filteringTime>
    <!--opt, int, filtering time, range:[0,120], unit:min-->1
  </filteringTime>
  <relatedShieldRegionNo>
    <!--opt, string, numbers of linked shielded areas, desc:Multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
  </relatedShieldRegionNo>
  <relatedType>
    <!--opt, enum, subType:string-->area
  </relatedType>
</PedestrianEvent>

```

12.6.3.10 Get the configuration parameters of roadblock detection

Request URL

GET /ISAPI/Traffic/channels/<channelID>/AID/<SID>/roadBlock

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<RoadBlockEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, ro, coordinate ratio between the upper Level and the device, the default value is 1000, see details in the message of XML_NormlizedScreenSize, attr:version{req, string, protocolVersion}-->
    <scenesID>
        <!--ro, req, string, scene ID,ranges from 1 to 16, range:[1,16]-->1
    </scenesID>
    <enabled>
        <!--ro, req, bool-->true
    </enabled>
    <amplifyDetectionEnabled>
        <!--ro, opt, bool-->true
    </amplifyDetectionEnabled>
    <durationTime>
        <!--ro, req, int, duration of roadblock detection,ranges from 1 to 60,the default value is 10,unit: second, range:[1,60], unit:s-->1
    </durationTime>
    <filteringTime>
        <!--ro, req, int, filtering time,ranges from 1 to 60,the default value is 30,unit: minute, range:[0,120], unit:min-->1
    </filteringTime>
    <SnapImageParam>
        <!--ro, opt, object-->
        <imgNum>
            <!--ro, req, int-->1
        </imgNum>
        <snapSensitivity>
            <!--ro, req, int, range:[1,100]-->1
        </snapSensitivity>
        <ImgGroupInfoList>
            <!--ro, req, array, subType:object-->
            <ImgGroupInfo>
                <!--ro, opt, object-->
                <imgInterval>
                    <!--ro, opt, int-->1
                </imgInterval>
                <imgType>
                    <!--ro, req, enum, subType:string-->far
                </imgType>
                <times>
                    <!--ro, opt, int-->1
                </times>
            </ImgGroupInfo>
        </ImgGroupInfoList>
    </SnapImageParam>
    <RelatedLaneParam>
        <!--ro, opt, object-->
        <RelatedLaneNoList>
            <!--ro, req, array, subType:object-->
            <relatedLaneNo>
                <!--ro, opt, int-->1
            </relatedLaneNo>
        </RelatedLaneNoList>
    </RelatedLaneParam>
    <NormalizedScreenSize>
        <!--ro, opt, object-->
        <normalizedScreenWidth>
            <!--ro, req, int-->1000
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!--ro, req, int-->1000
        </normalizedScreenHeight>
    </NormalizedScreenSize>
    <RelatedAreaParam>
        <!--ro, opt, object-->
        <relatedAreaNo>
            <!--ro, opt, string-->1
        </relatedAreaNo>
    </RelatedAreaParam>
    <relatedShieldRegionNo>
        <!--ro, opt, string-->1
    </relatedShieldRegionNo>
    <onlyReportOnceBeforeRevocationEnable>
        <!--ro, opt, bool-->true
    </onlyReportOnceBeforeRevocationEnable>
    <relatedType>
        <!--ro, opt, enum, subType:string-->area
    </relatedType>
</RoadBlockEvent>

```

12.6.3.11 Get the configuration capability of roadblock detection

Request URL

GET /ISAPI/Traffic/channels/<channelID>/AID/<SID>/roadBlock/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

None

Response Message

```

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

<RoadBlockEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, attr:version{opt, string, protocolVersion}-->
  <scenesID min="1" max="16">
    <!--ro, req, string, scene ID, which is between 1 and 16, attr:min{req, int},max{req, int}-->1
  </scenesID>
  <enabled opt="false,true">
    <!--ro, req, bool, whether to enable the function, attr:opt{req, string}-->true
  </enabled>
  <amplifyDetectionEnabled opt="false,true" def="false">
    <!--ro, opt, bool, attr:opt{req, string},def{req, bool}-->true
  </amplifyDetectionEnabled>
  <durationTime min="1" max="60" def="30">
    <!--ro, req, int, duration of roadblock detection, range:[0,60], unit:s, attr:min{req, int},max{req, int},def{req, int}-->1
  </durationTime>
  <filteringTime min="0" max="120" def="2">
    <!--ro, req, int, filtering time, range:[0,120], unit:min, attr:min{req, int},max{req, int},def{req, int}-->1
  </filteringTime>
  <SnapImageParam>
    <!--ro, opt, object, parameters for capturing pictures-->
    <imgNum>
      <!--ro, req, int, each rule corresponds to a group of parameters for capturing pictures-->1
    </imgNum>
    <snapSensitivity min="1" max="100" def="50">
      <!--ro, req, int, capture sensitivity, range:[1,100], attr:min{req, int},max{req, int},def{req, int}-->1
    </snapSensitivity>
    <ImgGroupInfoList size="1">
      <!--ro, req, array, List of the picture group information, subType:object, attr:size{req, int}-->
      <ImgGroupInfo>
        <!--ro, opt, object, picture group information-->
        <imgInterval>
          <!--ro, opt, int, capture interval-->1
        </imgInterval>
        <imgType opt="far">
          <!--ro, req, enum, picture type, subType:string, attr:opt{req, string}, desc:"far" (distant view)-->far
        </imgType>
        <times>
          <!--ro, opt, int, capture times-->1
        </times>
      </ImgGroupInfo>
    </ImgGroupInfoList>
  </SnapImageParam>
  <RelatedLaneParam>
    <!--ro, opt, object, Linked Lane parameters-->
    <RelatedLaneNoList size="4">
      <!--ro, req, array, List of Linked Lane numbers, subType:object, attr:size{req, int}-->
      <relatedLaneNo min="1" max="16">
        <!--ro, opt, int, related Lane No., attr:min{req, int},max{req, int}-->1
      </relatedLaneNo>
      <RelatedParkingNoList size="8">
        <!--ro, opt, array, List of Linked parking space No., subType:object, attr:size{req, int}-->
        <relatedParkingNo min="1" max="16">
          <!--ro, opt, int, Linked parking space No., attr:min{req, int},max{req, int}-->1
        </relatedParkingNo>
      </RelatedParkingNoList>
    </RelatedLaneNoList>
  </RelatedLaneParam>
  <NormalizedScreenSize>
    <!--ro, opt, object, coordinate ratio between the upper Level and the device-->
    <normalizedScreenWidth>
      <!--ro, req, int, normalized width-->1000
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--ro, req, int, normalized height-->1000
    </normalizedScreenHeight>
  </NormalizedScreenSize>
  <RelatedAreaParam>
    <!--ro, opt, object, Linked areas-->
    <relatedAreaNo opt="1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16">
      <!--ro, opt, string, Linked area No., attr:opt{req, string}, desc:multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
    </relatedAreaNo>
    <RelatedAreaParam>
      <relatedShieldRegionNo min="1" max="6">
        <!--ro, opt, string, numbers of Linked shielded areas, attr:min{req, int},max{req, int}, desc:multiple numbers should be separated by commas, e.g., 1,2,3,4.-->1
      </relatedShieldRegionNo>
      <onlyReportOnceBeforeRevocationEnable opt="false,true" def="true">
        <!--ro, opt, bool, attr:opt{req, string},def{req, int}-->true
      </onlyReportOnceBeforeRevocationEnable>
      <relatedType opt="area,lane">
        <!--ro, opt, string, attr:opt{opt, string}-->area
      </relatedType>
    </RelatedAreaParam>
  </RoadBlockEvent>

```

12.6.3.12 Get the screen parameter capability

Request URL

GET /ISAPI/Traffic/channels/<channelID>/eventRule/<SID>/laneRule/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

None

Response Message

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

<LaneParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, ro, coordinate ratio between the upper Level and the device, the default value is 1000, see details in the message of XML_Cap_NormalizedScreenSize, attr:version{req, string, protocolVersion}-->
    <scenesID min="1" max="16">
        <!--ro, req, string, scene ID, range:[1,16], attr:min{req, int},max{req, int}-->test
    </scenesID>
    <VolumeLane>
        <!--ro, opt, object-->
        <RegionCoordinatesList>
            <!--ro, req, array, subType:object, range:[0,2]-->
            <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:[0,1000]-->0
                </positionX>
                <positionY>
                    <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
                </positionY>
            </RegionCoordinates>
        </RegionCoordinatesList>
        <distanceOfLines min="1" max="200" def="80">
            <!--ro, opt, int, range:[1,200], unit:m, attr:min{req, int},max{req, int},def{req, int}-->80
        </distanceOfLines>
    </VolumeLane>
    <LaneRuleInfo>
        <!--ro, opt, object-->
        <laneNum min="1" max="12" def="4">
            <!--ro, req, int, number of lanes, range:[1,10], attr:min{req, int},max{req, int},def{req, int}, desc:number of lanes-->1
        </laneNum>
        <SingleLaneList size="10">
            <!--ro, req, array, subType:object, attr:size{req, int}-->
            <SingleLane>
                <!--ro, opt, object-->
                <laneNo min="1" max="10">
                    <!--ro, req, int, Lane No., range:[1,10], attr:min{req, int},max{req, int}-->1
                </laneNo>
                <relatedLaneEnabled opt="false,true">
                    <!--ro, opt, bool, attr:opt{req, string}-->true
                </relatedLaneEnabled>
                <laneType
                    opt="banLeft,banPasser,banRight,banTrucks,bus,carriageWay,emgLane,fast,highway,motor,nonmotor,other,reverse,slow,unknown,urbanFastWay,dangerousGoodsLane">
                        <!--ro, req, enum, Lane type, subType:string, attr:opt{req, string}, desc:Lane type-->banLeft
                    </laneType>
                    <laneDirection opt="unknown,up,down">
                        <!--ro, opt, enum, Lane direction, subType:string, attr:opt{req, string}, desc:Lane direction-->unknown
                    </laneDirection>
                    <laneDirectionType opt="0,1,2,3,4,5,6,7,8">
                        <!--ro, req, enum, Lane direction type, subType:int, attr:opt{req, string}, desc:0 (unknown),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)-->0
                    </laneDirectionType>
                    <voiceLinkageEnabled opt="false,true">
                        <!--ro, opt, bool, attr:opt{req, string}-->true
                    </voiceLinkageEnabled>
                    <leftLaneLine>
                        <!--ro, req, object, Left Lane Line-->
                        <laneLineType opt="unknown,white,singleYellow,doubleYellow,solidDotted,dottedSolid">
                            <!--ro, opt, enum, subType:string, attr:opt{req, string}-->unknown
                        </laneLineType>
                    </leftLaneLine>
                    <!--ro, req, object, Lane Line-->
                    <RegionCoordinatesList size="2">
                        <!--ro, req, array, subType:object, range:[0,2], attr:size{req, int}-->
                        <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:[0,1000]-->0
                            </positionX>
                            <positionY>
                                <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
                            </positionY>
                        </RegionCoordinates>
                    </RegionCoordinatesList>
                </laneType>
            </SingleLane>
        </SingleLaneList>
    </LaneRuleInfo>
</LaneParam>
```

```
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</LaneLine>
</LeftLaneLine>
<RightLaneLine>
  <!--ro, opt, object, right Lane Line, desc:right Lane Line-->
  <laneLineType opt="unknown,white,singleYellow,doubleYellow,solidDotted,dottedSolid">
    <!--ro, opt, enum, subType:string, attr:opt{req, string}-->unknown
  </laneLineType>
  <LaneLine>
    <!--ro, req, object, Lane Line-->
    <RegionCoordinatesList size="2">
      <!--ro, req, array, subType:object, range:[0,2], attr:size{req, int}-->
      <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:[0,1000]-->0
        </positionX>
        <positionY>
          <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
        </positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
  </LaneLine>
</RightLaneLine>
<relatedDriveWay min="1" max="99">
  <!--ro, opt, int, linked Lane number, range:[1,99], attr:min{req, int},max{req, int}-->1
</relatedDriveWay>
<IOOutList size="8">
  <!--ro, opt, array, subType:object, attr:size{req, int}-->
  <IOOut>
    <!--ro, req, object-->
    <id min="1" max="8">
      <!--ro, req, int, range:[1,8], attr:min{req, int},max{req, int}-->0
    </id>
    <enabled opt="true,false">
      <!--ro, req, bool, attr:opt{req, string}-->true
    </enabled>
  </IOOut>
</IOOutList>
<flashMode opt="together,alternation">
  <!--ro, opt, enum, subType:string, attr:opt{req, string}-->together
</flashMode>
<maxVehicleNum min="0" max="100">
  <!--ro, opt, int, range:[0,100], attr:min{req, int},max{req, int}-->6
</maxVehicleNum>
</SingleLane>
</SingleLaneList>
<RightLaneLine>
  <!--ro, req, object, right Lane Line-->
  <laneLineType opt="unknown,white,singleYellow,doubleYellow,solidDotted,dottedSolid">
    <!--ro, opt, enum, subType:string, attr:opt{req, string}-->unknown
  </laneLineType>
  <LaneLine>
    <!--ro, req, object, Lane Line-->
    <RegionCoordinatesList size="2">
      <!--ro, req, array, subType:object, range:[0,2], attr:size{req, int}-->
      <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:[0,1000]-->0
        </positionX>
        <positionY>
          <!--ro, req, int, Y-coordinate, range:[0,1000]-->0
        </positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
  </LaneLine>
</RightLaneLine>
</LaneRuleInfo>
<NormalizedScreenSize>
  <!--ro, opt, object-->
  <normalizedScreenWidth>
    <!--ro, req, int, normalized width-->0
  </normalizedScreenWidth>
  <normalizedScreenHeight>
    <!--ro, req, int, normalized height-->0
  </normalizedScreenHeight>
</NormalizedScreenSize>
<captureLineList size="2">
  <!--ro, opt, object, attr:size{req, int}-->
  <singleCaptureLine>
    <!--ro, opt, object-->
    <captureLineID min="1" max="2">
      <!--ro, opt, int, range:[1,2], attr:min{req, int},max{req, int}-->1
    </captureLineID>
    <captureLineType opt="motorVehicle,nonmotorVehicle">
      <!--ro, opt, enum, subType:string, attr:opt{req, string}-->motorVehicle
    </captureLineType>
  </captureLineList>
  <RegionCoordinatesList size="2">
    <!--ro, req, array, subType:object, range:[0,2], attr:size{req, int}-->
    <RegionCoordinates>
      <!--ro, opt, object-->

```

```

<positionX>
    <!--req, int, range:[0,1000]-->0
</positionX>
<positionY>
    <!--req, int, range:[0,1000]-->0
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</singleCaptureLine>
</captureLineList>
</LaneParam>

```

12.6.3.13 Configure the parameters of the polygon area

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/eventRule/<SID>/ploygonRule

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

```

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

<PloygonRule xmlns="http://www.isapi.org/ver20/XMLSchema" version="1.0">
<!--opt, object, parameters of the polygon area, attr:version{req, string, protocolVersion}-->
<DetectionRegionList>
    <!--opt, array, list of polygon detection areas, subType:object, desc:Up to 6 areas can be configured.-->
    <DetectionRegion>
        <!--opt, object, a single polygon detection area-->
        <RegionCoordinatesList>
            <!--opt, array, List of area points, subType:object, range:[3,10]-->
            <RegionCoordinates>
                <!--opt, object, area point, desc:The origin of coordinates is the upper-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>
        <rightLine>
            <!--opt, object-->
            <lineType>
                <!--req, enum, subType:string-->unknown
            </lineType>
        </rightLine>
        <leftLine>
            <!--opt, object-->
            <lineType>
                <!--req, enum, subType:string-->unknown
            </lineType>
        </leftLine>
    </DetectionRegion>
</DetectionRegionList>
<RuleRegionList>
    <!--opt, array, list of rule areas, subType:object-->
    <RuleRegion>
        <!--opt, object, rule area-->
        <RuleCoordinatesList>
            <!--opt, array, List of area points, subType:object, desc:It is a rectangle area.-->
            <RuleCoordinates>
                <!--opt, object, area point, desc:The origin of coordinates is the upper-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>
            </RuleCoordinates>
        </RuleCoordinatesList>
    </RuleRegion>
</RuleRegionList>

```

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 (Rebooting Required).-->0
    <statusString>
      <!--ro, req, enum, status information, subType:string, desc:OK (succeeded), Device Busy (the device is busy), Device Error (device exception), Invalid Operation (invalid operation), ;Invalid XML Format (invalid XML format), Invalid XML Content (invalid XML content), Reboot (reboot the device).-->OK
    </statusString>
    <subStatusCode>
      <!--ro, req, string, error code details, desc:error code details-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

12.6.3.14 Get the parameters of the polygon area

Request URL

GET /ISAPI/Traffic/channels/<channelID>/eventRule/<SID>/polygonRule

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

None

Response Message

```

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

<PloygonRule xmlns="http://www.isapi.org/ver20/XMLSchema" version="1.0">
  <!--ro, opt, object, parameters of the polygon area, attr:version{req, string, protocolVersion}-->
  <DetectionRegionList size="16">
    <!--ro, opt, array, List of polygon detection areas, subType:object, range:[1,16], attr:size{req, int}-->
    <DetectionRegion>
      <!--ro, opt, object, a single polygon detection area-->
      <RegionCoordinatesList>
        <!--ro, opt, array, List of area points, subType:object, range:[3,10]-->
        <RegionCoordinates>
          <!--ro, opt, object, area point, desc:The origin of coordinates is the upper-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>
    </DetectionRegion>
    <rightLine>
      <!--ro, opt, object-->
      <lineType>
        <!--ro, req, enum, subType:string-->unknown
      </lineType>
    </rightLine>
    <leftLine>
      <!--ro, opt, object-->
      <lineType>
        <!--ro, req, enum, subType:string-->unknown
      </lineType>
    </leftLine>
    <regionType>
      <!--ro, opt, enum, subType:string-->normal
    </regionType>
    <regionTrackDir>
      <!--ro, opt, enum, subType:string-->leftToRight
    </regionTrackDir>
    <regionNo>
      <!--ro, opt, int, range:[1,16]-->1
    </regionNo>
    <regionName>
      <!--ro, opt, string, range:[0,128]-->test
    </regionName>
  </DetectionRegion>
</DetectionRegionList>
<RuleRegionList size="1">
  <!--ro, opt, array, List of rule areas, subType:object, range:[0,4], attr:size{req, int}-->
  <RuleRegion>
    <!--ro, opt, object, rule area-->
    <RuleCoordinatesList size="4">
      <!--ro, opt, array, List of area points, subType:object, attr:size{req, int}, desc:It is a rectangle area.-->
      <RuleCoordinates>
        <!--ro, opt, object, area point, desc:The origin of coordinates is the upper-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>
      </RuleCoordinates>
    </RuleCoordinatesList>
  </RuleRegion>
</RuleRegionList>
</PloygonRule>

```

12.6.3.15 Traffic Incident Detection

EventType:AID

```

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

<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, result, attr:version{req, string}-->
  <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 number of the device that triggers the alarm-->80
  </portNo>
  <protocol>
    <!--ro, opt, enum, transmission communication 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, int, channel number of the device that triggers the alarm, desc:video channel number that triggers the alarm-->1
</channelID>
<dateTime>
    <!--ro, req, datetime, alarm triggering 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:AID (traffic incident detection)-->AID
</eventType>
<eventState>
    <!--ro, req, enum, continuous event status, subType:string, desc:active (valid event); inactive (invalid event)-->active
</eventState>
<eventDescription>
    <!--ro, req, string, event description-->automatic incident detection
</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>
<AID>
    <!--ro, opt, object, information about traffic incident detection-->
<illegalTrafficEvent>
    <!--ro, req, enum, traffic violation events, subType:string, desc:abandonedObject (fallen object), checkPoint (checkpoint detection), conflagration (fire), congestion, construction, crosslane (weaving out of the lane), edfManual (manual enforcement), fogDetection (fog detection), gasser (queue jumping), group (people gathering), illegalParking (illegal parking), intersectionStranded (vehicle overstay), LaneChange (illegal lane change), objectDroppedDown (object falling down), occupyOvertakingLane (occupy overtaking lane), parallelParking (parallel parking), pedestrian (pedestrian detection), polyJam (polygonal congestion area), prohibitionMarkViolation (prohibition violation), roadBlock (roadblock), slowMoving (slow moving), smoke, speed (speeding), suddenSpeedDrop (sudden speed drop), trafficAccident (traffic incident detection), trafficConflict (traffic flow conflict), turnRound (illegal U-turn), unknown, vehicleExist (vehicle on non-motor vehicle lane), wrongDirection (wrong-way driving), singleVehicleBreakdown (single vehicle breakdown, the event that one vehicle stops due to breakdown and a person gets off the vehicle to check the vehicle), vehNoYieldPedest (motor vehicles failing to yield for pedestrians), illegalMannedVeh (illegal manned motor vehicle), illegalMannedNonMotorVeh (illegal manned non-motor vehicle), umbrellaTentInstall (umbrella tent installation violation on non-motor vehicle), nonMotorVehOnVehLane (non-motor vehicle on motor vehicle lane), wearingNoHelmet (wearing no helmet), pedestrianRedLightRunning (pedestrian red light running), pedestOnNonMotorVehLane (pedestrian walking on non-motor vehicle lane), pedestOnVehLane (pedestrian walking on motor vehicle lane)-->abandonedObject
</illegalTrafficEvent>
<VehicleInfo>
    <!--ro, opt, object, vehicle information-->
<vehicleSpeed>
    <!--ro, opt, int, vehicle speed, range:[0,255], unit:km/h-->1
</vehicleSpeed>
<vehicleEnterState>
    <!--ro, opt, enum, vehicle entering status, subType:string, desc:vehicleEnter (vehicle entered), vehicleExit (vehicle exited)-->vehicleEnter
</vehicleEnterState>
<vehicleConfidence>
    <!--ro, opt, float, vehicle confidence, range:[0.0,100.0]-->0.0
</vehicleConfidence>
<vehicleType>
    <!--ro, req, enum, vehicle type, subType:string, desc:SUVMPV (SUV/MPV), buggy (small-sized truck), bus, concreteMixer (concrete mixer), containerTruck (container truck), coupe, crane, hatchback, LargeBus (large-sized bus), LightTruck (light truck), mediumBus (middle-sized bus), mediumHeavyTruck (medium and heavy truck), miniCar (mini sedan (it will be converted to vehicle)), minibus, minitruck, motorVehicle (motor vehicle (it will be converted to vehicle)), normotorVehicle (non-motor vehicle (it will be converted to threeWheelVehicle)), oilTankTruck (oil tank truck), pedestrian, pickupTruck (pickup truck (it will be converted to buggy)), platformTrailer (platform trailer), saloon, slagTruck (dump truck), smallCar (small sedan (it will be converted to vehicle)), threeWheelVehicle (tricycle), truck, twoWheelVehicle (two wheeler), unknown, van, vehicle (sedan)-->SUVMPV
</vehicleType>
<vehicleColor>
    <!--ro, req, enum, vehicle color, subType:string, desc:black, blue, brown, cyan, deepBlue (dark blue), deepGray (dark gray), gray, green, orange, pink, purple, red, silver, unknown, white, yellow-->black
</vehicleColor>
</VehicleInfo>
<PlateInfo>
    <!--ro, opt, object, license plate information-->
<plate>
    <!--ro, req, string, license plate number, desc:noPlate (vehicle without license plate), unknown (no license plate recognized), XXXXXXXX (license plate number. For motor vehicles, it is a 16-byte string; for non-motor vehicles, it is a 48-byte string)-->蓝浙A12345
</plate>
<plateType>
    <!--ro, req, enum, license plate type, subType:string, desc:02TypePersonalized (type 02 custom vehicle), 92FarmVehicle (two-line license plate civil vehicle), 92TypeArm (type 92 armed police vehicle), 92TypeCivil (type 92 civil vehicle), arm (police vehicle), civilAviation (civil aviation license plate), coach (driver-training vehicle), consulate (consular vehicle), embassy (embassy vehicle), emergency (emergency license plate), green1325FarmVehicle (agricultural vehicle (green 1325)), hongKongMacao (Hong Kong/Macao entrance vehicle), motorola (motorcycle), newEnergy (new energy vehicle license plate), oneLineArm (new armed police vehicle (one-line)), oneLineArmHeadquarters (armed police headquarter license plate (one-line)), tempEntry (temporary entry car), tempTravl (temporary license plate car), trailer, twoLineArm (new armed police vehicle (two-line)), twoLineArmHeadquarters (armed police headquarter license plate (two-line)), unknown, yellow1225FarmVehicle (agricultural vehicle (yellow 1225)), yellow1325FarmVehicle (agricultural vehicle (yellow 1325)), yellowTwoLine (yellow two-line rear license plate)-->02TypePersonalized
</plateType>
<plateColor>
    <!--ro, req, enum, license plate color, subType:string, desc:black, blue, civilAviationBlack (civil aviation black), civilAviationGreen (civil aviation green), golden, green, mixedColor (mixed color), newEnergyGreen (new energy green), newEnergyYellowGreen (new energy green and yellow), orange, other (other color), red, unknown, white, yellow-->black
</plateColor>
<licenseBright>
    <!--ro, opt, int, license plate brightness, range:[0,255]-->1
</licenseBright>

```

```

</confidence>
<!--ro, opt, float, License plate confidence, range:[0.0,100.0]-->0.0
</confidence>
<plateCharacterConfidenceList>
<!--ro, opt, array, subType:object-->
<plateCharacterConfidence>
<!--ro, opt, float, range:[0.00,100.00], step:0.01-->0.01
</plateCharacterConfidence>
</plateCharacterConfidenceList>
<comprehensiveEvaluation>
<!--ro, opt, int, range:[0,1000]-->1
</comprehensiveEvaluation>
</PlateInfo>
<region>
<!--ro, opt, enum, region, subType:string, desc:AFandAM (Africa and America), APAC (Asia-Pacific Region), ALL (all regions), ER (Russian Region), EU (Europe Region), EUandCIS (Europe and Russia), HKandMO (Hong Kong and Macao China), ME (Middle East), THAandLA (Thailand and Laos), other (other)-->AFandAM
</region>
<country>
<!--ro, opt, enum, country or region, subType:string, desc:0 (Unsupported. It indicates that the algorithm library does not support recognizing license plates of this country), 1 (Czech Republic), 10 (Moldova), 100 (Reserved), 101 (Reserved), 102 (Reserved), 103 (Reserved), 104 (Egypt), 105 (Libya), 106 (Sudan), 107 (Tunisia), 108 (Algeria), 109 (Morocco), 11 (Russia), 110 (Ethiopia), 111 (Eritrea), 112 (Somalia Democratic), 113 (Djibouti), 114 (Kenya), 115 (Tanzania), 116 (Uganda), 117 (Rwanda), 118 (Burundi), 119 (Seychelles), 12 (Ukraine), 120 (Chad), 121 (Central African), 122 (Cameroon), 123 (Equatorial Guinea), 124 (Gabon), 125 (Republic of the Congo (Congo-Brazzaville)), 126 (Democratic Republic of the Congo (Congo-Kinshasa)), 127 (São Tomé and Príncipe), 128 (Mauritania), 129 (Western Sahara (Saharaui)), 13 (Belgium), 130 (Senegal), 131 (Gambia), 132 (Mali), 133 (Burkina Faso), 134 (Guinea), 135 (Guinea-Bissau), 136 (Cape Verde), 137 (Sierra Leone), 138 (Liberia), 139 (Ivory Coast), 14 (Bulgaria), 140 (Ghana), 141 (Togo), 142 (Benin), 143 (Niger), 144 (Zambia), 145 (Angola), 146 (Zimbabwe), 147 (Malawi), 148 (Mozambique), 149 (Botswana), 15 (Denmark), 150 (Namibia), 151 (South Africa), 152 (Swaziland), 153 (Lesotho), 154 (Madagascar), 155 (Comoros), 156 (Mauritius), 157 (Nigeria), 158 (South Sudan), 159 (Saint Helena), 16 (Finland), 160 (Mayotte), 161 (Reunion), 162 (Canary Islands), 163 (AZORES), 164 (Madeira), 165 (Reserved), 166 (Reserved), 167 (Reserved), 168 (Reserved), 169 (Canada), 17 (United Kingdom Great Britain), 170 (Greenland), 171 (St. Pierre and Miquelon), 172 (United States), 173 (Bermuda), 174 (Mexico), 175 (Guatemala), 176 (Belize), 177 (El Salvador), 178 (Honduras), 179 (Nicaragua), 18 (Greece), 180 (Costa Rica), 181 (Panama), 182 (Bahamas), 183 (Turks and Caicos Islands), 184 (Cuba), 185 (Jamaica), 186 (Cayman Islands), 187 (Haiti), 188 (Dominican), 189 (Puerto Rico), 19 (Croatia), 190 (United States Virgin Islands), 191 (British Virgin Islands), 192 (Anguilla), 193 (Antigua and Barbuda), 194 (Collectivité de Saint-Martin), 195 (Autonomous country), 196 (Saint-Barthélemy), 197 (Saint Kitts and Nevis), 198 (Montserrat), 199 (Guadeloupe), 2 (France), 20 (Hungary), 200 (Dominica), 201 (Martinique), 202 (St. Lucia), 203 (Saint Vincent and the Grenadines), 204 (Grenada), 205 (Barbados), 206 (Trinidad and Tobago), 207 (Curacao), 208 (Aruba), 209 (Netherlands Antilles), 21 (Israel), 210 (Colombia), 211 (Venezuela), 212 (Guyana), 213 (Suriname), 214 (French Guiana), 215 (Ecuador), 216 (Peru), 217 (Bolivia), 218 (Paraguay), 219 (Chile), 22 (Luxembourg), 220 (Brazil), 221 (Uruguay), 222 (Argentina), 223 (Reserved), 224 (Reserved), 225 (Reserved), 226 (Reserved), 227 (Australia), 228 (New Zealand ), 229 (Papua New Guinea), 23 (Macedonia, North Macedonia after 2018), 230 (Solomon Islands), 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), 24 (Norway), 240 (Fiji Islands), 241 (Tonga), 242 (Tuvalu), 243 (Wallis et Futuna), 244 (Samoa), 245 (Eastern Samoa), 246 (Tokelau), 247 (Niue), 248 (Cook Islands), 249 (French Polynesia), 25 (Portugal), 250 (Pitcairn Islands), 251 (Hawaii State), 252 (Reserved), 253 (invalid), 254 (Unrecognized), 255 (ALL), 26 (Romania), 27 (Serbia), 28 (Azerbaijan), 29 (Georgia), 3 (Germany), 30 (Kazakhstan), 31 (Lithuania), 32 (Turkmenistan), 33 (Uzbekistan), 34 (Latvia), 35 (Estonia), 36 (Albania), 37 (Austria), 38 (Bosnia and Herzegovina), 39 (Ireland), 4 (Spain), 40 (Iceland), 41 (Vatican), 42 (Malta), 43 (Sweden), 44 (Switzerland), 45 (Cyprus), 46 (Turkey), 47 (Slovenia), 48 (Montenegro), 49 (Kosovo), 5 (Italy), 50 (Andorra), 51 (Armenia), 52 (Monaco), 53 (Liechtenstein), 54 (San Marino), 55 (Reserved), 56 (Reserved), 57 (Reserved), 58 (Reserved), 59 (China), 6 (Netherlands), 60 (Bahrain), 61 (South Korea), 62 (Lebanon), 63 (Nepal), 64 (Thailand), 65 (Pakistan), 66 (United Arab Emirates), 67 (Bhutan), 68 (Oman), 69 (North Korea), 7 (Poland), 70 (Philippines), 71 (Cambodia), 72 (Qatar), 73 (Kyrgyzstan), 74 (Maldives), 75 (Malaysia), 76 (Mongolia), 77 (Saudi Arabia), 78 (Brunei), 79 (Laos), 8 (Slovakia), 80 (Japan), 81 (Turkey), 82 (Palestinian), 83 (Tajikistan), 84 (Kuwait), 85 (Syria), 86 (India), 87 (Indonesia), 88 (Afghanistan), 89 (Sri Lanka), 9 (Belarus), 90 (Iraq), 91 (Vietnam), 92 (Iran), 93 (Yemen), 94 (Jordan), 95 (Burma/Myanmar), 96 (Sikkim), 97 (Bangladesh), 98 (Singapore), 99 (Democratic Republic of Timor-Leste)-->0
</country>
<parkingSerialNo>
<!--ro, opt, string, parking serial number, range:[1,16], desc:This node is valid when the value of <illegalTrafficEvent> is parallelParking, and the related URI is /ISAPI/Traffic/channels/<ID>/AID/<SID>/parallelParking.-->1
</parkingSerialNo>
<relatedLaneNo>
<!--ro, opt, string, Linked Lane number-->test
</relatedLaneNo>
<targetType>
<!--ro, opt, enum, target type, subType:string, desc:unknown, invalidTarget (invalid target), largeSizedVehicle (large-sized vehicle), smallSizedVehicle (small-sized vehicle), vehicle, nonMotor (non-motor vehicle), motorcycle, pedestrian, nonQuadricycle (non-quadricycle), twoWheelVehicle (two wheeler), threeWheelVehicle (tricycle), face, triangleRack (triangle rack), bus, truck, smoke, fire, banner, stall, headandShoulder (head and shoulders), head, trafficSign (traffic sign (speed limit sign, weight limit sign, etc.)), trafficLight (traffic Light), laneLine (Lane Line), curb, roadSurfaceMarking (road marking), wheel, vehicleHead (vehicle head), vehicleTail (vehicle tail), vehicleFront (vehicle front), vehicleRear (vehicle rear), licensePlate (license plate), vehicleWindow (vehicle window), roadSurface (road surface), zebraCrossing (zebra crossing), generalObstacle (general obstacle), airPlane, UAV, ship, police, policeVehicle (police vehicle), constructionStaff (construction worker), constructionVehicle (construction vehicle)-->unknown
</targetType>
<CRIndex>
<!--ro, opt, enum, country/region, subType:int, desc:0 (Unsupported. It indicates that the algorithm library does not support recognizing license plates of this country), 1 (Czech Republic), 10 (Moldova), 100 (Reserved), 101 (Reserved), 102 (Reserved), 103 (Reserved), 104 (Egypt), 105 (Libya), 106 (Sudan), 107 (Tunisia), 108 (Algeria), 109 (Morocco), 11 (Russia), 110 (Ethiopia), 111 (Eritrea), 112 (Somalia Democratic), 113 (Djibouti), 114 (Kenya), 115 (Tanzania), 116 (Uganda), 117 (Rwanda), 118 (Burundi), 119 (Seychelles), 12 (Ukraine), 120 (Chad), 121 (Central African), 122 (Cameroon), 123 (Equatorial Guinea), 124 (Gabon), 125 (Republic of the Congo (Congo-Brazzaville)), 126 (Democratic Republic of the Congo (Congo-Kinshasa)), 127 (São Tomé and Príncipe), 128 (Mauritania), 129 (Western Sahara (Saharaui)), 13 (Belgium), 130 (Senegal), 131 (Gambia), 132 (Mali), 133 (Burkina Faso), 134 (Guinea), 135 (Guinea-Bissau), 136 (Cape Verde), 137 (Sierra Leone), 138 (Liberia), 139 (Ivory Coast), 14 (Bulgaria), 140 (Ghana), 141 (Togo), 142 (Benin), 143 (Niger), 144 (Zambia), 145 (Angola), 146 (Zimbabwe), 147 (Malawi), 148 (Mozambique), 149 (Botswana), 15 (Denmark), 150 (Namibia), 151 (South Africa), 152 (Swaziland), 153 (Lesotho), 154 (Madagascar), 155 (Comoros), 156 (Mauritius), 157 (Nigeria), 158 (South Sudan), 159 (Saint Helena), 16 (Finland), 160 (Mayotte), 161 (Reunion), 162 (Canary Islands), 163 (AZORES), 164 (Madeira), 165 (Reserved), 166 (Reserved), 167 (Reserved), 168 (Reserved), 169 (Canada), 17 (United Kingdom Great Britain), 170 (Greenland), 171 (St. Pierre and Miquelon), 172 (United States), 173 (Bermuda), 174 (Mexico), 175 (Guatemala), 176 (Belize), 177 (El Salvador), 178 (Honduras), 179 (Nicaragua), 18 (Greece), 180 (Costa Rica), 181 (Panama), 182 (Bahamas), 183 (Turks and Caicos Islands), 184 (Cuba), 185 (Jamaica), 186 (Cayman Islands), 187 (Haiti), 188 (Dominican), 189 (Puerto Rico), 19 (Croatia), 190 (United States Virgin Islands), 191 (British Virgin Islands), 192 (Anguilla), 193 (Antigua and Barbuda), 194 (Collectivité de Saint-Martin), 195 (Autonomous country), 196 (Saint-Barthélemy), 197 (Saint Kitts and Nevis), 198 (Montserrat), 199 (Guadeloupe), 2 (France), 20 (Hungary), 200 (Dominica), 201 (Martinique), 202 (St. Lucia), 203 (Saint Vincent and the Grenadines), 204 (Grenada), 205 (Barbados), 206 (Trinidad and Tobago), 207 (Curacao), 208 (Aruba), 209 (Netherlands Antilles), 21 (Israel), 210 (Colombia), 211 (Venezuela), 212 (Guyana), 213 (Suriname), 214 (French Guiana), 215 (Ecuador), 216 (Peru), 217 (Bolivia), 218 (Paraguay), 219 (Chile), 22 (Luxembourg), 220 (Brazil), 221 (Uruguay), 222 (Argentina), 223 (Reserved), 224 (Reserved), 225 (Reserved), 226 (Reserved), 227 (Australia), 228 (New Zealand ), 229 (Papua New Guinea), 23 (Macedonia, North Macedonia after 2018), 230 (Solomon Islands), 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), 24 (Norway), 240 (Fiji Islands), 241 (Tonga), 242 (Tuvalu), 243 (Wallis et Futuna), 244 (Samoa), 245 (Eastern Samoa), 246 (Tokelau), 247 (Niue), 248 (Cook Islands), 249 (French Polynesia), 25 (Portugal), 250 (Pitcairn Islands), 251 (Hawaii State), 252 (Reserved), 253 (invalid), 254 (Unrecognized), 255 (ALL), 26 (Romania), 27 (Serbia), 28 (Azerbaijan), 29 (Georgia), 3 (Germany), 30 (Kazakhstan), 31 (Lithuania), 32 (Turkmenistan), 33 (Uzbekistan), 34 (Latvia), 35 (Estonia), 36 (Albania), 37 (Austria), 38 (Bosnia and Herzegovina), 39 (Ireland), 4 (Spain), 40 (Iceland), 41 (Vatican), 42 (Malta), 43 (Sweden), 44 (Switzerland), 45 (Cyprus), 46 (Turkey), 47 (Slovenia), 48 (Montenegro), 49 (Kosovo), 5 (Italy), 50 (Andorra), 51 (Armenia), 52 (Monaco), 53 (Liechtenstein), 54 (San Marino), 55 (Reserved), 56 (Reserved), 57 (Reserved), 58 (Reserved), 59 (China), 6 (Netherlands), 60 (Bahrain), 61 (South Korea), 62 (Lebanon), 63 (Nepal), 64
```

```

    (Thailand), 65 (Pakistan), 66 (United Arab Emirates), 67 (Bhutan), 68 (Oman), 69 (North Korea), 7 (Poland), 70 (Philippines), 71 (Cambodia), 72 (Qatar), 73
    (Kyrgyzstan), 74 (Maldives), 75 (Malaysia), 76 (Mongolia), 77 (Saudi Arabia), 78 (Brunei), 79 (Laos), 8 (Slovakia), 80 (Japan), 81 (Turkey), 82
    (Palestinian), 83 (Tajikistan), 84 (Kuwait), 85 (Syria), 86 (India), 87 (Indonesia), 88 (Afghanistan), 89 (Sri Lanka), 9 (Belarus), 90 (Iraq), 91 (Vietnam),
    92 (Iran), 93 (Yemen), 94 (Jordan), 95 (Burma/Myanmar), 96 (Sikkim), 97 (Bangladesh), 98 (Singapore), 99 (Democratic Republic of Timor-Leste)-->
</CRIndex>
<alarmLevel>
    <!--ro, opt, enum, alarm priority, subType:string, desc:alert (pre-alarm)-->alert
</alarmLevel>
<alertStartTime>
    <!--ro, opt, datetime, pre-alarm start time-->1970-01-01T00:00:00+08:00
</alertStartTime>
<alertEndTime>
    <!--ro, opt, datetime, pre-alarm end time-->1970-01-01T00:00:00+08:00
</alertEndTime>
<uid>
    <!--ro, opt, string, 64-byte unique identifier of VCA event uploading linkage, which can be generated by the time (it is accurate to millisecond) and
    a random number, range:[1,64], desc:The unique identifier indicates the uniqueness of the recognized target in a capture event.-->test
</uid>
<targetID>
    <!--ro, opt, int, ID of the captured target, desc:The ID indicates the uniqueness of the recognized target in a capture event.-->0
</targetID>
<laneNo>
    <!--ro, opt, int, number of the Lane where the target is located-->1
</laneNo>
<illegalType>
    <!--ro, opt, string, violation type, desc:N/A-->1001A
</illegalType>
<isOverSpeed>
    <!--ro, opt, bool, whether the vehicle is speeding-->true
</isOverSpeed>
<isRapidDeceleration>
    <!--ro, opt, bool, whether the vehicle is decelerating suddenly-->true
</isRapidDeceleration>
<illegalParkingEvent>
    <!--ro, opt, enum, illegal parking event of the vehicle, subType:string, desc:crossLane (parking over the Lane Line), crossBerth (occupying two
    parking spaces), crossLaneBerth (parking over the Lane Line and occupying two parking spaces), illegalParking (illegal parking)-->crossLane
</illegalParkingEvent>
<illegalTrafficEventSubType>
    <!--ro, opt, enum, event sub-type, subType:string, desc:passengers, abandonedObject (fallen object), construction, trafficAccident (traffic accident),
    roadBlock (roadBlock), unknown-->passengers
</illegalTrafficEventSubType>
<Position>
    <!--ro, opt, object, offset of the target center relative to the camera (unit: meter), desc:An X-coordinate is positive if it is to the right of the
    Y-axis and negative if it is to the left of the Y-axis. A Y-coordinate is positive if it is above of the X-axis and negative values are not supported. The
    origin is the center point (the center point of all Lane Lines) of the horizontal Line where the camera is located and which is perpendicular to the Lane
    Line.-->
    <x>
        <!--ro, req, float, X-coordinate-->0.000
    </x>
    <y>
        <!--ro, req, float, Y-coordinate-->0.000
    </y>
</Position>
<AIDGPSInfo>
    <!--ro, opt, object, GPS information of the location where the AID event (except the congestion event) occurred-->
    <longitudeType>
        <!--ro, req, enum, longitude, subType:string, desc:E (eastern longitude), W (western longitude)-->E
    </longitudeType>
    <latitudeType>
        <!--ro, req, enum, latitude, subType:string, desc:S (southern Latitude), N (northern Latitude)-->S
    </latitudeType>
    <longitude>
        <!--ro, req, object, longitude-->
        <degree>
            <!--ro, req, int, degree-->1
        </degree>
        <minute>
            <!--ro, req, int, minute-->1
        </minute>
        <sec>
            <!--ro, req, float, description, desc:The value is accurate to six decimal places.-->0.000000
        </sec>
    </longitude>
    <latitude>
        <!--ro, req, object, latitude-->
        <degree>
            <!--ro, req, int, degree-->1
        </degree>
        <minute>
            <!--ro, req, int, minute-->1
        </minute>
        <sec>
            <!--ro, req, float, description, desc:The value is accurate to six decimal places.-->0.000000
        </sec>
    </latitude>
</AIDGPSInfo>
<CongestionHeadGPSInfo>
    <!--ro, opt, object-->
    <longitudeType>
        <!--ro, req, enum, subType:string-->E
    </longitudeType>
    <latitudeType>
        <!--ro, req, enum, subType:string-->S
    </latitudeType>

```

```

</latitudeType>
<Longitude>
  <!--ro, req, object-->
  <degree>
    <!--ro, req, int-->1
  </degree>
  <minute>
    <!--ro, req, int-->1
  </minute>
  <sec>
    <!--ro, req, float-->0.000000
  </sec>
</Longitude>
<Latitude>
  <!--ro, req, object-->
  <degree>
    <!--ro, req, int-->1
  </degree>
  <minute>
    <!--ro, req, int-->1
  </minute>
  <sec>
    <!--ro, req, float-->0.000000
  </sec>
</Latitude>
</CongestionHeadGPSInfo>
<CongestionTailGPSInfo>
  <!--ro, opt, object-->
  <longitudeType>
    <!--ro, req, enum, subType:string-->E
  </longitudeType>
  <latitudeType>
    <!--ro, req, enum, subType:string-->S
  </latitudeType>
  <Longitude>
    <!--ro, req, object-->
    <degree>
      <!--ro, req, int-->1
    </degree>
    <minute>
      <!--ro, req, int-->1
    </minute>
    <sec>
      <!--ro, req, float-->0.000000
    </sec>
  </Longitude>
  <Latitude>
    <!--ro, req, object-->
    <degree>
      <!--ro, req, int-->1
    </degree>
    <minute>
      <!--ro, req, int-->1
    </minute>
    <sec>
      <!--ro, req, float-->0.000000
    </sec>
  </Latitude>
</CongestionTailGPSInfo>
<CongestionLaneGPSInfo>
  <!--ro, opt, object, GPS location information of the center point of the lane in which the congestion event occurred-->
  <longitudeType>
    <!--ro, req, enum, longitude, subType:string, desc:E (eastern Longitude), W (western Longitude)-->E
  </longitudeType>
  <latitudeType>
    <!--ro, req, enum, latitude, subType:string, desc:S (southern Latitude), N (northern Latitude)-->S
  </latitudeType>
  <Longitude>
    <!--ro, req, object, longitude-->
    <degree>
      <!--ro, req, int, degree-->1
    </degree>
    <minute>
      <!--ro, req, int, minute-->1
    </minute>
    <sec>
      <!--ro, req, float, description, desc:The value is accurate to six decimal places.-->0.000000
    </sec>
  </Longitude>
  <Latitude>
    <!--ro, req, object, latitude-->
    <degree>
      <!--ro, req, int, degree-->1
    </degree>
    <minute>
      <!--ro, req, int, minute-->1
    </minute>
    <sec>
      <!--ro, req, float, description, desc:The value is accurate to six decimal places.-->0.000000
    </sec>
  </Latitude>
</CongestionLaneGPSInfo>
<visibility>
  <!--ro, opt, int, visibility in thick fog, range:[0,30000], unit:m, dep:and,{$.EventNotificationAlert.AID.illegalTrafficEvent,eq,fogDetection}-->30

```

```

</visibility>
<RuleInfo>
    <!--ro, opt, object, AID rule information-->
    <ruleID>
        <!--ro, opt, int, ID of the rule according to which the event is detected, range:[1,16]-->1
    </ruleID>
    <ruleName>
        <!--ro, opt, string, name of the rule according to which the event is detected, range:[1,32]-->name
    </ruleName>
</RuleInfo>
<DirectPosition>
    <!--ro, opt, object, direction Line of alarm detection, desc:The origin is the upper-left corner of the screen.-->
    <StartPoint>
        <!--ro, opt, object, coordinates of the start point-->
        <x>
            <!--ro, req, int, X-coordinate, range:[0,1000]-->1
        </x>
        <y>
            <!--ro, req, int, Y-coordinate, range:[0,1000]-->1
        </y>
    </StartPoint>
    <EndPoint>
        <!--ro, opt, object, coordinates of the end point-->
        <x>
            <!--ro, req, int, X-coordinate, range:[0,1000]-->1
        </x>
        <y>
            <!--ro, req, int, Y-coordinate, range:[0,1000]-->1
        </y>
    </EndPoint>
</DirectPosition>
<speedLimit>
    <!--ro, opt, int, current speed Limit, range:[0,255], unit:km/h-->100
</speedLimit>
<state>
    <!--ro, opt, enum, event uploading status, subType:int, desc:0 (upload changes), 1 (upload auto-switch), 2 (upload the data captured by the current device by schedule). The value 2 is valid when the value of dwAIDType (alarm/event type) is the parking event.-->0
    <state>
    <fogUploadMode>
        <!--ro, opt, enum, method of uploading the thick fog event, subType:string, dep:and,
        ${.EventNotificationAlert.AID.illegalTrafficEvent,eq,fogDetection}, desc:trigger (upload the alarm when it is triggered), timing (upload the alarm by schedule). If this node is not returned, the default value is trigger.-->trigger
        </fogUploadMode>
    <trafficAccidentInfoList>
        <!--ro, opt, array, subType:object, dep:or, ${.EventNotificationAlert.TFS.illegalTrafficEvent,eq,trafficAccident}-->
        <trafficAccidentInfo>
            <!--ro, opt, object-->
            <PlateInfo>
                <!--ro, opt, object-->
                <plate>
                    <!--ro, req, string-->蓝浙A12345
                </plate>
                <plateType>
                    <!--ro, req, enum, subType:string-->02TypePersonalized
                </plateType>
                <plateColor>
                    <!--ro, req, enum, subType:string-->black
                </plateColor>
            </PlateInfo>
            <trafficAccidentStartTime>
                <!--ro, opt, datetime-->2004-05-03T17:30:08+08:00
            </trafficAccidentStartTime>
            <trafficAccidentGPSInfo>
                <!--ro, opt, object-->
                <longitudeType>
                    <!--ro, req, enum, subType:string-->E
                </longitudeType>
                <latitudeType>
                    <!--ro, req, enum, subType:string-->S
                </latitudeType>
                <Longitude>
                    <!--ro, req, object-->
                    <degree>
                        <!--ro, req, int-->1
                    </degree>
                    <minute>
                        <!--ro, req, int-->1
                    </minute>
                    <sec>
                        <!--ro, req, float-->0.000000
                    </sec>
                </Longitude>
                <Latitude>
                    <!--ro, req, object-->
                    <degree>
                        <!--ro, req, int-->1
                    </degree>
                    <minute>
                        <!--ro, req, int-->1
                    </minute>
                    <sec>
                        <!--ro, req, float-->0.000000
                    </sec>
                </Latitude>
            </trafficAccidentGPSInfo>
        </trafficAccidentInfo>
    </trafficAccidentInfoList>

```

```
</Latitude>
</trafficAccidentGPSInfo>
<targetMovementInfoList>
  <!--ro, opt, array, subType:object-->
  <targetMovementInfo>
    <!--ro, opt, object-->
    <longitudeType>
      <!--ro, req, enum, subType:string-->E
    </longitudeType>
    <latitudeType>
      <!--ro, req, enum, subType:string-->S
    </latitudeType>
    <Longitude>
      <!--ro, req, object-->
      <degree>
        <!--ro, req, int-->1
      </degree>
      <minute>
        <!--ro, req, int-->1
      </minute>
      <sec>
        <!--ro, req, float-->0.00000
      </sec>
    </Longitude>
    <Latitude>
      <!--ro, req, object-->
      <degree>
        <!--ro, req, int-->1
      </degree>
      <minute>
        <!--ro, req, int-->1
      </minute>
      <sec>
        <!--ro, req, float-->0.00000
      </sec>
    </Latitude>
    <targetMovementDateTime>
      <!--ro, opt, datetime-->2004-05-03T17:30:08+08:00
    </targetMovementDateTime>
  </targetMovementInfo>
</targetMovementInfoList>
</trafficAccidentInfo>
</trafficAccidentInfoList>
<alarmDistance>
  <!--ro, opt, int, range:[20,220], unit:cm-->20
</alarmDistance>
<lowSpeedLimit>
  <!--ro, opt, int, range:[20,200], unit:km/h-->20
</lowSpeedLimit>
<averageSpeed>
  <!--ro, opt, float, range:[0,200], unit:km/h-->60.0
</averageSpeed>
<congestionLevel>
  <!--ro, opt, int, range:[1,3)-->3
</congestionLevel>
<congestionQueueLen>
  <!--ro, opt, int, range:[0,10000], unit:m-->10
</congestionQueueLen>
<visibilityLevel>
  <!--ro, opt, int, range:[0,4)-->3
</visibilityLevel>
<abandonedObjectVehicleProportion>
  <!--ro, opt, float, range:[0.00,10.00)-->0.500
</abandonedObjectVehicleProportion>
<relateObjectInfoList>
  <!--ro, opt, array, subType:object, range:[0,32)-->
  <relateObjectInfo>
    <!--ro, opt, object-->
    <targetType>
      <!--ro, req, enum, subType:string-->triangleRack
    </targetType>
    <objectRect>
      <!--ro, opt, object-->
      <X>
        <!--ro, req, int, range:[0,1000)-->1
      </X>
      <Y>
        <!--ro, req, int, range:[0,1000)-->1
      </Y>
      <width>
        <!--ro, req, int, range:[0,1000)-->1
      </width>
      <height>
        <!--ro, req, int, range:[0,1000)-->1
      </height>
    </objectRect>
    <targetID>
      <!--ro, opt, int, range:[1,1000000)-->1
    </targetID>
  </relateObjectInfo>
</relateObjectInfoList>
<multiTargetList>
  <!--ro, opt, array, subType:object, range:[0,8)-->
  <multiTarget>
```

```

----->
<!--ro, opt, object-->
<targetType>
  <!--ro, opt, enum, subType:string-->unknown
</targetType>
</multiTarget>
</multiTargetList>
</AID>
<detectionPictureTransType>
  <!--ro, opt, enum, picture transmission method, subType:string, desc:url (picture URL), binary (binary picture data)-->url
</detectionPictureTransType>
<detectionPicturesNumber>
  <!--ro, opt, int, number of pictures contained in the alarm received in arming mode-->1
</detectionPicturesNumber>
<contentIDSence1>
  <!--ro, opt, string, scene picture ID of the vehicle (it indicates the uniqueness of the picture data), range:[1,32], desc:The generation rule is the device's serial number + the time elapsed since the device was started + a random number-->D6A18E41308CE9934DCDDD1111
</contentIDSence1>
<contentIDSence2>
  <!--ro, opt, string, scene picture ID of the vehicle (it indicates the uniqueness of the picture data), range:[1,32], desc:The generation rule is the device's serial number + the time elapsed since the device was started + a random number-->D6A18E41308CE9934DCDDD1111
</contentIDSence2>
<contentIDSence3>
  <!--ro, opt, string, scene picture ID of the vehicle (it indicates the uniqueness of the picture data), range:[1,32], desc:The generation rule is the device's serial number + the time elapsed since the device was started + a random number-->D6A18E41308CE9934DCDDD1111
</contentIDSence3>
<contentIDVehiclePlate>
  <!--ro, opt, string, vehicle picture ID (it indicates the uniqueness of the picture data), range:[1,32], desc:The generation rule is the device's serial number + the time elapsed since the device was started + a random number-->D6A18E41308CE9934DCDDD2222
</contentIDVehiclePlate>
<contentIDComposite>
  <!--ro, opt, string, composite vehicle picture (it indicates the uniqueness of the picture data), range:[1,32], desc:The generation rule is the device's serial number + the time elapsed since the device was started + a random number-->D6A18E41308CE9934DCDDD2222
</contentIDComposite>
<PictureURLInfoList>
  <!--ro, opt, array, list of picture URL information, subType:object, desc:This node is returned when the value of <detectionPictureTransType> is url.-->
<PictureURLInfo>
  <!--ro, opt, object, picture URL information-->
<pictureName>
  <!--ro, req, enum, picture name, subType:string, desc:detectionPicture (captured background picture), licensePlatePicture (license plate close-up)-->
</detectionPicture>
</pictureName>
<pictureURL>
  <!--ro, req, string, picture URL-->http://127.0.0.1
</pictureURL>
<plateRect>
  <!--ro, opt, object, coordinates of the license plate frame, desc:This node is valid when the value of <type> is detectionPicture. The origin is the upper-left corner of the screen.-->
<x>
  <!--ro, req, int, X-coordinate of the upper-left corner of the boundary frame, range:[0,1000]-->1
</x>
<y>
  <!--ro, req, int, Y-coordinate of the upper-left corner of the boundary frame, range:[0,1000]-->1
</y>
<width>
  <!--ro, req, int, width of the boundary frame, range:[0,1000]-->1
</width>
<height>
  <!--ro, req, int, height of the boundary frame, range:[0,1000]-->1
</height>
</plateRect>
<objectInfo>
  <!--ro, opt, object, target information-->
<targetType>
  <!--ro, req, enum, target type, subType:string, desc:unknown, invalidTarget (invalid target), largeSizedVehicle (large-sized vehicle), smallSizedVehicle (small-sized vehicle), vehicle, nonMotor (non-motor vehicle), motorcycle, pedestrian, nonQuadricycle (non-quadricycle), twoWheelVehicle (two wheeler), threeWheelVehicle (tricycle), face, triangleRack (triangle rack), bus, truck, smoke, fire, banner, stall, headAndShoulder (head and shoulders), head, trafficSign (traffic sign (speed limit sign, weight limit sign, etc.)), trafficLight (traffic light), laneLine (lane line), curb, roadSurfaceMarking (road marking), wheel, vehicleHead (vehicle head), vehicleTail (vehicle tail), vehicleFront (vehicle front), vehicleRear (vehicle rear), licensePlate (license plate), vehicleWindow (vehicle window), roadSurface (road surface), zebraCrossing (zebra crossing), generalObstacle (general obstacle), airplane, UAV, ship-->unknown
</targetType>
<confidence>
  <!--ro, opt, float, confidence level, range:[0.00,100.00]-->0.00
</confidence>
<objectRect>
  <!--ro, opt, object, coordinates of the target, desc:The origin is the upper-left corner of the screen.-->
<x>
  <!--ro, req, int, X-coordinate of the upper-left corner of the boundary frame, range:[0,1000]-->1
</x>
<y>
  <!--ro, req, int, Y-coordinate of the upper-left corner of the boundary frame, range:[0,1000]-->1
</y>
<width>
  <!--ro, req, int, width of the boundary frame, range:[0,1000]-->1
</width>
<height>
  <!--ro, req, int, height of the boundary frame, range:[0,1000]-->1
</height>
</objectRect>
</objectInfo>
</PictureURLInfo>
</PictureURLInfoList>
<UUID>

```

```

<!--ro, opt, string, UUID, desc:If the vehicle is captured many times, the UUID is the same.-->test
</UUID>
<targetAttrs>
    <!--ro, opt, string, target attributes. This node is only supported by data center products and will be transparently transmitted., desc:target
    attributes. This node is only supported by data center products and will be transparently transmitted.-->{"deviceId": "1","deviceChannel": 1,"deviceName": "shebei"}
</targetAttrs>
<monitoringSiteID>
    <!--ro, opt, string, camera number-->test
</monitoringSiteID>
<monitorDescription>
    <!--ro, opt, string, camera information-->test
</monitorDescription>
<SceneInfo>
    <!--ro, opt, object, scene information-->
<scenesID>
    <!--ro, opt, string, scene ID, range:[1,16]-->test
</scenesID>
<sceneName>
    <!--ro, opt, string, scene name, range:[0,128]-->test
</sceneName>
<PTZPos>
    <!--ro, opt, object, PTZ position-->
<elevation>
    <!--ro, opt, int, elevation, range:[-900,2700]-->1
</elevation>
<azimuth>
    <!--ro, opt, int, azimuth, range:[0,3600]-->1
</azimuth>
<absoluteZoom>
    <!--ro, opt, int, zooming ratio, range:[1,1000]-->1
</absoluteZoom>
</PTZPos>
<relatedAreaNo>
    <!--ro, opt, string, Linked area number, range:[1,8], desc:This number is generated by calling /ISAPI/Traffic/channels/<channelID>/eventRule/<SID> and
    corresponds to <ruleId>. The URI for configuring congestion event parameters is /ISAPI/Traffic/channels/<channelID>/AID/<SID>/playCongestion in which the
    Linked area number (<relatedAreaNo>) needs to be configured.-->1
</relatedAreaNo>
<RelatedRegionCoordinatesList>
    <!--ro, opt, array, coordinate List of Linked areas, subType:object, range:[3,10], desc:This List contains coordinates of areas in <relatedAreaNo>.-->
<RegionCoordinates>
    <!--ro, opt, object, coordinates of the area, desc:The origin is the upper-left corner of the screen.-->
<positionX>
    <!--ro, req, float, X-coordinate, range:[0.000,1.000]-->0.000
</positionX>
<positionY>
    <!--ro, req, float, Y-coordinate, range:[0.000,1.000]-->0.000
</positionY>
</RegionCoordinates>
</RelatedRegionCoordinatesList>
<regionType>
    <!--ro, opt, enum, subType:string-->normal
</regionType>
<regionName>
    <!--ro, opt, string, range:[0,128]-->test
</regionName>
<SceneInfo>
<detectDir>
    <!--ro, opt, enum, detection direction, subType:int, desc:1 (upward), 2 (downward), 3 (bidirection), 4 (from east to west), 5 (from south to north), 6
    (from west to east), 7 (from north to south), 8 (other)-->1
</detectDir>
<dateMillisecondTime>
    <!--ro, opt, datetime, time when the alarm is triggered. The time can be accurate to millisecond.-->2004-05-03T17:30:08.000+08:00
</dateMillisecondTime>
<deviceUUID>
    <!--ro, opt, string, device number, 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>
<vehiclepicture>
    <!--ro, opt, string, The generation rule is the device's serial number + the time elapsed since the device was started + a random number, range:[1,32],
    desc:The generation rule is the device's serial number + the time elapsed since the device was started + a random number-->D6A18E41308CE9934DCDD01111
</vehiclepicture>
<batteryInfo>
    <!--ro, opt, object-->
<batterStatus>
    <!--ro, opt, enum, subType:string-->normal
</batterStatus>
<batteryPower>
    <!--ro, opt, int, range:[0,100], step:1, unit:%, dep:and,{$.EventNotificationAlert.batteryInfo.batterStatus,eq,normal}-->1
</batteryPower>
</batteryInfo>
<isAlert>
    <!--ro, opt, bool-->true
</isAlert>
<binaryPictureObjectInfo>
    <!--ro, opt, object, dep:and,{$.EventNotificationAlert.detectionPictureTransType,eq,binary}-->
<targetType>
    <!--ro, req, enum, subType:string-->unknown
</targetType>
<confidence>
    <!--ro, opt, float, range:[0.00,100.00]-->0.00
</confidence>
<objectRect>
    <!--ro, opt, object-->

```

```
<!--ro, opt, object-->
<X>
  <!--ro, req, int, range:[0,1000]-->1
</X>
<Y>
  <!--ro, req, int, range:[0,1000]-->1
</Y>
<width>
  <!--ro, req, int, range:[0,1000]-->1
</width>
<height>
  <!--ro, req, int, range:[0,1000]-->1
</height>
</objectRect>
</binaryPictureObjectInfo>
<AIDBackgroundImageResolution>
  <!--ro, opt, object-->
  <height>
    <!--ro, req, int, unit:px-->1
  </height>
  <width>
    <!--ro, req, int, unit:px-->1
  </width>
</AIDBackgroundImageResolution>
<plateInfoList>
  <!--ro, opt, object-->
  <plateInfo>
    <!--ro, opt, object-->
    <plateRect>
      <!--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>
    </plateRect>
    <plateColor>
      <!--ro, opt, enum, subType:string-->black
    </plateColor>
    <plateNo>
      <!--ro, req, string, range:[1,32]-->蓝浙A12345
    </plateNo>
    <confidence>
      <!--ro, opt, float, range:[0.0,100.0]-->0.0
    </confidence>
    <province>
      <!--ro, opt, enum, subType:int-->0
    </province>
  </plateInfo>
</plateInfoList>
<isDataRetransmission>
  <!--ro, opt, bool-->true
</isDataRetransmission>
<ultrasonicRawData>
  <!--ro, opt, string-->test
</ultrasonicRawData>
<relaLaneDirectionType>
  <!--ro, opt, enum, subType:int-->1
</relaLaneDirectionType>
<regionTrackDir>
  <!--ro, opt, enum, subType:string-->leftToRight
</regionTrackDir>
<congestionVehicleNum>
  <!--ro, opt, int, range:[0,10000]-->100
</congestionVehicleNum>
<congestionOverTimeVehicleNum>
  <!--ro, opt, int, range:[0,10000]-->80
</congestionOverTimeVehicleNum>
</EventNotificationAlert>
```

Parameter Name	Parameter Value	Parameter Type(Content-Type)	Content-ID	File Name	Description
AID	[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	compositePicture	compositePicture.jpg	--
.value	[Binary picture data]	image/jpeg	vehiclePicture	vehiclePicture.jpg	--
.value	[Binary picture data]	image/jpeg	contentIDSence1	contentIDSence1.jpg	--
.value	[Binary picture data]	image/jpeg	contentIDSence2	contentIDSence2.jpg	--
.value	[Binary picture data]	image/jpeg	contentIDSence3	contentIDSence3.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.

12.6.4 Non-Motor Vehicle Detection by Video

12.6.4.1 Get the capability of non-motor vehicle control parameters

Request URL

GET /ISAPI/ITC/TriggerMode/nonMotorCon/capabilities

Query Parameter

None

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<NonMotorCon xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, non-motor vehicle control parameters, attr:version{req, string, protocolVersion}-->
  <enable>
    <!--ro, req, bool, whether to enable the function-->true
  </enable>
  <stackTargetEnable>
    <!--ro, req, bool, overlay target frame (on the pedestrian running the red Light at the first capture)-->true
  </stackTargetEnable>
  <post>
    <!--ro, req, bool, checkpoint-->true
  </post>
  <reverse>
    <!--ro, req, bool, converse running-->true
  </reverse>
  <reverseCapNo>
    <!--ro, req, int, captured picture amount of converse running, range:[2,3]-->2
  </reverseCapNo>
  <TriggerLineList size="3">
    <!--ro, opt, array, triggering line list, subType:object, attr:size{req, int}-->
    <TriggerLine>
      <!--ro, req, object, triggering line-->
      <lineName opt="triggerLine">
        <!--ro, req, enum, triggering line, subType:string, attr:opt{req, string}, desc:"triggerLine" (triggering line)-->triggerLine
      </lineName>
      <lineType opt="unknown,white,singleYellow,doubleYellow,guardRail,noCross">
        <!--ro, req, enum, Line type, subType:string, attr:opt{req, string}, desc:"unknown", "white" (solid white line), "singleYellow" (single yellow line), "doubleYellow" (double yellow line), "guardRail" (Lane line with guardrail), "noCross" (Lane line that cannot be crossed)-->unknown
      </lineType>
      <TriggerLineCoordinatesList size="2">
        <!--ro, opt, array, triggering line coordinate list, subType:object, attr:size{req, int}, desc:supports drawing multiple triggering lines (subject to the capability) currently max. 3 lines are supported-->
        <TriggerLineCoordinates>
          <!--ro, opt, object, triggering line coordinates, desc:the origin is the upper-left corner of the screen-->
          <positionX>
            <!--ro, req, int, X-coordinate, range:[0,2]-->0
          </positionX>
          <positionY>
            <!--ro, req, int, Y-coordinate, range:[0,2]-->0
          </positionY>
        </TriggerLineCoordinates>
      </TriggerLineCoordinatesList>
    </TriggerLine>
  </TriggerLineList>
</NonMotorCon>

```

12.6.5 Road Pre-Alarm

12.6.5.1 Get the voice prompt parameters

Request URL

GET /ISAPI/ITC/TriggerMode/roadWarning/voiceBroadcast?format=json

Query Parameter

None

Request Message

None

Response Message

```
{
    "VoiceBroadcastList": [
        /*ro, opt, array, subType:object*/
        {
            "screenID": 1,
            /*ro, opt, int, range:[1,2]*/
            "enabled": true,
            /*ro, req, bool*/
            "volume": 50,
            /*ro, opt, int, range:[0,100]*/
            "durationTime": 30,
            /*ro, opt, int, range:[20,100], unit:s*/
            "TimeBlockList": [
                /*ro, opt, array, subType:object*/
                {
                    "dayOfWeek": 1,
                    /*ro, opt, int, range:[1,7]*/
                    "TimeRange": {
                        /*ro, req, object*/
                        "beginTime": "10:00:00",
                        /*ro, req, time*/
                        "endTime": "12:00:00"
                        /*ro, req, time*/
                    },
                    "volume": 50
                    /*ro, opt, int, range:[0,100]*/
                }
            ]
        }
    ]
}
```

12.6.5.2 Set the voice prompt parameters

Request URL

PUT /ISAPI/ITC/TriggerMode/roadWarning/voiceBroadcast?format=json

Query Parameter

None

Request Message

```
{
    "VoiceBroadcastList": [
        /*opt, array, subType:object*/
        {
            "screenID": 1,
            /*opt, int, range:[1,2]*/
            "enabled": true,
            /*req, bool*/
            "volume": 50,
            /*opt, int, range:[0,100]*/
            "durationTime": 30,
            /*opt, int, range:[20,100], unit:s*/
            "TimeBlockList": [
                /*opt, array, subType:object*/
                {
                    "dayOfWeek": 1,
                    /*opt, int, range:[1,7]*/
                    "TimeRange": {
                        /*req, object*/
                        "beginTime": "10:00:00",
                        /*req, time*/
                        "endTime": "12:00:00"
                        /*req, time*/
                    },
                    "volume": 50
                    /*opt, int, range:[0,100]*/
                }
            ]
        }
    ]
}
```

Response Message

```
{
    "statusCode": 1,
    /*ro, opt, int*/
    "statusString": "ok",
    /*ro, opt, string, range:[1,64]*/
    "subStatusCode": "ok",
    /*ro, opt, string, range:[1,64]*/
    "errorCode": 1,
    /*ro, opt, int*/
    "errorMsg": "ok"
    /*ro, opt, string*/
}
```

12.7 Video Recognition

12.7.1 VCA Abnormal Event Recognition

12.7.1.1 Get the intelligent capability of device

Request URL

GET /ISAPI/Intelligent/channels/<channelID>/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<IntelliCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, attr:version{req, string, protocolVersion}-->
    <isFaceSupport>
        <!--ro, req, bool, whether it supports face detection, desc:whether it supports face detection-->true
    </isFaceSupport>
    <isFieldDetectionSupport>
        <!--ro, req, bool, whether it supports intrusion detection, desc:whether it supports intrusion detection-->true
    </isFieldDetectionSupport>
    <isSupportAlgParam>
        <!--ro, opt, bool-->true
    </isSupportAlgParam>
    <isSupportFaceRule>
        <!--ro, opt, bool-->true
    </isSupportFaceRule>
    <isSupportIntelliResource>
        <!--ro, opt, bool-->true
    </isSupportIntelliResource>
    <isSupportShield>
        <!--ro, opt, bool-->true
    </isSupportShield>
    <isLeavePositionSupport>
        <!--ro, opt, bool-->true
    </isLeavePositionSupport>
    <isSupportPersonDensityDetection>
        <!--ro, opt, bool, whether the device supports people density detection, desc:whether the device supports people density detection-->true
    </isSupportPersonDensityDetection>
    <isSupportMixedTargetDetection>
        <!--ro, opt, bool, whether the device supports multi-target-type detection events, desc:whether the device supports multi-target-type detection events-->true
    </isSupportMixedTargetDetection>
    <isSupportRadarVideoTargetUpload>
        <!--ro, opt, bool-->true
    </isSupportRadarVideoTargetUpload>
</IntelliCap>
```

12.8 Two-Way Audio

12.8.1 Two-Way Audio

12.8.1.1 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>
        <audioInboundCompressionType>
            <!--ro, opt, enum, encoding type of the audio input, subType:string, desc:"G.711alaw", "G.711ulaw", "G.726", "G.729", "G.729a", "G.729b", "PCM", "MP3", "AC3", "AAC", "ADPCM"-->G.711alaw
        </audioInboundCompressionType>
        <speakerVolume>
            <!--ro, opt, int, input volume, range:[1,100]-->100
        </speakerVolume>
        <microphoneVolume>
            <!--ro, opt, int, output volume, range:[1,100]-->100
        </microphoneVolume>
        <noisereduce>
            <!--ro, opt, bool, whether to enable the environmental noise filter or not-->true
        </noisereduce>
        <audioBitRate>
            <!--ro, opt, int, audio frame rate, unit:kbs-->100
        </audioBitRate>
        <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>
        <associateVideoInputs>
            <!--ro, opt, object, Linked video channel-->
            <enabled>
                <!--ro, req, bool, whether to enable the Linked video input channel or not-->true
            </enabled>
            <videoInputChannelList>
                <!--ro, req, array, List of Linked video channels, subType:object-->
                <videoInputChannelID>
                    <!--ro, opt, string, Linked video channel-->1
                </videoInputChannelID>
            </videoInputChannelList>
        </associateVideoInputs>
        <audioSamplingRate>
            <!--ro, opt, float, audio sampling rate, unit:kHz-->48.00
        </audioSamplingRate>
        <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>
        <muteDuringPanTilt>
            <!--ro, opt, bool, whether to mute during motion-->true
        </muteDuringPanTilt>
        <audioOutputType>
            <!--ro, opt, enum, audio output type, subType:string, desc:"Close" (output closed), "LineOut" (Line-Level output), "Speaker" (speaker output), "selfAdaptive" (self-adaptive), "LineOut_Speaker" (both Line-Level output and speaker output)-->LineOut
        </audioOutputType>
        <matrixPickUp>
            <!--ro, opt, bool, whether to use the wired matrix pickup or not (this node will not be returned if the wired matrix pickup is not supported, and it is valid only when the audio input type is "Linein")-->true
        </matrixPickUp>
    </TwoWayAudioChannel>
</TwoWayAudioChannelList>
```

12.8.1.2 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>
        <audioInboundCompressionType>
            <!--req, enum, encoding type of the audio input, subType:string, desc:"G.711alaw", "G.711ulaw", "G.726", "G.729", "G.729a", "G.729b", "PCM", "MP3", "AC3", "AAC", "ADPCM"-->G.711alaw
        </audioInboundCompressionType>
        <speakerVolume>
            <!--opt, int, input volume-->100
        </speakerVolume>
        <microphoneVolume>
            <!--opt, int, output volume-->100
        </microphoneVolume>
        <noisereduce>
            <!--opt, bool, whether to enable the environmental noise filter or not-->true
        </noisereduce>
        <audioBitRate>
            <!--opt, int, audio frame rate, unit:kbs-->100
        </audioBitRate>
        <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>
        <associateVideoInputs>
            <!--opt, object, Linked video channel-->
            <enabled>
                <!--req, bool, whether to enable the linked video input channel or not-->true
            </enabled>
            <videoInputChannelList>
                <!--req, array, List of linked video channels, subType:object-->
                <videoInputChannelID>
                    <!--opt, string, linked video channel-->1
                </videoInputChannelID>
            </videoInputChannelList>
        </associateVideoInputs>
        <audioSamplingRate>
            <!--opt, float, audio sampling rate, unit:kHz-->48.00
        </audioSamplingRate>
        <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>
        <muteDuringPanTilt>
            <!--opt, bool, whether to mute during motion-->true
        </muteDuringPanTilt>
        <audioOutputType>
            <!--req, enum, audio output type, subType:string, desc:"Close" (output closed), "LineOut" (line-level output), "Speaker" (speaker output), "selfAdaptive" (self-adaptive), "LineOut_Speaker" (both line-level output and speaker output)-->LineOut
        </audioOutputType>
        <matrixPickUp>
            <!--opt, bool, whether to use the wired matrix pickup or not (this node is valid only when the audio input type is "Linein")-->true
        </matrixPickUp>
    </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>
  <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>

```

12.8.1.3 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-->
    </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>
    <audioInboundCompressionType>
        <!--opt, enum, encoding type of the audio input, subType:string, desc:"G.711alaw", "G.711ulaw", "G.726", "G.729", "G.729a", "G.729b", "PCM", "MP3", "AC3", "AAC", "ADPCM"-->G.711alaw
    </audioInboundCompressionType>
    <speakerVolume>
        <!--opt, int, input volume, range:[1,100]-->100
    </speakerVolume>
    <microphoneVolume>
        <!--opt, int, output volume, range:[1,100]-->100
    </microphoneVolume>
    <noisereduce>
        <!--opt, bool, whether to enable the environmental noise filter or not-->true
    </noisereduce>
    <audioBitRate>
        <!--opt, int, audio frame rate, unit:kbs-->100
    </audioBitRate>
    <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>
    <associateVideoInputs>
        <!--opt, object, linked video channel-->
        <enabled>
            <!--req, bool, whether to enable the linked video input channel or not-->true
        </enabled>
        <videoInputChannelList>
            <!--req, array, list of linked video channels, subType:object-->
            <videoInputChannelID>
                <!--opt, string, linked video channel-->1
            </videoInputChannelID>
        </videoInputChannelList>
    </associateVideoInputs>
    <audioSamplingRate>
        <!--opt, float, audio sampling rate, unit:kHz-->48.00
    </audioSamplingRate>
    <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>
    <micInForbidden>
        <!--opt, bool, whether the audio input is not supported. If this node is not returned or its value is "false", audio input is supported; if the value is "true", audio input is not supported-->true
    </micInForbidden>
    <muteDuringPanTilt>
        <!--opt, bool, whether to mute during motion-->true
    </muteDuringPanTilt>
    <audioOutputType>
        <!--opt, enum, audio output type, subType:string, desc:"Close" (output closed), "LineOut" (line-level output), "Speaker" (speaker output), "selfAdaptive" (self-adaptive), "LineOut_Speaker" (both line-level output and speaker output)-->LineOut
    </audioOutputType>
    <timeOut>
        <!--opt, int, timeout during two-way audio, range:[10,600], unit:s, desc:when the Web Client initiates the two-way audio actively towards the radar device, the noise data will be transmitted to the radar device although the person on the Web Client does not speak, because there is always some noise. When this node is configured, the device will stop the two-way audio after a specific time period, regardless of whether the person on the Web Client stops speaking or not.-->10
    </timeOut>
    <matrixPickUp>
        <!--opt, bool, whether to use the wired matrix pickup or not (this node is valid only when the audio input type is "Linein")-->true
    </matrixPickUp>
</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-->OK
    </subStatusCode>
  </statusCode>
</ResponseStatus>

```

12.8.1.4 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	Two-way audio channel No.
audioInputType	string	Audio input type, corresponding to the "audioInputType" in the message. This parameter is mainly used to obtain the parameters corresponding to the specified audio input type.

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>
    <audioInboundCompressionType opt="G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM,G.722.1">
        <!--ro, opt, enum, encoding type of the audio input, subType:string, attr:opt{opt, string}, desc:"G.711alaw", "G.711ulaw", "G.726", "G.729", "G.729a", "G.729b", "PCM", "MP3", "AC3", "AAC", "ADPCM"-->G.711alaw
    </audioInboundCompressionType>
    <speakerVolume min="0" max="100">
        <!--ro, opt, int, input volume, attr:min{opt, int},max{opt, int}-->100
    </speakerVolume>
    <microphoneVolume min="0" max="100">
        <!--ro, opt, int, output volume, attr:min{opt, int},max{opt, int}-->100
    </microphoneVolume>
    <noisereduce opt="true,false">
        <!--ro, opt, bool, whether to enable the environmental noise filter or not, attr:opt{opt, string}-->true
    </noisereduce>
    <audioBitRate min="1" max="100">
        <!--ro, opt, int, audio frame rate, unit:kbs, attr:min{opt, int},max{opt, int}-->100
    </audioBitRate>
    <audioInputType opt="MicIn,LineIn,selfAdaptive,wirelessPickUp,doubleMic,networkPickUp">
        <!--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>
    <associateVideoInputs>
        <!--ro, opt, object, linked video channel-->
        <enabled opt="true,false">
            <!--ro, req, bool, whether to enable the linked video input channel or not, attr:opt{opt, string}-->true
        </enabled>
        <videoInputChannelList>
            <!--ro, req, array, list of linked video channels, subType:object-->
            <videoInputChannelID min="1" max="2048">
                <!--ro, opt, string, linked video channel, attr:min{opt, string},max{opt, string}-->1
            </videoInputChannelID>
        </videoInputChannelList>
    </associateVideoInputs>
    <audioSamplingRate min="1.00" max="48.00">
        <!--ro, opt, float, audio sampling rate, unit:kHz, attr:min{opt, float},max{opt, float}-->48.00
    </audioSamplingRate>
    <lineOutForbidden opt="true,false">
        <!--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, attr:opt{opt, string}-->true
    </lineOutForbidden>
    <muteDuringPanTilt opt="true,false">
        <!--ro, opt, bool, whether to mute during motion, attr:opt{opt, string}-->true
    </muteDuringPanTilt>
    <audioOutputType opt="Close,LineOut,Speaker,selfAdaptive,LineOut_Speaker,HDMI" def="LineOut">
        <!--ro, opt, enum, audio output type, subType:string, attr:opt{opt, string},def{opt, string}, desc:"Close" (output closed), "LineOut" (line-level output), "Speaker" (speaker output), "selfAdaptive" (self-adaptive), "LineOut_Speaker" (both line-level output and speaker output)-->LineOut
    </audioOutputType>
    <isSupportEchoCancellation>
        <!--ro, opt, bool, whether the device supports echo cancellation-->true
    </isSupportEchoCancellation>
    <timeOut min="10" max="600">
        <!--ro, opt, int, timeout during two-way audio, attr:min{req, int},max{req, int}, desc:when the Web Client initiates the two-way audio actively towards the radar device, the noise data will be transmitted to the radar device although the person on the Web Client does not speak, because there is always some noise. When this node is configured, the device will stop the two-way audio after a specific time period, regardless of whether the person on the Web Client stops speaking or not.-->300
    </timeOut>
    <matrixPickUp opt="true,false">
        <!--ro, opt, bool, whether to use the wired matrix pickup or not (this node will not be returned if the wired matrix pickup is not supported, and it is valid only when the audio input type is "Linein"), attr:opt{opt, string}-->false
    </matrixPickUp>
    <isSupportRestore>
        <!--ro, opt, bool, whether the device supports restoring audio parameters of the two-way audio channel, desc:corresponding API: /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>/restore?format=json-->true
    </isSupportRestore>
    <networkPickUpStatus opt="online,offline">
        <!--ro, opt, enum, subType:string, dep:and,{$.TwoWayAudioChannel.audioInputType,eq,networkPickUp}, attr:opt{opt, string}-->online
    </networkPickUpStatus>
    <noisereduceType opt="normal,smart">
        <!--ro, opt, enum, subType:string, attr:opt{req, string}-->normal
    </noisereduceType>
</TwoWayAudioChannel>

```

12.8.1.5 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>
  <description>
    <!--ro, opt, string, Custom Error Information Description, range:[0,1024], desc:Custom detailed error information returned by device application, used for quick positioning and diagnosis.-->badXmlFormat
  </description>
</ResponseStatus>
```

12.8.1.6 Start two-way audio

Request URL

PUT /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>/open?type=<type>

Query Parameter

Parameter Name	Parameter Type	Description
twoWayAudioChannelID	string	Two-way audio channel No.
type	string	If this field does not exist, it indicates two-way audio via channel, and the audioid is the channel ID. When the type is childDevID, it indicates two-way audio with a single sub device, and the audioid is the gateway intercom resource index (obtained from /ISAPI/System/DynamicResource/SearchResources?format=json); when the type is childDevList, it indicates broadcasting to multiple sub devices, with multiple childDevID values specified in the message, and the audioid in the URL is invalid.

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}-->
    <audioLevel>
        <!--opt, int, audio Level, range:[0,15], desc:range: [0,15]-->0
    </audioLevel>
    <microphoneVolume>
        <!--opt, int, output volume-->100
    </microphoneVolume>
    <audioCompressionType>
        <!--opt, 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>
    <audioSamplingRate>
        <!--opt, float, audio sampling rate, unit:kHz-->48.00
    </audioSamplingRate>
    <isBroadcast>
        <!--opt, bool-->true
    </isBroadcast>
    <broadcastType>
        <!--opt, enum, subType:string-->alarm
    </broadcastType>
    <childList>
        <!--opt, array, subType:object, range:[0,8], dep:and,{$.TwoWayAudio.isBroadCast,eq,true}-->
        <childDevID>
            <!--opt, string, range:[9,32]-->test
        </childDevID>
        <voiceTalkIndex>
            <!--opt, int, range:[1,256]-->test
        </voiceTalkIndex>
    </childList>
</TwoWayAudio>

```

Response Message

```

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

<TwoWayAudioSession xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--no, opt, object, result returned when starting two-way audio succeeds, attr:version{opt, string, protocolVersion}-->
    <sessionId>
        <!--no, req, string, two-way audio session ID-->1
    </sessionId>
</TwoWayAudioSession>

```

12.8.1.7 Get the audio parameters of a specified two-way audio channel

Request URL

GET /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>?audiolnputType=<audiolnputType>

Query Parameter

Parameter Name	Parameter Type	Description
twoWayAudioChannelID	string	Two-way audio channel No.
audiolnputType	string	Audio input type, corresponding to the "audiolnputType" in the message. This parameter is mainly used to obtain the parameters corresponding to the specified audio input type.

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-->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.722.1"-->G.711alaw
  </audioCompressionType>
  <audioInboundCompressionType>
    <!--ro, opt, enum, encoding type of the audio input, subType:string, desc:"G.711alaw", "G.711uLaw", "G.726", "G.729", "G.729a", "G.729b", "PCM", "MP3", "AC3", "AAC", "ADPCM"-->G.711alaw
  </audioInboundCompressionType>
  <speakerVolume>
    <!--ro, opt, int, input volume, range:[1,100]-->100
  </speakerVolume>
  <microphoneVolume>
    <!--ro, opt, int, output volume, range:[1,100]-->100
  </microphoneVolume>
  <noisereduce>
    <!--ro, opt, bool, whether to enable the environmental noise filter or not-->true
  </noisereduce>
  <audioBitRate>
    <!--ro, opt, int, audio frame rate, unit:kbs-->100
  </audioBitRate>
  <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>
  <associateVideoInputs>
    <!--ro, opt, object, linked video channel-->
    <enabled>
      <!--ro, req, bool, whether to enable the linked video input channel or not-->true
    </enabled>
    <videoInputChannelList>
      <!--ro, req, array, list of linked video channels, subType:object-->
      <videoInputChannelID>
        <!--ro, opt, string, linked video channel-->1
      </videoInputChannelID>
    </videoInputChannelList>
  </associateVideoInputs>
  <audioSamplingRate>
    <!--ro, opt, float, audio sampling rate, unit:kHz-->48.00
  </audioSamplingRate>
  <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>
  <micInForbidden>
    <!--ro, opt, bool, whether the audio input is not supported. If this node is not returned or its value is "false", audio input is supported; if the value is "true", audio input is not supported-->true
  </micInForbidden>
  <muteDuringPanTilt>
    <!--ro, opt, bool, whether to mute during motion-->true
  </muteDuringPanTilt>
  <audioOutputType>
    <!--ro, opt, enum, audio output type, subType:string, desc:"Close" (output closed), "LineOut" (line-level output), "Speaker" (speaker output), "selfAdaptive" (self-adaptive), "LineOut_Speaker" (both line-level output and speaker output)-->LineOut
  </audioOutputType>
  <timeOut>
    <!--ro, opt, int, timeout during two-way audio, range:[10,600], unit:s, desc:when the Web Client initiates the two-way audio actively towards the radar device, the noise data will be transmitted to the radar device although the person on the Web Client does not speak, because there is always some noise. When this node is configured, the device will stop the two-way audio after a specific time period, regardless of whether the person on the Web Client stops speaking or not.-->10
  </timeOut>
  <matrixPickUp>
    <!--ro, opt, bool, whether to use the wired matrix pickup or not (this node is valid only when the audio input type is "Linein")-->true
  </matrixPickUp>
  <networkPickUpStatus>
    <!--ro, opt, enum, subType:string, dep:and,{$.TwoWayAudioChannel.audioInputType,eq,networkPickUp}-->online
  </networkPickUpStatus>
  <noisereduceType>
    <!--ro, opt, enum, subType:string-->normal
  </noisereduceType>
  </TwoWayAudioChannel>

```

12.9 Entrance & Exit Management and Control

12.9.1 Entrance/Exit Cabinet

12.9.1.1 Get the cabinet parameters

Request URL

GET /ISAPI/ITC/cabinetParam

Query Parameter

None

Request Message

None

Response Message

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

<CabinetParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <cabinetValidEnable>
    <!--ro, req, bool-->true
  </cabinetValidEnable>
  <CabinetParamList>
    <!--ro, opt, array, subType:object-->
    <singleCabinetParam>
      <!--ro, opt, object-->
      <cabinetId>
        <!--ro, req, int, range:[1,4]-->0
      </cabinetId>
      <associateIO>
        <!--ro, req, int, range:[1,255]-->0
      </associateIO>
      <cabinetName>
        <!--ro, opt, string, range:[0,32]-->test
      </cabinetName>
      <cabinetStatus>
        <!--ro, req, enum, subType:int-->0
      </cabinetStatus>
      <alarmIntervalTime>
        <!--ro, req, int, range:[1,300], step:30, unit:s-->1
      </alarmIntervalTime>
    </singleCabinetParam>
  </CabinetParamList>
</CabinetParam>
```

12.9.1.2 Set the cabinet parameters

Request URL

PUT /ISAPI/ITC/cabinetParam

Query Parameter

None

Request Message

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

<CabinetParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, attr:version{req, string, protocolVersion}-->
  <cabinetValidEnable>
    <!--req, bool-->true
  </cabinetValidEnable>
  <CabinetParamList>
    <!--opt, array, subType:object-->
    <singleCabinetParam>
      <!--opt, object-->
      <cabinetId>
        <!--req, int, range:[1,4]-->0
      </cabinetId>
      <associateIO>
        <!--req, int, range:[1,255]-->0
      </associateIO>
      <cabinetName>
        <!--opt, string, range:[0,32]-->test
      </cabinetName>
      <cabinetStatus>
        <!--req, enum, subType:int-->0
      </cabinetStatus>
      <alarmIntervalTime>
        <!--req, int, range:[1,300], step:30, unit:s-->1
      </alarmIntervalTime>
    </singleCabinetParam>
  </CabinetParamList>
</CabinetParam>
```

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>
```

12.9.1.3 Get the capability of cabinet parameters

Request URL

GET /ISAPI/ITC/cabinetParam/capabilities

Query Parameter

None

Request Message

None

Response Message

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

<CabinetParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <isSupportCabinetEnable>
    <!--ro, req, bool-->true
  </isSupportCabinetEnable>
  <cabinetId min="1" max="4">
    <!--ro, req, int, range:[1,4], attr:min{req, int},max{req, int}-->0
  </cabinetId>
  <associateIO opt="1,2,3,4,5,6,7,8,0xff">
    <!--ro, req, int, range:[1,255], attr:opt{req, string}-->0
  </associateIO>
  <cabinetName min="0" max="32">
    <!--ro, opt, string, range:[0,32], attr:min{req, int},max{req, int}-->test
  </cabinetName>
  <cabinetStatus min="0" max="1">
    <!--ro, req, int, range:[0,1], attr:min{req, int},max{req, int}-->0
  </cabinetStatus>
  <alarmIntervalTime min="1" max="300">
    <!--ro, req, int, range:[1,300], step:30, attr:min{req, int},max{req, int}-->1
  </alarmIntervalTime>
</CabinetParam>
```

12.9.2 Entrance/Exit Display Screen

12.9.2.1 Get the configuration capability of LCD (Liquid Crystal Display) parameters

Request URL

GET /ISAPI/Parking/channels/<channelID>/LCD/capabilities?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

{
  "LCDCap": {
    /*ro, req, object, LCD configuration capability*/
    "allowListDisplayEnabled": {
      /*ro, opt, object, whether to automatically display the custom content configured by the user whose vehicle is in the authorized list, desc:If enabled, if the passing vehicle is in the authorized list, the device will automatically overwrite the custom content configured by the user. If not enabled, the display content will not be changed.*/
      "@opt": [true, false],
      /*ro, req, array, range, subType:bool*/
      "@def": false
      /*ro, opt, bool, default value*/
    },
    "blockListDisplayEnabled": {
      /*ro, opt, object, whether to automatically display the custom content configured by the user whose vehicle is in the unauthorized list, desc:If enabled, if the passing vehicle is in the unauthorized list, the device will automatically overwrite the custom content configured by the user. If not enabled, the display content will not be changed.*/
      "@opt": [true, false],
      /*ro, req, array, range, subType:bool*/
      "@def": false
      /*ro, opt, bool, the default value*/
    },
    "temporaryListDisplayEnabled": {
      /*ro, opt, object, whether to automatically display the custom content configured by the user whose vehicle is a temporary vehicle, desc:If enabled, if the passing vehicle is a temporary vehicle and is not in the unauthorized/authorized list, the device will automatically overwrite the custom content configured by the user. If not enabled, the display content will not be changed.*/
      "@opt": [true, false],
      /*ro, req, array, range, subType:bool*/
      "@def": false
      /*ro, opt, bool, the default value*/
    },
    "timeFormat": {
      /*ro, opt, object, time format*/
      "@opt": ["YYYY-MM-DD", "hh:mm:ss", "hh:mm:ss", "YYYY-MM-DD hh:mm:ss", "YYYY-MM-DD hh-mm-ss", "YYYY:MM:DD", "YYYY:MM:DD hh-mm:ss", "YYYY:MM:DD hh:mm:ss"],
      /*ro, req, array, options, subType:string*/
      "#text": "YYYY-MM-DD"
      /*ro, opt, string, example*/
    },
    "CustomContentList": {
      /*ro, opt, object, list of custom contents*/
      "maxSize": 8,
      /*ro, req, int, the maximum number of custom contents*/
      "CustomContent": {
        /*ro, opt, object, custom content*/
        "id": {
          /*ro, req, object, No., desc:starts from 1*/
          "@min": 1,
          /*ro, req, int, the minimum value*/
          "@max": 8
          /*ro, req, int, the maximum value*/
        },
        "content": {
          /*ro, opt, object, custom content*/
          "@min": 1,
          /*ro, req, int, the minimum value*/
          "@max": 16
          /*ro, req, int, the maximum value*/
        },
        "value": {
          /*ro, opt, object, options of display content*/
          "@opt": ["plate", "parkingLotNum", "triggerTime", "vehicleType", "plateType", "vehicleCategory"],
          /*ro, req, array, range, subType:string, desc:"plate", "parkingLotNum" (remaining parking space), "triggerTime" (vehicle passing time), "vehicleType", "plateType", "vehicleCategory)*/
          "#text": "plate"
          /*ro, opt, string, the sample value*/
        },
        "fontSizeLine": {
          /*ro, opt, object, font size*/
          "@min": 1,
          /*ro, req, int, the minimum value*/
          "@max": 100
          /*ro, req, int, the maximum value*/
        },
        "fontColorLine": {
          /*ro, opt, object, font color*/
          "R": {
            /*ro, req, object, R*/
            "@min": 0,
            /*ro, req, int, the minimum value*/
            "@max": 255
            /*ro, req, int, the maximum value*/
          },
          "G": {
            /*ro, req, object, G*/
            "@min": 0,
            /*ro, req, int, the minimum value*/
            "@max": 255
            /*ro, req, int, the maximum value*/
          },
          "B": {
            /*ro, req, object, B*/
            "@min": 0,
            /*ro, req, int, the minimum value*/
          }
        }
      }
    }
  }
}

```

```

        "@max": 255
        /*ro, req, int, the maximum value*/
    },
    "fontBoldLine": {
        /*ro, opt, object, whether to enable bold font*/
        "@opt": [true, false],
        /*ro, req, array, options, subType:bool*/
        "@def": false
        /*ro, opt, bool, default value*/
    },
    "passVehicleDisplayEnabled": {
        /*ro, opt, object, whether to display the content when a vehicle is passing*/
        "@opt": [true, false],
        /*ro, req, array, options, subType:bool*/
        "@def": false
        /*ro, opt, bool, default value*/
    }
}
},
"fontSize": {
/*ro, opt, object, font size*/
    "@min": 1,
    /*ro, req, int, the minimum value*/
    "@max": 100
    /*ro, req, int, the maximum value*/
},
"MediaDataInfo": {
/*ro, opt, object, settings of playing the media data*/
    "cycleIntervalTime": {
        /*ro, req, object, cycle display interval,unit: second, desc:unit: second*/
        "@min": 1,
        /*ro, req, int, the minimum value*/
        "@max": 2
        /*ro, req, int, the maximum value*/
    },
    "cyclePlayOrder": {
        /*ro, req, object, playing order of cycle display*/
        "@min": 1,
        /*ro, req, int, the minimum value*/
        "@max": 2
        /*ro, req, int, the maximum value*/
    }
},
"ctrlMode": {
/*ro, opt, object, control mode,
desc:It setting the mode is not supported, by default it is valid for both camera and platform.
Camera (automatic control by device): The device recognizes vehicles in the block and authorized list and temporary license plates by itself, and automatically displays the vehicle passing information according to the strategy configured in displayPassingVehicleInfoEnabled, allowListDisplayEnabled, blockListDisplayEnabled, and temporaryListDisplayEnabled.
Platform (manual control): Display the content in CustomContentList applied by the user. the mode, The automatic control strategy configured in displayPassingVehicleInfoEnabled, allowListDisplayEnabled, blockListDisplayEnabled, and temporaryListDisplayEnabled will not take effect.
Camera and platform: the above two control modes take effect at the same time. When CustomContentList is applied, its content is displayed immediately; when the CustomContentList is not applied, it will automatically display the default content on the device.*/
        "@opt": ["camera", "platform", "cameraAndplatform"],
        /*ro, req, array, options, subType:string, desc:"camera", "platform", "cameraAndplatform" (takes effect in camera and platform mode)*/
        "#text": "camera"
        /*ro, opt, string, example*/
    },
    "fanSpeedLevel": {
/*ro, opt, object, fan speed level*/
        "@opt": ["low", "middle", "high", "close"],
        /*ro, req, array, option, subType:string*/
        "#text": "low"
        /*ro, opt, string, example*/
    },
    "screenWidth": 1,
    /*ro, opt, int, screen width*/
    "screenHeight": 1,
    /*ro, opt, int, screen height*/
    "softwareVersion": "test",
    /*ro, opt, string, Android version of the LCD*/
    "BackBright": {
        /*ro, opt, object, backlight brightness management*/
        "enabled": {
            /*ro, opt, object, whether to enable*/
            "@opt": [true, false]
            /*ro, req, array, option, subType:bool*/
        },
        "value": {
            /*ro, opt, object, brightness*/
            "@min": 0,
            /*ro, req, int, the minimum value*/
            "@max": 100
            /*ro, req, int, the maximum value*/
        },
        "timeCtrlEnabled": {
            /*ro, opt, object, whether to enable adjustment by time period*/
            "@opt": [true, false]
            /*ro, req, array, option, subType:bool*/
        },
        "TimeCtrlList": {
            /*ro, opt, object, time period settings*/
            ...
        }
    }
}
}

```

```

    "maxSize": 8,
    /*ro, req, int, the maximum number of custom contents*/
    "TimeCtrl": {
        /*ro, opt, object, time period*/
        "startTime": "17:30:08",
        /*ro, req, string, start time*/
        "endTime": "18:30:08",
        /*ro, req, string, end time*/
        "value": {
            /*ro, opt, object, Brightness*/
            "@min": 0,
            /*ro, req, int, the minimum value*/
            "@max": 100
            /*ro, req, int, the maximum value*/
        },
    },
},
"displayTime": {
/*ro, opt, object, display duration*/
    "@min": 1,
    /*ro, opt, int, the minimum value*/
    "@max": 300
    /*ro, opt, int, the maximum value*/
},
"MediaDataImport": {
/*ro, opt, object, settings for importing media data*/
    "maxPictureNum": 10,
    /*ro, req, int, maximum number of pictures that can be imported*/
    "maxVideoNum": 10,
    /*ro, req, int, maximum number of videos that can be imported*/
    "pictureType": {
        /*ro, opt, object, picture type*/
        "@opt": ["backgroundPicture", "contentPicture", "logo"],
        /*ro, req, array, options, subType:string*/
        "#text": "backgroundPicture"
        /*ro, opt, string, example*/
    }
},
"HintContentList": {
/*ro, opt, object, list of prompt contents*/
    "maxSize": 2,
    /*ro, req, int, the maximum number of custom contents*/
    "HintContent": {
        /*ro, opt, object, prompt content*/
        "id": {
            /*ro, req, object, No.*/
            "@min": 1,
            /*ro, req, int, the minimum value*/
            "@max": 2
            /*ro, req, int, the maximum value*/
        },
        "content": {
            /*ro, opt, object, prompt content*/
            "@min": 1,
            /*ro, req, int, the minimum value*/
            "@max": 16
            /*ro, opt, int, the maximum value*/
        },
        "value": {
            /*ro, opt, object, options of display content*/
            "@opt": ["plate", "parkingLotNum", "triggerTime", "vehicleType", "plateType", "vehicleCategory"],
            /*ro, req, array, range, subType:string, desc:"plate", "parkingLotNum" (remaining parking space), "triggerTime" (vehicle passing time),
            "vehicleType", "plateType", "vehicleCategory"*/
            "#text": "plate"
            /*ro, opt, string, the sample value*/
        },
        "contentType": {
            /*ro, opt, object, content type*/
            "@opt": ["free", "passingVehicle"],
            /*ro, req, array, option, subType:string*/
            "#text": "free"
            /*ro, opt, string, example*/
        },
        "fontSizeLine": {
            /*ro, opt, object, font size*/
            "@min": 1,
            /*ro, req, int, the minimum value*/
            "@max": 100
            /*ro, req, int, the maximum value*/
        },
        "fontColorLine": {
            /*ro, opt, object, font color*/
            "R": {
                /*ro, req, object, R*/
                "@min": 0,
                /*ro, req, int, the minimum value*/
                "@max": 255
                /*ro, req, int, the maximum value*/
            },
            "G": {
                /*ro, req, object, G*/
                "@min": 0,
                /*ro, req, int, the minimum value*/
            }
        }
    }
}

```

```

        , /*..., etc, the maximum value,
        "@max": 255
        /*ro, req, int, the maximum value*/
    },
    "B": {
        /*ro, req, object, B*/
        "@min": 0,
        /*ro, req, int, the minimum value*/
        "@max": 255
        /*ro, req, int, the maximum value*/
    }
},
"fontBoldLine": {
/*ro, opt, object, whether to enable bold font*/
    "@opt": [true, false],
    /*ro, req, array, options, subType:bool*/
    "@def": true
    /*ro, opt, bool, default value*/
}
},
"parkingLotDisplayEnable": {
/*ro, opt, object, whether to display vacant parking spaces*/
    "@opt": [true, false]
    /*ro, req, array, option, subType:bool*/
},
"displayPlateTypeEnable": {
/*ro, opt, object, whether to display the license plate type*/
    "@opt": [true, false]
    /*ro, req, array, option, subType:bool*/
},
"powerOffSaveEnabled": {
/*ro, opt, object, Whether that the parameters can be saved after power failure is supported in the URL,
desc:Whether the parameters are saved after the power is off;
Since the parameters need to be written to flash after power failure, if this node is not applied, the parameters will not be saved in flash; if the
value is false, the LCD will be restored to the default parameters after the power is off and be restarted.
For example: /ISAPI/Parking/channels/<channelID>/LCD?format=json&powerOffSaveEnabled=true*/
    "@opt": [true, false],
    /*ro, req, array, options, subType:bool*/
    "@def": false
    /*ro, opt, bool, default value*/
},
"mediaOccupyConfigListCap": {
/*ro, opt, object, configure the display partition of passing vehicle information and the free scene of LCD*/
    "@size": 2,
    /*ro, opt, int, display the maximum number supported in the array of display partition*/
    "mediaOccupyMode": {
/*ro, opt, object, the display mode of LCD media content and custom content*/
        "@opt": ["mediaOccupy", "customOccupy", "mediaRatioOccupy"],
        /*ro, req, array, options, subType:string, desc:"mediaOccupy" (only display media information which can be configured via
ISAPI/Parking/channels/<channelID>/LCD/mediaData/import), "customOccupy" (only display custom information configured in the CustomContentList),
"mediaRatioOccupy" (display according to the percentage)*/
    },
    "mediaOccupyRatio": {
/*ro, opt, object, percentage of area occupied by the media content of the LCD screen*/
        "@min": 0,
        /*ro, req, int, the minimum value*/
        "@max": 100
        /*ro, req, int, the maximum value*/
    },
    "contentType": {
/*ro, opt, object, content type*/
        "@opt": ["free", "passingVehicle"],
        /*ro, req, array, options, subType:string*/
    }
},
"logoDisplayEnable": {
/*ro, opt, object, whether to display the logo which can be configured via ISAPI/Parking/channels/<channelID>/LCD/mediaData/import*/
    "@opt": [true, false]
    /*ro, req, array, options, subType:bool*/
},
}
}
}

```

12.9.2.2 Get a piece of LCD (Liquid Crystal Display) media data

Request URL

GET /ISAPI/Parking/channels/<channelID>/LCD/mediaData/<FileID>?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
FileID	string	--

Request Message

None

Response Message

```
{  
    "MediaData": {  
        /*ro, req, object, media data*/  
        "id": 1,  
        /*ro, req, int, No., desc:it starts from 1 and corresponds to the ID returned after the media data is imported*/  
        "dataType": "picture",  
        /*ro, req, enum, type of the imported data, subType:string, desc:"picture", "video"*/  
        "name": "test",  
        /*ro, req, string, file name, range:[1,64]*/  
        "pictureType": "backgroundPicture",  
        /*ro, opt, enum, picture type, subType:string, desc:"backgroundPicture" (background picture), "contentPicture" (content picture)*/  
        "size": 1  
        /*ro, opt, int, file size, range:[1,1024000], desc:unit: KB*/  
    }  
}
```

12.9.2.3 Delete a piece of LCD (Liquid Crystal Display) media data

Request URL

DELETE /ISAPI/Parking/channels/<channelID>/LCD/mediaData/<FileID>?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
FileID	string	--

Request Message

None

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 corresponds to subStatusCode when statusCode is not 1*/  
    "errorMsg": "ok",  
    /*ro, opt, string, error information, desc:this field is required when the value of statusCode is not 1*/  
}
```

12.9.2.4 Get all LCD (Liquid Crystal Display) media data

Request URL

GET /ISAPI/Parking/channels/<channelID>/LCD/mediaData?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

{
    "MediaDataList": [
        /*ro, req, array, media data list, subType:object*/
        {
            "MediaData": {
                /*ro, opt, object, media data*/
                "id": 1,
                /*ro, req, int, No., desc:it starts from 1 and corresponds to the ID returned after the media data is imported*/
                "dataType": "picture",
                /*ro, req, enum, type of the imported data: "picture", "video", subType:string, desc:"picture", "video"*/
                "name": "test",
                /*ro, req, string, file name, range:[1,64]*/
                "pictureType": "backgroundPicture",
                /*ro, opt, enum, picture type, subtype:string, desc:"backgroundPicture" (background picture), "contentPicture" (content picture)*/
                "size": 1
                /*ro, opt, int, data size, range:[1,1024000], desc:unit: KB*/
            }
        }
    ]
}

```

12.9.2.5 Delete all LCD (Liquid Crystal Display) media data

Request URL

DELETE /ISAPI/Parking/channels/<channelID>/LCD/mediaData?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

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, it corresponds to subStatusCode*/
    "errorMsg": "ok"
    /*ro, opt, string, error description, desc:this field is required when the value of statusCode is not 1*/
}

```

12.9.2.6 Set the LCD (Liquid Crystal Display) parameters.

Request URL

PUT /ISAPI/Parking/channels/<channelID>/LCD?format=json&powerOffSaveEnabled=<powerOffSaveEnabled>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
powerOffSaveEnabled	string	Whether the parameters are saved after the power is off; Since the parameters need to be written to flash after power failure, if this node is not applied, the parameters will not be saved in flash; if the value is false, the LCD will be restored to the default parameters after the power is off and be restarted. For example: /ISAPI/Parking/channels//LCD? format=json&powerOffSaveEnabled=true

Request Message

```

{
  "LCD": {
    /*req, object, LCD parameters*/
    "displayPassingVehicleInfoEnabled": true,
    /*opt, bool, whether to enable the display of passing vehicle information*/
    "allowListDisplayEnabled": false,
    /*opt, bool, whether to automatically display the custom content configured by the user whose vehicle is in the authorized list, desc:If enabled, if the passing vehicle is in the authorized list, the device will automatically overwrite the custom content configured by the user. If not enabled, the display content will not be changed.*/
    "blockListDisplayEnabled": false,
    /*opt, bool, whether to automatically display the custom content configured by the user whose vehicle is in the unauthorized list, desc:If enabled, if the passing vehicle is in the unauthorized list, the device will automatically overwrite the custom content configured by the user. If not enabled, the display content will not be changed.*/
    "temporaryListDisplayEnabled": false,
    /*opt, bool, whether to automatically display the custom content configured by the user whose vehicle is a temporary vehicle, desc:If enabled, if the passing vehicle is a temporary vehicle and is not in the unauthorized/authorized list, the device will automatically overwrite the custom content configured by the user. If not enabled, the display content will not be changed.*/
    "displayPlateEnable": true,
    /*opt, bool, whether to display the license plate*/
    "displayTimeEnable": true,
    /*opt, bool, whether to enable time display*/
    "timeFormat": "YYYY-MM-DD",
    /*opt, enum, time display format, subType:string, desc:"YYYY-MM-DD", "hh-mm-ss"*/
    "backgroundDisplayEnabled": true,
    /*opt, bool, whether to display background picture, desc:If enabled, when a vehicle is passing, the passing vehicle information will be displayed with the background picture*/
    "backgroundDisplayTime": 1,
    /*opt, int, display duration of the background picture, range:[1,180], unit:s, desc:If another vehicle is passing, the duration will be refreshed*/
    "CustomContentList": [
      /*opt, array, custom content list, subType:object, range:[1,8]*/
      {
        "CustomContent": {
          /*opt, object, custom content*/
          "id": 1,
          /*req, int, No., range:[1,8], desc:it increases from 1*/
          "content": "test",
          /*opt, string, custom content, range:[1,16]*/
          "value": "plate",
          /*opt, enum, options of display content, subType:string, desc:"plate", "parkingLotNum" (remaining parking space), "triggerTime" (vehicle passing time), "vehicleType", "plateType", "vehicleCategory"*/
          "contentType": "advertisement",
          /*opt, enum, content type, subType:string, desc:"advertisement", "redLight", "greenLight"*/
          "fontSizeline": 1,
          /*opt, int, font size, range:[1,100]*/
          "fontColorLine": {
            /*opt, object, font color*/
            "R": 1,
            /*req, int, R, range:[0,255]*/
            "G": 1,
            /*req, int, G, range:[0,255]*/
            "B": 1,
            /*req, int, B, range:[0,255]*/
          },
          "fontTypeLine": "Microsoft YaHei",
          /*opt, enum, font type, subType:string, desc:"Microsoft YaHei", "SimSun", "SimHei", "FangSong", "KaiTi"*/
          "fontBoldLine": false,
          /*opt, bool, whether to enable bold font*/
          "passVehicleDisplayEnabled": false
          /*opt, bool, whether to display the content when a vehicle is passing, desc:true (display the content when a vehicle is passing), false (display the content for free time), it is false by default*/
        }
      }
    ],
    "fontSize": 1,
    /*opt, int, font size, range:[1,100]*/
    "fontColor": {
      /*opt, object, font color*/
      "R": 1,
      /*req, int, R, range:[0,255]*/
      "G": 1,
      /*req, int, G, range:[0,255]*/
      "B": 1
      /*req, int, B, range:[0,255]*/
    },
    "fontMovementSpeed": 20,
    /*opt, int, moving speed threshold of font, range:[0,30], desc:default value: 20*/
    "MediaDataInfo": {
      /*opt, object, settings of playing the media data*/
      "cycleIntervalTime": 1,
      /*req, int, cycle display interval, unit:s*/
      "cyclePlayOrder": [1, 3, 4]
      /*req, array, playing order of cycle display, subType:int*/
    },
    "ctrlMode": "cameraAndplatform",
    /*opt, enum, control mode, subType:string, desc:"camera", "platform", "cameraAndplatform" (takes effect in camera and platform mode); If setting the mode is not supported, by default it is valid for both camera and platform.
    Camera (automatic control by device): The device recognizes vehicles in the block and authorized list and temporary license plates by itself, and automatically displays the vehicle passing information according to the strategy configured in displayPassingVehicleInfoEnabled, allowListDisplayEnabled, blockListDisplayEnabled, and temporaryListDisplayEnabled.
    Platform (manual control): Display the content in CustomContentList applied by the user. the mode, The automatic control strategy configured in displayPassingVehicleInfoEnabled, allowListDisplayEnabled, blockListDisplayEnabled, and temporaryListDisplayEnabled will not take effect.
    Camera and platform: the above two control modes take effect at the same time. When CustomContentList is applied, its content is displayed immediately when the CustomContentList is not applied, it will automatically display the default content on the device */
  }
}

```

```

immediately, when the customContentList is not applied, it will automatically display the default content on the device.//
    "fanSpeedLevel": "low",
    /*opt, enum, fan speed, subType:string, desc:"low", "middle", "high", "close"*/
    "temperatureThreshold": 1,
    /*opt, int, screen temperature threshold, range:[0,100], unit:°C*/
    "screenWidth": 1,
    /*ro, opt, int, screen width*/
    "screenHeight": 1,
    /*ro, opt, int, screen height*/
    "softwareVersion": "test",
    /*ro, opt, string, Android version of the LCD*/
    "BackBright": {
        /*opt, object, backlight brightness management*/
        "enabled": true,
        /*opt, bool, whether to enable the function*/
        "value": 1,
        /*opt, int, brightness*/
        "timeCtrlEnabled": true,
        /*opt, bool, whether to enable adjustment by time period*/
        "TimeCtrlList": [
            /*opt, array, time period configuration, subType:object, range:[1,8]*/
            {
                "TimeCtrl": {
                    /*opt, object, time period*/
                    "startTime": "17:30:08",
                    /*req, time, start time*/
                    "endTime": "18:30:08",
                    /*req, time, end time*/
                    "value": 1,
                    /*req, int, brightness, range:[0,100]*/
                    "vehiclePassingValue": 1
                    /*opt, int, passing vehicle brightness, range:[0,100], desc:When a vehicle passes, the brightness is the value of this node; if no vehicle passes, the brightness is the value of the node value. If configuring this node is not supported, the brightness is the value of the node value*/
                }
            }
        ]
    },
    "SceneUIList": [
        /*opt, array, UI template management of multiple scenes, subType:object*/
        {
            "SceneUI": {
                /*opt, object, ID of scene UI template*/
                "scenemode": "entranceWithPlate",
                /*req, enum, scene type, subType:string, desc:"entranceWithPlate" (UI template type for entered vehicles with license plate), "entranceWithoutPlate" (UI template type for entered vehicles without license plate), "exitWithoutCharge" (UI template type for exited vehicles with unpaid fee), "exitWithCharge" (UI template type for exited vehicles without unpaid fee), "free" (free UI template type)*/
                "UUID": 1
                /*req, int, ID of scene UI template*/
            }
        }
    ],
    "displayTime": 1,
    /*opt, int, display duration, range:[1,300]*/
    "HintContentList": [
        /*opt, array, list of prompt contents, subType:object, range:[1,2]*/
        {
            "HintContent": {
                /*opt, object, prompt content*/
                "id": 1,
                /*req, int, No., range:[1,2], desc:it increases from 1*/
                "content": "test",
                /*opt, string, custom content, range:[1,16]*/
                "value": "plate",
                /*opt, enum, options of display content, subType:string, desc:"plate", "parkingLotNum" (remaining parking space), "triggerTime" (vehicle passing time), "vehicleType", "plateType", "vehicleCategory"*/
                "contentType": "free",
                /*opt, enum, content type, subType:string, desc:"free", "passingVehicle"*/
                "fontSizeLine": 1,
                /*opt, int, font size, range:[1,100]*/
                "fontColorLine": {
                    /*opt, object, font color*/
                    "R": 1,
                    /*req, int, R, range:[0,255]*/
                    "G": 1,
                    /*req, int, G, range:[0,255]*/
                    "B": 1
                    /*req, int, B, range:[0,255]*/
                },
                "fontTypeLine": "Microsoft YaHei",
                /*opt, enum, font type, subType:string, desc:"Microsoft YaHei", "SimSun", "SimHei", "FangSong", "KaiTi"*/
                "fontBoldLine": true
                /*opt, bool, whether to enable bold font*/
            }
        }
    ],
    "parkingLotDisplayEnable": true,
    /*opt, bool, whether to enable the display of vacant parking spaces*/
    "sceneMode": "entryScene",
    /*opt, enum, scene mode, subType:string, desc:"entryScene", "exitScene", "normalScene"*/
    "displayPlateTypeEnable": true,
    /*opt, bool, whether to enable the display of the license plate type*/
    "displayValidityPeriodEnable": true,
    /*opt, bool, whether to enable the display of validity period*/
    "logoDisplayEnable": true,

```

```

/*opt, bool, whether to display the logo which can be configured via ISAPI/Parking/channels/<channelID>/LCD/mediaData/import*/
"mediaOccupyConfigList": [
    /*ro, opt, array, parameters of configuring the area for displaying the content when a vehicle is displaying and when it is free time on the LCD
screen, subType:object*/
    {
        "mediaOccupyMode": "mediaOccupy",
        /*ro, opt, enum, display mode of media information and custom information on LCD screen, subType:string, desc:"mediaOccupy" (only display
media information), "customOccupy" (only display custom information configured in the CustomContentList), "mediaRatioOccupy" (display according to the
percentage); "mediaOccupy" (only display media information which can be configured via ISAPI/Parking/channels/<channelID>/LCD/mediaData/import),
"customOccupy" (only display custom information configured in the CustomContentList), "mediaRatioOccupy" (display according to the percentage)*/
        "mediaOccupyRatio": 0,
        /*ro, opt, int, percentage of area occupied by the media content of the LCD screen, range:[0,100]*/
        "contentType": "free"
        /*ro, opt, enum, content type, subType:string, desc:"free", "passingVehicle"*/
    }
],
"SceneUI": {
/*opt, object, UI template of multiple scenes*/
    "id": 1,
    /*req, int, valid template ID, desc:1, 2, 3*/
    "RegionList": [
        /*opt, array, template area list, subType:object*/
        {
            "Region": {
                /*opt, object, area*/
                "id": 1,
                /*req, int, No., desc:it increases from 1*/
                "trafficLightEnabled": true,
                /*opt, bool, whether to switch the traffic light*/
                "contentEnabled": true,
                /*opt, bool, whether to enable the advertisement text*/
                "pictureEnabled": true,
                /*opt, bool, whether to enable the advertisement picture*/
                "videoEnabled": true,
                /*opt, bool, whether to enable the advertisement video*/
                "time": 1,
                /*opt, int, display interval, unit:s*/
                "previewEnabled": true,
                /*opt, bool, live view*/
                "localTimeEnabled": true
                /*opt, bool, whether to display the local time*/
            }
        }
    ]
},
"characterMovementMode": "scrolling"
/*opt, enum, subType:string*/
}
}

```

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 node is required when the value of statusCode is not 1*/
}

```

12.9.2.7 Get the LCD (Liquid Crystal Display) parameters

Request URL

GET /ISAPI/Parking/channels/<channelID>/LCD?format=json&powerOffSaveEnabled=<powerOffSaveEnabled>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
powerOffSaveEnabled	string	Whether the parameters are saved after the power is off; Since the parameters need to be written to flash after power failure, if this node is not applied, the parameters will not be saved in flash; if the value is false, the LCD will be restored to the default parameters after the power is off and be restarted. For example: /ISAPI/Parking/channels//LCD?format=json&powerOffSaveEnabled=true

Request Message

None

Response Message

```
{
    "LCD": {
        /*ro, req, object, LCD*/
        "displayPassingVehicleInfoEnabled": true,
        /*ro, opt, bool, whether to enable the display of passing vehicle information*/
        "allowListDisplayEnabled": false,
        /*ro, opt, bool, whether to automatically display the custom content configured by the user whose vehicle is in the authorized list, desc:If enabled, if the passing vehicle is in the authorized list, the device will automatically overwrite the custom content configured by the user. If not enabled, the display content will not be changed.*/
        "blockListDisplayEnabled": false,
        /*ro, opt, bool, whether to automatically display the custom content configured by the user whose vehicle is in the unauthorized list, desc:If enabled, if the passing vehicle is in the unauthorized list, the device will automatically overwrite the custom content configured by the user. If not enabled, the display content will not be changed.*/
        "temporaryListDisplayEnabled": false,
        /*ro, opt, bool, whether to automatically display the custom content configured by the user whose vehicle is a temporary vehicle, desc:If enabled, if the passing vehicle is a temporary vehicle and is not in the unauthorized/authorized list, the device will automatically overwrite the custom content configured by the user. If not enabled, the display content will not be changed.*/
        "displayPlateEnable": true,
        /*ro, opt, bool, whether to display the license plate*/
        "displayTimeEnable": true,
        /*ro, opt, bool, whether to enable time display*/
        "timeFormat": "YYYY-MM-DD",
        /*ro, opt, enum, time display format, subType:string, desc:time format: "YYYY-MM-DD", "hh-mm-ss"*/
        "backgroundDisplayEnabled": true,
        /*ro, opt, bool, whether to display background picture, desc:If enabled, when a vehicle is passing, the passing vehicle information will be displayed with the background picture.*/
        "backgroundDisplayTime": 60,
        /*ro, opt, int, display duration of the background picture, range:[1,180], unit:s, desc:If another vehicle is passing, the duration will be refreshed.*/
        "CustomContentList": [
            /*ro, opt, array, custom content list, subType:object, range:[1,8]*/
            {
                "CustomContent": {
                    /*ro, opt, object, custom content*/
                    "id": 1,
                    /*ro, req, int, No., range:[1,8], desc:it increases from 1*/
                    "content": "test",
                    /*ro, opt, string, custom content, range:[1,16]*/
                    "value": "plate",
                    /*ro, opt, enum, options of display content, subType:string, desc:"plate", "parkingLotNum" (remaining parking space), "triggerTime" (vehicle passing time), "vehicleType", "plateType", "vehicleCategory"*/
                    "contentType": "advertisement",
                    /*ro, opt, enum, content type, subType:string, desc:"advertisement", "redLight", "greenLight"*/
                    "fontSizeLine": 1,
                    /*ro, opt, int, font size, range:[1,100]*/
                    "fontColorLine": {
                        /*ro, opt, object, font color*/
                        "R": 1,
                        /*ro, req, int, R, range:[0,255]*/
                        "G": 1,
                        /*ro, req, int, G, range:[0,255]*/
                        "B": 1
                        /*ro, req, int, B, range:[0,255]*/
                    },
                    "fontTypeLine": "Microsoft YaHei",
                    /*ro, opt, enum, font type, subType:string, desc:"Microsoft YaHei", "SimSun", "SimHei", "FangSong", "KaiTi"*/
                    "fontBoldLine": false,
                    /*ro, opt, bool, whether to enable bold font*/
                    "passVehicleDisplayEnabled": false
                    /*ro, opt, bool, whether to display the content when a vehicle is passing, desc:true (display the content when a vehicle is passing), false (display the content for free time), it is false by default.*/
                }
            }
        ],
        "fontSize": 1,
        /*ro, opt, int, font size, range:[1,100]*/
        "fontColor": f
    }
}
```

```

    "fontColor": [
        /*ro, opt, object, font color*/
        "R": 1,
        /*ro, req, int, R, range:[0,255]*/
        "G": 1,
        /*ro, req, int, G, range:[0,255]*/
        "B": 1
        /*ro, req, int, B, range:[0,255]*/
    ],
    "fontMovementSpeed": 20,
    /*ro, opt, int, moving speed threshold of font, range:[0,30], desc:default value: 20*/
    "MediaDataInfo": {
        /*ro, opt, object, settings of playing the media data*/
        "cycleIntervalTime": 1,
        /*ro, req, int, cycle display interval, unit:s*/
        "cyclePlayOrder": [1, 3, 4]
        /*ro, req, array, playing order of cycle display, subType:int*/
    },
    "ctrlMode": "cameraAndplatform",
    /*ro, opt, enum, control mode, subType:string,
    desc:"camera", "platform", "cameraAndplatform" (takes effect in camera and platform mode); If setting the mode is not supported, by default it is valid for both camera and platform.
    Camera (automatic control by device): The device recognizes vehicles in the block and authorized list and temporary license plates by itself, and automatically displays the vehicle passing information according to the strategy configured in displayPassingVehicleInfoEnabled, allowListDisplayEnabled, blockListDisplayEnabled, and temporaryListDisplayEnabled.
    Platform (manual control): Display the content in CustomContentList applied by the user. the mode, The automatic control strategy configured in displayPassingVehicleInfoEnabled, allowListDisplayEnabled, blockListDisplayEnabled, and temporaryListDisplayEnabled will not take effect.
    Camera and platform: the above two control modes take effect at the same time. When CustomContentList is applied, its content is displayed immediately; when the CustomContentList is not applied, it will automatically display the default content on the device.*/
        "fanSpeedLevel": "low",
        /*ro, opt, enum, fan speed gear, subType:string, desc:fan speed level: "low", "middle", "high", "close"*/
        "temperatureThreshold": 1,
        /*ro, opt, int, screen temperature threshold, range:[0,100], unit:°C*/
        "screenWidth": 1,
        /*ro, opt, int, screen width*/
        "screenHeight": 1,
        /*ro, opt, int, screen height*/
        "softwareVersion": "test",
        /*ro, opt, string, Android version of the LCD*/
        "BackBright": {
            /*ro, opt, object, backlight brightness management*/
            "enabled": true,
            /*ro, opt, bool, whether to enable the function*/
            "value": 1,
            /*ro, opt, int, brightness*/
            "timeCtrlEnabled": true,
            /*ro, opt, bool, whether to enable adjustment by time period*/
            "TimeCtrlList": [
                /*ro, opt, array, Time Period Configuration, subType:object, range:[1,8]*/
                {
                    "TimeCtrl": {
                        /*ro, opt, object, time period*/
                        "startTime": "17:30:08",
                        /*ro, req, time, start time*/
                        "endTime": "18:30:08",
                        /*ro, req, time, end time*/
                        "value": 1,
                        /*ro, req, int, brightness, range:[0,100]*/
                        "vehiclePassingValue": 1
                        /*ro, opt, int, passing vehicle brightness, range:[0,100], desc:When a vehicle passes, the brightness is the value of this node; if no vehicle passes, the brightness is the value of the node value. If configuring this node is not supported, the brightness is the value of the node value*/
                    }
                }
            ]
        },
        "SceneUIList": [
            /*ro, opt, array, UI template management of multiple scenes, subType:object*/
            {
                "SceneUI": {
                    /*ro, opt, object, ID of scene UI template*/
                    "scenemode": "entranceWithPlate",
                    /*ro, req, enum, scene type, subType:string, desc:"entranceWithPlate" (UI template type for entered vehicles with license plate), "entranceWithoutPlate" (UI template type for entered vehicles without license plate), "exitingWithoutCharge" (UI template type for exited vehicles with unpaid fee), "exitingWithCharge" (UI template type for exited vehicles without unpaid fee), "free" (free UI template type)*/
                    "UUID": 1
                    /*ro, req, int, ID of scene UI template*/
                }
            ],
            "displayTime": 1,
            /*ro, opt, int, display duration, range:[1,300]*/
            "HintContentList": [
                /*ro, opt, array, list of prompt contents, subType:object, range:[1,2]*/
                {
                    "HintContent": {
                        /*ro, opt, object, prompt content*/
                        "id": 1,
                        /*ro, req, int, No., range:[1,2], desc:it increases from 1*/
                        "content": "test",
                        /*ro, opt, string, prompt content, range:[1,16]*/
                        "value": "plate",
                        /*ro, opt, enum, options of display content, subType:string, desc:"plate", "parkingLotNum" (remaining parking space), "triggerTime" (vehicle passing time), "vehicleType", "plateType", "vehicleCategory"*/
                        "contentType": "free",
                    }
                }
            ]
        ]
    }
}

```

```

/*ro, opt, enum, content type, subType:string, desc:"advertisement", "redLight", "greenLight"*/
"fontSizeLine": 1,
/*ro, opt, int, font size, range:[1,100]*/
"fontColorLine": {
/*ro, opt, object, font color*/
    "R": 1,
    /*ro, req, int, R, range:[0,255]*/
    "G": 1,
    /*ro, req, int, G, range:[0,255]*/
    "B": 1
    /*ro, req, int, B, range:[0,255]*/
},
"fontTypeLine": "Microsoft YaHei",
/*ro, opt, enum, font type, subType:string, desc:"Microsoft YaHei", "SimSun", "SimHei", "FangSong", "KaiTi"*/
"fontBoldLine": true
/*ro, opt, bool, whether to enable bold font*/
}
}
],
"parkingLotDisplayEnable": true,
/*ro, opt, bool, whether to enable the display of vacant parking spaces*/
"sceneMode": "entryScene",
/*ro, opt, enum, scene mode, subType:string, desc:"entryScene", "exitScene", "normalScene"*/
"displayPlateTypeEnable": true,
/*ro, opt, bool, whether to enable the display of the license plate type*/
"displayValidityPeriodEnable": true,
/*ro, opt, bool, whether to enable the display of validity period*/
"mediaOccupyConfigList": [
/*ro, opt, array, parameters of configuring the area for displaying the content when a vehicle is displaying and when it is free time on the LCD screen, subType:object*/
{
    "mediaOccupyMode": "mediaOccupy",
    /*ro, opt, enum, display mode of media information and custom information on LCD screen, subType:string, desc:"mediaOccupy" (only display media information), "customOccupy" (only display custom information configured in the CustomContentList), "mediaRatioOccupy" (display according to the percentage); "mediaOccupy" (only display media information which can be configured via ISAPI/Parking/channels/<channelID>/LCD/mediaData/import), "customOccupy" (only display custom information configured in the CustomContentList), "mediaRatioOccupy" (display according to the percentage)*/
    "mediaOccupyRatio": 0,
    /*ro, opt, int, percentage of area occupied by the media content of the LCD screen, range:[0,100]*/
    "contentType": "free"
    /*ro, opt, enum, content type, subType:string, desc:"free", "passingVehicle"*/
}
],
"logoDisplayEnable": true,
/*ro, opt, bool, whether to display the logo which can be configured via ISAPI/Parking/channels/<channelID>/LCD/mediaData/import*/
"SceneUI": {
/*ro, opt, object, object, UI template of multiple scenes*/
    "id": 1,
    /*ro, req, int, No. which starts from 1, value range: [1,4], desc:1, 2, 3*/
    "RegionList": [
/*ro, opt, array, template area list, subType:object*/
{
    "Region": {
        "id": 1,
        /*ro, opt, object, area*/
        "trafficLightEnabled": true,
        /*ro, opt, bool, whether to switch the traffic light*/
        "contentEnabled": true,
        /*ro, opt, bool, advertisement text*/
        "pictureEnabled": true,
        /*ro, opt, bool, advertisement picture*/
        "videoEnabled": true,
        /*ro, opt, bool, advertisement video*/
        "time": 1,
        /*ro, opt, int, display interval, unit:s*/
        "previewEnabled": true,
        /*ro, opt, bool, live view*/
        "localTimeEnabled": true
        /*ro, opt, bool, display local time*/
    }
}
]
},
"characterMovementMode": "scrolling"
/*ro, opt, enum, subType:string*/
}
}

```

12.9.2.8 Set the parameters for displaying a specified scene on the LED screen

Request URL

PUT /ISAPI/Parking/channels/<channelID>/LEDConfigurations/multiScene/<SID>?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
SID	string	--

Request Message

```
{
  "SingleSceneLEDConfigurations": {
    /*opt, object*/
    "sid": 1,
    /*req, int*/
    "mode": "passingVehicle",
    /*req, enum, subType:string*/
    "showFreeEnabled": true,
    /*opt, bool, dep:and, ${$.MultiSceneLEDConfigurations[*].SingleSceneLEDConfigurations.mode,eq,passingVehicle}*/
    "displayTime": 1,
    /*opt, int, range:[1,300], unit:s, dep:and, ${$.MultiSceneLEDConfigurations[*].SingleSceneLEDConfigurations.mode,eq,passingVehicle}*/
    "vehicleDisplayEnabled": true,
    /*opt, bool*/
    "LEDConfigurationList": [
      /*opt, array, subType:object*/
      {
        "LEDConfiguration": {
          /*opt, object*/
          "id": 1,
          /*req, int*/
          "enabled": true,
          /*req, bool*/
          "ShowInfoList": [
            /*opt, array, subType:object*/
            {
              "ShowInfo": {
                /*opt, object*/
                "id": 1,
                /*req, int*/
                "fontSize": 16,
                /*opt, enum, subType:int*/
                "fontColor": "red",
                /*opt, enum, subType:string*/
                "speedType": "fast",
                /*req, enum, subType:string*/
                "displayMode": "left",
                /*req, enum, subType:string*/
                "LineInfoList": [
                  /*opt, array, subType:object*/
                  {
                    "LineInfo": {
                      /*opt, object*/
                      "id": 1,
                      /*req, int*/
                      "value": "plate",
                      /*opt, enum, subType:string*/
                      "customValue": "欢迎光临"
                      /*opt, string*/
                    }
                  }
                ]
              }
            }
          ]
        }
      ],
    }
  }
}
```

Response Message

```
{
  "statusCode": 1,
  /*ro, opt, int*/
  "statusString": "ok",
  /*ro, opt, string, range:[1,64]*/
  "subStatusCode": "ok",
  /*ro, opt, string, range:[1,64]*/
  "errorCode": 1,
  /*ro, opt, int*/
  "errorMsg": "ok"
  /*ro, opt, string*/
}
```

12.9.2.9 Get the parameters for displaying a specified scene on the LED screen

Request URL

GET /ISAPI/Parking/channels/<channelID>/LEDConfigurations/multiScene/<SID>?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	channel No.
SID	string	scene No.

Request Message

None

Response Message

```
{
    "SingleSceneLEDConfigurations": {
        /*ro, opt, object, the parameters for displaying a specified scene on the LED screen*/
        "sid": 1,
        /*ro, req, int, scene ID*/
        "mode": "passingVehicle",
        /*ro, req, enum, scene mode, subType:string, desc:"passingVehicle", "noVehicle"*/
        "showFreeEnabled": true,
        /*ro, opt, bool, whether to enable displaying in free time, dep:and,
        {$ .MultiSceneLEDConfigurations[*].SingleSceneLEDConfigurations.mode,eq,passingVehicle}*/
        "displayTime": 1,
        /*ro, opt, int, display duration, range:[1,300], unit:s, dep:and,
        {$ .MultiSceneLEDConfigurations[*].SingleSceneLEDConfigurations.mode,eq,passingVehicle}*/
        "vehicleDisplayEnabled": true,
        /*ro, opt, bool, whether enable automatic displaying of passing vehicle information*/
        "allowListDisplayEnabled": false,
        /*ro, opt, bool, whether to automatically display the custom content configured by the user whose vehicle is in the authorized list, desc:If enabled, if the passing vehicle is in the authorized list, the device will automatically overwrite the custom content configured by the user. If not enabled, the display content will not be changed.*/
        "blockListDisplayEnabled": false,
        /*ro, opt, bool, whether to automatically display the custom content configured by the user whose vehicle is in the unauthorized list, desc:If enabled, if the passing vehicle is in the unauthorized list, the device will automatically overwrite the custom content configured by the user. If not enabled, the display content will not be changed.*/
        "temporaryListDisplayEnabled": false,
        /*ro, opt, bool, whether to automatically display the custom content configured by the user whose vehicle is a temporary vehicle, desc:If enabled, if the passing vehicle is a temporary vehicle and is not in the unauthorized/authorized list, the device will automatically overwrite the custom content configured by the user. If not enabled, the display content will not be changed.*/
        "LEDConfigurationList": [
            /*ro, opt, array, the parameters list of LED screen, subType:object*/
            {
                "LEDConfiguration": {
                    /*ro, opt, object, the parameters of LED screen*/
                    "id": 1,
                    /*ro, req, int, screen ID*/
                    "enabled": true,
                    /*ro, req, bool, enable displaying multiple lines on LED screen*/
                    "ShowInfoList": [
                        /*ro, opt, array, display content, subType:object*/
                        {
                            "ShowInfo": {
                                /*ro, opt, object, display content*/
                                "id": 1,
                                /*ro, req, int, row No., desc:it increases from 1*/
                                "fontSize": 16,
                                /*ro, opt, enum, font size, subType:int, desc:16 (16 × 16 dot matrix), 32 (32 × 32 dot matrix)*/
                                "fontColor": "red",
                                /*ro, opt, enum, font color, subType:string, desc:"red", "green", "yellow"*/
                                "speedType": "fast",
                                /*ro, req, enum, speed type, subType:string, desc:"fast", "medium", "slow"*/
                                "displayMode": "left",
                                /*ro, req, enum, display mode, subType:string, desc:"left", "right", "instant"*/
                                "LineInfoList": [
                                    /*ro, opt, array, the list of combination of rows, subType:object*/
                                    {
                                        "LineInfo": {
                                            /*ro, opt, object, combination of rows, desc:value and customValue are mutually exclusive.*/
                                            "id": 1,
                                            /*ro, req, int, No., desc:it increases from 1*/
                                            "value": "plate",
                                            /*ro, opt, enum, options of display content, subType:string, desc:"plate", "YYYY-MM-DD", "hh-mm-ss"*/
                                            "customValue": "欢迎光临"
                                            /*ro, opt, string, custom display contents*/
                                        }
                                    }
                                ]
                            }
                        }
                    ]
                }
            }
        ]
    }
}
```

```

        }
    ],
    "LEDGeneralConfiguration": {
        /*ro, opt, object, the general parameters of LED screen, desc:The parameters of LEDConfigurationList will take effect if LEDConfigurationList is configured.*/
        "FontSize": 16,
        /*ro, opt, enum, font size, subType:int, desc:16 (16 × 16 dot matrix), 32 (32 × 32 dot matrix)*/
        "FontColor": "red",
        /*ro, opt, enum, font color, subType:string, desc:"red", "green", "yellow"*/
        "Frequency": 5,
        /*ro, req, int, the frequency of screen refresh, desc:it increases from 1*/
        "FreeTimeShowInfo": {
            /*ro, opt, object, display in free time*/
            "CustomContentList": [
                /*ro, opt, array, content list displayed in free time, subType:object*/
                {
                    "CustomContent": {
                        /*ro, opt, object, display content in free time*/
                        "id": 1,
                        /*ro, req, int, No., range:[1,5], desc:it increases from 1*/
                        "content": "test"
                        /*ro, opt, string, display content, desc:the maximum size is 40*/
                    }
                }
            ],
            "TimeBlockList": [
                /*ro, opt, array, parameters for time control configuration, subType:object*/
                {
                    "TimeBlock": {
                        /*ro, opt, object, time period*/
                        "id": 1,
                        /*ro, opt, int, No.*/
                        "beginTime": "00:00",
                        /*ro, opt, string, start time, desc:control in different times during 24 hours, accurate to the minute, format: hh:MM*/
                        "endTime": "24:00"
                        /*ro, opt, string, end time, desc:control in different times during 24 hours, accurate to the minute, format: hh:MM*/
                    }
                }
            ],
            "showTime": 15
            /*ro, req, int, display duration, range:[1,30]*/
        },
        "IllegalShowInfo": {
            /*ro, opt, object, violation information for display*/
            "IllegalList": [
                /*ro, opt, array, violation information list for display, subType:object*/
                {
                    "IllegalContent": {
                        /*ro, opt, object, violation information for display*/
                        "id": 1,
                        /*ro, req, int, No., range:[1,5], desc:it increases from 1*/
                        "content": "test"
                        /*ro, opt, string, display content, desc:the maximum size is 40*/
                        "contentType": "nonHelmet"
                        /*ro, opt, enum, content type, subType:string, desc:"nonHelmet", "bicycleManned" (carrying passengers on non-motor vehicle),
                    }
                }
            ],
            "TimeBlockList": [
                /*ro, opt, array, whether to enable parameters of configuring time control, subType:object*/
                {
                    "TimeBlock": {
                        /*ro, opt, object, time period*/
                        "id": 1,
                        /*ro, opt, int, No.*/
                        "beginTime": "00:00",
                        /*ro, opt, string, start time, desc:control in different times during 24 hours, accurate to the minute, format: hh:MM*/
                        "endTime": "24:00"
                        /*ro, opt, string, end time, desc:control in different times during 24 hours, accurate to the minute, format: hh:MM*/
                    }
                }
            ],
            "showTime": 15
            /*ro, req, int, display times, range:[1,30]*/
        },
        "speedMode": "slowMode"
        /*ro, opt, enum, speed mode, subType:string, desc:"fastMode", "slowMode"*/
    }
}

```

12.9.2.10 Get the capability of configuring the parameters for displaying a specified scene on the LED screen

Request URL

GET /ISAPI/Parking/channels/<channelID>/LEDConfigurations/multiScene/capabilities?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	channel No.

Request Message

None

Response Message

```
{
    "MultiSceneLEDConfigurationsCap": {
        /*ro, opt, object, the capability of configuring the parameters for displaying multiple scenes on the LED screen*/
        "SingleSceneLEDConfigurations": {
            /*ro, opt, object, configuring the parameters for displaying a specified scene on the LED screen*/
            "sid": {
                /*ro, req, object, scene ID*/
                "#maxSize": 2
                /*ro, req, int, the maximum number of scenes*/
            },
            "mode": {
                /*ro, req, object, scene mode*/
                "@opt": ["passingVehicle", "noVehicle"],
                /*ro, req, array, options, subType:string, desc:"passingVehicle", "noVehicle"*/
                "#text": "passingVehicle"
                /*ro, opt, string, the sample value*/
            },
            "showFreeEnabled": {
                /*ro, opt, object, whether to enable vacancy display mode, desc:it is valid when the value of mode is "passingVehicle" (passing-vehicle scene)*/
                "@opt": [true, false]
                /*ro, req, array, options, subType:bool*/
            },
            "displayTime": {
                /*ro, opt, object, display duration, desc:it is valid when the value of mode is "passingVehicle"*/
                "@min": 1,
                /*ro, req, int, the minimum value*/
                "@max": 300
                /*ro, req, int, the maximum value*/
            },
            "LEDConfigurationList": {
                /*ro, opt, object, the list of LED configuration parameters*/
                "LEDConfiguration": {
                    /*ro, opt, object, LED configuration parameters*/
                    "id": {
                        /*ro, req, object, display screen ID, desc:it increases from 1*/
                        "#maxSize": 1
                        /*ro, req, int, the maximum number of scenes*/
                    },
                    "enabled": {
                        /*ro, req, object, whether to enable the LED screen to display multiple lines*/
                        "@opt": [true, false]
                        /*ro, req, array, options, subType:bool*/
                    },
                    "ShowInfoList": {
                        /*ro, opt, object, list of display information*/
                        "ShowInfo": {
                            /*ro, opt, object, display information*/
                            "id": {
                                /*ro, req, object, it increases from 1*/
                                "#maxSize": 4
                                /*ro, req, int, the maximum number of rows*/
                            },
                            "fontSize": {
                                /*ro, opt, object, font size*/
                                "@opt": ["16", "32"]
                                /*ro, opt, array, options, subType:string, desc:16 (16 × 16 dot matrix), 32 (32 × 32 dot matrix)*/
                            },
                            "fontColor": {
                                /*ro, opt, object, font color*/
                                "@opt": ["red", "green", "yellow"],
                                /*ro, req, array, options, subType:string, desc:red, green, yellow, off*/
                                "#text": "red"
                                /*ro, opt, string, the sample value*/
                            },
                            "speedType": {
                                /*ro, req, object, speed type*/
                                "@opt": ["fast", "medium", "slow"],
                                /*ro, req, array, options, subType:string, desc:fast, medium, slow*/
                                "#text": "fast"
                                /*ro, opt, string, the sample value*/
                            },
                            "displayMode": {
                                /*ro, req, object, display mode*/
                                "@opt": ["left", "right", "instant"],
                                /*ro, req, array, options, subType:string, desc:left, right, instant*/
                                "#text": "left"
                                /*ro, opt, string, the sample value*/
                            },
                            "LineInfoList": {
                                /*ro, opt, object, the list of line information*/
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```
/*ro, opt, object, the list of line information*/
"LineInfo": {
    /*ro, opt, object, line information, desc:value and customValue cannot be configured simultaneously, and one ID can only be
configured with one of these two parameters*/
    "id": {
        /*ro, req, object, index, desc:it increases from 1*/
        "@maxSize": 4
        /*ro, req, int, the maximum number of rows*/
    },
    "value": {
        /*ro, opt, object, display information*/
        "@opt": ["plate", "YYYY-MM-DD", "hh-mm-ss"],
        /*ro, req, array, options, subType:string, desc:"plate", "YYYY-MM-DD", "hh-mm-ss"*/
        "#text": "plate"
        /*ro, opt, string, the sample value*/
    },
    "customValue": {
        /*ro, opt, object, custom display information*/
        "@min": 1,
        /*ro, req, int, the minimum value*/
        "@max": 100
        /*ro, req, int, the maximum value*/
    }
},
},
},
},
},
}
}
```

12.9.2.11 Create screen parameters

Request URL

POST /ISAPI/Parking/server/deviceManagement/led?format=json

Query Parameter

None

Request Message

```

{
    "LedInfo": {
        /*opt, object, screen information*/
        "deviceID": "test",
        /*opt, string, unique device ID*/
        "deviceType": "led",
        /*opt, string, device type*/
        "deviceName": "显示屏",
        /*opt, string, device name*/
        "displayType": "enterLocation",
        /*opt, enum, display type, subType:string, desc:"enterLocation" (screen for entrance location), "enterPrompt" (screen for prompt at entrance),
        "exitCharge" (screen for charge at exit)*/
        "communicateMode": "serial",
        /*req, enum, communication mode, subType:string, desc:"serial", "network"*/
        "SerialCtrlInfo": {
            /*opt, object, serial port control information*/
            "comNo": 1,
            /*req, int, serial port No.*/
            "ctrlCode": 1
            /*req, int, control code*/
        },
        "NetworkCtrlInfo": {
            /*opt, object, network control information*/
            "ipaddr": "10.10.112.250",
            /*req, string, IP address*/
            "portNo": 1
            /*opt, int, port No., range:[0,65535], step:1*/
        },
        "Params": {
            /*opt, object, parameters*/
            "lcdType": "monochrome",
            /*req, enum, screen type, subType:string, desc:"monochrome", "twocolor", "colorful"*/
            "ScreenSize": {
                /*opt, object, screen size*/
                "height": 1,
                /*req, int, height*/
                "width": 1
                /*req, int, width*/
            },
            "color": "red",
            /*req, enum, color of prompt content, subType:string, desc:"red", "green", "yellow", "blue", "pink", "white", "cyan"*/
            "scrolltype": "none",
            /*req, enum, scroll type, subType:string, desc:"none" (not scroll), "leftScroll" (scroll left), "rightScroll" (scroll right), "upScroll" (scroll
            up), "downScroll" (scroll down)*/
            "isfreeShowLeft": true,
            /*req, bool, display vacant parking spaces in free time*/
            "textSize": 1,
            /*req, int, font size*/
            "FixedContent": {
                /*opt, object, fixed content*/
                "enabled": true,
                /*req, bool, whether to enable displaying fixed content*/
                "istimeShow": true,
                /*req, bool, whether to enable displaying the time*/
                "content": "test"
                /*req, string, fixed display content*/
            }
        },
        "NoticeInfo": [
            /*opt, array, prompt content, subType:object*/
            {
                "line": 1,
                /*req, int, number of lines of content*/
                "textType": "plateNum",
                /*req, string, prompt content, desc:supports selecting multiple content types, "plateNum" (plate number), "cardNum" (card number),
                "vehicleType" (vehicle type), "parkingType" (parking type), "timeYearMonthDay" (year-month-day), "timeHourMinSec" (hour-min-sec), "leftParkSpace" (vacant
                parking spaces), "validTime" (validity)*/
                "textColor": "red",
                /*req, enum, color of prompt content, subType:string, desc:"red", "green", "yellow", "blue", "pink", "white", "cyan"*/
                "textSize": "test",
                /*req, string, font size of prompt content*/
                "scrolltype": "none"
                /*req, enum, scroll type, subType:string, desc:"none", "leftScroll", "rightScroll", "upScroll", "downScroll"*/
            }
        ],
        "fullControl": "none"
        /*opt, enum, full-screen control, subType:string, desc:"none" (no control), "forceShowFull" (force showing no more vehicles allowed)*/
    }
}

```

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*/
}

```

12.9.2.12 Configure screen parameters

Request URL

PUT /ISAPI/Parking/server/deviceManagement/led?format=json&id=<deviceID>

Query Parameter

Parameter Name	Parameter Type	Description
deviceID	string	unique device ID

Request Message

```

{
    "LedInfo": {
        /*opt, object, screen information*/
        "deviceID": "test",
        /*opt, string, unique device ID*/
        "deviceType": "led",
        /*opt, string, device type*/
        "deviceName": "显示屏",
        /*opt, string, device name*/
        "displayType": "enterLocation",
        /*opt, enum, display type, subType:string, desc:"enterLocation" (screen for entrance location), "enterPrompt" (screen for prompt at entrance),
        "exitCharge" (screen for charge at exit)*/
        "communicateMode": "serial",
        /*req, enum, communication mode, subType:string, desc:"serial", "network"*/
        "SerialCtrlInfo": {
            /*opt, object, serial port control information*/
            "comNo": 1,
            /*req, int, serial port No.*/
            "ctrlCode": 1
            /*req, int, control code*/
        },
        "networkCtrlInfo": {
            /*opt, object, network control information*/
            "ipaddr": "10.10.112.250",
            /*req, string, IP address*/
            "portNo": 1
            /*opt, int, port No., range:[0,65535], step:1*/
        },
        "Params": {
            /*opt, object, parameters*/
            "lcdType": "monochrome",
            /*req, enum, screen type, subType:string, desc:"monochrome", "twocolor", "colorful"*/
            "ScreenSize": {
                /*opt, object, screen size*/
                "height": 1,
                /*req, int, height*/
                "width": 1
                /*req, int, width*/
            },
            "color": "red",
            /*req, enum, color of prompt content, subType:string, desc:"red", "green", "yellow", "blue", "pink", "white", "cyan"*/
            "scrolltype": "none",
            /*req, enum, scroll type, subType:string, desc:"none", "leftScroll", "rightScroll", "upScroll", "downScroll"*/
            "isfreeShowLeft": true,
            /*req, bool, display vacant parking spaces in free time*/
            "textSize": 1,
            /*req, int, font size*/
            "FixedContent": {
                /*opt, object, fixed content*/
                "enabled": true,
                /*req, bool, whether to enable displaying fixed content*/
                "istimeShow": true,
                /*req, bool, whether to enable displaying the time*/
                "content": "test"
                /*req, string, fixed display content*/
            }
        },
        "NoticeInfo": [
            /*opt, array, prompt content, subType:object*/
            {
                "line": 1,
                /*req, int, number of lines of content*/
                "textType": "plateNum",
                /*req, string, prompt content, desc:supports selecting multiple content types, "plateNum" (plate number), "cardNum" (card number),
                "vehicleType" (vehicle type), "parkingType" (parking type), "timeYearMonthDay" (year-month-day), "timeHourMinSec" (hour-min-sec), "leftParkSpace" (vacant
                parking spaces), "validTime" (validity)*/
                "textColor": "red",
                /*req, enum, color of prompt content, subType:string, desc:"red", "green", "yellow", "blue", "pink", "white", "cyan"*/
                "textSize": "test",
                /*req, string, font size of prompt content*/
                "scrolltype": "none"
                /*req, enum, scroll type, subType:string, desc:"none", "leftScroll", "rightScroll", "upScroll", "downScroll"*/
            }
        ],
        "fullControl": "none"
        /*opt, enum, full-screen control, subType:string, desc:"none" (no control), "forceShowFull" (force showing no more vehicles allowed)*/
    }
}

```

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*/
}

```

12.9.2.13 Get screen parameters

Request URL

GET /ISAPI/Parking/server/deviceManagement/led?format=json&id=<deviceID>

Query Parameter

Parameter Name	Parameter Type	Description
deviceID	string	unique device ID

Request Message

None

Response Message

```
{
    "LedInfo": {
        /*ro, opt, object, screen information*/
        "deviceID": "test",
        /*ro, opt, string, unique device ID*/
        "deviceType": "led",
        /*ro, opt, string, device type*/
        "deviceName": "显示屏",
        /*ro, opt, string, device name*/
        "displayType": "enterLocation",
        /*ro, opt, enum, display type, subType:string, desc:"enterLocation" (screen for entrance location), "enterPrompt" (screen for prompt at entrance),
        "exitCharge" (screen for charge at exit)*/
        "communicateMode": "serial",
        /*ro, req, enum, communication mode, subType:string, desc:"serial", "network"*/
        "SerialCtrlInfo": {
            /*ro, opt, object, serial port control information*/
            "comNo": 1,
            /*ro, req, int, serial port No.*/
            "ctrlCode": 1
            /*ro, req, int, control code*/
        },
        "networkCtrlInfo": {
            /*ro, opt, object, network control information*/
            "ipaddr": "10.10.112.250",
            /*ro, req, string, IP address*/
            "portNo": 1
            /*ro, opt, int, port No., range:[0,65535], step:1*/
        },
        "Params": {
            /*ro, opt, object, parameters*/
            "lcdType": "monochrome",
            /*ro, req, enum, screen type, subType:string, desc:"monochrome", "twocolor", "colorful"*/
            "ScreenSize": {
                /*ro, opt, object, screen size*/
                "height": 1,
                /*ro, req, int, height*/
                "width": 1
                /*ro, req, int, Width.*/
            },
            "color": "red",
            /*ro, req, enum, color of prompt content, subType:string, desc:"red", "green", "yellow", "blue", "pink", "white", "cyan"*/
            "scrolltype": "none",
            /*ro, req, enum, scroll type, subType:string, desc:"none", "leftScroll", "rightScroll", "upScroll", "downScroll"*/
            "isfreeShowLeft": true,
            /*ro, req, bool, display vacant parking spaces in free time*/
            "textSize": 1,
            /*ro, req, int, font size*/
            "FixedContent": {
                /*ro, opt, object, fixed content*/
                "enabled": true,
                /*ro, req, bool, whether to enable displaying fixed content*/
                "istimeShow": true,
                /*ro, req, bool, whether to display the time*/
                "content": "test"
                /*ro, req, string, fixed display content*/
            }
        },
        "NoticeInfo": [
            /*ro, opt, array, prompt content, subType:object*/
            {
                "line": 1,
                /*ro, req, int, number of lines of content*/
                "textType": "plateNum",
                /*ro, req, string, prompt content, desc:supports selecting multiple content types, "plateNum", "cardNum", "vehicleType", "parkingType",
                "timeYearMonthDay", "timeHourMinSec", "leftParkSpace" (vacant parking spaces), "validTime"*/
                "textColor": "red",
                /*ro, req, enum, color of prompt content, subType:string, desc:"red", "green", "yellow", "blue", "pink", "white", "cyan"*/
                "textSize": "test",
                /*ro, req, string, font size of prompt content*/
                "scrolltype": "none"
                /*ro, req, enum, scroll type, subType:string, desc:"none", "leftScroll", "rightScroll", "upScroll", "downScroll"*/
            }
        ],
        "fullControl": "none"
        /*ro, opt, enum, full-screen control, subType:string, desc:"none" (no control), "forceShowFull" (force showing no more vehicles allowed)*/
    }
}
```

12.9.2.14 Get the capability set of displaying information on the LCD screen

Request URL

GET /ISAPI/System/LCDScreen/capabilities?format=json

Query Parameter

None

Request Message

None

Response Message

```
{
    "LCDScreenCap": {
        /*ro, opt, object, capability set of displaying information on the LCD screen*/
        "DisplayInfoCap": {
            /*ro, opt, object, capability of the LCD screen*/
            "QRCodeImg": {
                /*ro, opt, object, QR code information*/
                "QRCodeNum": "1",
                /*ro, opt, string, the maximum number of QR code pictures that can be displayed*/
                "QRCodeImgFormat": {
                    /*ro, opt, object, payment method and picture type of the QR code defined by the file name and the extension name*/
                    "@opt": "weixin.jpg,zhifubao.bmp"
                    /*ro, req, string, optional values*/
                },
                "paymentType": {
                    /*ro, opt, object, payment method (multiple methods can be separated by commas)*/
                    "@opt": "WeChatPay,Alipay"
                    /*ro, req, string, optional values*/
                }
            },
            "sence": {
                /*ro, opt, object, scene*/
                "@opt": "1,2,3,4,5"
                /*ro, req, string, optional values, desc:"1" (entered vehicles with license plate), "2" (entered vehicles without license plate), "3" (exited vehicles with license plate and without unpaid fee), "4" (exited vehicles with license plate and unpaid fee), "5" (exited vehicles without license plate)*/
            },
            "amounts": true,
            /*ro, opt, bool, whether it supports displaying the account*/
            "licenseLen": 64,
            /*ro, opt, int, the maximum string length of the license plate characters, desc:If this node exists, it indicates that displaying the license plate information is supported*/
            "phoneNumberLen": 20,
            /*ro, opt, int, the maximum string length of the phone No., desc:If this node exists, it indicates that displaying the contact information is supported*/
            "noticeLen": 256,
            /*ro, opt, int, the maximum string length of the notice information*/
            "enterTime": true,
            /*ro, opt, bool, whether it supports displaying the entering time*/
            "leaveTime": true,
            /*ro, opt, bool, whether it supports displaying the exiting time*/
            "customInfoLen": 128,
            /*ro, opt, int, the maximum string length of the displayed custom information, desc:If this node exists, it indicates that displaying the custom information is supported*/
            "plateType": {
                /*ro, opt, object, license plate type*/
                "@opt": "fixedPlate,temporaryPlate,blackPlate"
                /*ro, req, string, optional values, desc:"fixedPlate" (registered vehicle), "temporaryPlate" (temperatory vehicle). "blackPlate" (vehicles in the blocklist)*/
            },
            "validityPeriod": {
                /*ro, opt, object, validity period*/
                "@min": 0,
                /*ro, req, int, the minimum value*/
                "@max": 29
                /*ro, req, int, the maximum value*/
            },
            "remainingCharge": "30",
            /*ro, opt, string, account balance*/
            "parkingLot": 2,
            /*ro, opt, int, number of vacant parking spaces*/
            "titleLen": 128,
            /*ro, opt, int, the maximum string length of the title, desc:if this node exists, it indicates that setting the title is supported*/
            "laneID": {
                /*ro, opt, object*/
                "@min": 1,
                /*ro, req, int*/
                "@max": 32
                /*ro, req, int*/
            }
        }
    }
}
```

Parameter Name	Parameter Value	Parameter Type(Content-Type)	Content-ID	File Name	Description
	[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.

12.9.2.15 Configure the information to be displayed on the screen.

Request URL

POST /ISAPI/System/LCDScreen/displayInfo?format=json

Query Parameter

None

Request Message

```
{
    "DisplayInfo": {
        /*opt, object, displayed information*/
        "QRCodeImg": [
            /*opt, array, array of object, list of QR codes, subType:object*/
            {
                "QRCodeBase64": "abcdeadfadsf==",
                /*req, string, QR code encoded via Base64*/
                "QRCodeFileName": "weixin.jpg",
                /*req, string, picture format of the QR code, desc:"weixin.jpg" (WeChat), "zhifubao.bmp" (Alipay)*/
                "paymentType": "WeChatPay,Alipay",
                /*req, string, payment method, desc:"WeChatPay", "Alipay". Multiple methods can be separated by commas*/
                "QRContent": "http://"
            }
        ],
        "sense": 1,
        /*req, enum, scene, subType:int, desc:1 (entered vehicles with license plate),2 (entered vehicles without license plate),3 (exited vehicles without unpaid fee),4 (exited vehicles with unpaid fee),5 (exited vehicles without license plate),6 (the fee is paid),7 (exited vehicles without license plate and unpaid fee),8 (exited vehicles without license plate and with unpaid fee),9 (paying failed),10 (free). This node is used to control the content and layout displayed on the screen*/
        "amounts": 30.0,
        /*opt, float, float,amount, desc:float,amount*/
        "license": "ABC12345",
        /*opt, string, license plate number*/
        "phoneNumber": "13012345678",
        /*opt, string, phone number*/
        "notice": "abcdefg",
        /*opt, string, notice information*/
        "enterTime": "1970-01-01T00:00:00+08:00",
        /*opt, datetime, entering time*/
        "leaveTime": "1970-01-01T00:00:00+08:00",
        /*opt, datetime, exiting time*/
        "customInfo": "abcdefg",
        /*opt, string, displayed custom information*/
        "plateType": "fixedPlate",
        /*opt, enum, license plate type, subType:string, desc:"fixedPlate" (registered vehicle), "temporaryPlate" (temporary vehicle), "blackPlate" (vehicles in the blocklist)*/
        "validityPeriod": "2020年11月12日-2020年12月11日",
        /*opt, string, validity period*/
        "remainingCharge": "30",
        /*opt, string, the account balance, desc:the account balance*/
        "parkingLot": 2,
        /*opt, int, number of vacant parking spaces*/
        "tideLaneStatus": "opening",
        /*opt, enum, reversible lane status, subType:string, desc:"opening", "closed"*/
        "title": "abc",
        /*opt, string, displayed title*/
        "laneID": "1"
        /*opt, string, range:[1,32]*/
    }
}
```

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*/
}
```

12.9.3 Parking Lot Barrier Gate

12.9.3.1 Set the barrier gate control parameters supported by the device to open, close, lock or unlock the barrier gate

Request URL

PUT /ISAPI/Parking/channels/<channelID>/barrierGate

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

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

<BarrierGate xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, object, barrier gate control, attr:version{req, string, protocolVersion}-->
    <ctrlMode opt="open,close,lock,unlock">
        <!--req, enum, barrier gate control mode, subType:string, attr:opt{req, string}, desc:"open" (open the barrier gate), "close" (close the barrier gate), "Lock" (lock the barrier gate), "unlock" (unlock the barrier gate)-->open
    </ctrlMode>
    <cardNo>
        <!--opt, string, range:[1,64]-->1234567890
    </cardNo>
</BarrierGate>
```

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, opt, 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 description-->OK
    </subStatusCode>
    <description>
        <!--ro, opt, string, range:[0,1024]-->badXmlFormat
    </description>
    <MErrCode>
        <!--ro, opt, string-->0x00000000
    </MErrCode>
    <MErrDevSelfEx>
        <!--ro, opt, string-->0x00000000
    </MErrDevSelfEx>
</ResponseStatus>
```

12.9.3.2 Get the entrance/exit barrier gate status

Request URL

GET /ISAPI/Parking/channels/<channelID>/barrierGate/barrierGateStatus

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

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

<BarrierGate xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, entrance/exit barrier gate, attr:version{req, string, protocolVersion}-->
  <barrierGateStatus>
    <!--ro, opt, enum, barrier gate status, subType:int, desc:0 (no signal), 1 (closed), 2 (open)-->1
  </barrierGateStatus>
</BarrierGate>
```

12.9.3.3 Get the capability of controlling the barrier gate

Request URL

GET /ISAPI/Parking/channels/<channelID>/barrierGate/capabilities

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

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

<BarrierGate xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <ctrlMode opt="open,close,lock,unlock">
    <!--ro, req, enum, control mode, subType:string, attr:opt{req, string}, desc:control mode-->open
  </ctrlMode>
  <isSupportBarrierGateStatus>
    <!--ro, opt, bool, whether it supports getting the barrier gate status-->true
  </isSupportBarrierGateStatus>
  <cardNo min="1" max="64">
    <!--ro, opt, string, range:[1,64], attr:min{opt, string},max{opt, string}-->123456789
  </cardNo>
</BarrierGate>
```

12.9.4 Parking Lot

12.9.4.1 Get the parking lot service capability

Request URL

GET /ISAPI/Parking/capabilities

Query Parameter

None

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ParkingCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, parking lot service capability, attr:version{req, string, protocolVersion}-->
  <isSupportTME>
    <!--ro, opt, bool, whether it supports TME nodes-->true
  </isSupportTME>
  <isSupportLampCtrl>
    <!--ro, opt, bool, whether it supports configuring and controlling parking space indicators-->true
  </isSupportLampCtrl>
  <isSupportParkingStatus>
    <!--ro, opt, bool, whether it supports getting parking space status-->true
  </isSupportParkingStatus>
  <isSupportPicDevision>
    <!--ro, opt, bool, whether it supports parking space picture division-->true
  </isSupportPicDevision>
  <isSupportAlterCtrlLampCap>
    <!--ro, opt, bool, whether it supports alternate indicator control-->true
  </isSupportAlterCtrlLampCap>
  <isSupportParkingSpace>
    <!--ro, opt, bool, whether it supports parking area configuration-->true
  </isSupportParkingSpace>
  <isSupportVideoParkingPole>
    <!--ro, opt, bool, whether it supports parking video pole-->true
  </isSupportVideoParkingPole>
  <isSupportBollard>
    <!--ro, opt, bool-->true
  </isSupportBollard>
  <isSupportLEDConfigurationsMultiScene>
    <!--ro, opt, bool-->true
  </isSupportLEDConfigurationsMultiScene>
  <isSupportVoiceCombinantBroadcast>
    <!--ro, opt, bool-->true
  </isSupportVoiceCombinantBroadcast>
  <isSupportLCD>
    <!--ro, opt, bool-->true
  </isSupportLCD>
  <isSupportPackingSpaceRecognition>
    <!--ro, opt, bool, whether it supports parking space detection, desc:corresponding URL:/ISAPI/Parking/channels/<channelID>/packingSpaceRecognition?format=json-->true
  </isSupportPackingSpaceRecognition>
  <isSupportPackingSpaceIndicatorLamp>
    <!--ro, opt, bool-->true
  </isSupportPackingSpaceIndicatorLamp>
  <isSupportManualPackingLock>
    <!--ro, opt, bool-->true
  </isSupportManualPackingLock>
  <isSupportAutoPackingLock>
    <!--ro, opt, bool-->true
  </isSupportAutoPackingLock>
  <isSupportMultiPackingSpaceIndicatorLamp>
    <!--ro, opt, bool-->true
  </isSupportMultiPackingSpaceIndicatorLamp>
</ParkingCap>

```

12.9.4.2 Get the capability of configuring the parameters of combined voice broadcasts

Request URL

GET /ISAPI/Parking/channels/<channelID>/voiceBroadcastInfo/combinantBroadcast/capabilities?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```
{
  "CombinantBroadcastCap": {
    /*ro, opt, object, configuration capability of combined broadcast*/
    "enabled": {
      /*ro, req, object, whether to enable combined broadcasts*/
      "@opt": [true, false]
      /*ro, req, array, range, subType:bool*/
    },
    "volume": {
      /*ro, opt, object, voice volume*/
      "@min": 0,
      /*ro, req, int, the minimum value*/
      "@max": 10
    }
  }
}
```

```

    /*ro, req, int, the maximum value*/
},
"ctrlMode": {
/*ro, opt, object*/
    "@opt": ["camera", "platform", "cameraAndplatform"],
    /*ro, req, array, subType:string*/
    "#text": "camera"
    /*ro, opt, string*/
},
"vehicleBroadcastEnabled": {
/*ro, opt, object*/
    "@opt": [true, false],
    /*ro, req, array, range, subType:bool*/
    "@def": false
    /*ro, opt, bool, the default value*/
},
"allowListBroadcastEnabled": {
/*ro, opt, object*/
    "@opt": [true, false],
    /*ro, req, array, range, subType:bool*/
    "@def": false
    /*ro, opt, bool, the default value*/
},
"blockListBroadcastEnabled": {
/*ro, opt, object*/
    "@opt": [true, false],
    /*ro, req, array, range, subType:bool*/
    "@def": false
    /*ro, opt, bool, the default value*/
},
"temporaryListBroadcastEnabled": {
/*ro, opt, object*/
    "@opt": [true, false],
    /*ro, req, array, range, subType:bool*/
    "@def": false
    /*ro, opt, bool, the default value*/
},
"BroadcastInfoList": {
/*ro, opt, object, list of combined broadcasts information*/
    "BroadcastInfo": {
/*ro, opt, object, combined broadcasts information*/
        "id": {
/*ro, req, object, No., desc:it starts from 1*/
            "@maxSize": 8
            /*ro, req, int, starts from one*/
        },
        "value": {
/*ro, opt, object, object, desc:object*/
            "@opt": ["plate", "fullSpace", "expireRemind", "triggerTime", "vehicleType", "parkingType", "plateType", "parkPeriodTime",
"chargeAmount"],
            /*ro, req, array, range, subType:string*/
            "#text": "plate"
            /*ro, opt, string, a value for example*/
        },
        "customValue": {
/*ro, opt, object, custom broadcast information*/
            "@min": 1,
            /*ro, req, int, the minimum value*/
            "@max": 100
            /*ro, req, int, the maximum value*/
        },
        }
    },
    "volumeTimeEnabled": {
/*ro, opt, object, whether to enable volume time control*/
    "@opt": [true, false]
    /*ro, req, array, range, subType:bool*/
},
"VolumeTimeBlockList": {
/*ro, opt, object, the list of volume time control configuration information*/
    "TimeBlock": {
/*ro, opt, object, time period*/
        "id": {
/*ro, req, object, No., desc:object, ID*/
            "@maxSize": 5
            /*ro, req, int, maximum number of elements in the array*/
        },
        "volume": {
/*ro, opt, object, voice volume*/
            "@min": 0,
            /*ro, req, int, the minimum value*/
            "@max": 10
            /*ro, req, int, the maximum value*/
        },
        "beginTime": {
/*ro, opt, object, start time, desc:different time periods for control during 24 hours,it should be accurate to minute and in hh:MM format*/
            "@min": "00:00",
            /*ro, req, string, the minimum value*/
            "@max": "24:00"
            /*ro, req, string, maximum value*/
        },
        "endTime": {
/*ro, opt, object, end time, desc:different time periods for control during 24 hours,it should be accurate to minute and in hh:MM format*/
            "@min": "00:00",
            /*ro, req, string, the minimum value*/
            "@max": "24:00"
            /*ro, req, string, maximum value*/
        }
    }
}

```

```

        /*ro, req, string, the minimum value*/
        "@max": "24:00"
        /*ro, req, string, maximum value*/
    }
},
"broadcastPort": {
/*ro, opt, object, broadcast port, desc:object,broadcast port*/
    "@opt": ["rs485", "audioPort"],
    /*ro, req, array, range, subType:string*/
    "#text": "audioPort"
    /*ro, opt, string, a value for example*/
},
}
}

```

12.9.4.3 Set the parameters of combined voice broadcast

Request URL

PUT /ISAPI/Parking/channels/<channelID>/voiceBroadcastInfo/combinanteBroadcast?
format=json&powerOffSaveEnabled=<powerOffSaveEnabled>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
powerOffSaveEnabled	string	--

Request Message

```

{
    "CombinanteBroadcast": {
        /*req, object, combined broadcasts*/
        "enabled": true,
        /*req, bool, whether to enable combined broadcasts*/
        "volume": 1,
        /*opt, int, voice volume, range:[0,10]*/
        "ctrlMode": "cameraAndplatform",
        /*ro, opt, enum, subType:string*/
        "vehicleBroadcastEnabled": true,
        /*opt, bool*/
        "allowListBroadcastEnabled": false,
        /*opt, bool*/
        "blockListBroadcastEnabled": false,
        /*opt, bool*/
        "temporaryListBroadcastEnabled": false,
        /*opt, bool*/
        "BroadcastInfoList": [
            /*opt, array, list of combined broadcasts information, subType:object, range:[1,8]*/
            {
                "BroadcastInfo": {
                    /*opt, object, combined broadcasts information*/
                    "id": 1,
                    /*req, int, ID, range:[1,8], desc:it increases from 1*/
                    "value": "plate",
                    /*opt, enum, options, subType:string, desc:"plate" (license plate number prompt)*/
                    "customValue": "欢迎光临"
                    /*opt, string, custom broadcast information, range:[1,100], desc:value and customValue cannot be configured simultaneously, and one ID
can only be configured with one of these two parameters*/
                }
            }
        ],
        "volumeTimeEnabled": true,
        /*opt, bool, whether to enable volume time control*/
        "VolumeTimeBlockList": [
            /*opt, array, the list of volume time control configuration information, subType:object, range:[1,5]*/
            {
                "TimeBlock": {
                    /*opt, object, time period*/
                    "id": 1,
                    /*req, int, ID, range:[1,5], desc:it increases from 1, and supports up to 5 time periods*/
                    "volume": 1,
                    /*opt, int, volume, range:[0,10]*/
                    "beginTime": "00:00",
                    /*opt, time, start time, desc:different time periods for control during 24 hours, it should be accurate to minute and in hh:MM format*/
                    "endTime": "00:00"
                    /*opt, time, end time, desc:different time periods for control during 24 hours, it should be accurate to minute and in hh:MM format*/
                }
            }
        ],
        "soundMode": "child",
        /*opt, enum, types of voice mode, subType:string, desc:"child", "cartoon", "alto", "bass", "soprano", "tenor"*/
        "broadcastPort": "audioPort"
    }
}

```

```
broadcastPort : audioPort ,  
/*ro, opt, enum, broadcast port, subType:string, desc:"rs485", "audioPort"*/  
"Illegal": {  
/*opt, object, illegal*/  
"broadcastInfo": "test"  
/*opt, string, broadcast information, range:[0,64]*/  
},  
"PedestrianAlert": {  
/*opt, object, pedestrian pre-alarm*/  
"enabled": true,  
/*req, bool, whether to enable combined broadcasts*/  
"broadcastInfo": "test"  
/*opt, string, broadcast information, range:[0,64]*/  
},  
"TrafficLight": {  
/*opt, object, traffic light*/  
"enabled": true,  
/*req, bool, whether to enable combined broadcasts*/  
"broadcastMode": "single"  
/*opt, enum, broadcast mode, subType:string, desc:"single", "cycle"*/  
},  
"FreeTimeBroadcast": {  
/*opt, object*/  
"CustomContentList": [  
/*ro, opt, array, subType:object*/  
{  
"CustomContent": {  
/*ro, opt, object, broadcast information*/  
"id": 1,  
/*ro, req, int, ID, range:[1,5], desc:it increases from 1*/  
"content": "test"  
/*ro, opt, string, broadcast information*/  
}  
}  
],  
"TimeBlockList": [  
/*ro, opt, array, subType:object*/  
{  
"TimeBlock": {  
/*ro, opt, object, time period*/  
"id": 1,  
/*ro, opt, int, ID*/  
"beginTime": "00:00",  
/*ro, opt, string, start time, desc:different time periods for control during 24 hours, it should be accurate to minute and in hh:MM  
format*/  
"endTime": "24:00"  
/*ro, opt, string, end time, desc:different time periods for control during 24 hours, it should be accurate to minute and in hh:MM  
format*/  
}  
}  
],  
"times": 1,  
/*ro, req, int*/  
"cycleIntervalTimes": 15  
/*ro, req, int, loop interval*/  
},  
"IllegalBroadcast": {  
/*opt, object*/  
"IllegalList": [  
/*ro, opt, array, subType:object*/  
{  
"IllegalContent": {  
/*ro, opt, object, broadcast information*/  
"id": 1,  
/*ro, req, int, ID, range:[1,5], desc:it increases from 1*/  
"content": "test",  
/*ro, opt, string, broadcast information*/  
"contentType": "nonHelmet"  
/*opt, enum, content type, subType:string*/  
}  
}  
],  
"TimeBlockList": [  
/*ro, opt, array, subType:object*/  
{  
"TimeBlock": {  
/*ro, opt, object, time period*/  
"id": 1,  
/*ro, opt, int, ID*/  
"beginTime": "00:00",  
/*ro, opt, string, start time, desc:different time periods for control during 24 hours, it should be accurate to minute and in hh:MM  
format*/  
"endTime": "24:00"  
/*ro, opt, string, end time, desc:different time periods for control during 24 hours, it should be accurate to minute and in hh:MM  
format*/  
}  
}  
],  
"times": 1,  
/*ro, req, int*/  
"cycleIntervalTimes": 15  
/*ro, req, int, loop interval*/  
}  
}
```

```
}
```

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*/
}
```

12.9.4.4 Get the parameters of combined voice broadcast

Request URL

GET /ISAPI/Parking/channels/<channelID>/voiceBroadcastInfo/combinanteBroadcast?

format=json&powerOffSaveEnabled=<powerOffSaveEnabled>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--
powerOffSaveEnabled	string	Whether to save parameter settings in flash after power off; if this field is not applied, the parameter setting will not be saved in flash. If it is false, the device will restore it to the default value after rebooting. For example, /ISAPI/Parking/channels//voiceBroadcastInfo/combinanteBroadcast? format=json&powerOffSaveEnabled=true.

Request Message

None

Response Message

```

{
    "CombinatBroadcast": {
        /*ro, req, object, combined broadcasts*/
        "enabled": true,
        /*ro, req, bool, whether to enable combined broadcasts*/
        "volume": 1,
        /*ro, int, voice volume, range:[0,10]*/
        "ctrlMode": "cameraAndplatform",
        /*ro, enum, subType:string*/
        "vehicleBroadcastEnabled": true,
        /*ro, bool*/
        "allowListBroadcastEnabled": false,
        /*ro, bool*/
        "blockListBroadcastEnabled": false,
        /*ro, bool*/
        "temporaryListBroadcastEnabled": false,
        /*ro, bool*/
        "BroadcastInfoList": [
            /*ro, array, list of combined broadcasts information, subType:object*/
            {
                "BroadcastInfo": {
                    /*ro, object, combined broadcasts information*/
                    "id": 1,
                    /*ro, int, No., range:[1,8], desc:it starts from 1*/
                    "value": "plate",
                    /*ro, enum, broadcast content, subType:string, desc:plate: license plate number prompt*/
                    "customValue": "欢迎光临",
                    /*ro, string, custom broadcast information, range:[1,100], desc:the value and customValue cannot be configured simultaneously,one
ID can only be configured with one of these two parameters*/
                }
            },
            ],
        "volumeTimeEnabled": true,
        /*ro, bool, whether to enable volume time control*/
        "VolumeTimeBlockList": [
            /*ro, array, the list of volume time control configuration information, subType:object, range:[1,5]*/
            {
                "TimeBlock": {
                    /*ro, object, time period*/
                    "id": 1,
                    /*ro, int, No., range:[1,5], desc:read-only,ID,value range: [1,8],starts from one*/
                    "volume": 1,
                    /*ro, int, voice volume, range:[0,10]*/
                    "beginTime": "00:00",
                    /*ro, time, Start time, desc:different time periods for control during 24 hours,it should be accurate to minute and in hh:MM
format*/
                    "endTime": "00:00"
                    /*ro, time, end time, desc:different time periods for control during 24 hours,it should be accurate to minute and in hh:MM format*/
                }
            },
            ],
        "broadcastPort": "audioPort",
        /*ro, enum, broadcast port, subType:string, desc:rs485: RS-485; audioPort: audio port*/
    }
}

```

12.9.5 Vehicle Entry and Exit

12.9.5.1 Get the entrance and exit capability

Request URL

GET /ISAPI/ITC/Entrance/capabilities

Query Parameter

None

Request Message

None

Response Message

```

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

<EntranceCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, attr:version{req, string, protocolVersion}-->
  <supportEntrance>
    <!--ro, opt, bool, whether it supports entrance and exit control-->true
  </supportEntrance>
  <supportBarrierGateNum>
    <!--ro, opt, int-->1
  </supportBarrierGateNum>
  <supportAlarmINNum>
    <!--ro, opt, int-->1
  </supportAlarmINNum>
  <supportRelayNum>
    <!--ro, opt, int-->1
  </supportRelayNum>
  <isSupportWizardDeviceInfo>
    <!--ro, req, bool, whether to support configuring device information in wizard-->true
  </isSupportWizardDeviceInfo>
  <isSupportWizardFirstLogin>
    <!--ro, req, bool, whether to support judging if it is login for the first time in wizard-->true
  </isSupportWizardFirstLogin>
  <supportEntranceParam>
    <!--ro, opt, int-->1
  </supportEntranceParam>
</EntranceCap>

```

12.9.5.2 Get the entrance and exit parameters of traffic cameras

Request URL

GET /ISAPI/ITC/Entrance/entranceParam

Query Parameter

None

Request Message

None

Response Message

```

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

<EntranceParamList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, subType:object, attr:version{req, string, protocolVersion}-->
  <EntranceParam>
    <!--ro, opt, object-->
    <laneNum>
      <!--ro, req, int, barrier gate No.-->1
    </laneNum>
    <bEnable>
      <!--ro, req, bool, whether to enable the parameters-->true
    </bEnable>
    <ctrlMode>
      <!--ro, opt, enum, subType:int-->1
    </ctrlMode>
    <relateTriggerMode>
      <!--ro, req, enum, Linked trigger mode, subType:int, desc:0 (checkpoint single I/O trigger), 1 (checkpoint vehicle detector), 2 (MPR)-->1
    </relateTriggerMode>
    <vehControlMeasure>
      <!--ro, opt, object, vehicle management and control method-->
      <plateNumOnlyEnable>
        <!--ro, req, bool, only match license plate number-->true
      </plateNumOnlyEnable>
      <plateNumColorEnable>
        <!--ro, req, bool, match license plate number and license plate color simultaneously-->true
      </plateNumColorEnable>
    </vehControlMeasure>
  <vehInfoManagList>
    <!--ro, req, array, subType:object-->
    <vehInfoManag>
      <!--ro, req, object, vehicle information management-->
      <vehInfoManagNum>
        <!--ro, req, enum, vehicle information management, subType:int, desc:0 (temporary vehicle configuration), 1 (blocklist vehicle configuration), 2 (allowlist vehicle configuration)-->1
        </vehInfoManagNum>
        <barrierGateOper>
          <!--ro, req, enum, barrier gate operation, subType:int, desc:0 (no operation), 1 (open barrier gate)-->1
        </barrierGateOper>
        <relayOutAlarmEnable>
          <!--ro, req, bool, whether to enable relay output alarm-->true
        </relayOutAlarmEnable>
        <upAlarmEnable>
          <!--ro, req, bool, upload arming alarm information-->true
        </upAlarmEnable>
      </vehInfoManagNum>
    </vehInfoManag>
  </vehInfoManagList>
</EntranceParamList>

```

```

<hostUpAlarmEnable>
    <!--ro, req, bool, alarm host uploads alarm information-->true
</hostUpAlarmEnable>
</vehInfoManag>
</vehInfoManagList>
<relayList>
    <!--ro, req, array, relay list, subType:object-->
<relay>
    <!--ro, req, object, relay-->
<relayNum>
    <!--ro, req, int, relay ID-->1
</relayNum>
<relayFunction>
    <!--ro, req, enum, relay function, subType:int, desc:0-none,1-open barrier gate, 2-close barrier gate, 3-stop barrier gate, 4-alarm signal, 5-solid light-->1
</relayFunction>
</relay>
</relayList>
<IOAlarmList>
    <!--ro, req, array, subType:object-->
<IOAlarm>
    <!--ro, req, object-->
<IOAlarmNum>
    <!--ro, req, int, alarm port No.-->1
</IOAlarmNum>
<IOAlarmType>
    <!--ro, req, enum, alarm type, subType:int, desc:alarm type: 0-none,1-fire alarm-->1
</IOAlarmType>
</IOAlarm>
</IOAlarmList>
<notCloseCarFollow>
    <!--ro, opt, bool, whether to enable the barrier gate not to close when the vehicle is followed by another vehicle-->true
</notCloseCarFollow>
<bigCarKeepOpen>
    <!--ro, opt, object, the barrier gate remains open when the Large-sized vehicle passes-->
<enabled>
    <!--ro, req, bool, whether to enable the function-->true
</enabled>
<duration>
    <!--ro, req, int, duration of the barrier gate remaining open, unit:s-->1
</duration>
</bigCarKeepOpen>
<ParkingDetection>
    <!--ro, opt, object-->
<enabled>
    <!--ro, req, bool, whether to enable the function-->true
</enabled>
<judgeTime>
    <!--ro, req, int, unit:s-->1
</judgeTime>
</ParkingDetection>
</EntranceParam>
</EntranceParamList>

```

12.9.5.3 Set the entrance and exit parameters

Request URL

PUT /ISAPI/ITC/Entrance/entranceParam

Query Parameter

None

Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<EntranceParamList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--opt, array, subType:object, attr:version{req, string, protocolVersion}-->
<EntranceParam>
    <!--opt, object-->
<laneNum>
    <!--req, int, barrier gate No.-->1
</laneNum>
<bEnable>
    <!--req, bool, whether to enable this function-->true
</bEnable>
<ctrlMode>
    <!--opt, enum, subType:int-->1
</ctrlMode>
<relateTriggerMode>
    <!--req, enum, linked trigger mode, subType:int, desc:0 (checkpoint single I/O trigger), 1 (checkpoint vehicle detector), 2 (MPR)-->1
</relateTriggerMode>
<vehControlMeasure>
    <!--opt, object, vehicle management and control-->
<plateNumOnlyEnable>
    <!--req, bool, match license plate number only-->true
</plateNumOnlyEnable>
<plateNumColorEnable>

```

```

<!--req, bool, match both license plate number and license plate color-->true
</plateNumColorEnable>
</vehControlMeasure>
<vehInfoManagList>
  <!--req, array, subType:object-->
  <vehInfoManag>
    <!--req, object-->
    <vehInfoManagNum>
      <!--req, enum, vehicle information management, subType:int, desc:0 (temporary vehicle configuration),1 (blocklist vehicle configuration), 2
(allowlist vehicle configuration)-->1
      </vehInfoManagNum>
      <barrierGateOper>
        <!--req, enum, barrier gate operation, subType:int, desc:0 (no operation), 1 (open barrier gate)-->1
      </barrierGateOper>
      <relayOutAlarmEnable>
        <!--req, bool, whether to enable relays to output alarms-->true
      </relayOutAlarmEnable>
      <upAlarmEnable>
        <!--req, bool, upload arming alarm information-->true
      </upAlarmEnable>
      <hostUpAlarmEnable>
        <!--req, bool, alarm host uploads alarm information-->true
      </hostUpAlarmEnable>
    </vehInfoManag>
  </vehInfoManagList>
  <relayList>
    <!--req, array, subType:object-->
    <relay>
      <!--req, object-->
      <relayNum>
        <!--req, int, relay ID-->1
      </relayNum>
      <relayFunction>
        <!--req, enum, relay function, subType:int, desc:0 (none), 1 (open barrier gate), 2 (close barrier gate), 3 (stop barrier gate), 4 (alarm signal),
5 (solid light)-->1
      </relayFunction>
    </relay>
  </relayList>
  <IOAlarmList>
    <!--req, array, subType:object-->
    <IOAlarm>
      <!--req, object-->
      <IOAlarmNum>
        <!--req, int, alarm port No.-->1
      </IOAlarmNum>
      <IOAlarmType>
        <!--req, enum, alarm type, subType:int, desc:0 (none), 1(fire alarm)-->1
      </IOAlarmType>
    </IOAlarm>
  </IOAlarmList>
  <notCloseCarFollow>
    <!--opt, bool, whether to enable the barrier gate not to close when the vehicle is followed by another vehicle-->true
  </notCloseCarFollow>
  <bigCarKeepOpen>
    <!--opt, object, the barrier gate remains open when the large-sized vehicle passes-->
    <enabled>
      <!--req, bool, whether to enable the function-->true
    </enabled>
    <duration>
      <!--req, int, duration of the barrier gate remaining open, unit:s-->1
    </duration>
  </bigCarKeepOpen>
  <ParkingDetection>
    <!--opt, object-->
    <enabled>
      <!--req, bool, whether to enable the function-->true
    </enabled>
    <judgeTime>
      <!--req, int, unit:s-->1
    </judgeTime>
  </ParkingDetection>
</EntranceParam>
</EntranceParamList>

```

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>

```

12.9.6 Vehicle List Comparison

12.9.6.1 Get the capability of deleting the license plate information in the blocklist or allowlist

Request URL

GET /ISAPI/Traffic/channels/<channelID>/DelLicensePlateAuditData/capabilities?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

```

{
    "id": {
        /*ro, req, object, ID list*/
        "@size": 1000
        /*ro, req, int, the maximum number of members in the array (the maximum number of IDs in the list)*/
    },
    "licensePlate": {
        /*ro, opt, object*/
        "@size": 1
        /*ro, req, int, the maximum number of members in the array (the maximum number of IDs in the list)*/
    },
    "cardNo": {
        /*ro, opt, object*/
        "@size": 1
        /*ro, req, int, the maximum number of members in the array (the maximum number of IDs in the list)*/
    },
    "deleteAllEnabled": {
        /*ro, opt, object*/
        "@opt": [true, false]
        /*ro, req, array, subType:bool*/
    },
    "plateColor": {
        /*ro, opt, object*/
        "@size": 1,
        /*ro, req, int*/
        "@opt": ["black", "blue", "civilAviationBlack", "civilAviationGreen", "golden", "green", "mixedColor", "newEnergyGreen", "newEnergyYellowGreen", "orange", "other", "red", "unknown", "white", "yellow"]
        /*ro, opt, array, options, subType:string*/
    },
    "plateType": {
        /*ro, opt, object*/
        "@size": 1,
        /*ro, req, int*/
        "@opt": ["02TypePersonalized", "04NewMilitay", "92FarmVehicle", "92TypeArm", "92TypeCivil", "arm", "civilAviation", "coach", "consulate", "embassy", "emergency", "green1325FarmVehicle", "hongKongMacao", "leftRightMilitay", "motorola", "newEnergy", "oneLineArm", "oneLineArmHeadquarters", "tempEntry", "tempTravl", "trailer", "twoLineArm", "twoLineArmHeadquarters", "unknown", "upDownMilitay", "yellow1225FarmVehicle", "yellow1325FarmVehicle", "yellowTwoline"]
        /*ro, opt, array, options, subType:string*/
    },
    "name": {
        /*ro, opt, object*/
        "@size": 1
        /*ro, req, int*/
    },
    "certificateType": {
        /*ro, opt, object*/
        "@size": 1,
        /*ro, req, int*/
        "@opt": ["officerID", "ID", "passportID", "other"]
        /*ro, opt, array, subType:string*/
    },
    "certificateNumber": {
        /*ro, opt, object*/
        "@size": 1
        /*ro, req, int*/
    },
    "virtualParkingNum": {
        /*ro, opt, object*/
        "@size": 1
        /*ro, req, int*/
    },
    "CompoundCond": {
        /*ro, opt, object*/
        "@size": 1,
        /*ro, req, int*/
        "licensePlate": {
            /*ro, opt, object*/
            "@min": 1,
            /*ro, opt, int*/
            "@max": 32
            /*ro, opt, int*/
        },
        "plateColor": {
            /*ro, opt, object*/
            "@opt": ["black", "blue", "civilAviationBlack", "civilAviationGreen", "golden", "green", "mixedColor", "newEnergyGreen", "newEnergyYellowGreen", "orange", "other", "red", "unknown", "white", "yellow"]
            /*ro, opt, array, options, subType:string*/
        }
    }
}

```

12.9.6.2 Get the capability of adding license plate information to the blocklist or allowlist, or editing the license plate information in the blocklist or allowlist

Request URL

GET /ISAPI/Traffic/channels/<channelID>/licensePlateAuditData/record/capabilities?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Channel No.

Request Message

None

Response Message

```
{
  "LicensePlateInfoCap": {
    /*ro, req, object, list of license plates to be added to the list*/
    "id": {
      /*ro, req, object, index, desc:the value of this node is the ID of the license plate information, and the current ID can be obtained by calling the
      URI /ISAPI/Traffic/channels/<channelID>/searchLPListAudit*/
      "@min": 1,
      /*ro, opt, int*/
      "@max": 16
      /*ro, opt, int*/
    },
    "listType": {
      /*ro, opt, object, list type, desc:"blockList", "allowList"*/
      "@opt": ["blockList", "allowList"]
      /*ro, opt, array, subType:string*/
    },
    "licensePlate": {
      /*ro, req, object, license plate number*/
      "@min": 1,
      /*ro, opt, int*/
      "@max": 32
      /*ro, opt, int*/
    },
    "cardID": {
      /*ro, opt, object, card ID (Wiegand protocol), desc:object,card ID (Wiegand protocol),which consists of 8 decimal digits*/
      "@min": 1,
      /*ro, req, int*/
      "@max": 9
      /*ro, req, int*/
    },
    "createTime": "1970-01-01T00:00:00+08:00",
    /*ro, opt, datetime, created time*/
    "effectiveTime": "1970-01-01T00:00:00+08:00",
    /*ro, opt, datetime, effective time*/
    "effectiveStartDate": "1970-01-01",
    /*ro, opt, date, start date of the validity period*/
    "cardNo": {
      /*ro, opt, object, card No.*/
      "@min": 1,
      /*ro, req, int*/
      "@max": 64
      /*ro, req, int*/
    },
    "plateColor": {
      /*ro, req, object*/
      "@opt": ["black", "blue", "civilAviationBlack", "civilAviationGreen", "golden", "green", "mixedColor", "newEnergyGreen", "newEnergyYellowGreen",
      "orange", "other", "red", "unknown", "white", "yellow"]
      /*ro, opt, array, subType:string*/
    },
    "plateType": {
      /*ro, req, object*/
      "@opt": ["02TypePersonalized", "04NewMilitay", "92FarmVehicle", "92TypeArm", "92TypeCivil", "arm", "civilAviation", "coach", "consulate",
      "embassy", "emergency", "green1325FarmVehicle", "hongKongMacao", "leftRightMilitay", "motorola", "newEnergy", "oneLineArm", "oneLineArmHeadquarters",
      "tempEntry", "tempTrav1", "trailer", "twoLineArm", "twoLineArmHeadquarters", "unknown", "upDownMilitay", "yellow1225FarmVehicle", "yellow1325FarmVehicle",
      "yellowTwoLine"]
      /*ro, opt, array, subType:string*/
    },
    "operationType": {
      /*ro, opt, object*/
      "@opt": ["add", "modify"]
      /*ro, opt, array, subType:string*/
    },
    "name": {
      /*ro, opt, object*/
      "@min": 1,
      /*ro, req, int*/
      "@max": 32
      /*ro, req, int*/
    },
    "certificateType": {
      /*ro, req, object*/
      "@opt": ["officerID", "ID", "passportID", "other"]
      /*ro, opt, array, subType:string*/
    },
    "certificateNumber": {
      /*ro, opt, object*/
      "@min": 1,
      /*ro, req, int*/
    }
  }
}
```

```

/*ro, req, int*/
"@max": 32
/*ro, req, int*/
},
"virtualParkingNum": {
/*ro, opt, object*/
"@min": 1,
/*ro, req, int*/
"@max": 16
/*ro, req, int*/
},
"plateDescription": {
/*ro, opt, object*/
"@min": 0,
/*ro, req, int*/
"@max": 128
/*ro, req, int*/
},
"groupName": {
/*ro, opt, object*/
"@min": 1,
/*ro, req, int*/
"@max": 64
/*ro, req, int*/
}
}
}
}

```

12.9.6.3 Add license plate information to the blocklist or allowlist, or edit the license plate information in the blocklist or allowlist

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/licensePlateAuditData/record?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Channel No.

Request Message

```

{
    "LicensePlateInfoList": [
        /*req, array, subType:object*/
        {
            "id": "1",
            /*req, string, ID, range:[1,16], desc:index,the maximum string size is 16 bytes. The value of this node is the ID of the license plate information, and the current ID can be obtained by calling the URI /ISAPI/Traffic/channels/<channelID>/searchLPLListAudit*/
            "listType": "blockList",
            /*req, enum, list type, subType:string, desc:list type: "blockList","allowList"*/
            "LicensePlate": "test",
            /*req, string, license plate No., range:[1,32]*/
            "createTime": "1970-01-01T00:00:00+08:00",
            /*opt, datetime, creation time*/
            "effectiveTime": "1970-01-01T00:00:00+08:00",
            /*opt, datetime, end time of the validity period*/
            "cardID": "test",
            /*opt, string, card ID (Wiegand protocol), range:[1,9], desc:card ID by Wiegand rule,which is a number consisting of 8 digits in decimal format. The maximum string size is 9 bytes*/
            "effectiveStartDate": "1970-01-01",
            /*opt, string, start date of the validity period*/
            "cardNo": "abcd1234",
            /*opt, string, card No., range:[1,64]*/
            "plateColor": "blue",
            /*req, enum, license plate color, subType:string, desc:license plate color*/
            "plateType": "92TypeCivil",
            /*req, enum, license plate type, subType:string, desc:license plate type*/
            "operationType": "add",
            /*opt, string*/
            "name": "张三",
            /*ro, opt, string, range:[0,96]*/
            "certificateType": "officerID",
            /*ro, opt, enum, subType:string*/
            "certificateNumber": "test",
            /*ro, opt, string, range:[0,32]*/
            "virtualParkingNum": "123",
            /*opt, string, range:[1,16]*/
            "plateDescription": "abc",
            /*opt, string, range:[0,128]*/
            "groupName": "group1"
            /*opt, string, range:[1,64]*/
        }
    ]
}

```

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 details, desc:this node is required when the value of statusCode is not 1*/
    "MerrCode": "0x00000000",
    /*ro, opt, string*/
    "MerrDevSelfEx": "0x00000000"
    /*ro, opt, string*/
}

```

12.9.6.4 Update the vehicle blocklist and allowlist file

Request URL

PUT /ISAPI/Traffic/channels/<channelID>/licensePlateAuditData?fileType=<fileType>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Channel No.
fileType	enum	When "FileType" is xml, it is an XML file. If no "fileType" field or "fileType" is excel, it is an Excel file of XLS format. When "fileType" is csv, it is a Excel file of CSV format.

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, sub status code, desc:sub status code-->OK
    </subStatusCode>
    <description>
        <!--ro, opt, string, range:[0,1024]-->badXmlFormat
    </description>
    <AttachInfo>
        <!--ro, opt, object-->
    <StatusList>
        <!--ro, req, object-->
        <Status>
            <!--ro, req, object-->
            <index>
                <!--ro, req, int-->0
            </index>
            <statusCode>
                <!--ro, req, enum, subType:int-->0
            </statusCode>
            <statusString>
                <!--ro, req, enum, subType:string-->OK
            </statusString>
            <subStatusCode>
                <!--ro, req, string, range:[1,64]-->ok
            </subStatusCode>
            <description>
                <!--ro, opt, string, range:[0,1024]-->test
            </description>
            <plateNo>
                <!--ro, opt, string, range:[1,64]-->test
            </plateNo>
            <licensePlateListType>
                <!--ro, opt, enum, subType:string-->allowlist
            </licensePlateListType>
        </Status>
    </StatusList>
    </AttachInfo>
    <successNum>
        <!--ro, opt, int-->100
    </successNum>
</ResponseStatus>
```

12.9.6.5 Export the file of vehicle blocklist and allowlist data

Request URL

GET /ISAPI/Traffic/channels/<channelID>/licensePlateAuditData?fileType=<fileType>

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	Channel No.
fileType	enum	When "FileType" is xml, it is an XML file. If no "fileType" field or "fileType" is excel, it is an Excel file of XLS format. When "fileType" is csv, it is a Excel file of CSV format.

Request Message

None

Response Message

None

12.10 Parking Space Detection

12.10.1 Parking Space Indicator Control

12.10.1.1 Set parameters of parking space indicator

Request URL

PUT /ISAPI/Parking/parkingSpace/indicatorLamp?format=json&security=<security>&iv=<iv>

Query Parameter

None

Request Message

```
{
    "lampType": "internal",
    /*opt, enum, indicator type, subType:string, desc:by default, the value is "internal"; internal: internal indicator; external: external indicator;
    internalAndExternal: internal and external indicator*/
    "VehicleNotExist": {
        /*req, object, parameters of parking spaces without parked vehicles*/
        "enabled": true,
        /*req, bool, whether to enable*/
        "flashEnabled": true,
        /*req, bool, whether to enable flashing*/
        "lampColor": "close"
        /*req, enum, indicator color, subType:string, desc:;close; red; green; yellow; blue; magenta; cyan; white*/
    },
    "VehicleExist": {
        /*req, object, parameters of parking spaces with parked vehicles*/
        "enabled": true,
        /*req, bool, whether to enable*/
        "flashEnabled": true,
        /*req, bool, whether to enable flashing*/
        "lampColor": "close"
        /*req, enum, indicator color, subType:string, desc:;close; red; green; yellow; blue; magenta; cyan; white*/
    },
    "VehicleEntry": {
        /*opt, object, parameters of parking spaces which vehicles entering, desc:The color of indicator is same as that of indicator for parking spaces without
        parked vehicles when vehicles are entering the parking spaces.*/
        "enabled": true,
        /*req, bool, whether to enable*/
        "flashEnabled": true
        /*req, bool, whether to enable flashing*/
    },
    "pressLine": {
        /*req, object, parameters of parking over lane line detection*/
        "enabled": true,
        /*req, bool, whether to enable*/
        "flashEnabled": true
        /*req, bool, whether to enable flashing*/
        "lampColor": "close"
        /*req, enum, indicator color, subType:string, desc:;close; red; green; yellow; blue; magenta; cyan; white*/
    },
    "specialParking": {
        /*req, object, parameters of special parking spaces, desc:the special parking space will be configured with indicators which are different with that
        configured for the normal parking spaces, e.g., VIP parking space; 0: normal parking space; 1: special parking space*/
        "enabled": true,
        /*req, bool, whether to enable*/
        "flashEnabled": true
        /*req, bool, whether to enable flashing*/
        "lampColor": "close"
        /*req, enum, indicator color, subType:string, desc:;close; red; green; yellow; blue; magenta; cyan; white*/
    },
    "replaceLampCtrl": {
        /*opt, object, indicator control*/
        "enabled": true,
        /*req, bool, whether to enable*/
        "ipV4Address": "192.168.1.64"
        /*req, string, indicator control, range:[0,32], desc:used for configuring an IP address for the device that control indicates*/
    }
}
```

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:This field 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.*/  
}
```

12.10.1.2 Get parameters of parking space indicator

Request URL

GET /ISAPI/Parking/parkingSpace/indicatorLamp?format=json

Query Parameter

None

Request Message

None

Response Message

```

{
    "lampType": "internal",
    /*ro, opt, enum, Indicator type, subType:string, desc:by default, the value is "internal"; internal: internal indicator; external: external indicator;
internalAndExternal: internal and external indicator*/
    "VehicleNotExist": {
        /*ro, req, object, parameters of parking spaces without parked vehicles*/
        "enabled": true,
        /*ro, req, bool, whether to enable*/
        "flashEnabled": true,
        /*ro, req, bool, whether to enable flashing*/
        "lampColor": "close"
        /*ro, req, enum, indicator color, subType:string, desc:;close; red; green; yellow; blue; magenta; cyan; white*/
    },
    "VehicleExist": {
        /*ro, req, object, parameters of parking spaces with parked vehicles*/
        "enabled": true,
        /*ro, req, bool, whether to enable*/
        "flashEnabled": true,
        /*ro, req, bool, whether to enable flashing*/
        "lampColor": "close"
        /*ro, req, enum, indicator color, subType:string, desc:;close; red; green; yellow; blue; magenta; cyan; white*/
    },
    "VehicleEntry": {
        /*ro, opt, object, parameters of parking spaces which vehicles are entering, desc:The color of indicator is same as that of indicator for parking spaces
without parked vehicles when vehicles are entering the parking spaces.*/
        "enabled": true,
        /*ro, req, bool, whether to enable*/
        "flashEnabled": true
        /*ro, req, bool, whether to enable flashing*/
    },
    "pressLine": {
        /*ro, req, object, parameters of parking over lane line detection*/
        "enabled": true,
        /*ro, req, bool, whether to enable*/
        "flashEnabled": true,
        /*ro, req, bool, whether to enable flashing*/
        "lampColor": "close"
        /*ro, req, enum, indicator color, subType:string, desc:;close; red; green; yellow; blue; magenta; cyan; white*/
    },
    "specialParking": {
        /*ro, req, object, parameters of special parking spaces, desc:the special parking space will be configured with indicators which are different with that
configured for the normal parking spaces, e.g., VIP parking space; 0: normal parking space; 1: special parking space*/
        "enabled": true,
        /*ro, req, bool, whether to enable*/
        "flashEnabled": true,
        /*ro, req, bool, whether to enable flashing*/
        "lampColor": "close"
        /*ro, req, enum, indicator color, subType:string, desc:;close; red; green; yellow; blue; magenta; cyan; white*/
    },
    "replaceLampCtrl": {
        /*ro, opt, object, indicator control*/
        "enabled": true,
        /*ro, req, bool, whether to enable*/
        "ipV4Address": "192.168.1.64",
        /*ro, req, string, IP address, range:[0,32], desc:used for getting the IP address for the device that control indicates*/
        "port": 80,
        /*ro, opt, int, range:[0,65535]*/
        "username": "test"
        /*ro, opt, string, range:[1,32]*/
    },
    "lampChangeEnabled": false
    /*ro, opt, bool*/
}

```

12.10.2 Parking Space Patrol

12.10.2.1 Get the capability of configuring the trigger mode parameters of parking detection

Request URL

GET /ISAPI/ITC/TriggerMode/parkingIncident/capabilities

Query Parameter

None

Request Message

None

Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<ParkingIncidentCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, opt, object, parking detection, attr:version{req, string, protocolVersion}-->
    <relatedParkCount min="1" max="8">

```

```

<!--ro, req, int, total number of linked parking spaces, range:[1,8], attr:min{req, int},max{req, int}-->0
</relatedParkCount>
<relatedDriveWay min="1" max="16">
    <!--ro, req, string, Lane No., range:[1,16], attr:min{req, int},max{req, int}-->test
</relatedDriveWay>
<laneDirectionType opt="0,1,2">
    <!--ro, req, enum, types of Lane directions, subType:int, attr:opt{req, string}, desc:0 (unknown), 1 (driving into or out of the parking space from its front), 2 (driving into or out of the parking space from its back)-->0
</laneDirectionType>
<parkingDirectionType opt="0,1,2">
    <!--ro, req, enum, subType:int, attr:opt{req, string}-->0
</parkingDirectionType>
<parkSight opt="parallel,vertical,blindSpot">
    <!--ro, opt, enum, types of parking scenes, subType:string, attr:opt{req, string}, desc:"parallel" (parking in parallel way), "vertical" (parking in vertical way), "blind spot" (parking in a blind spot)-->parallel
</parkSight>
<Parallel>
    <!--ro, opt, object, parameters of parking in a parallel way-->
<relatedDriveWay opt="parkingLane,virtualLane,triggerLane,alertLane,1,2,3,4,5,6,7,8">
    <!--ro, req, enum, Lane No., subType:string, attr:opt{req, string}-->parkingLane
</relatedDriveWay>
<snapTimes min="3" max="3">
    <!--ro, opt, int, capture times, range:[2,3], attr:min{req, int},max{req, int}, desc:the default value is 3-->0
</snapTimes>
<maxParkCount>
    <!--ro, opt, int, maximum number of parking space Lines, range:[1,8]-->0
</maxParkCount>
</Parallel>
<Vertical>
    <!--ro, opt, object, parameters of parking in a vertical way-->
<relatedDriveWay opt="virtualLane,triggerLane,1,2,3,4,5">
    <!--ro, req, enum, Lane No., subType:string, attr:opt{req, string}-->virtualLane
</relatedDriveWay>
<snapTimes min="2" max="3">
    <!--ro, opt, int, capture times, range:[2,3], attr:min{req, int},max{req, int}, desc:the default value is 2-->0
</snapTimes>
<maxParkCount>
    <!--ro, opt, int, maximum number of parking space Lines, range:[1,8]-->0
</maxParkCount>
</Vertical>
<IllegalParking>
    <!--ro, opt, object, parameters of illegal parking-->
<enabled opt="true,false">
    <!--ro, req, bool, whether to enable the function, attr:opt{req, string}-->true
</enabled>
<sensitivity min="0" max="100">
    <!--ro, req, int, sensitivity, range:[0,100], attr:min{req, int},max{req, int}, desc:the default value is 50-->0
</sensitivity>
<duration min="1" max="900">
    <!--ro, req, int, duration, range:[1,900], unit:s, attr:min{req, int},max{req, int}, desc:the default value is 15-->1
</duration>
</IllegalParking>
<OnLaneLine>
    <!--ro, opt, object, parameters of driving on the Lane Line-->
<enabled opt="true,false">
    <!--ro, req, bool, whether to enable the function, attr:opt{req, string}-->true
</enabled>
<sensitivity min="0" max="100">
    <!--ro, req, int, sensitivity, range:[0,100], attr:min{req, int},max{req, int}, desc:the default value is 50-->0
</sensitivity>
<duration min="1" max="180">
    <!--ro, req, int, duration, range:[1,180], unit:s, attr:min{req, int},max{req, int}, desc:the default value is 15-->1
</duration>
</OnLaneLine>
<CrossBerth>
    <!--ro, opt, object, parameters of crossing the parking space-->
<enabled opt="true,false">
    <!--ro, req, bool, whether to enable the function, attr:opt{req, string}-->true
</enabled>
<sensitivity min="0" max="100">
    <!--ro, req, int, sensitivity, range:[0,100], attr:min{req, int},max{req, int}, desc:the default value is 50-->0
</sensitivity>
<duration min="1" max="180">
    <!--ro, req, int, duration, range:[1,180], unit:s, attr:min{req, int},max{req, int}, desc:the default value is 15-->1
</duration>
</CrossBerth>
<parkingSnapTime>
    <!--ro, opt, object-->
<enabled opt="true,false">
    <!--ro, req, bool, attr:opt{req, string}-->true
</enabled>
<time min="0" max="180">
    <!--ro, req, int, range:[0,180], attr:min{req, int},max{req, int}-->0
</time>
</parkingSnapTime>
<reverseParkingCaptureEnabled opt="true,false">
    <!--ro, opt, bool, attr:opt{req, string}-->true
</reverseParkingCaptureEnabled>
</ParkingIncidentCap>

```

12.10.3 Parking Space Status Detection

12.10.3.1 Get parking space detection parameters

Request URL

GET /ISAPI/Parking/channels/<channelID>/packingSpaceRecognition?format=json

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	channel No.

Request Message

None

Response Message

```
{  
    "PackingSpaceRecognition": {  
        /*ro, req, object, parking space detection parameter*/  
        "channelID": 1,  
        /*ro, req, int, channel No.*/  
        "sensitivity": 1,  
        /*ro, opt, int, sensitivity, range:[1,100], desc:the default value is 50*/  
        "parkingSnapTime": 1,  
        /*ro, opt, int, 停车抓拍时间,range:[1,100],默认15, range:[1,100], unit:s, desc:停车抓拍时间,range:[1,100],默认15*/  
        "packingSpaceTotalNum": 1,  
        /*ro, opt, int, 车位总数量,range:[0,4], range:[0,4]*/  
        "PackingSpaceInfoList": [  
            /*ro, opt, array, array,车位信息列表,subType:object,range:[0,4], subType:object, range:[0,4]*/  
            {  
                "PackingSpaceInfo": {  
                    /*ro, opt, object, object,车位信息*/  
                    "absoulteParkingNum": "123",  
                    /*ro, req, string, 车位编号,range:[1,16], range:[1,16]*/  
                    "virtualParkingNum": "123",  
                    /*ro, opt, string, 虚拟车位编号,range:[1,16], range:[1,16]*/  
                    "Region": [  
                        /*ro, opt, array, array,车位区域坐标,subType:object,range:[1,4], subType:object, range:[1,4]*/  
                        {  
                            "Point": {  
                                /*ro, opt, object, point coordinates*/  
                                "positionX": 0.123,  
                                /*ro, req, float, x-coordinate, range:[0,1], desc:X-coordinate*/  
                                "positionY": 0.123  
                                /*ro, req, float, y-coordinate, range:[0,1], desc:accurate to three decimal places. The origin is the upper-left corner of  
the image*/  
                            }  
                        }  
                    ],  
                    "parkingType": 1,  
                    /*ro, opt, enum, parking space type, subType:int, desc:parking space type*/  
                }  
            }  
        ],  
        "occlusionOccupancyParams": {  
            /*ro, opt, object*/  
            "enabled": true  
            /*ro, req, bool, whether to enable the detection*/  
        },  
    }  
}
```

12.10.3.2 Get parking space status information

Request URL

GET /ISAPI/Parking/channels/<channelID>/parkingStatus

Query Parameter

Parameter Name	Parameter Type	Description
channelID	string	--

Request Message

None

Response Message

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

<ParkingStatusCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, parking space status capability, attr:version{req, string, protocolVersion}-->
  <ParkingNum>
    <!--ro, req, int, total number of parking spaces, range:[1,5]-->4
  </ParkingNum>
  <ParkingStatusList>
    <!--ro, req, array, subType:object-->
    <ParkingStatus>
      <!--ro, req, object-->
      <parkingNo>
        <!--ro, req, string, range:[0,64]-->test
      </parkingNo>
      <plateNo>
        <!--ro, opt, enum, subType:string-->noPlate
      </plateNo>
      <carStatus>
        <!--ro, opt, bool-->true
      </carStatus>
      <clampBlinking>
        <!--ro, opt, bool-->true
      </clampBlinking>
      <clampColor>
        <!--ro, opt, enum, subType:string-->red
      </clampColor>
      <vehicleColor>
        <!--ro, opt, enum, subType:string-->black
      </vehicleColor>
      <vehicleLogo>
        <!--ro, opt, string, range:[0,64]-->Audi
      </vehicleLogo>
      <inTime>
        <!--ro, opt, datetime-->2004-05-03T17:30:08+08:00
      </inTime>
      <outTime>
        <!--ro, opt, datetime-->2004-05-03T17:30:08+08:00
      </outTime>
      <vehicleType>
        <!--ro, opt, enum, subType:string-->unknown
      </vehicleType>
      <confidenceLevel>
        <!--ro, opt, int, range:[0,100]-->50
      </confidenceLevel>
      <plateColor>
        <!--ro, opt, enum, subType:string-->black
      </plateColor>
      <vehicleTargetRect>
        <!--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>
      </vehicleTargetRect>
      <parkingNumber>
        <!--ro, opt, int-->0
      </parkingNumber>
      <parkingTime>
        <!--ro, opt, int, range:[0,2592000], unit:s-->1
      </parkingTime>
      <virtualParkingNum>
        <!--ro, opt, string, range:[1,32]-->1
      </virtualParkingNum>
    </ParkingStatus>
  </ParkingStatusList>
  <backgroundImage>
    <!--ro, opt, object-->
    <resourcesContentType>
      <!--ro, opt, enum, subType:string-->url
    </resourcesContentType>
    <resourcesContent>
      <!--ro, opt, string, range:[1,128]-->test
    </resourcesContent>
  </backgroundImage>
</ParkingStatusCap>
```

Parameter Name	Parameter Value	Parameter Type(Content-Type)	Content-ID	File Name	Description
ParkingStatusCap	[Message content]	application/xml	--	--	--
.value	[Binary picture data]	image/jpeg	detectionPicture	detectionPicture.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.

12.10.3.3 Search for parking space detection events

Request URL

POST /ISAPI/Parking/packingSpaceRecognition/search?format=json

Query Parameter

None

Request Message

```
{
    "PackingSpaceRecognitionSearchDescription": {
        /*req, object, search conditions*/
        "searchID": "fgdtge5456",
        /*req, string, search ID, range:[1,32], desc:it is used to confirm the upper-level platform or system. If the platform or the system is the same one
        during two searches, the search history will be saved in the memory to speed up next search*/
        "searchResultPosition": 0,
        /*req, int, the start position of the search result in the result list, range:[0,4294967295], desc:in a single search, if you cannot get all the
        records in the result list, you can mark the end position and get the following records after the marked position in the next search*/
        "maxResults": 500
        /*req, int, the maximum number of search results this time, range:[0,4294967295]*/
    }
}
```

Response Message

```
{
    "PackingSpaceRecognitionSearchResult": {
        /*ro, req, object, search results*/
        "searchID": "fgdtge5456",
        /*ro, req, string, search ID, range:[1,32], desc:it is used to confirm the upper-level platform or system. If the platform or the system is the same
        one during two searches, the search history will be saved in the memory to speed up next search*/
        "responseStatusStrg": "OK",
        /*ro, req, string, searching status description, range:[1,32], desc:searching status description "OK" (search completed), "MORE" (searching for more
        data), "NO MATCH" (no matched data)*/
        "numOfMatches": 1,
        /*ro, opt, int, number of records returned for one time, range:[0,4294967295]*/
        "totalMatches": 1,
        /*ro, opt, int, total number of matched results, range:[0,4294967295]*/
        "PackingSpaceRecognition": [
            /*ro, opt, array, subType:object*/
            {
                "channelID": 1,
                /*ro, req, int, channel No. of the device that triggers the alarm*/
                "RegionInfo": [
                    /*ro, opt, array, region information, subType:object*/

```

```

{
    "regionalID": 1,
    /*ro, req, int, detection region ID, range:[1,255]*/
    "actualPackingSpaceNum": 1,
    /*ro, opt, int, range:[1,65535]*/
    "detectedPackingSpaceNum": 1,
    /*ro, opt, int, range:[1,65535]*/
    "PackingSpace": [
        /*ro, opt, array, subType:object*/
        {
            "absoulteParkingNum": "123",
            /*ro, req, string, parking space No., range:[1,16]*/
            "virtualParkingNum": "123",
            /*ro, opt, string, virtual parking space No., range:[1,32]*/
            "isParked": "yes",
            /*ro, req, enum, parking space status, subType:string, desc:"yes" (cars on the parking space), "no" (no car)*/
            "illegalParkingEvent": "crossLane",
            /*ro, opt, enum, illegal parking event of the vehicle, subType:string, desc:crossLane (parking over the lane line),
            crossBerth (occupying two parking spaces), crossLaneBerth (parking over the lane line and occupying two parking spaces), reverse (reverse parking)*/,
            "IllegalParkingEventExtendInfo": [
                /*ro, opt, array, subType:object*/
                {
                    "illegalParkingEvent": "crossLane"
                    /*ro, opt, enum, illegal parking event of the vehicle, subType:string, desc:crossLane (parking over the lane line),
                    crossBerth (occupying two parking spaces), crossLaneBerth (parking over the lane line and occupying two parking spaces), reverse (reverse parking)*/
                }
            ],
            "plateNo": "test",
            /*ro, opt, string, "license plate number", range:[0,32], desc:license plate number (For motor vehicles, it is a 16-byte
            string; for non-motor vehicles, it is a 48-byte string)*/
            "arriveTime": "1970-01-01T00:00:00+08:00",
            /*ro, opt, datetime*/
            "leaveTime": "1970-01-01T00:00:00+08:00",
            /*ro, opt, datetime*/
            "plateColor": "black",
            /*ro, opt, enum, license plate color, subType:string*/
            "plateType": "unknown",
            /*ro, opt, enum, license plate type, subType:string*/
            "parkingLockStatus": "open",
            /*ro, opt, enum, subType:string*/
            "vehicleLogoRecog": 1025,
            /*ro, opt, int, vehicle main brand, range:[1025,1801]*/
            "vehicleSubLogoRecog": 0,
            /*ro, opt, int, vehicle sub-brand, range:[0,100]*/
            "vehicleModel": 0,
            /*ro, opt, int, index of model year of vehicle sub-brand, range:[0,100]*/
            "lampFlash": "yes",
            /*ro, opt, enum, subType:string, desc:"yes", "no"*/
            "lampColor": "close",
            /*ro, opt, enum, subType:string*/
            "vehicleColor": "white",
            /*ro, opt, enum, vehicle color, subType:string, desc:"black", "blue", "brown", "cyan", "deepBlue" (dark blue), "deepGray"
            (dark gray), "gray", "green", "orange", "pink", "purple", "red", "silver", "unknown", "white", "yellow"*/
            "parkingInfoListEnabled": true,
            /*ro, opt, bool*/
            "plateInParkingInfoList": true,
            /*ro, opt, bool*/
            "ParkingPicture": {
                /*ro, opt, object*/
                "resourcesContentType": "url",
                /*ro, opt, enum, resource transmission type, subType:string, desc:"url" (transmitted via URL). When the alarm message
                contains additional resources (such as picture), this field should be returned*/
                "resourcesContent": "test"
                /*ro, opt, string, resource ID, range:[1,256], desc:when the returned alarm message contains additional resources (e.g.,
                pictures), this field should be returned. When the value of resourcesContentType is "url", its value is the resource URL*/
            }
        }
    ]
},
"PicResources": {
    /*ro, opt, object, real-time capture of the parking space, desc:resource transmission module,when the alarm message contains additional
    resources (such as picture),this node must be returned*/
    "resourcesContentType": "url",
    /*ro, opt, enum, resource transmission type, subType:string, desc:"url" (transmitted via URL). When the alarm message contains
    additional resources (such as picture), this field should be returned*/
    "resourcesContent": "test"
    /*ro, opt, string, resource ID, range:[1,256], desc:when the returned alarm message contains additional resources (e.g., pictures), this
    field should be returned. When the value of resourcesContentType is "url", its value is the resource URL*/
}
}

```

13 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.

flavio.gonzalez@segulab.edu.uy
Segulab