PoE & WiFi IP Cameras

HTTP CGI User Manual

Document Version : V5.00.11

Update : 2021.04.10

Jerry Lin

Abstract

HTTP CGI is mainly for the integration with the third-party network management devices or software. The document describes the HTTP CGI of the IP cameras, the developer can use the document to finish the integration the IP cameras into your management system.

Who Can Use the Document

The developer who are making the IP Cameras integration application development.

Development Platform

The interfaces can be used in cross platforms development like ARM, Windows, Mac OS, Linux on PC etc. and in the mobile integration for Android and IOS platforms.

Update History

Updated History			
NO.	**Updated Contents**	**Responsible**	**Time**
1	The first release	Tony	2015.04.08
2	Audio capacity add: bit width, please check the red fonts in "media";	Jerry	2015.06.23
3	Add: Get, Add, Modify, Delete User information parameters Command; please refer to 1.5 Interface of user management in 1. System;	Jerry	2015.07.23
4	Add: Get and Set OSD parameters command; please refer to 3.12 Get and Set OSD parameters in 3. Media;	Jerry	2015.07.23
5	Add: Command for the device maintenance; please refer to 5.1 device maintenance in 5. Maintain;	Tony	2015.07.23
6	Add: Defined audio and video network transmission procedure and interfaces; Add: two-way speaking procedure; please refer to 6.1 and 6.2 items in 6. Stream;	Jerry	2015.07.23
7	Add: Interface definition of the storage device status to get. Add: Interface definition of the storage device formatting. Please refer to 5.2 item in 5. Maintains;	Jerry	2015.07.24

8	Add: Interface definition of recording parameters' get and set; Add: Interface definition of scheduled task parameters get and set; Add: Cycle recording of Recording configuration parameters; please refer to 7. Scheduled recording items	Jerry	2015.07.24
9	Bug fixed: error to definition of presec in recording configuration;	Jerry	2015.07.25
10	Modified: the date format changed from YYYY-MM-DD to YYYYMMDD, month format changed from YYYY-MM to YYYYMM; The record searching time range changed from time to beginning time and ending time like the below: begin: beginning time, unit: second; end: ending time, unit: second; please refer to 6. Stream 6.3 record searching	Tony	2015.07.29
11	Mistake correction: 6.2 2-way speaking procedure: The Procedure of requesting for 2 way speaking: 1. Client: Send the request for speaking; 2. Client: the client will send the audio data to the device after around 1 second when the client received the success code "200 returned from the device for the request for speaking from the client.(The first frame of the data sent to the client is metadata. The client can start the corresponding decoder according to the value of the metadata, the data after the first frame is the audio data.) 3. Client: disconnected, the device release the resource;	Jerry	2015.07.29
12	Modified: 5.1 Device Maintenance, correct the typo from maintain to maintain." /action/set?subject=maintain"	Tony	2015.07.31
13	Modified: 6.3 Record Searching, change the time format from 2015-08 to 201508.	Jerry	2015.08.01
14	Modified: 6.3 and 6.4 command, added channel item, like " <chn> 0 <chn>"	Jerry	2015.08.13
15	Modified: 6.3, 6.4 changed the tag of "month" to "date".	Tony	2015.08.14
16	Modified: 6.3 The DD should be not 00 on searching.	Tony	2015.08.18
17	Modified: 3.5 add AAC audio codec.	Tony	2015.10.10
18	Added: 4.0 Alarm, the example how to calculate the outmask value is added in the further description.	Tony	2015.11.05
19	Added: 3.13 Get and Set ROI parameters	Tony	2015.12.03
20	Added: 5.4 Firmware upgrade	Tony	2015.12.03
21	Added: 4.5 Alarm notification, 4.6 Get and set I/O alarm parameters	TONY	2016.03.21
22	Revised: demo code: <stream/> should be </stream>	TONY	2016.03.31

23	Added: 1.6 Interfaces of acquiring the device running status	Jerry	2016.05.03
24	Updated: 1.6 Interfaces of acquiring the device running status, added: uptime, IO alarm input and IO alarm output status;	Jerry	2016.06.20
25	Updated: 4.6 Get and Set I/O alarm parameters, added: the current output level parameter	Jerry	2016.06.20
26	Added: 8.1 PTZ Control Command	Jerry	2016.06.28
27	Added: 3.14 Get and Set Audio Input and Output Volume	Tony	2016.09.01
28	Make the more detailed clarification for the 5.2 – Status value	Jian Qin	2016.09.24
29	Added: 2.10 Get and Set RTMP Parameters Definition Added: 2.11 Get the RTSP URL Interface Definition; Added: 2.12 Get the MJPEG SNAPSHOT URL Interface Definition; Added: 2.13 Get and Set VoIP (SIP) parameters Definition.	Jian Qin	2016.11.11
30	Correct some mistakes, and adjusted the document format	Jerry Lin	2017.01.10
31	Added: Find the password and Configurations file Output and Input	Jerry Lin	2017.01.16
32	Updated 1.5 Interfaces of the user management description about the user's role, user's password encryption description;	Jerry Lin	2017.02.10
33	Updated: 2.1 and 2.2 IPV6, MTU information description.	Jerry Lin	2017.02.10
34	Added: 4.7 Trigger Alarm Event	Jian Qin	2017.02.17
35	Added: 4.8 Alarm Data Definition	Jian Qin	2017.02.17
36	Added: 6.5 Download the archives	Jian Qin	2017.02.17
37	Updated: 6.3 Record Searching: added path of the archives	Jian Qin	2017.02.17
38	Updated: 2.10 Get and Set RTMP parameters: to support mainstream and sub stream video can be pushed to the different server.	Jian Qin	2017.03.02
39	Updated 3.9: added LDC and Rotate mode;	Jerry Lin	2017/04/01
40	Added: 10 HTTP CGI for Request the LOG	Tony	2017.04.17

41	Modify 2.1: add the network cards enable or disable parameter.	Tony	2017.05.20
42	Modify: 3.9 added the item “defog” enable and disable;	Tony	2017.05.21
43	Updated the document version from V2.00.06 to V3.00.01 because of new message protocol updated.	Lin Qing	2017.06.05
44	Updated: Version number is added into all commands. If there is no special notes, the default protocol version is V1.0, otherwise there is notification: it's suitable for V2.0 Added: DST parameters, and cancel auto DST; Updated: OSD pictures parameters, and revised the OSD coordinate from pixels to permillage unit. Added: AAC audio codec description in audio capacitor command.	Lin Qing	2017.06.05
45	Added: new alarm type – Network Disconnection; Added: Alarm processing new method – Notify the alarm server; Added: The alarm server parameters; Updated: Stream type in snapshot parameters. Added: Get and Set UPNP parameters; Added: FTP Testing Command Added: SMTP Testing Command Added: RTSP Parameters Added: IP Filter Parameters Updated: add “stream type” into VoIP parameters	Lin Qing	2017.06.12
46	Added: Status of “Motion Detection” Alarm; Updated: Upload data format of motion detection alarm; Updated: add “threshold” of motion; Added: Auto Reboot Command;	Lin Qing	2017.06.19
47	Added: Get PPPoE Status	Lin Qing	2017.06.23

48	Added: Devability ONVIF Added: Devability Audio Input; Added: DevPara P2P Enable/Disable	Lin Qing	2017.06.26
49	Optimized PTZ control and command	LIN QING	2017.08.25
50	Added: Get and set NFS parameters 2.19	LIN QING	2017.08.30
51	Updated: Get the storage device information 5.2 (add NFS) ; Updated: Get and set the snapshot codec parameters 3.2 (add NFS path) Updated: Get and set the recording parameters 7.2 (add NFS path)	LIN QING	2017.09.02
52	Added: 2.20 Get and set IEEE802.1x parameters Added: 2.21 Get and set SNMP parameters	LIN QING	2017.09.11
53	Updated: 2.15 Get and Set uPNP parameters: added “active” to support enabling and disabling the service; Updated: 3.1 Video parameters: added “active” to support enabling and disabling the stream.	LIN QING	2017.09.15
54	Updated: 5.1 added parameter “2: restore all the parameters except TCP/IP parameters"; Updated: 2.1 added parameter “type” which is used to support define the network adaptor work mode;	LIN QING	2017.11.15
	Updated 3.9: added the parameter “marter” which is used to support Smart IR enable or disable; Updated 1.2: added the parameter “marter” which is used to get the value if the camera supports Smart IR or not.	QIN Jian	2018.03.16
56	Updated 8.1 added PTZ parameters including cruise, baud rate, protocols type etc. Updated 8.2 added cruise command.	Qin Jian	2018.03.22
57	Updated 8.2 Updated Preset data format;	Qin Jian	2018.04.08
58	Updated 8.2 added RS-485 parameters	Qin Jian	2018.04.27
59	Updated 6.5 delete the interface to get the Raw data	Qin Jian	2018.06.12

60	Updated 3.1Get and Set Video encode parameters, added: Enable or Disable Audio	QING LIN	2018.06.15
61	Updated: Limitation for MSTAR M313E and M316DM for CGI like below APIs: 2.13 GET the MJPEG Snapshot URL; 3.2 Get and Set Snapshot stream codec parameters 3.4 Get Video codec capacity parameters 3.9 Get and Set Image parameters 4.1 Get and Set Alarm Parameters	QING LIN	2018.12.06
62	Updated the device's capacity information: change onvif segment to “nvrproto” segment; Update the device system information: add nvr protocols enable and disable segment; Updated the device system information: add “telnet” enable and disable segment;	QING LIN	2018.12.06
63	Update Image parameters settings. 3.3 added g.726: codec: codec type (0: g711u 1: g711a 2:AAC 3: mp2 4: pcm 5: g726) 3.9 added: imgstyle: Display mode: 0: standard; 1: bright; 2: vivid; 3: Gentle; 4: custom 3.15 Get and Set Image Parameters Templates 4,1 added new alarm: /action/get?subject=alarm&type=0 [type: Alarm Type (0:IO alarm 1: Device Startup 2: Motion Detection 3: Video Shield 4: PIR 5: Disconnection 10: Line crossing 11: Intrusion 12: Human detection 13: Face detection 14: Object left and remove 15: Wondering)]	QING LIN	2019.04.02
64	Integrated the IVA (Intelligent Video Analytics) CGI into the document. Please refer to chapter 11.	QING LIN	2019.04.02
65	Updated Face detection and recognition APIs as below:	QING LIN	2019.06.25

	Facedetect: the parameters related with face detection and facial recognition; Facepic for getting the snapshot of the detected faces; API facereco to get the result of the face recognition and comparision; API facereginfo for face database management: adding, deleting, modifying and searching etc.		
66	API line crossing, add the enable/disable “show the detection line” parameter; API Intrusion, add the enable/disable “show the detection frame” parameter; API for download the device information certificate files. API for upload the FD/FR authorized files; API for upload the FR model files.	QING LIN	2019.07.23
67	API 11.21 LPR/ANPR related added	QING LIN	2019.08.16
68	Adjust the Smart Detection, IVA (Intelligent Video Analytics) and AI based functions in different parts.	QING LIN	2019.11.06
69	Adjust the sensitivity value range of the crossing line to 1-4;	Jerry	2019.11.18
70	Update the license plate effective time, adding and deletion; Update the license plate number searching;	Jerry	2020.12.05
71	Add the API to search the people count statistic file per day. Add the API to download the statistic files to the computer; Add the API to upload the statistic files to the FTP server;	Jerry	2020.02.27
72	Add the API to set the video working mode: HDR and 50/60FPS in media; Add the capability of the device to support HDR and 50/60FPS;	Jerry	2020.02.28
73	Add the API: Preparation for the FW update using the FW update API.	Jerry	2020.03.04
74	Recheck and make full AI functions supported. Update to Version 5.0.01	Jerry	2020.03.01

75	Update 13.5.2 to support " bw: 0:black 1:white"	Jerry	2020.04.20
76	Update 13.3 to support masking wearing detection. Update 4.1 add the alarm type: the face with mask or without mask.	Jerry	2020.05.10
77	Added the API: 2.22 to 2.34 related with Bonjour, QoS, HTTPS and 4G connection	Jerry	2020.05.10
78	Updated 4.3 Alarm data to support more type of alarm: /FACERECO_FAILURE/FACERECO_BNAME/FACERECO_WNAME/FACEDECTION_MASK /FACEDECTION_NOMASK	Jerry	2020.06.05
79	Update 1.2 /action/get?subject=devability add the new item which is used for extended Live stream extended protocol.	Jerry	2020.06.15
80	Update 6.1 the video and audio stream definition, to support the new frame definition when the liveproto =1 in the API 1.2	Jerry	2020.06.15
81	Add: 6.7 API to get the shared data in camera buffer	Jerry	2020.06.15
82	Add: LPR support FTP upload DBF file.	Jerry	2020.07.22
83	Change the API 4.6 - Alarm IO parameter to support manual/time and linkage night and day method.	Zhaofan wang	2020.08.25
84	Added Amazon Alexa authorization code settings. 1.8 Added Amazon Alexa authorization status settings.1.9	Xin Tan	2020.11.10
85	Change the device capability for acquiring the Full time Color capability and LEDs qty. 3.15	Xin Tan	2020.11.23

86	Update the “Find the password” set and get command	Xin Tan	2021.03.15

.

Table of Contents

❑ System part

1 System

1.1 Get the Device Information

- **Description**

The interface to get the device information including: device name, firmware version, hardware version, seqno, device's uuid, model name and device's QR code etc.

- **Request URL**

```
/action/get?subject=devinfo
```

- **Request Body**

None.

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <devinfo>
        <name>HDIPCAM</name>
        <softver>V1.2</softver>
        <hardver>V1.2</hardver>
        <seqno>xxxxxxxxxxxxx</seqno>
        <uuid>xxxxxxxxxxxxx</uuid>
        <model>xxxxxxxxxxxxx</model>
        <manufacture>xxxxxxxxxxxxx</manufacture>
        <qrcode>xxxxxxxxxxxxx</qrcode>
     </devinfo>
</response>
```

- **Further description**

name: Device Name

softver: Firmware Version

hardver: Hardware Version

seqno: Device's Serial Number (Sequence number)

uuid: Device's UUID

model: Device's Model Name

manufacture: The Manufacture

qrcode: Device's QR code

1.2 Get Device's Capacity Information

- **Description**

The interface to get the capacity or ability information like IO alarm in/out interfaces, video in/out interfaces, audio in/out interfaces, PTZ, USB interfaces, PIR, video stream and features of the functionalities like motion detection, privacy, tamper and OSD etc.

- **Request URL**

```
/action/get?subject=devability
```

- **Request Body**

None.

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <devability>
        <adapter>1</adapter>
        <ioin>1</ioin>
        <ioout>1</ioout>
        <vin>1</vin>
        <vout>0</vout>
        <ain>1</ain>
        <aout>1</aout>
        <ptz>0</ptz>
        <serial>0</serial>
        <usb>0</usb>
        <disk>0</disk>
```

```
        <stream>3</stream>
        <motion>4</motion>
        <privacy>4</privacy>
        <tamper>1</tamper>
        <osd>5</osd>
        <smartva>1</smartva>
        <smartva_alg>1</smartva_alg>
        <nvrproto>63</nvrproto>
        <aimode>2</aimode>
        <smartir>0</smartir>
        <porch>1</porch>
        <atto_ab>1</atto_ab>
        <hdr>1</hdr>
        <highfps>1</highfps>
        <liveproto>1</liveproto>
        <irled>1</irled>
        <fullcolor>1</fullcolor>
    </devability>
</response>
```

- **Further description**

adapter: Number of network adapters supported.

ioio: Number of IO input supported.

ioout: Nummer of IO output supported.

vin: Number of video input supported.

vout: Number of video ouput supported.

ain: Number of audio input supported.

aout: Number of audio output supported.

ptz: If the PTZ supported or not (0: NO, 1: YES)

serial: Number of the serial ports supported.

usb: Number of the USB devices supported.

disk: Number of the storage devices supported.

pir: If the PIR supported or not (0: No, 1: YES)

stream: Number of the video streams supported.

motion: Number of the motion detection zones supported.

privacy: Number of the privacy masking zones supported.

tamper: Number of the video shield zones supported.

osd: Number of the OSD supported.

Smartva: If the Intelligent video analysis supported or not. (0: No, 1: YES)

nvrproto: NVR compatibility protocols supported (bit0: onvif bit1: DH, bit2: HK bit3: HB bit4:AL bit5: XM)

Aimode: Audio Input mode. (1: Auto Mode, 2: Manual Mode.)

Smarteir: if the Smart IR supported or not. (0: No, 1: YES)

Hdr: if HDR video working mode supported or not (0: No, 1: YES)

Highfps: if HIGHFPS video working mode supported or not. (0:NO, 1: YES)

Liveproto: the type protocol version for realtime streaming, 0: version 0; 1: version 1.0, only version 1.0 can support frame head extended data.

Irled: LED capability: 0: no LED; 1: 1 White or IR LED installed; 2: 1 White LED + 1 IR LED;

Fullcolor: Full time color model or not, 0: No. 1: yes, it's full time color model

Smart_alg: definition of the smart video analytics algorithm supported; it includes:

```
enum
{
  SMARTVIDEO_ALG_UNSUPPORT = 0,
  SMARTVIDEO_ALG_IVE,
  SMARTVIDEO_ALG_JABSCO,
  SMARTVIDEO_ALG_HSFD,
  SMARTVIDEO_ALG_HSFR,
  SMARTVIDEO_ALG_LPRJIC,
  SMARTVIDEO_ALG_LPRIV
};
```

atto_ab: It's customization FW kinds definition, it is internal parameter used.

Note:

When “aimode” is manual mode, it means the device supports both Mic and Line audio input at the same time, and also support both switches manually.

1.3 Get and Set the device system information

- **Description**

The interface to get and set the data, time and system language of the device.

- **Request URL**

GET:

```
/action/get?subject=devpara
```

SET:

```
/action/set?subject=devpara
```

■ Request Body

GET:

None.

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <devpara>
        <name>HDIPCAM</name>
        <datefmt>0</datefmt>
        <timefmt>0</timefmt>
        <language>0</language>
        <p2p>0</p2p>
        <telnet>0</telnet>
        <nvrproto>1</nvrproto>
     </devpara>
</request>
```

■ Response

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <devpara>
        <name>HDIPCAM</name>
        <datefmt>0</datefmt>
        <timefmt>0</timefmt>
        <language>0</language>
```

```
        <p2p>0</p2p>
        <telnet>1</telnet>
        <nvrproto>1</nvrproto>
    </devpara>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

- **Further description**

name: Device Name

datefmt: System Data Format(0: Year-Month-Day, 1: M-Y-D, 2: D-M-Y)

timefmt: System Time Format (0: 24 Hours, 1: 12 Hours)

langauge: System Language (Not used now)

p2p: Enable/Disable P2P feature. (0: Disable, 1: Enable)

telnet: enable/disable Telnet service. (0: Disable, 1: Enable)

nvrproto: enable the NVR/CMS compatibility protocols supported. (bit0: onvif, bit1:DH bit2:HIK bit3: HB bit4: AL bit5: XM)

1.4 Get and Set the device system time

- **Description**

The interface to get and set the device system time.

- **Request URL**

GET:

```
/action/get?subject=systime
```

SET:

```
/action/set?subject=systime
```

■ **Request Body**

GET:

None.

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <systime>
        <mode>0</mode>
        <tz>CST-8</tz>
        <datetime>2015-07-08T12:15:25</datetime>
        <dst>
            <active>0</active>
            <begin>
                <month>0</month>
                <week>0</week>
                <day>0</day>
                <second>0</second>
             </begin>
            <end>
                <month>0</month>
                <week>0</week>
                <day>0</day>
                <second>3600</second>
             </end>
        </dst>
        <ntp>
            <host>ntp.server</host>
            <port>123</port>
            <interval>1</interval>
        </ntp>
```

```
    </systime>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <systime>
        <mode>0</mode>
        <tz>CST-8</tz>
        <datetime>2015-07-08T12:15:25</datetime>
        <dst>
            <active>0</active>
            <begin>
                <month>0</month>
                <week>0</week>
                <day>0</day>
                <second>0</second>
            </begin>
            <end>
                <month>0</month>
                <week>0</week>
                <day>0</day>
                <second>3600</second>
            </end>
        </dst>
        <ntp>
            <host>ntp.server</host>
            <port>123</port>
            <interval>1</interval>
        </ntp>
    </systime>
</response>
```

SET:

200: Succeed to Set.

400: Error to Request.

403: No authorized.

500: Failed to Set.

■ **Further description**

mode: The Time Set Mode (0: Timing Manual, 1: NTP)

tz: Time Zone

datetime: Current System Time (Format is YYYY-MM-DD Thh:mm:ss)

dst.active: DST(Daylight saving time) or not (0: NO, 1: YES)

ntp.host: NTP Server Address

ntp.port: NTP Server Ports

ntp.interval: NTP Timing Interval, Unite: Hour, Valid value [1, 5]

dst.active: If enable Auto DST or not. (0: Disable, 1: Enable)

dst.begin.month: the start month of DST, the valid value range is [0,11] (from Jan to Dec.)

dst.begin.week: the start week of DST, the valid value range is [0,4] (0, the first week, 4, the last week)

dst.begin.day: the start day of DST, the valid value range is [0,6] (Monday to Sunday)

dst.begin.second: the start second of DST, the unit is second.

dst.end: the end of DST. Please refer to “dst.begin” for the parameters definition.

below is the standard time zone for the reference.

```
<OPTION VALUE="<GMT+14>+14" name="x_timezoneOpt1"></OPTION>
<OPTION VALUE="<GMT+13>+13" name="x_timezoneOpt2"></OPTION>
<OPTION VALUE="<GMT+12>+12" name="x_timezoneOpt3"></OPTION>
<OPTION VALUE="SST11" name="x_timezoneOpt4"></OPTION>
<OPTION VALUE="HAST10HADT,M3.2.0,M11.1.0" name="x_timezoneOpt5"></OPTION>
<OPTION VALUE="AKST9AKDT,M3.2.0,M11.1.0" name="x_timezoneOpt6"></OPTION>
<OPTION VALUE="PST8PDT,M3.2.0,M11.1.0" name="x_timezoneOpt7"></OPTION>
<OPTION VALUE="MST7MDT,M3.2.0,M11.1.0" name="x_timezoneOpt8"></OPTION>
<OPTION VALUE="CST6CDT,M3.2.0,M11.1.0" name="x_timezoneOpt9"></OPTION>
<OPTION VALUE="CST6CDT,M4.1.0,M10.5.0" name="x_timezoneOpt10"></OPTION>
<OPTION VALUE="CST5CDT,M3.2.0/0,M11.1.0/1" name="x_timezoneOpt11"></OPTION>
<OPTION VALUE="EST5EDT,M3.2.0,M11.1.0" name="x_timezoneOpt12"></OPTION>
<OPTION VALUE="VET4:30" name="x_timezoneOpt13"></OPTION>
<OPTION VALUE="PYT4PYST,M10.3.0/0,M3.2.0/0" name="x_timezoneOpt14"></OPTION>
<OPTION VALUE="CLT4CLST,M10.2.0/0,M3.2.0/0" name="x_timezoneOpt15"></OPTION>
<OPTION VALUE="AST4ADT,M3.2.0,M11.1.0" name="x_timezoneOpt16"></OPTION>
<OPTION VALUE="NST3:30NDT,M3.2.0/0:01,M11.1.0/0:01" name="x_timezoneOpt17"></OPTION>
<OPTION VALUE="BRT3BRST,M10.3.0/0,M2.3.0/0" name="x_timezoneOpt18"></OPTION>
<OPTION VALUE="FNT2" name="x_timezoneOpt19"></OPTION>
<OPTION VALUE="AZOT1AZOST,M3.5.0/0,M10.5.0/1" name="x_timezoneOpt20"></OPTION>
```

<OPTION VALUE="GMT0BST,M3.5.0/1,M10.5.0" name="x_timezoneOpt21"></OPTION>
<OPTION VALUE="CET-1CEST,M3.5.0,M10.5.0/3" name="x_timezoneOpt22"></OPTION>
<OPTION VALUE="EET-2EEST,M3.5.0/3,M10.5.0/4" name="x_timezoneOpt23"></OPTION>
<OPTION VALUE="EET-2EEST,M4.4.5/0,M8.4.5/2" name="x_timezoneOpt24"></OPTION>
<OPTION VALUE="IST-2IDT,M3.5.5/2,M9.3.0/2" name="x_timezoneOpt25"></OPTION>
<OPTION VALUE="SAST-2" name="x_timezoneOpt26"></OPTION>
<OPTION VALUE="EET-2EEST,M3.5.0/0,M10.5.0/0" name="x_timezoneOpt27"></OPTION>
<OPTION VALUE="EET-2EEST,M4.1.5/0,J305/0" name="x_timezoneOpt28"></OPTION>
<OPTION VALUE="MSK-3MSD,M3.5.0,M10.5.0/3" name="x_timezoneOpt29"></OPTION>
<OPTION VALUE="AST-3ADT,J91/3,J274/4" name="x_timezoneOpt30"></OPTION>
<OPTION VALUE="IRST-3:30IRDT-4:30,80/0,264/0" name="x_timezoneOpt31"></OPTION>
<OPTION VALUE="AZT-4AZST,M3.5.0/4,M10.5.0/5" name="x_timezoneOpt32"></OPTION>
<OPTION VALUE="AFT-4:30" name="x_timezoneOpt33"></OPTION>
<OPTION VALUE="PKT-5" name="x_timezoneOpt34"></OPTION>
<OPTION VALUE="IST-5:30" name="x_timezoneOpt35"></OPTION>
<OPTION VALUE="NPT-5:45" name="x_timezoneOpt36"></OPTION>
<OPTION VALUE="OMST-6OMSST,M3.5.0,M10.5.0/3" name="x_timezoneOpt37"></OPTION>
<OPTION VALUE="MMT-6:30" name="x_timezoneOpt38"></OPTION>
<OPTION VALUE="WIT-7" name="x_timezoneOpt39"></OPTION>
<OPTION VALUE="CST-8" SELECTED name="x_timezoneOpt40"></OPTION>
<OPTION VALUE="WST-8WDT,M10.5.0/2/3,M3.5.0/3" name="x_timezoneOpt41"></OPTION>
<OPTION VALUE="JST-9" name="x_timezoneOpt42"></OPTION>
<OPTION VALUE="CST-9:30" name="x_timezoneOpt43"></OPTION>
<OPTION VALUE="CST-9:30CST,M10.1.0,M4.1.0/3" name="x_timezoneOpt44"></OPTION>
<OPTION VALUE="EST-10EST,M10.1.0,M4.1.0/3" name="x_timezoneOpt45"></OPTION>
<OPTION VALUE="SBT-11" name="x_timezoneOpt46"></OPTION>
<OPTION VALUE="NFT-11:30" name="x_timezoneOpt47"></OPTION>
<OPTION VALUE="NZST-12NZDT,M9.5.0,M4.1.0/3" name="x_timezoneOpt48"></OPTION>
<OPTION VALUE="FJT-12" name="x_timezoneOpt49"></OPTION>
<OPTION VALUE="PETT-12PETST,M3.5.0,M10.5.0/3" name="x_timezoneOpt50"></OPTION>
<OPTION VALUE="MHT-12" name="x_timezoneOpt51"></OPTION>
<OPTION VALUE="CHAST-12:45CHADT,M9.5.0/2:45,M4.1.0/2:45" name="x_timezoneOpt52"></OPTION>
<OPTION VALUE="TOT-13" name="x_timezoneOpt53"></OPTION>

1.5 Interfaces of the User Management

- **Description**

The interface is for the user management including getting the user information, user adding, deleting and modification etc.

- **Request URL**

Get the user information:

```
/action/get?subject=user
```

User adding:

```
/action/set?subject=user&do=add
```

User deleting:

```
/action/set?subject=user&do=del
```

User modification:

```
/action/set?subject=user&do=modify
```

- **Request Body**

Get the user information: None

User adding:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <user>
        <name>tester</name>
        <password>12345</password>
        <group>0</group>
        <permit>
            <config>255</config>
            <operation>255</operation>
        </permit>
     </user>
</request>
```

User deleting:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <user>
        <name>tester</name>
     </user>
</request>
```

User modification:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <user>
        <name>tester</name>
        <password>54321</password>
        <group>0</group>
        <permit>
            <config>255</config>
            <operation>255</operation>
        </permit>
     </user>
</request>
```

■ **Response**

Get the user information:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <user>
        <name>admin</name>
        <password>12345</password>
        <group>0</group>
        <permit>
            <config>255</config>
            <operation>255</operation>
        </permit>
     </user>
       ...
      <user>
        <name>test1</name>
        <password>12345</password>
        <group>0</group>
        <permit>
```

```
            <config>255</config>
            <operation>255</operation>
        </permit>
    </user>
</response>
```

User adding:

User deleting:

User modification:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

name: user name

password: user's password, here please use base64 encrypted password.

group: user group ((0: administrator 1: Operator 2: Viewer)

perimt.config: permission configuration, each bit means each permission for camera configuration;

perimt.operation: permission operation, each bit means each permission for camera operating;

NOTE:

1. All of the existed user information will be returned, and the password returned is NULL.

2. The current right information is valid, please use the user group to assign the user's permission right. The right definition of the three type users: Administrator, Operator and Viewer is like the blow:

Administrator's Right:

Parameters Settings including: System Parameters, Network Parameters, Audio and Video Parameters, Alarm and Event Parameters, Recording Parameters, User Management, Intelligent Video Analytics Parameters, Video Preview, Playback, Two-way speaking, SD Operation, PTZ control, Firmware Upgrade, Reboot and Restore etc.

Operator's Right:

Parameters Settings including: System Parameters, Network Parameters, Audio and Video Parameters, Alarm and Event Parameters, Recording Parameters, Intelligent Video Analytics Parameters, Video Preview, Playback, Two-way speaking, SD Operation, PTZ control, Reboot and Restore etc.

Without: User Management and Firmware Upgrade

Viewer Right:

ONLY get the following parameters: System Information, Network Parameters, Audio and Video Parameters, Alarm and Event Parameters, Recording Parameters, Intelligent Video Analytics Parameters, Video Preview;

3. The password should be encrypted using BASE-64 when the user is added or modified.

1.6 Interfaces of acquiring the device running status

- **Description**

The interface is for the user to acquiring the device running status and server status etc.

- **Request URL**

```
/action/get?subject=sysstatus
```

- **Request Body**

None

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <systemstat>
        <cpu>80</cpu>
        <uptime>80</uptime>
        <video>1</video>
        <audio>1</audio>
        <onvif>1</onvif>
        <voip>1</voip>
        <rtmp>1</rtmp>
        <rtsp>1</rtsp>
        <upnp>1</upnp>
        <ftp>1</ftp>
        <record>1</record>
        <snap>1</snap>
        <tcp>1</tcp>
```

```
        <ddns>1</ddns>
        <ioin>1</ioin>
        <ioout>1</ioout>
    </systemstat>
</response>
```

■ **Further description**

cpu: cpu loading percentage，value: [0, 100]

uptime: the running time from the camera started, value: seconds

video: video service status: 1: running 0: stop

audio: audio service status: 1: running 0: stop

onvif: onvif service status: 1: running 0: stop

voip: voip service status: 1: running 0: stop

ftp: ftp service status: 1: running 0: stop

rtmp: rtmp service status: 1: running 0: stop

upnp: upnp service status: 1: running 0: stop

rtsp: rtsp service status: 1: running 0: stop

record: recording service status: 1: running 0: stop

snap: snapshot service status: 1: running 0: stop

tcp: tcp service status: 1: running 0: stop

ddns: ddns service status: 1: running 0: stop

ioin: the current IO alarm input status, each bit represents each IO, maximum support 32 IO, 1: high level 0: low level

ioout: the current IO alarm output status, each bit represents each IO, maximum support 32 IO, 1: high level 0: low level

1.7 Interfaces of acquiring and setting IP filter parameters

■ **Description**

The interface is for the user to acquire and set the IP address filter list in the camera.

■ **Request URL**

Get:

```
/action/get?subject=ipfilter
```

Set:

```
/action/set?subject=ipfilter
```

- **Request Body**

GET:

None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <ipfilter>
        <active>0</active>
        <filter>0</filter>
        <iprule>
            <name>12312</name>
            <start>192.168.1.35</start>
            <end>192.168.1.155</end>
         </iprule>
         <iprule>
            <name>12312</name>
            <start>192.168.1.35</start>
            <end>192.168.1.155</end>
         </iprule>
         ...
     </ipfilter>
</request>
```

- **Response**

GET

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

```
<response>
    <ipfilter>
        <active>0</active>
        <filter>0</filter>
        <iprule>
            <name>12312</name>
            <start>192.168.1.35</start>
            <end>192.168.1.155</end>
         </iprule>
         <iprule>
            <name>12312</name>
            <start>192.168.1.35</start>
            <end>192.168.1.155</end>
         </iprule>
         ...
     </ipfilter>
</response>
```

Set:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

active: If enable IP filter list or not.

Filter: IP address filter type: 0: White list, 1: Black list.

iprule.name: the name of filter

iprule.start: the start IP address which should be filtered.

iprule.end: the end IP address which should be filtered.

Note:

1. Max. 4 filters can be supported.

2. The starting and ending IP address should be the same IP segment.

1.8 Interfaces of acquiring the device ID code

- **Description**

The interface is for the user to acquire the device ID code. Each device including the IP cameras and NVRs have a unique device ID code for some application.

- **Request URL**

```
/action/get?subject=ident_code
```

- **Request Body**

None

- **Response**

GET

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <ident_code ver="2.0">
        <data>ZjA6MjM6Yjk6NTA6NTA6NTItODg4ODg4ODg4ODg4ODg4ODg4OC0w</data>
</ident_code>
</response>
```

- **Further description**

Data is the base64 code string of the device identification code. (Alexa authorization binding device requires license, identification code, authorization identification, etc.)

1.9 Interfaces of acquiring and setting Alexa authorization information

- **Description**

The interface is for the user to acquire and set Alexa authorization information.

■ **Request URL**

Get:

```
/action/get?subject=alexa_lic
```

Set:

```
/action/set?subject=alexa_lic
```

■ **Request Body**

GET:

None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <license>
    <lic>RHZL-00-ZR4Q-T8X5-00000107</lic>
    </license>
</request>
```

■ **Response**

GET

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <lic_stat ver="2.0">
        <stat>0</stat>
        <active_sec>1604993763</active_sec>
        <expire_sec>120</expire_sec>
        <remain_sec>70</remain_sec>
        <lic>1111</lic>
    </lic_stat>
```

```
</response>
```

Set:

```
200: Succeed to Set.

400: Error to Request.

403: No authorized.

500: Failed to Set.
```

- **Further description**

lic is the authorization code that is bound to modify the device characteristics by the authorized system.

stat: 0: authorized -1: not authorized

active_sec: authorization activation time, UTC seconds (0 when there is no authorization)

expire_sec: authorization validity period set by the system, in seconds (no authorization is 0)

remain_sec: How long is the authorization valid, in seconds (no authorization is 0)

lic: authorization code

1.10 Interfaces for resetting the user's password

- **Description**

The interfaces are used for resetting the user's password. The interface can be used for the end user to change the new password when the old password is forgot.

- **Request URL**

Get:

```
/action/get?subject=changepasswd
```

Set:

```
/action/set?subject=changepasswd
```

- **Request Body**

GET:

None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <changepasswd>
    <key>12345</key>
    <newpwd>abcde</newpwd>
    </changepasswd>
</request>
```

■ **Response**

GET

```
<?xml version="1.0" encoding="utf-8"?>
<?xml version="1.0" encoding="utf-8"?>
<response>
    <changepasswd ver="2.0">
        <key>B87AEFFA72D4F3105A223634C7445460BAB9CA1CDCA6</key>
        <time>86400</time>
    </changepasswd>
</response>
```

Set:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

/action/get?subject=changepasswd:

Key is the string of reset password, which consists of two parts.

The first part is MD5 + AES encryption of a random string.

The second part uses six numbers of MAC address as metadata. Use the MD5 value of the serial number to do AES encryption to get the string.

These two parts are combined to form a key.

The key is sent to the manufacturer as a credential. The manufacturer uses the password tool to compare the key with the MAC address and serial number provided by the user. In order to confirm whether the key belongs to the device with the MAC address and SN.

/action/set?subject=changepasswd

Key is the provided key number, please be noted that this key should be the returned result by the password tool, it's different as the above “key”.

Newpwd is the Base64 new password.

❑ Network part

2 Network

2.1 Get and Set the Network Parameters

■ **Description**

The interface to get and set the device's network parameters like the network adaptor's name, Mac address, IP protocols, IP address, gateway address and DNS etc.

■ **Request URL**

GET:

```
/action/get?subject=network&adapter=0    [adapter: Network Adapter's ID number (0: local, 1: wifi)]
```

SET:

```
/action/set?subject=network&adapter=0
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <network>
        <network>
        <name>eth0</name>
        <mac>d0:22:12:88:88:88</mac>
        <proto>1</proto>
        <index>0</index>
        <active>1</active>
        <mtu>64</mtu>
        <type>0</type>
        <ipv4>
            <mode>0</mode>
```

```
            <ip>192.168.1.120</ip>
            <mask>255.255.255.0</mask>
            <gateway>192.168.1.1</gateway>
            <dns>8.8.8.8</dns>
            <dns>192.168.1.1</dns>
        </ipv4>
        <ipv6>
            <mode>0</mode>
            <ip>2001:da8:2004:1000:202:116:160:41</ip>
            <prefixlen>64</prefixlen>
            <gateway>2001:da8:2004:1000::1</gateway>
            <dns>2001:da8:2004:1000::1</dns>
            <dns>2001:da8:2004:1000::2</dns>
        </ipv6>
    </network>
    </network>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <network>
        <name>eth0</name>
        <mac>d0:22:12:88:88:88</mac>
        <proto>1</proto>
        <index>0</index>
        <active>1</active>
        <mtu>64</mtu>
        <type>0</type>
        <ipv4>
            <mode>0</mode>
            <ip>192.168.1.120</ip>
            <mask>255.255.255.0</mask>
```

```
            <gateway>192.168.1.1</gateway>
            <dns>8.8.8.8</dns>
            <dns>192.168.1.1</dns>
        </ipv4>
        <ipv6>
            <mode>0</mode>
            <ip>2001:da8:2004:1000:202:116:160:41</ip>
            <prefixlen>64</prefixlen>
            <gateway>2001:da8:2004:1000::1</gateway>
            <dns>2001:da8:2004:1000::1</dns>
            <dns>2001:da8:2004:1000::2</dns>
        </ipv6>
    </network>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

name: Network Adapter's Name (Read Only)

mac: Mac Address (Read Only)

index: Index number of the network cards (Read only);

active: Enable or Disable network cards; 0: disable, 1: enable;

proto: Network Protocol (bit0: IPV4, bit1:IPV6)

mtu: MTU Value, the value range [500, 1500];

type: Network adapter's work mode: 0: adaptive 1: 10M HALF, 2: 10M FULL 3: 100M HALF 4: 100M FULL;

ipv4.mode: IP Address Setting Mode(0: Static, 1: DHCP)

ipv4.ip: IP Address.

ipv4.mask: IP V4 Subnetwork mask address

ipv4.gateway: Gateway address

ipv4.dns: Primary DNS address

ipv4.dns: Secondary DNS address

ipv6.mode: IP Address Setting Mode(0: Static, 1: DHCP)

ipv6.ip: IP Address.

Ipv6.prefixlen: IPV6 mask value

Ipv6.gateway: Gateway address

Ipv6.dns: Primary DNS address

Ipv6.dns: Secondary DNS address

NOTE:
1. The name of the network adaptor and MAC address are fixed, and cannot be set.
2. The adapter should be the same as the physical index of the network adapter, and it should be greater than or equal to 0;
3. Now the address of IPV6 is supported as Global type address;
4. The local wired network is always enabled, and WiFi network can be configured enable or disable;

2.2 Get Network Link Status

- **Description**

The interface to get the network link status like to check whether the device is connected or not, MTU value, WiFi Parameters like WiFi SSID, Authentication method, WiFi Encryption Method and the Wireless signal strength status etc.

- **Request URL**

```
/action/get?subject=netstat&adapter=0     [adapter: Network Adapter's ID number (0: local, 1: wifi)]
```

- **Request Body**

None

- **Response**

```
<response>
    <netstat>
        <link>1</link>
        <network>
            <mtu>1500</mtu>
            <name>eth0</name>
```

```
        <mac>d0:22:12:88:88:88</mac>
        <proto>1</proto>
        <ipv4>
          <mode>0</mode>
          <ip>192.168.1.120</ip>
          <mask>255.255.255.0</mask>
          <gateway>192.168.1.1</gateway>
          <dns>8.8.8.8</dns>
          <dns>192.168.1.1</dns>
        </ipv4>
         <ipv6>
          <mode>0</mode>
          <ip>2001:da8:2004:1000:202:116:160:41</ip>
          <prefixlen>64</prefixlen>
          <gateway>2001:da8:2004:1000::1</gateway>
          <dns>2001:da8:2004:1000::1</dns>
          <dns>2001:da8:2004:1000::2</dns>
        </ipv6>
      </network>
      <wifiap>
          <essid>WIFI-AP</essid>
          <auth>3</auth>
          <alg>3</alg>
          <signal>-41</signal>
      </wifiap>
  </netstat>
  </response>
```

■ **Further description**

link: Link status（0: Not Connected, 1: Connected)

mtu: MTU value

network: Prefer to the Network parameters definition.

wifiap.essid: AP's ESSID of the current WIFI connected.

wifiap.auth: AP's Password Authentication Method(0: NONE, 1:WEP　2:WPA 3:WPA2 ）

wifiap.alg: AP's Password Encryption Method(0:OPEN 1:SHARED 2:TKIP 3:AES）

wifiap.signal: AP's Signal Strength Status

NOTE:
The “wifiap” of the answered package can be valid only on adapter=1 when getting WIFI network status.

2.3 Get and Set Network Port Parameters

■ **Description**

The interface to get and set the network port parameters including HTTP port, RTSP port and Private TCP port.

■ **Request URL**

GET:

```
/action/get?subject=netserv
```

SET:

```
/action/set?subject=netserv
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <netserv>
        <http>80</http>
        <rtsp>554</rtsp>
        <tcp>6000</tcp>
    </netserv>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <netserv>
        <http>80</http>
        <rtsp>554</rtsp>
        <tcp>6000</tcp>
    </netserv>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

http: http port

rtsp: rtsp port

tcp: private tcp port

2.4 Get and Set SMTP Parameters

■ **Description**

The interface to get and set the SMTP parameters like the SMTP host server address, SMTP host server port number, the user name, password, SSL encryption status, the sender name and the receptor's mail address.

■ **Request URL**

GET:

```
/action/get?subject=smtp
```

SET:

```
/action/set?subject=smtp
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <smtp>
        <host/>
        <port>25</port>
        <username/>
        <password/>
        <ssl>0</ssl>
        <sender/>
        <recipient/>
        <recipient/>
        <recipient/>
        <recipient/>
    </smtp>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <smtp>
        <host>smtp.126.com<host/>
        <port>25</port>
        <username>test@126.com</username>
        <password>123456</password>
        <ssl>0</ssl>
        <sender>Tester</sender>
```

```
            <recipient>test1@163.com</recipient>
            <recipient>test2@163.com</recipient>
            <recipient>test3@163.com</recipient>
            <recipient>test4@163.com</recipient>
        </smtp>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

host: smtp host server address

port: smtp host server port number

username: username

password: password

ssl: enable ssl encryption （0: Disable 1: Enable）

sender: sender name

recipient: the receptor's mail address, the maximum number of the address is 4.

2.5 Get and Set DDNS Parameters

■ **Description**

The interface to get and set the DDNS parameters like whether to enable DDNS or not, the DDNS type, the domain name, the domain user name, the password and the DDNS updating interval time etc.

■ **Request URL**

GET:

```
/action/get?subject=ddns
```

SET:

```
/action/set?subject=ddns
```

■ Request Body

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <ddns>
        <active>1<active/>
        <type>25</type>
        <domain>www.ipcam.com</domain>
        <username>test@126.com</username>
        <password>123456<password/>
        <interval>0</interval>
    </ddns>
</request>
```

■ Response

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <ddns>
        <active>1<active/>
        <type>25</type>
        <domain>www.ipcam.com</domain>
        <username>test@126.com</username>
        <password>123456</password>
        <interval>0</interval>
    </ddns>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

- **Further description**

active: Enable DDNS or not（0: Disable 1: Enbale）

type: DDNS Service Type（0:3322　1:oray）

domain: Porting Domain Name

username: Domain User name

password: Domain password

interval: Updating interval, valid range [1, 600], unite: second

2.6 Get and Set PPPoE Parameters

- **Description**

The interface to get and set the PPPoE parameters like whether to enable PPPoE or not, the PPPoE user name, the password etc.

- **Request URL**

GET:

```
/action/get?subject=pppoe
```

SET:

```
/action/set?subject=pppoe
```

- **Request Body**

GET:　None

SET:

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

```
    <pppoe>
        <active>1<active/>
        <username>test@126.com</username>
        <password>123456<password/>
    </pppoe>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <pppoe>
        <active>1<active/>
        <username>test@126.com</username>
        <password>123456</password>
    </pppoe>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

active: Enable PPPoE or not（0: Disable, 1: Enable）

username: The user name

password: The password

2.7 Get PPPoE working status

■ **Description**

The interface to get the PPoE connection status.

■ **Request URL**

```
/action/get?subject=pppoestat
```

■ **Request Body**

None.

■ **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <pppoestat>
        <link>1</link>
        <addr>192.168.1.1</addr>
        <mask>192.168.1.1</mask>
        <gateway>192.168.1.1</gateway>
    </pppoestat>
</response>
```

■ **Further description**

Link: Link status. 0: not connected; 1: connected;

Addr: working IP address.

Mask: working network mask;

Gateway: working network gateway.

2.8 Get and Set FTP Parameters

■ **Description**

The interface to get and set the FTP parameters like the FTP server address, FTP server port, FTP user name and the password, whether to enable Anonymous login or not, whether to enable resume the broken transfer or not and the FTP Server remote path etc.

■ **Request URL**

GET:

```
/action/get?subject=ftp
```

SET:

```
/action/set?subject=ftp
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <ftp>
        <host>ftp.server</host>
        <port>25</port>
        <anonymous>0</anonymous>
        <username>test@126.com</username>
        <password>123456</password>
        <resume>1</resume>
        <rpath>/ipcam_root</rpath>
    </ftp>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <ftp>
        <host>ftp.server</host>
        <port>25</port>
        <anonymous>0</anonymous>
        <username>test@126.com</username>
        <password>123456</password>
        <resume>1</resume>
```

```
        <rpath>/ipcam_root</rpath>
    </ftp>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

- **Further description**

host: ftp servers address

port: ftp server port

username: ftp username

password: ftp password

anonymous: Enable Anonymous login or not（0:Disable 1:Enable）

resume: Enable Resume broken transfer or not（0: Disable 1:Enable）

rpath: ftp server remote path

2.9 Get and Set WIFI Connection Parameters

- **Description**

The interface to get and set the WiFi connection parameters like the WiFi AP SSID, the authentication method, encryption method and the WiFi AP's password etc.

- **Request URL**

GET:

```
/action/get?subject=wifi
```

SET:

```
/action/set?subject=wifi
```

- **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <wifi>
        <essid>IPC-AP</essid>
        <auth>3</auth>
        <alg>3</alg>
        <password>123456789</password>
    </wifi>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <wifi>
        <essid>IPC-AP</essid>
        <auth>3</auth>
        <alg>3</alg>
        <password>123456789</password>
    </wifi>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

essid: AP ESSID Connected

auth: AP Connected Password Authentication Method(0:NONE 1:WEP 2:WPA 3:WPA2)

alg: AP connected Password Encryption Method(0:OPEN 1:SHARED 2:TKIP 3:AES）

password: WIFI AP's Password

2.10 WIFI Scanning

- **Description**

The interface to make the camera to scan the WiFi signal, and show the scanned result like the scanned WiFi SSID, the authentication method, encryption method and the wireless signal strength status etc.

- **Request URL**

```
/action/get?subject=wifiscan
```

- **Request Body**

None

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <wifiap>
        <essid>WIFI-AP</essid>
        <auth>3</auth>
        <alg>3</alg>
        <signal>-41</signal>
    </wifiap>
   <wifiap>
        <essid>TPLINK</essid>
        <auth>3</auth>
        <alg>3</alg>
        <signal>-41</signal>
    </wifiap>
</response>
```

- **Further description**

wifi Scanning Result, there will be multiple WIFI AP if multiple Aps are scanned.

essid: AP ESSID

auth: AP's password authentication method (0: NONE 1: WEP 2: WPA 3: WPA2)

alg: AP's password encryption method（0: OPEN 1:SHARED 2: TKIP 3: AES）

signal: AP connected signal strength status

2.11 Get and Set RTMP Video Parameters

- **Description**

The interface to get and set the RTMP video parameters including the related stream type pushed using RTMP and the schedule of pushing.

- **Request URL**

GET:

```
/action/get?subject=rtmp
```

SET:

```
/action/set?subject=rtmp
```

- **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <rtmp>
        <port>1935<port/>
        <push>
            <active>1<active/>
            <url>rtmp://server:port/live/cam0</url>
            <url>rtmp://server:port/live/cam1</url>
            <tsection>0-1200</tsection>
            <tsection>1200-3600</tsection>
```

```
            <tsection>3600-7200</tsection>
            <tsection>36000-86400</tsection>
        </push>
    </rtmp>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <rtmp>
        <port>1935<port/>
        <push>
            <active>1<active/>
            <url>rtmp://server:port/live/cam0</url>
            <url>rtmp://server:port/live/cam1</url>
            <tsection>0-1200</tsection>
            <tsection>1200-3600</tsection>
            <tsection>3600-7200</tsection>
            <tsection>36000-86400</tsection>
        </push>
    </rtmp>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

Port: RTMP service port

push.active: Enable or Disable RTMP push service: 0: Disable, 1: Enable

push.url: the URL address which the video (mainstream and substream) are pushed to.

push.tsection: The schedule of the stream pushing: the time format is: starting time – ending time

Note:

1. The service port is fixed 1935 now, it cannot be changed.
2. Push.url: the maximum channels supported are 2 channels (mainstream, substream as sequence)
3. There is maximum 4-time segments for the pushing schedule supported. The value is the second of the current time, for example, 16:00:00 should be using 57600 (16*3600 second)

2.12 GET the RTSP video URL

- **Description**

The interface to get the RTSP video url address for the RTSP video player to play network video.

- **Request URL**

```
/action/get?subject=rtspurl&stream=0
```

- **Request Body**

Stream: 0: mainstream 1: substream

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <refer>
        <url>rtsp://192.168.1.120:554/live/main<url/>
        <expire>0<expire/>
    </refer>
</response>
```

- **Further description**

url: the RTSP video request address

expire: the effective time of address, the unit is second.

Note:

Expire: when it's 0, it means the RTSP address is never expired.

2.13 GET the MJPEG Snapshot URL

- **Description**

The interface to get the address of the snapshot using MJEPG.

- **Request URL**

```
/action/get?subject=snapurl
```

- Request Body

None.

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <refer>
        <url>http://192.168.1.120:80/action/snap<url/>
        <expire>0<expire/>
    </refer>
</response>
```

- **Further description**

url: the HTTP snapshot request address.

expire: the effective time of address, the unit is second.

Note:

Expire: when it's 0, it means the RTSP address is never expired.

MSTAR MSC313E 2.0MP cameras did not support this API.

2.14 Get and Set VoIP (SIP) Parameters

■ **Description**

The interface to get and set the VoIP (SIP) parameters including VoIP server address, port number, local service port, VoIP call ID number, the username and password etc.

■ **Request URL**

GET:

```
/action/get?subject=voip
```

SET:

```
/action/set?subject=voip
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <voip>
        <active>1<active/>
        <auth>1<auth/>
        <servhost>voip.host<servhost/>
        <servport>5060</servport>
        <lport>5060</lport>
        <callid>876543210</callid>
        <username>user</username>
        <password>passwd</password>
        <calltime>600</calltime>
        <expire>3600</expire>
    </voip>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <voip>
        <active>1<active/>
        <auth>1<auth/>
        <servhost>voip.host<servhost/>
        <servport>5060</servport>
        <lport>5060</lport>
        <callid>876543210</callid>
        <username>user</username>
        <password>passwd</password>
        <calltime>600</calltime>
        <expire>3600</expire>
    </voip>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

active: Disable or Enable VoIP (SIP) service, 0: Disable, 1: Enable

auth: Disable or Enable the security Authentication, 0: Disable, 1: Enable

servhost: VoIP (SIP) register server address

servport: VoIP (SIP) register server port

lport: local VoIP service monitoring port

callid: the ID number called.

Username: the registered user name of the callid

Password: the registered password of the callid;

Calltime: the maximum called time, the unit is second;

Expire: the maximum service effective time, the unit is second.

Note:

The camera will re-register the service for the VoIP server again when the current service (call) is expired.

2.15 Get and Set UPnP Parameters

- **Description**

The interface to get and set the UPnP parameters.

- **Request URL**

GET:

```
/action/get?subject=upnp
```

SET:

```
/action/set?subject=upnp
```

- **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
        <upnp>
          <active>1</active>
           <fname>123123</fname>
           <pmap>
               <active>1</active>
               <protocol>0</protocol>
               <interport>80</interport>
               <exterport>80</exterport>
```

```
            <status>1</status>
        </pmap>
        <pmap>
            <active>1</active>
            <protocol>0</protocol>
            <interport>801</interport>
            <exterport>801</exterport>
            <status>1</status>
        </pmap>
        ....
    </upnp>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <upnp>
        <active>1</active>
        <fname>123123</fname>
        <pmap>
            <active>1</active>
            <protocol>0</protocol>
            <interport>80</interport>
            <exterport>80</exterport>
            <status>1</status>
        </pmap>
        <pmap>
            <active>1</active>
            <protocol>0</protocol>
            <interport>801</interport>
            <exterport>801</exterport>
            <status>1</status>
        </pmap>
```

```
        ....
    </upnp>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

- **Further description**

Active: Enable/disable the UPNP service: 0: Disable, 1: Enable

fname: the device name discovered by UPnP;

pmap.active: Enable/disable port mapping: 0: Disable, 1: Enable.

pmap.protocol: Protocol type of port mapping: 0: TCP, 1: UDP

pmap.interport: Internal port of port mapping;

pmap.exterport: External port of port mapping;

pmap.status: Status of port mapping, it's only read, 0: port mapping failed. 1: port mapping successfully.

Note:

1. Max. 4 items port mapping are supported.
2. Pmap.status is the parameter which is read only.

2.16 FTP Testing

- **Description**

The interface to execute FTP connection testing. The user can use the interface to test if the FTP server parameters and FTP connection are ok or not.

- **Request URL**

```
/action/get?subject=ftp
```

- **Request Body**

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <ftp>
        <host>ftp.server</host>
        <port>25</port>
        <anonymous>0</anonymous>
        <username>test@126.com</username>
        <password>123456</password>
        <resume>1</resume>
        <rpath>/ipcam_root</rpath>
    </ftp>
</request>
```

- **Response**

```
200: Test successfully.
400: Error to Request.
403: No authorized.
500: Test failed.
```

- **Further description**

2.17 SMTP Testing

- **Description**

The interface to execute SMTP connection testing. The user can use the interface to test if the SMTP server parameters and the connection are ok or not.

- **Request URL**

```
/action/get?subject=smtp
```

- **Request Body**

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

```
    <smtp>
        <host>smtp.126.com<host/>
        <port>25</port>
        <username>test@126.com</username>
        <password>123456</password>
        <ssl>0</ssl>
        <sender>Tester</sender>
        <recipient>test1@163.com</recipient>
        <recipient>test2@163.com</recipient>
        <recipient>test3@163.com</recipient>
        <recipient>test4@163.com</recipient>
    </smtp>
</request>
```

- **Response**

```
200: Test successfully.
400: Error to Request.
403: No authorized.
500: Test failed.
```

- **Further description**

2.18 Get and Set RTSP Parameters

- **Description**

The interface to get and set the RTSP parameters.

- **Request URL**

GET:

```
/action/get?subject=rtsp
```

SET:

```
/action/set?subject=rtsp
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <rtsp>
        <auth>1<auth/>
        <mcast>
            <active>1<active>
            <port>1000</port>
            <addr>224.0.0.1</addr>
            <ttl>64</ttl>
        </mcast>
    </rtsp>
</request>
```

■ **Response**

GET:

```
 <?xml version="1.0" encoding="utf-8"?>
<response>
    <rtsp>
        <auth>1<auth/>
        <mcast>
            <active>1<active>
            <port>1000</port>
            <addr>224.0.0.1</addr>
            <ttl>64</ttl>
        </mcast>
    </rtsp>
```

```
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

auth: enable/disable RTSP authentication, 0: disable, 1: enable;

mcast.active: enable/disable RTSP multicasting, 0: Disable, 1: Enable.

mcast.port: multicasting port.

mcast.addr: multicasting address.

mcast.ttl: multicasting TTL value, the effective value is [64, 255].

2.19 Get and Set NFS Parameters

■ **Description**

The interface to get and set the NFS parameters.

■ **Request URL**

GET:

```
/action/get?subject=nfs
```

SET:

```
/action/set?subject=nfs
```

■ **Request Body**

GET: None

SET:

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

```
<request>
      <nfs>
          <active>1<active/>
          <host>192.168.1.25<host>
          <path>/mnt/nfs</path>
     </nfs>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
      <nfs>
          <active>1<active/>
          <host>192.168.1.25<host>
          <path>/mnt/nfs</path>
     </nfs>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

auth: enable/disable NFS service, 0: disable, 1: enable;

nfs.host: NFS server address, it should be IP address.

nfs.path.: NFS server's mounted path, it should not be empty.

2.20 Get and Set IEEE802.1x Parameters

■ **Description**

The interface to get and set the IEEE802.1x parameters.

- **Request URL**

GET:

```
/action/get?subject=ieee8021x
```

SET:

```
/action/set?subject=ieee8021x
```

- **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <ieee8021x>
        <active>1<active/>
        <identity>123123<identity>
        <password>123123s</password>
        <eapver>0s</eapver>
        <eaptype>0s</eaptype>
    </ieee8021x>
</request>
```

- **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <ieee8021x>
        <active>1<active/>
        <identity>123123<identity>
        <password>123123s</password>
```

```
        <eapver>0s</eapver>
        <eaptype>0s</eaptype>
    </ieee8021x>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

- **Further description**

active: enable/disable IEEE802.1x service, 0: disable, 1: enable;

identity: user name.

password: password.

Eaptype: EAP encryption method: 0: MD5 1: TLS

Eapver: EAP version 0: version 1 1: version 2;

2.21 Get and Set SNMP Parameters

- **Description**

The interface to get and set the SNMP parameters.

- **Request URL**

GET:

```
/action/get?subject=snmp
```

SET:

```
/action/set?subject=snmp
```

- **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
        <snmp>
            <v1active>1<v1active/>
            <v2active>1<v2active/>
            <servport>161<servport/>
            <rcommunity>public<rcommunity/>
            <wcommunity>private<wcommunity/>
            <trapip>192.168.1.25<trapip/>
            <trapport>161<trapport/>
            <tcommunity>public<tcommunity/>
            <v3active>1<v3active/>
            <rouser>
                <name>rouser<name/>
                <securty>1<securty/>
                <authtype>1<authtype/>
                <algtype>1<algtype/>
                <authpwd>1234<authpwd/>
                <algpwd>1234<algpwd/>
            </rouser>
            <rwuser>
                <name>rwuser<name/>
                <securty>1<securty/>
                <authtype>1<authtype/>
                <algtype>1<algtype/>
                <authpwd>1234<authpwd/>
                <algpwd>1234<algpwd/>
            </rwuser>
     </snmp>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
        <snmp>
            <v1active>1<v1active/>
            <v2active>1<v2active/>
            <servport>161<servport/>
            <rcommunity>public<rcommunity/>
            <wcommunity>private<wcommunity/>
            <trapip>192.168.1.25<trapip/>
            <trapport>161<trapport/>
            <tcommunity>public<tcommunity/>
            <v3active>1<v3active/>
            <rouser>
                <name>rouser<name/>
                <securty>1<securty/>
                <authtype>1<authtype/>
                <algtype>1<algtype/>
                <authpwd>1234<authpwd/>
                <algpwd>1234<algpwd/>
            </rouser>
            <rwuser>
                <name>rwuser<name/>
                <securty>1<securty/>
                <authtype>1<authtype/>
                <algtype>1<algtype/>
                <authpwd>1234<authpwd/>
                <algpwd>1234<algpwd/>
            </rwuser>
      </snmp>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
```

403: No authorized.

500: Failed to Set.

- **Further description**

v1active: enable/disable SNMPv1 service, 0: disable, 1: enable;

v2active: enable/disable SNMPv2c service, 0: disable, 1: enable;

servport: SNMP port number.

rcommunity: the name of read community;

wcommunity: the name of write community;

trapip: Trap address

trapport:Trap port

v3active: enable/disable SNMPv3, 0: disable, 1: enable

rouser.name: the name of the security for reading

rouser.securty: security class:0:no auth,no priv 1:auth,no priv 2:auth, priv

rouser.authtype: Authentication method: 0:MD5 1:SHA

rouser.authpwd: Password of authentication method;

rouser.algtype: Type Key of authentication method, 0:DES 1:AES

rouser.algpwd: Key

rwuser.name: the name of the security of writing

rwuser.securty: security class:0:no auth,no priv 1:auth,no priv 2:auth, priv

rwuser.authtype: authentication method, 0:MD5 1:SHA

rwuser.authpwd: authentication password

rwuser.algtype: Type Key of authentication method, 0:DES 1:AES

rwuser.algpwd: Key

2.22 Get and Set Bonjour Parameters

- **Description**

The interface to get and set the Bonjour parameters.

- **Request URL**

GET:

```
/action/get?subject=bonjour
```

SET:

```
/action/set?subject=bonjour
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
   <bonjour ver="2.0">
      <enable>0</enable>
      <servname>1212121</servname>
   </bonjour>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
      <bonjour ver="2.0">
            <enable>1</enable>
            <servname>IPCAM</servname>
      </bonjour>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

enable: enable/disable bonjour service, 0: disable, 1: enable;

servname: the server name (camera) for bonjour using.

2.23 Get and Set QoS Parameters

■ **Description**

The interface to get and set the QoS parameters.

■ **Request URL**

GET:

```
/action/get?subject=qos
```

SET:

```
/action/set?subject=qos
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
   <qos ver="2.0">
       <enable>1</enable>
       <desp>44</desp>
   </qos>
</request>
```

■ **Response**

GET:

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

```
<response>
    <qos ver="2.0">
        <enable>1</enable>
        <desp>44</desp>
    </qos>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

enable: enable/disable QoS service, 0: disable, 1: enable;

desp: QoS value, the range is [0-63].

2.24 Get and Set HTTPS Parameters

■ **Description**

The interface to get and set the HTTPS parameters.

■ **Request URL**

GET:

```
/action/get?subject=https
```

SET:

```
/action/set?subject=https
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
        <https ver="2.0">
        <enable>1</enable>
    </https>
</request>
```

- **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
      <https ver="2.0">
            <enable>1</enable>
      </https>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

- **Further description**

enable: enable/disable HTTPS service, 0: disable, 1: enable;

2.25 Get and Set HTTPS Information

- **Description**

The interface to get and set the HTTPS information.

- **Request URL**

GET:

```
/action/get?subject=https_info
```

SET:

```
/action/set?subject=https_info
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <https_info ver="2.0">
      <nation>1</nation>
      <ip>1</ip>
      <expire>1</expire>
      <provice>1</provice>
      <location>1</location>
      <org>1</org>
      <org_unit>1</org_unit>
      <email>1</email>
  </https_info>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <https_info ver="2.0">
        <info>C=CN</info>
        <install_info>C=CN</install_info>
        <install_attr>C=CN</install_attr>
```

```
    </https_info>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

Info: the created HTTPS request information

Install_info: the installed certificate information

Install_attr: the installed certificate Attributes

/action/set?subject=https_info is used to create HTTPS certificate as below information:

1) Nation: country;
2) IP: IP address of domain name;
3) Expire: the time for expired;
4) Province: the name of the province;
5) Location: the location name;
6) Org: the name of the organization;
7) Org_unit: the unit name of the organization;
8) Email: the email address for the certificate;

2.26 To delete the HTTPS certificate

■ **Description**

The interface to delete the HTTPS certificate.

■ **Request URL**

```
/action/https.crt?subject=del
```

■ **Request Body**

None

- **Response**

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

2.27 To delete the HTTPS installed certificate

- **Description**

The interface to delete the HTTPS certificate.

- **Request URL**

```
/action/https.crt?subject=del_install
```

- **Request Body**

None

- **Response**

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

2.28 To download the HTTPS certificate

- **Description**

The interface to download the HTTPS certificate.

- **Request URL**

```
/action/https.crt?subject=download_cert
```

- **Request Body**

None

- **Response**

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

2.29 To install the HTTPS certificate

- **Description**

The interface to install the HTTPS certificate.

- **Request URL**

```
/action/https.crt?subject=install
```

- **Request Body**

None

- **Response**

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

2.30 To upload the crt file of the HTTPS certificate

- **Description**

The interface to download the crt file of the HTTPS certificate.

- **Request URL**

```
/action/upload?file=https_cert
```

- **Request Body**

None

- **Response**

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

2.31 To upload the key file of the HTTPS certificate

- **Description**

The interface to download the key file of the HTTPS certificate.

- **Request URL**

```
/action/upload?file=https_key
```

- **Request Body**

None

- **Response**

```
200: Succeed to Set.
400: Error to Request.
```

```
403: No authorized.
500: Failed to Set.
```

2.32 Get and Set 4G connection parameters

■ **Description**

The interface to get and set the 4G connection parameters.

■ **Request URL**

GET:

```
/action/get?subject=mobile
```

SET:

```
/action/set?subject=mobile
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <mobile ver="2.0">
        <active>1</active>
        <dialmode>1</dialmode>
        <apn>CMNET</apn>
        <dialno>*98*1#</dialno>
        <name></name>
        <passwd></passwd>
        <auth>0</auth>
    </mobile>
</response>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
   <mobile ver="2.0">
            <active>1</active>
            <dialmode>1</dialmode>
            <apn>CMNET</apn>
            <dialno>*98*1#</dialno>
            <name></name>
            <passwd></passwd>
      </mobile>
</request>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
404: No data
500: Error in server
```

■ **Further description**

active: enable 4G connection: 0: disable, 1: enable

dialmode: the dial working mode: 0: auto 1: Manual (in auto mode, the following parameters apn, dialno, name, passwd, auth etc. are no needed.)

apn: the apn name of the dial;

dialno: the dial number

name: the dial account name

passwd: the dial account password

2.33 To get the 4G connection status

■ **Description**

The interface to get the 4G connection status.

- **Request URL**

```
/action/get?subject=mobile_status
```

- **Request Body**

None

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <mobile_status ver="2.0">
        <ip></ip>
        <gateway></gateway>
        <netmask></netmask>
        <dns></dns>
        <rssi></rssi>
        <cimi></cimi>
        <status>0</status>
    </mobile_status>
</response>.
```

- **Further description**

ip: the IP address of the 4G connection

gateway: the 4G connection gateway IP address.

netmask: the 4G connection sub mask

dns: the DNS address of the 4G connection

rssi: the operator

status: 0: no network; 1: connected to the network

2.34 Get and Set 4G Dial Schedule

- **Description**

The interface to get and set the 4G dialing schedule.

- **Request URL**

GET:

```
/action/get?subject=mobile_schedule
```

SET:

```
/action/set?subject=mobile_schedule
```

- **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
<schetask ver="2.0">
<active>1</active>
<schedule>
        <day>
            <tsection>0-0</tsection>
            <tsection>0-0</tsection>
            <tsection>0-0</tsection>
            <tsection>0-0</tsection>
            <tsection>0-0</tsection>
            <tsection>0-0</tsection>
        </day>
        <day>
            <tsection>0-0</tsection>
            <tsection>0-0</tsection>
            <tsection>0-0</tsection>
            <tsection>0-0</tsection>
            <tsection>0-0</tsection>
            <tsection>0-0</tsection>
```

```
            </day>
            ...
        </schedule>
 </schetask>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <schetask ver="2.0">
        <active>1</active>
        <schedule>
            <day>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
            </day>

            ...
            <day>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
            </day>
        </schedule>
    </schetask>
```

```
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
404: No data
500: Error in server
```

■ **Further description**

active: enable 4G connection schedule (0: disable, 1: 7*24 hours, 2: per schedule)

❑ Media part

3 Media

3.1 Get and Set Video encode parameters

■ **Description**

The interface to get and set the video encode parameters like codec type like H.264 and H.265, video resolution, video frame rate, frame control type, key frame interval parameter, bit rate, codec quality and codec profile class etc.

■ **Request URL**

GET:

```
/action/get?subject=videoenc&stream=0 [stream: Encode stream Type (0: main stream 1: sub stream)]
```

SET:

```
/action/set?subject=videoenc&stream=0
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<requset>
    <videoenc>
        <active>1</active>
        <codec>1</codec>
        <resolution>2048x1520</resolution>
        <framerate>25</framerate>
        <rc>0</rc>
        <keygop>50</keygop>
        <bitrate>6000</bitrate>
        <quality>4</quality>
        <profile>4</profile>
```

```
            <audioen>1</audioen>
        </videoenc>
</requset>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
        <videoenc>
                <active>1</active>
                <codec>0</codec>
                <resolution>2048x1520</resolution>
                <framerate>25</framerate>
                <rc>0</rc>
                <keygop>50</keygop>
                <bitrate>6000</bitrate>
                <quality>4</quality>
                <profile>4</profile>
               <audioen>1</audioen>
        </videoenc>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

active: Enable/Disable the stream: 0: Disable 1: Enable

codec: Codec type (0: h264 1:h265）

resolution: Resolution Format: width*height

framerate: Frame Rate

rc: Frame control (0: Variable stream 1:)

keygop: key frame interval

bitrate: bit rate

quality: codec quality

profile: codec profile Class (0: base 1:main 2:high)

audioen: disable / enable audio (0: disable 1: enable)

NOTE:

profile: now the main profile is supported.

3.2 Get and Set Snapshot stream codec parameters

- **Description**

The interface to get and set the snapshot stream codec parameters like the frame rate, codec quality and snapshot interval etc.

- **Request URL**

GET:

```
/action/get?subject=snap
```

SET:

```
/action/set?subject=snap
```

- **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <snap>
        <framerate>1</framerate>
        <quality>4</quality>
```

```
        <interval>5</interval>
        <stream>0</stream>
        <path>0</path>
    </snap>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <snap>
        <framerate>1</framerate>
        <quality>4</quality>
        <interval>5</interval>
        <stream>0</stream>
        <path>0</path>
    </snap>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

framerate: frame rate

quality: codec quality

interval: snapshot interval, unit: second, valid range: [1, 600]

stream: the snapshot stream type, 0: mainstream, 1: substream.

Path: the snapshot storage direction 0: local storage; 1: NFS storage

Note: MSTAR MSC313E solution 2MP camera has no “snapshot” feature, and MSTAR MSC316DM 4.0/5.0MP cameras has the snapshot, and only “sub stream” can be selected.

3.3 Get and Set audio codec parameters

■ Description

The interface to get and set the audio codec parameters like the audio codec type (G.711 a law, u law and AAC), audio sampling rate, sampling width, bit rate, audio channel number and audio input mode etc.

■ Request URL

GET:

```
/action/get?subject=audioenc
```

SET:

```
/action/set?subject=audioenc
```

■ Request Body

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
<audioenc>
        <active>1</active>
        <codec>1</codec>
        <sample>8000</sample>
        <bitwidth>16</bitwidth>
        <bitrate>16000</bitrate>
        <channel>0</channel>
        <input>0</input>
    </audioenc>
</request>
```

■ Response

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<audioenc>
    <active>1</active>
        <codec>1</codec>
        <sample>8000</sample>
        <bitwidth>16</bitwidth>
        <bitrate>16000</bitrate>
        <channel>0</channel>
        <input>0</input>
    </audioenc>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

active: enable or disable audio (transmission and record etc.) (0: disable 1: enable)

codec: codec type (0: g711u 1: g711a 2:AAC 3: mp2 4: pcm 5: g726)

sample: audio sampling rate

bitwidth: sampling width

bitrate: bit rate, unit: kps

channel: audio channel number (0: single channel 1: stereo)

input: audio input mode (0: line 1: MIC)

NOTE:

Now only the Single Channel (0) is supported.

3.4 Get Video codec capacity parameters

- **Description**

The interface to get the video codec capacity parameters like the encode type supported, H.264 encode option, video resolution, frame rate, bitrate, codec control value range, key frame interval value range, h.264 profile etc.

- **Request URL**

```
/action/get?subject=videoencability&stream=0 [stream: encode stream type (0: mainstream 1: substream)]
```

- **Request Body**

None.

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <videoencability>
        <h264>
            <option>
                <resolution>1280x960</resolution>
                <framerate>5-25</framerate>
                <bitrate>1000-6000</bitrate>
                <rc>0-1</rc>
                <quality>0-5</quality>
                <keygop>5-100</keygop>
                <profile>1-1</profile>
            </option>
            ....
        </h264>
        <h265>
            <option>
                <resolution>1280x960</resolution>
                <framerate>5-25</framerate>
                <bitrate>1000-6000</bitrate>
```

```
                <rc>0-1</rc>
                <quality>0-6</quality>
                <keygop>5-100</keygop>
                <profile>1-1</profile>
            </option>
            ...
        </h265>
    </videoencability>
</response>
```

■ **Further description**

h264: Encode Type.

h264.option: H.264 encode option, there will be multiple option if multiple resolution supported.

h264.option.resolution: h.264 video resolution, format: width*height

h264.option.framerate: h.264 video frame rate range supported, format: minimum - maximum value

h264.option.bitrate: bit rate range supported: format: minimum - maximum value

h264.option.rc: codec control value range, format: minimum - maximum value

h264.option.quality: codec quality value range, format: minimum - maximum value

h264.option.keygop: key frame interval value range supported, format: minimum - maximum value

h264.option.profile: codec class range: format: minimum - maximum value

h265: encode type supported

h265.option: refer to h.264 option's parameters

Note:

MSTAR MSC316DM solution 4.0/5.0MP cameras support 5.0MP@15fps, 4.0MP/3.0MP/1080P/720P@20FPS, the frame is fixed 20FPS for 4MP, 3MP, 1080P and 720P.

3.5 Get audio codec capacity parameters

■ **Description**

The interface to get the audio codec capacity parameters like the encode type like G.711 or AAC, G.711a option, audio sampling, biwidth, bitrate, channel and G.711u and AAC parameters etc.

- **Request URL**

```
/action/get?subject=audioencability
```

- **Request Body**

None.

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <audioencability>
        <g711a>
            <option>
                <sample>8000</sample>
                <bitwidth>16</bitwidth>
                <bitrate>16000</bitrate>
                <channel>0</channel>
                <channel>1</channel>
            </option>
            <option>
                <sample>16000</sample>
                <bitwidth>16</bitwidth>
                <bitrate>32000</bitrate>
                <channel>0</channel>
                <channel>1</channel>
            </option>
        </g711a>
        <g711u>
            <option>
                <sample>8000</sample>
                <bitwidth>16</bitwidth>
                <bitrate>16000</bitrate>
```

```
                    <bitrate>32000</bitrate>
                    <channel>0</channel>
                    <channel>1</channel>
                </option>
            </g711u>
        </audioencability>
</response>
```

- **Further description**

g711a: audio encode type supported.

g711a.option: encode option g711a supported. There will be multiple option if there is multiple sampling methods.

g711a.option.sample: sampling rate supported.

g711a.option.bitwidth: sampling width supported.

g711a.option.bitrate: bit rates supported.

g711a.option.channel: channel list supported.

g711u: audio encode type supported.

g711u.option: refer to g711a's option

AAC: audio encode type supported.

AAC.option: refer to g711a's option.

Note:

In the current version, G.711a, G.711u, AAC, MP2, PCM ang G.726 are supported.

3.6 Get and Set Motion Detection Parameters

- **Description**

The interface to get and set motion detection parameters like whether to enable motion detection or not, relative zone display resolution, the sensitivity of the motion detection and motion detection zone range etc.

- **Request URL**

GET:

```
/action/get?subject=motion
```

SET:

```
/action/set?subject=motion
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <motion>
        <active>0</active>
        <resolution>640x360</resolution>
        <sensitivity>0</sensitivity>
        <threshold>0</threshold>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
    </motion>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <motion>
        <active>0</active>
        <resolution>640x360</resolution>
        <sensitivity>0</sensitivity>
        <threshold>0</threshold>
        <rect>0,0,0,0</rect>
```

```
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
    </motion>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

active: Enable Motion Detection or not (0:Disable 1: Enable)

resolution: Relative Zone display resolution, format: width*height

sensitivity: motion detection sensitivity, the bigger value, the high sensitivity, value range:[0,9]

threshold: the threshold value of motion detection, the value range is [0,100].

rect: motion detection zone range, maximum 4 zones supported, format: x coordinate, y coordinate, width, height.

3.7 Get and Set Privacy mask parameters

■ **Description**

The interface to get and set privacy mask parameters like whether to enable privacy mask or not, privacy mask set zone display resolution and zone range etc.

■ **Request URL**

GET:

```
/action/get?subject=privacy
```

SET:

```
/action/set?subject=privacy
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <privacy>
        <active>0</active>
        <resolution>640x360</resolution>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
    </privacy>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <privacy>
        <active>0</active>
        <resolution>640x360</resolution>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
    </privacy>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
```

```
403: No authorized.
500: Failed to Set.
```

- **Further description**

active: Enable privacy mask or not (0: disable 1: enable)

resolution: relative zone display resolution, format: width*height

rect: privacy mask zone range, maximum 4 zones supported, format: x coordinate, y coordinate, width, height.

3.8 Get and Set Video Shield Parameters

- **Description**

The interface to get and set video shield parameters like whether to enable video shield or not, the sensitivity of the video shield etc.

- **Request URL**

GET:

```
/action/get?subject=tamper
```

SET:

```
/action/set?subject=tamper
```

- **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <tamper>
        <active>0</active>
        <sensitivity>0</sensitivity>
    </tamper>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <tamper>
        <active>0</active>
        <sensitivity>0</sensitivity>
    </tamper>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

active: Enable Video Shield or not (0: Disable, 1: Enable)

sensitivity: Video Shield Sensitivity, the bigger value, the higher sensitivity, valid value range: [0,2]

3.9 Get and Set Image parameters

■ **Description**

The interface to get and set image parameters like the video frequency, image parameters like saturation, sharpness, contrast, brightness, and the features related with the image like image mirroring, De-noise status, day and night work mode, WDR status, light mattering mode, electronica shutter, Auto Iris and the white balance mode parameters etc.

■ **Request URL**

GET:

```
/action/get?subject=videoimage
```

SET:

```
/action/set?subject=videoimage
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <videoimage>
        <freq>0</freq>
        <imgstyle>0</imagestyle>
        <saturation>50</saturation>
        <sharpness>50</sharpness>
        <contrast>50</contrast>
        <brightness>50</brightness>
        <mirror>0</mirror>
        <noise>50</noise>
        <ldc>0</ldc>
        <defog>0</defog>
        <smartir>0</smartir>
        <rotate>0</rotate>
        <daynight>
            <mode>0</mode>
            <tsection>0-0</tsection>
            <color>35</color>
            <grey>20</grey>
        </daynight>
        <widedynamic>
            <wdr>0</wdr>
            <backlight>0</backlight>
        </widedynamic>
        <autoexposure>
            <metter>1</metter>
            <shutter>4</shutter>
```

```
            <iris>0</iris>
        </autoexposure>
        <whitebalance>
            <mode>0</mode>
            <rgain>50</rgain>
            <ggain>50</ggain>
            <bgain>50</bgain>
        </whitebalance>
    </videoimage>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <videoimage>
        <freq>0</freq>
        <imgstyle>0</imagestyle>
        <saturation>50</saturation>
        <sharpness>50</sharpness>
        <contrast>50</contrast>
        <brightness>50</brightness>
        <mirror>0</mirror>
        <noise>50</noise>
        <ldc>0</ldc>
        <defog>0</defog>
        <smartir>0</smartir>
        <rotate>0</rotate>
        <daynight>
            <mode>0</mode>
            <tsection>0-0</tsection>
            <color>35</color>
            <grey>20</grey>
        </daynight>
```

```
        <widedynamic>
            <wdr>0</wdr>
            <backlight>0</backlight>
        </widedynamic>
        <autoexposure>
            <metter>1</metter>
            <shutter>4</shutter>
            <iris>0</iris>
        </autoexposure>
        <whitebalance>
            <mode>0</mode>
            <rgain>50</rgain>
            <ggain>50</ggain>
            <bgain>50</bgain>
        </whitebalance>
    </videoimage>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

freq: Video Frequency (0:50hz 1:60hz)

imgstyle: Display mode: 0: standard; 1: bright; 2: vivid; 3: Gentle; 4: custom

saturation: Saturation, valid value range: [0, 100]

sharpness: Sharpness, valid value range: [0, 100]

contrast: Contrast, valid value range: [0, 100]

brightness: Brightness, valid value range: [0, 100]

mirror: Mirroring type: (0: horizontal vertical 1: horizontal 2: vertical 3: NONE)

noise: Denoise, valid value: [0, 100]

ldc: Lens Distortion Correction, valid value: [0, 100]

defog: Enable or Disable Defog: 0: disable 1: enable;

smarter: Enable or Disable Smart IR feature: 0: disable 1: enable;

rotate: Image rotate, 0: close, 1: rote 90 degree;

daynight.mode: Day & Night Mode (0: auto 1:color 2: Black & White 3:Timiing 4: inter-sync)

daynight.tsection: Color Time Segment on Timing work mode, format: starting time - ending time

daynight.color: B&W switch to Color threshold on inter-sync work mode

daynight.grey: Color switch to B&W threshold on inter-sync work mode

widedynamic.wdr: Enable WDR or not (0: Disable, 1: Enable)

widedynamic.backlight: Enable Back Light Compensation or not (0: Disable, 1: Enable)

autoexposure.metter: Light mattering mode (0: Global 1: middle)

autoexposure.shutter: Shutter

0: 1/2;

1: 1/4;

2: 1/8;

3: 1/10;

4: 1/12;

5: 1/15;

6: 1/25 or 1/30 (60Hz)

7: 1/50 or 1/60 (60Hz)

8: 1/100 or 1/120 (60Hz)

9: 1/125

10: 1/200

11: 1/500

12: 1/1000

autoexposure.iris: Enable Auto Iris (0: Disable 1: Enable)

whitebalance.mode: White Balance Mode (0: Auto 1: Outdoor 2: Indoor 3: Fluorescent Lamp 4: Manual）

whitebalance.rgain: Red Gain on manual mode

whitebalance.ggain: Green Gain on manual mode

whitebalance.bgain: Blue Gain on manual mode

Note：

MSTAR MSC316DM and MSC313E solution cameras has no “smart IR” feature.

3.10 Get RTSP Service Request URL

- **Description**

The interface to get the RTSP URL address of the mainstream and substream.

- **Request URL**

```
/action/set?subject=rtspurl&stream=0 [stream: Encode Stream Type(0:mainstream 1: substream)]
```

- **Request Body**

None.

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <refer>
        <url>rtsp://192.168.1.120:554/live/main</url>
        <expire>0</expire>
    </refer>
</response>
```

3.11 Get Snapshot Request URL

- **Description**

The interface to get the snapshot request URL address.

- **Request URL**

```
/action/get?subject=snapurl
```

- **Request Body**

None.

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <refer>
        <url>http://192.168.1.120:80/action/snap?cam=0</url>
        <expire>0</expire>
    </refer>
</response>
```

3.12 Get and Set OSD Parameters

- **Description**

The interface to get and set OSD parameters like the system information, user customized information, whether to display OSD or not and the zone range etc.

- **Request URL**

GET:

```
/action/get?subject=osd
```

SET:

```
/action/set?subject=osd
```

- **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <osd>
        <system>
```

```
            <active>1</active>
            <xpos>100<xpos>
            <ypos>100<ypos>
        </system>
        <datetime>
            <active>1</active>
            <xpos>200<xpos>
            <ypos>300<ypos>
        </datetime>
        <picture>
            <active>1</active>
            <xpos>200<xpos>
            <ypos>300<ypos>
        </picture>
        <custom>
            <active>1</active>
            <xpos>200<xpos>
            <ypos>300<ypos>
            <ctext>HDIPCAM<ctext>
        </custom>
    </osd>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <osd>
    <osd ver="2.0">
        <system>
            <active>1</active>
            <xpos>100<xpos>
            <ypos>100<ypos>
        </system>
```

```
        <datetime>
            <active>1</active>
            <xpos>200<xpos>
            <ypos>300<ypos>
        </datetime>
        <picture>
            <active>1</active>
            <xpos>200<xpos>
            <ypos>300<ypos>
        </picture>
        <custom>
            <active>1</active>
            <xpos>200<xpos>
            <ypos>300<ypos>
            <ctext>HDIPCAM<ctext>
        </custom>
    </osd>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

system: System Information

datetime: Date and Time information

picture: Pictures

custom: User defined information

active: if to be display or not (0: no, 1: yes)

xpos: Axis-X displayed.

ypos: Axis_y displayed.

ctext: The string the user defined.

NOTE:

1. Please refer to the system parameter command if the format of the day and time display need to be modified.

2. If “osd version” =2.0, xpos and ypos coordinator unit will use permillage, otherwise it will adopt the related pixel unit value to mainstream resolution.

3. “picture” is added when “osd version” =2;

3.13 Get and Set ROI Parameters

■ **Description**

The interface to get and set ROI parameters like the ROI related zone resolution, zone etc.

■ **Request URL**

GET:

```
/action/get?subject=videorio
```

SET:

```
/action/Set?subject=videorio
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <videoroi>
        <active>0</active>
        <resolution>640x480</resolution>
        <rect>0,0,200,200</rect>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
```

```
        <rect>0,0,0,0</rect>
    </videoroi>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <videoroi>
        <active>0</active>
        <resolution>640x480</resolution>
        <rect>0,0,200,200</rect>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
        <rect>0,0,0,0</rect>
    </videoroi>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

Active: 0: Disable, 1: Enable

Resolution: the display resolution of the relative set zone, the format is width * height

Rect: the ROI zone, ROI supports the maximum 4 ROI zones, the format is X rect, Y rect, width, hight.

3.14 Get and Set Audio Input and Output Volume

■ **Description**

The interface is used to control the volume of the Audio volume if the camera has the audio input device like built-in microphone, or the audio input device (like microphone, audio Pick-up device) which is connected to the cameras through audio input interface, and audio output device like built-in speaker, or the audio output device (like passive or active speaker etc.) which is connected to the cameras through audio output interface.

- **Request URL**

GET:

```
/action/get?subject=audiovolume
```

SET:

```
/action/Set?subject=audiovolume
```

- **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <audiovolume>
        <input>0</input>
        <output>0</output>
    </audiovolume>
</request>
```

- **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <audiovolume>
        <input>0</input>
        <output>0</output>
    </audiovolume>
```

```
</response>
```

SET:

```
200: Succeed to Set.

400: Error to Request.

403: No authorized.

500: Failed to Set.
```

- **Further description**

Input: the volume of the input audio, the value range is [0, 100]

Output: the volume of the output audio, the value range is [0,100]

3.15 Get and Set Image Parameters Templates

- **Description**

The interface is used to set the camera's image parameters by the templated method. It means the user can use the “Day” parameters for the working “Day” mode, and can use the “Night” parameters for the working “Night” mode.

- **Request URL**

GET:

```
/action/get?subject=cameraimage
```

SET:

```
/action/Set?subject=cameraimage
```

- **Request Body**

GET: None

SET:

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

<request>

  <cameraimage ver="2.0">
```

```
<freq>0</freq>
<rotate>0</rotate>
<mirror>0</mirror>
<mode>0</mode>
<daysect>21600-64800</daysect>
<imagescene>
  <imgstyle>0</imgstyle>
  <saturation>50</saturation>
  <sharpness>50</sharpness>
  <contrast>50</contrast>
  <brightness>50</brightness>
  <noise>50</noise>
  <ldc>0</ldc>
  <defog>0</defog>
  <smartir>0</smartir>
  <dnr2d>1</dnr2d>
  <dnr3d>1</dnr3d>
  <widedynamic>
    <wdr>0</wdr>
    <backlight>0</backlight>
  </widedynamic>
  <autoexposure>
    <mode>0</mode>
    <metter>0</metter>
    <shutter>4</shutter>
    <iris>0</iris>
  </autoexposure>
  <whitebalance>
    <mode>0</mode>
    <rgain>50</rgain>
    <ggain>50</ggain>
    <bgain>50</bgain>
  </whitebalance>
</imagescene>
```

```
<imagescene>
  <imgstyle>0</imgstyle>
  <saturation>50</saturation>
  <sharpness>50</sharpness>
  <contrast>50</contrast>
  <brightness>50</brightness>
  <noise>50</noise>
  <ldc>0</ldc>
  <defog>0</defog>
  <smartir>0</smartir>
  <dnr2d>1</dnr2d>
  <dnr3d>1</dnr3d>
  <widedynamic>
    <wdr>0</wdr>
    <backlight>0</backlight>
  </widedynamic>
  <autoexposure>
    <mode>0</mode>
    <metter>0</metter>
    <shutter>4</shutter>
    <iris>0</iris>
  </autoexposure>
  <whitebalance>
    <mode>0</mode>
    <rgain>50</rgain>
    <ggain>50</ggain>
    <bgain>50</bgain>
  </whitebalance>
</imagescene>
<imagescene>
  <imgstyle>0</imgstyle>
  <saturation>0</saturation>
  <sharpness>0</sharpness>
  <contrast>0</contrast>
```

```
        <brightness>0</brightness>
        <noise>50</noise>
        <ldc>0</ldc>
        <defog>0</defog>
        <smartir>-1091487120</smartir>
        <dnr2d>1</dnr2d>
        <dnr3d>1</dnr3d>
        <widedynamic>
          <wdr>0</wdr>
          <backlight>0</backlight>
        </widedynamic>
        <autoexposure>
          <mode>0</mode>
          <metter>0</metter>
          <shutter>0</shutter>
          <iris>0</iris>
        </autoexposure>
        <whitebalance>
          <mode>0</mode>
          <rgain>0</rgain>
          <ggain>0</ggain>
          <bgain>0</bgain>
        </whitebalance>
      </imagescene>
      <daynight>
        <mode>0</mode>
        <irled>0</irled>
        <daysect>21600-64800</daysect>
        <color>10</color>
        <grey>5</grey>
      </daynight>
    </cameraimage>
</response>
</response>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <cameraimage ver="2.0">
    <freq>0</freq>
    <rotate>0</rotate>
    <mirror>0</mirror>
    <mode>0</mode>
    <daysect>21600-64800</daysect>
    <imagescene>
      <imgstyle>0</imgstyle>
      <saturation>50</saturation>
      <sharpness>50</sharpness>
      <contrast>50</contrast>
      <brightness>50</brightness>
      <noise>50</noise>
      <ldc>0</ldc>
      <defog>0</defog>
      <smartir>0</smartir>
      <dnr2d>1</dnr2d>
      <dnr3d>1</dnr3d>
      <widedynamic>
        <wdr>0</wdr>
        <backlight>0</backlight>
      </widedynamic>
      <autoexposure>
        <mode>0</mode>
        <metter>0</metter>
        <shutter>4</shutter>
        <iris>0</iris>
      </autoexposure>
      <whitebalance>
```

```
        <mode>0</mode>
        <rgain>50</rgain>
        <ggain>50</ggain>
        <bgain>50</bgain>
    </whitebalance>
</imagescene>
<imagescene>
    <imgstyle>0</imgstyle>
    <saturation>50</saturation>
    <sharpness>50</sharpness>
    <contrast>50</contrast>
    <brightness>50</brightness>
    <noise>50</noise>
    <ldc>0</ldc>
    <defog>0</defog>
    <smartir>0</smartir>
    <dnr2d>1</dnr2d>
    <dnr3d>1</dnr3d>
    <widedynamic>
        <wdr>0</wdr>
        <backlight>0</backlight>
    </widedynamic>
    <autoexposure>
        <mode>0</mode>
        <metter>0</metter>
        <shutter>4</shutter>
        <iris>0</iris>
    </autoexposure>
    <whitebalance>
        <mode>0</mode>
        <rgain>50</rgain>
        <ggain>50</ggain>
        <bgain>50</bgain>
    </whitebalance>
```

```
</imagescene>
<imagescene>
   <imgstyle>0</imgstyle>
   <saturation>0</saturation>
   <sharpness>0</sharpness>
   <contrast>0</contrast>
   <brightness>0</brightness>
   <noise>50</noise>
   <ldc>0</ldc>
   <defog>0</defog>
   <smartir>-1091487120</smartir>
   <dnr2d>1</dnr2d>
   <dnr3d>1</dnr3d>
   <widedynamic>
      <wdr>0</wdr>
      <backlight>0</backlight>
   </widedynamic>
   <autoexposure>
      <mode>0</mode>
      <metter>0</metter>
      <shutter>0</shutter>
      <iris>0</iris>
   </autoexposure>
   <whitebalance>
      <mode>0</mode>
      <rgain>0</rgain>
      <ggain>0</ggain>
      <bgain>0</bgain>
   </whitebalance>
</imagescene>
<daynight>
   <mode>0</mode>
   <irled>0</irled>
   <daysect>21600-64800</daysect>
```

```
        <color>10</color>
        <grey>5</grey>
      </daynight>
    </cameraimage>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

freq: the system frequency value (0: 50Hz, 1: 60Hz)

rotate: the image rotation value. (0: Close rotation; 1: 90 degree)

mirror: the type of mirroring. (0: Horizontal + Vertical, 1: Horizontal, 2: Vertical,3: No Mirroring.)

mode: Image application scene mode (0: Normal mode, 1: Day mode, 2: Night mode, 3. Timing mode, 4. Auto mode)

daysect: The day time segment setting, its format is starting time-ending time, the unit is second, it's valid only when mode is “timing mode”.

Imagescene.imgstyle: the image style -0: standard, 1: Bright, 2: Vivid, 3: Gentle, 4: Custom.

Imagescene.saturation: the image saturation, the valid value range [0, 100], it is valid when imgstyle is 4.

Imagescene.sharpness: the image sharpness, the valid value range [0,100], it is valid when imgstyle is 4.

Imagescene.contrast: the image contrast, the valid value range [0, 100], it is valid when imgstyle is 4.

Imagescene.brightness: the image brightness, the valid value range [0,100], it is valid when imgstyle is 4.

Imagescene.noise: the value of the noise reduction, the valid value range [0, 100].

Imagescene.ldc: the value of the LDC (Lens Distortion Correction), the valid value range [0, 100].

Imagescene.dnr3d: Enable or Disable 3D noise reduction: 0: Disable; 1: Enable;

Imagescene.dnr2d: Enable or Disable 2D noise reduction: 0: Disable; 1: Enable;

Imagescene.defog: Enable or Disable Defog, 0: Disable; 1: Enable;

Imagescene.smartir: Enable or Disable Smart IR, 0: Disable; 1: Enable;

Imagescene.widedynamic.wdr: Enable or Disable WDR, 0: Disable; 1: Enable;

Imagescene.widedynamic.backlight: Enable or Disable back light compensation, 0: Disable; 1: Enable;

Imagescene.autoexposure.metter: Light metering mode, 0: Global; 1:Middle;

Imagescene.autoexposure.shutter: Shutter value;

0: 1/2;

1: 1/4;

2: 1/8;

3: 1/10;

4: 1/12;

5: 1/15;

6: 1/25 or 1/30 (60Hz)

7: 1/50 or 1/60 (60Hz)

8: 1/100 or 1/120 (60Hz)

9: 1/125

10: 1/200

11: 1/500

12: 1/1000

Imagescene.autoexposure.iris: Enable or Disable Auto IRIS, 0: Disable; 1: Enable;

Imagescene.whitebalance.mode: The mode of White Balance, 0: Auto; 1: Outdoor; 2:Indoor; 3: light lamp; 4: Manual;

Imagescene.whitebalance.rgain: Red Gain value when the white balance is “Manual”;

Imagescene.whitebalance.ggain: Green Gain value when the white balance is “Manual”;

Imagescene.whitebalance.bgain: Blue Gain value when the white balance is “Manual”;

Daynight.mode: The mode of day and night, 0: Auto; 1: Color; 2: Black and White; 3: Timing; 4: Internal sync;

Daynight.tsection: the time segment of the color mode when daynight.mode is 3. Its format is starting time – ending time.

Daynight.color: the threshold value of the light when the WB mode switched to Color mode during “daynight.mode is 4”;

Daynight.grey: the threshold value of the light when the color mode switch to BW mode during “daynight mode is 4”

Daynight.irled: the IR LED mode. It works different meaning as the model's LED capability (Please check the API 1.2). when irled=1 (IR model or FTC model), 0: Auto, 1: Open, 2: close; when irled=2 (Dual LED models) 0: smart switching by alarm or human detection. 1: IR mode; 2: White LED mode; 3 Close;

Daynight.ledlevel: LED's maximum brightness, range is [0,100], it's valid for Full Time Color models.

Daynight.smartevt: Smart LED's triggered event; 0: close; 1: open. It is valid for Full Time Color models.

Daynight.evtblink: Smart LED blinking. 0: close; 1: open; It's valid for Full Time Color models.

Note:

There are total 4 kinds of imagescenes, as sequence, they are Normal mode, Day mode, Night mode, and Timing mode.

3.16 Set and Get the IR-Cut working Parameters

- **Description**

The interface is to get and set the IR-CUT of the camera' working parameters. The client can use the interface to forcedly switch the IR-CUT to work on the day mode when the light condition is dark or low light, and the image is color.

- **Request URL**

GET:

```
/ action/get?subject=ircut
```

SET:

```
/action/set?subject=ircut
```

- **Request Body**

GET:

None.

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <ircut>
    <col2greydelay>2</col2greydelay>
    <grey2coldelay>5</grey2coldelay>
    <autotime>18000-65220</autotime>
  </ircut>
</request>
```

- **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<ircut>
      <col2greydelay>2</col2greydelay>
      <grey2coldelay>5</grey2coldelay>
      <autotime>18000-65220</autotime>
  </ircut>
</response>
```

SET:

```
200: Successful
400:   Error to request
403: without permission
500：Failed
```

■ **Further description**

col2greydelay: delay time of the color switching to B/W. it's fixed as 2.

grey2coldelay: delay time of the B/W switching to color, it's fixed as 5.

Autotimer: it means the working period of the AUTO mode of the IR-CUT. The unit is second, if the value is 0-86400, then during the full day, the IR-CUT is working on AUTO mode. If it's not in the period, the IR-CUT is working fixed day mode, it won't switch according to the light automatically.

3.17 Set and Get the Video Working Mode

■ **Description**

The interface is to get and set the video working mode which includes if the camera working on HDR mode, or high FPS mode.

■ **Request URL**

GET:

```
/action/get?subject=videowork
```

SET:

```
/action/set?subject=videowork
```

■ Request Body

GET:

None.

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
   <videowork>
      <hdr>0</hdr>
      <highfps>0</highfps>
   </videowork>
</request>
```

■ Response

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
   <videowork>
      <hdr>0</hdr>
      <highfps>0</highfps>
   </videowork>
</response>
```

SET:

```
200: Successful
400:   Error to request
403: without permission
500：Failed
```

■ **Further description**

hdr: enable or disable the camera is working on HDR mode, 0: disable, 1: enable.

highfps: enable or disable the camera is working on high FPS mode: 0: disable, 1: enable.

Note:

1. HDR mode cannot work with the HIGHFPS mode at the same time. only choose one from two.

2. HDR mode and HIGHFPS mode are not supported by all cameras, it can be decided by the API: 1.2 getting the device's capacity information.

3.18 Get the snapshot capability

■ **Description**

The interface is to get the camera's snapshot capability.

■ **Request URL**

GET:

```
/action/get?subject=snapability
```

■ **Request Body**

GET:

None.

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <snapability>
    <streamopt>3</streamopt>
    <stream0>1-5</stream0>
    <stream1>1-5</stream1>
  </snapability>
```

```
</response>
```

■ **Further description**

streamopt: the options of the snapshot stream, each bit represents each stream, 1 mean it's active. Bit0: mainstream, bit1: sub-stream.

stream0: the FPS range of the main stream snapshot, it's [1,5]

stream1: the FPS range of the sub stream snapshot, it's [1,5]

❑ Alarm & Event part

4 Alarm

4.1 Get and Set Alarm Parameters

■ **Description**

The interface to get and set the alarm parameters like alarm schedule mode, alarm input delay time, alarm output work way, and scheduled alarm information etc.

■ **Request URL**

GET:

```
/action/get?subject=alarm&type=0 [type: Alarm Type (0:IO alarm 1: Device Startup 2: Motion Detection 3: Video Shield 4: PIR 5: Disconnection 10: Line crossing 11: Intrusion 12: Human detection or Person detection 13: Face detection 14: Object left and remove 15: loitering 16. LPR 17. FR – failed recognition, 18: FR-White list, 19-FR-Blacklist, 20-Wearing mask, 21-Not Wearing mask)]
```

SET:

```
/action/set?subject=alarm&type=0
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <alarmevt>
        <active>0</active>
        <duration>10</duration>
        <outmask>256</outmask>
        <schedule>
            <day>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
```

```
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
            </day>
            <day>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
            </day>
            ...
        </schedule>
    </alarmevt>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <alarmevt>
        <active>0</active>
        <duration>10</duration>
        <outmask>256</outmask>
        <schedule>
            <day>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
            </day>
            <day>
```

```
                    <tsection>0-0</tsection>
                    <tsection>0-0</tsection>
                    <tsection>0-0</tsection>
                    <tsection>0-0</tsection>
                    <tsection>0-0</tsection>
                    <tsection>0-0</tsection>
                </day>
                ...
            </schedule>
        </alarmevt>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

active: Alarm Schedule mode (0: Disable 1:7*24 Hours 2: Scheduled)

duration: Alarm Input Delay Time

outmask: Alarm Processing mask code, every bit shows every alarm output type. The value is *decimal. The value is calculated through binary.*

bit0-bit7: IO Alarm output

bit8-bit11: LED alarm output

bit12: snapshot

bit13: recording

bit14: FTP upload

bit15: PTZ

bit16: Sending mail

bit17: audio alarm

bit18: push alarm message to the server

For example, you want to set the camera to make the snapshot, and then to upload the snapshot to the FTP server:

Bit0 **Bit16 - Bit17**

0 – 0 – 0 – 0 – 0 – 0 – 0 – 0 – 0 – 0 – 0 – 0 – 1 – 0 – 1 – 0 – 0 – 0 - 0

The hexadecimal value is 0x5000, so the decimal value is 20480. Now the outmask value is 20480.

schedule: alarm scheduled information, schedule is on week, 6-time segments per day, unit: second.

tsection: format, starting time - ending time, as example 0 clock to 1 clock is 0-3600.

Note:

bit12: snapshot：MSTAR MSC313E solution 2MP cameras do not support this action, and MSTAR MSC316DM 4.0MP/5.0MP cameras support the action, and the snapshot is using sub stream for snapshot.

4.2 Subscribe Alarm Information

- **Description**

The interface to subscribe the alarm.

- **Request URL**

```
/action/alarm?subject=subscript
```

- **Request Body**

None.

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <refer>
        <url>http://192.168.1.120:80/alarm?subject=query&=1</url>
        <expire>10</expire>
    </refer>
</response>
```

- **Further description**

url: URL of subscribe alarm request

expire: Valid period, unit: second

Alarm Subscribing procedure:

1. Send the request of alarm subscribing to the device.

2. it's needed to send alarm lookup to the device continuously within valid time after the URL of the alarm lookup got, thus the alarm real-time can be saved.

3. Send the request to delete the alarm subscribe to the device when exit.

4.3 Get Alarm Information

- **Description**

The interface to get alarm information like the alarm type, alarm source, the alarm happened time and the alarm status etc.

- **Request URL**

```
/action/alarm?subject=query&id=0
```

- **Request Body**

None.

- **Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
        <alarmmsg>
                <topic>MOTION</topic>
                        <source>0</source>
                        <status>1</status>
                        <datetime>2000-01-01T08:06:27</datetime>
                        <mdstatus>
                                <row>18</row>
                                <col>22</col>
                                <move>0</move>
                                <cell>0,0,0,0,0,0,0,0,0,0,0,0,0</cell>
```

```
                    </mdstatus>
        </alarmmsg>
        <alarmmsg>
                <topic>CROSSLINE</topic>
                <source>0</source>
                <status>0</status>
                <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
        <alarmmsg>
                <topic>INSTRUSION</topic>
                <source>0</source>
                <status>0</status>
                <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
        <alarmmsg>
                <topic>HUMANDECTION</topic>
                <source>0</source>
                <status>0</status>
                <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
        <alarmmsg>
                <topic>FACEDECTION</topic>
                <source>0</source>
                <status>0</status>
                <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
        <alarmmsg>
                <topic>OBJECTLEFTLOST</topic>
                <source>0</source>
                <status>0</status>
                <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
        <alarmmsg>
                <topic>LOITERING</topic>
```

```
            <source>0</source>
            <status>0</status>
            <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
        <alarmmsg>
            <topic>LPR</topic>
            <source>0</source>
            <status>0</status>
            <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
        <alarmmsg>
            <topic>FACERECO_FAILURE</topic>
            <source>0</source>
            <status>0</status>
            <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
        <alarmmsg>
            <topic>FACERECO_FAILURE</topic>
            <source>0</source>
            <status>0</status>
            <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
        <alarmmsg>
            <topic>FACERECO_BNAME</topic>
            <source>0</source>
            <status>0</status>
            <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
        <alarmmsg>
            <topic>FACERECO_WNAME</topic>
            <source>0</source>
            <status>0</status>
            <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
```

```
        <alarmmsg>
                <topic>FACEDETECTION_MASK</topic>
                <source>0</source>
                <status>0</status>
                <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
        <alarmmsg>
                <topic>FACEDETECTION_NOMASK</topic>
                <source>0</source>
                <status>0</status>
                <datetime>2000-01-01T08:06:27</datetime>
        </alarmmsg>
</response>
```

■ **Further description**

topic: alarm information topic: IO/MOTION/CROSSLINE/INTRUSION/HUMANDETECTION/FACEDETECTION/OBJECTLEFTLOST/LOITERING/LPR/FACERECO_FAILURE/FACERECO_BNAME/FACERECO_WNAME/FACEDECTION_MASK/FACEDECTION_NOMASK

source: alarm source

datetime: alarm time, format: YYYY-MM-DD hh:mm:ss

status: alarm status (0: alarm end 1: alarm happen)

data: alarm data. It's valid when the "status" = 1; please check the defined data format which is described in the 4.8 Alarm data format definition.

NOTE:

1. Request URL is that be returned when it is to subscribe alarm from the device.
2. It's need to ask the request on overtime period, otherwise the device will release automatically the alarm subscribed.
3. The request will postpone automatically the valid time of the alarm subscribed.

4.4 Delete the alarm subscription

■ **Description**

The interface to destroy the alarm subscription.

- **Request URL**

```
/action/alarm?subject=destroy&id=0
```

- **Request Body**

None.

- **Response**

```
200: Successful
400:  Error to request
500：Failed
```

- **Further description**

NOTE:

Request URL is that be returned when it is to subscribe alarm from the device.

4.5 Alarm notification

- **Description**

The interface is used to ask the device to report the alarm data.

- **Request URL**

```
/action/alarm?subject=notify
```

- **Request Body**

None.

- **Response**

```
200: Successful
400:  Error to request
500：Failed
```

- **Further description**

NOTE:

The work process of the device's alarm report is like the below:

1) Send the alarm report request to the device using this interface;

2) Waiting for the reported alarm data from the device after receiving the 200 response, now the connection should not be cut off. (Please refer to 4.8 alarm data definition)

3) The device send the alarm data to the client when there is the alarm event detected.

4) The client disconnect the device when quit, and release the resource.

4.6 Get and Set I/O alarm parameters

- **Description**

The interfaces are used to get and set the I/O alarm parameters if the device supports alarm input and output interfaces.

- **Request URL**

GET:

```
/action/get?subject=alarmio
```

SET:

```
/action/set?subject=alarmio
```

- **Request Body**

GET: None.

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <alarmio>
        <input>1</input>
        <output>1</output>
        <outcur>1</outcur>
```

```
        <dnswitch>1</dnswitch>
        <outmode>1</outmode>
        <outduty>
            <section>0-86400-1</section>
            <section>0-86400-1</section>
            <section>0-86400-1</section>
            <section>0-86400-1</section>
        </outduty>
    </alarmio>
</request>
```

■ Response

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <alarmio>
        <input>1</input>
        <output>1</output>
        <outcur>1</outcur>

        <dnswitch>1</dnswitch>
        <outmode>1</outmode>
        <outduty>
            <section>0-86400-1</section>
            <section>0-86400-1</section>
            <section>0-86400-1</section>
            <section>0-86400-1</section>
        </outduty>
    </alarmio>
</response>
```

SET:

```
200: Successful
```

```
400:  Error to request
500：Failed
```

■ **Further description**

input: IO input level, each bit represents an IO, supports up to 32

output: IO output level, each bit represents an IO, supports up to 32

outcur: IO current output level, each bit represents one IO, supports up to 32

dnswitch: IO input linkage day and night switch, 0: off, 1: on

outmode: IO output mode: 0: automatic 1: high level 2: low level 3: timing

outduty: timing mode, the schedule parameters, its format: start time-end time-output level. The start/end time is the number of seconds in 1 day, the range is [0,86400], the end time must be greater than the start time.The effective value of output level is 1: high level, 2: low level

Note:

1. bit0~bit31, 1: high level 0: low level

2. Currently only supports 1 I/O at most

3. Models that support mask detection, all face detection alarm types, shared face detection alarm types.

4.7 Trigger the Alarm Event Manually

■ **Description**

The user can use the interface to trigger the alarm event.

■ **Request URL**

```
/action/alarm?subject=trigger&event=manual
```

■ **Request Body**

```
Event: Event Name (Manual: Manual Event)
```

■ **Response**

```
200: Successful
400:  Error to request
500：Failed
503: Unused service
```

■ **Further description**

Note:

Current only “Manual” event can be supported.

4.8 Get and Set Alarm Sever parameters

■ **Description**

The interfaces are used to set and get the alarm event server's parameters including server name, server address and port etc.

■ **Request URL**

GET:

```
/action/get?subject=evtserver
```

SET:

```
/action/set?subject=evtserver
```

■ **Request Body**

GET: None.

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <evtserver>
        <eserver>
            <name>event1</name>
```

```
            <host>server</host>
            <port>1</port>
        <eserver>
        <eserver>
            <name>event2</name>
            <host>server</host>
            <port>2</port>
        <eserver>
        ...
    </evtserver>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <evtserver>
        <eserver>
            <name>event1</name>
            <host>server</host>
            <port>1</port>
        <eserver>
        <eserver>
            <name>event2</name>
            <host>server</host>
            <port>2</port>
        <eserver>
        ...
    </evtserver>
</response>
```

SET:

```
200: Successful
```

400: Error to request

500：Failed

- **Further description**

eserver.name: the name of alarm event server (it should be unique name.)

eserver.host: the address of the alarm event server.

Eserver.port: the port of the alarm event server.

Note:

1. Max. 4 servers can be supported.
2. The sequence of alarm server to receive the alarm pushing is the same as the servers list.

4.9 Get the status of motion detection

- **Description**

The interface is used to get the status of motion detection, the details data status can be refer to the alarm data definition in the next item.

- **Request URL**

```
/action/get?subject=mdstatus
```

- **Request Body**


```
<?xml version="1.0" encoding="utf-8"?>
<response>
 <mdstatus>
    <row>36</row>
    <col>44</col>
    <move>2720</move>
<cell>0,0,0,0,0,0,0,0,0,0,0,0,0,f0000000,7c00001f,c00001ff,1ff7,1ff7c,1ff7c0,1ff7c00,7e00000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0</cell>
</mdstatus>
</response>
```

■ **Response**

```
200: Successful
400:  Error to request
500：Failed
```

■ **Further description**

The data status definition can be referred to the 4.10 alarm data definition.

4.10 Alarm Data Definition

■ **Description**

The alarm data including motion detection, I/O alarm and PIR etc. is defined here. It defined the alarm data format. [Now only motion detection is defined.]

■ **Motion Detection**

■ **Date Format Defition**

```
<mdstatus>
    <row>22</row>
    <col>22</col>
    <move>122</move>
    <cell>0,0,0,0,0,0,0,0,0,0,0,0,f0000000,7c00001f,c00001ff,1ff7,1ff7c,1ff7c0,1ff7c00,
    7e00000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0</cell>
</mdstatus>
```

■ **Further description**

row: the row of the full motion detection zone; for example, it's 18 if the sub stream resolution is CIF. And it's 36 if the sub stream resolution is D1.

col: the column of the full motion detection zone; it's 22 if the sub stream resolution is CIF. And it's 44 if the sub stream resolution is D1.

Move: the maximum movement value in the motion detection zone.

cell: the status of the motion detection cells; (Every 4 bytes group, separated by commas, less than 4 bytes to complete 4 bytes (binary data, each bit unit represents the status of a cell, 0: no alarm, 1: alarm)

NOTE:

The value of cell is the string which is converted from binary to hexadecimal. 1 byte is 2 hexadecimals string. For example, if the binary of the cell is 0x12345, 0xabcdef, here it should be <cell>12345,abcdef</cell>.

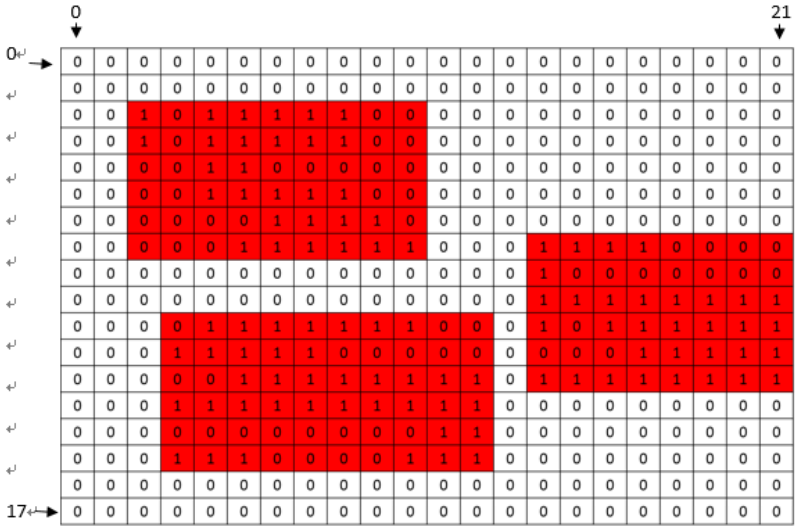
The relationship between the total cells with the configured motion detection zone.

The below (22*18 zones) shows the relationship:

The red cells are the configured motion detection zones.

The cell value (value "1" means movement, "0" - no movement) is only applied for the configured motion detection zone (RED ZONE), if there is movement, the cell value is 1, if there is no movement, the cell value is 0. At the same time, the cells which are not in the configured motion zone will not be applied even there is the motion, the cell value will be kept 0.

For example, the cell value in the row No.2 is: 00101111 10000000 00000000



❑ Maintenance part

5 Maintenance

5.1 Device Maintenance

- **Description**

The interface to restart or recovery the device to the factory configuration.

- **Request URL**

```
/action/set?subject=maintain
```

- **Request Body**

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <maintain>
        <type>1</type>
     </maintain>
</request>>
```

- **Response**

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

- **Further description**

type: Maintenance Type (0: reboot the device 1: restore the device settings to the factory 2: restore the parameter except the TCP/IP parameters)

5.2 Get Storage Device Information

- **Description**

The interface to get the storage device information like the device status, device name, the total capacity of the storage and the rest capacity.

■ **Request URL**

```
/action/get?subject=diskinfo&id=0 [id: index of storage device, 0: the first device, 1: NFS]
```

■ **Request Body**

None

■ **Response**

```
<?xml version="1.0" encoding="utf-8" ?>
- <response>
- <diskinfo>
<status>3</status>
<devname>/dev/disk0</devname>
<path>/mnt/mmc/</path>
<size>1024</size>
<free>1024</free>
</diskinfo>
</response>
```

■ **Further description**

status: Status Value, it's defined as the below:

bit0: existed or not? 0: NOT, 1: Existed

bit1: if the device is mounted or not? 0: NOT mounted, 1: mounted

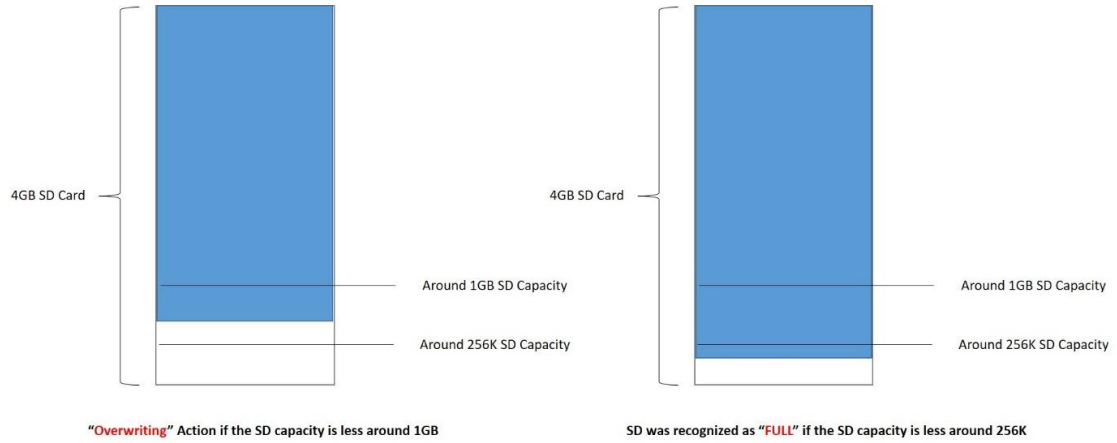
bit2: if the device is full? 0: Not Full, 1: Full

bit3: if there is the error to write or read?) 0: No Error, 1: Error

Example: 19: 0001 0011: On overwriting, the Disk is not Full, there is SD card installed.

23: 0001 0111: On overwriting, the Disk is Full, There is SD card installed.

The definition of “Start Overwriting” and “Disk Full” can be refer to the below JPG showed.



devname: the storage device name

path: the installation path of the storage device.

size: the total capacity of the storage device, unit: byte.

free: the rest capacity of the storage device, unit: byte.

5.3 Format the storage device

- **Description**

The interface to format the storage device.

- **Request URL**

```
/action/set?subject=diskfmt
```

- **Request Body**

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <diskfmt>
        <disk>0</disk>
        <fmt>0</fmt>
    </diskfmt>
</diskinfo>
```

- **Response**

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

- **Further description**

disk: index of the device, 0: the first device.

fmt: format type(0:fat32 1:ext2) , fat32 is supported only now.

NOTE:

Formatting operation will cost a little longer time (the more capacity, the longer time to wait).

5.4 Firmware upgrade

- **Description**

The interface to upgrade the firmware.

- **Request URL**

```
/action/upload?file=firmware
```

- **Request Body**

The firmware file.

- **Response**

```
200: Succeed to Set.
400: Error to Request.
415: Invalid firmware file.
416: Mismatched hardware.
```

- **Further description**

The firmware upload adopts the standard HTTP file uploading technology, for more details, please refer to HTTP file transmission protocols.

The device will be coming into file updating process after the firmware file uploaded. It will take a little long time for the camera to update the firmware, so it's suggested that the client set the minimum 200 second of the time out.

5.5 Firmware upgrade preparation

- **Description**

The interface to notify the camera that there is FW update. When the camera received the request, the camera will execute some cleaning works and stop some tasks inside which can make FW update safer.

- **Request URL**

```
/action/upgrade?op=prepare
```

- **Request Body**

None

- **Response**

```
200: Succeed to Set.
400: Error to Request.
```

5.6 Finding the device's password

- **Description**

The interface is to help the client to find the password. It will back the serial number when using the interface. And the client will send back the serial number to the technical support, the engineer can send out the password to the client using the special tool with this serial number.

- **Request URL**

```
/action/get?subject=findpwd
```

- **Request Body**

No.

- **Response**

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <findpasswd>
<verification>ABCDEFJHIJKLMNOPQRSTUVWXYZ123456</verification>
 </findpasswd>
</response>
```

.

- **Further description**

Verification: 32 bit verification code.

5.7 Output and Input the configurations file

- **Description**

The client can use this interface to copy the related parameters from one specified camera, and output to one bin file which will be copied to some cameras in batch.

- **Request URL**

Output:

```
/action/export
```

Input:

```
/action/input
```

- **Request Body**

Output: None.

Input: specified BIN file

- **Response**

Output: specified BIN file

Input:

```
200: Successful
400:  Error to request
500：Failed
```

■ **Further description**

1. Export file is using the standard HTTP file download method;
2. Import file is using the standard HTTP file upload method;
3. The device will be reboot after import the file successfully.

5.8 Set and Get the Device Auto Reboot parameters

■ **Description**

The interface is to get and set the device auto reboot related parameters.

■ **Request URL**

GET:

```
/action/get?subject=autoreboot
```

SET:

```
/action/set?subject=autoreboot
```

■ **Request Body**

GET:

None.

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <autoreboot>
        <mode>1</mode>
        <time>25</time>
    </autoreboot>
```

```
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <autoreboot>
        <mode>1</mode>
        <time>25</time>
    </autoreboot>
</response>
```

SET:

```
200: Successful
400:  Error to request
403: without permission
500：Failed
```

■ **Further description**

Mode: the work mode of auto reboot, 0: Disable, 1: Everyday, 2 – 8 : Monday – Sunday;

Time: the time of auto reboot, the unit is second;

❑ Stream part

6 Stream

6.1 Request Audio or Video stream

- **Description**

The interface to request the audio and video stream.

- **Request URL**

```
/action/stream?subject=liveplay&stream=0 [stream: stream type (0: main stream 1: substream)]
```

- **Request Body**

None

- **Response**

```
200: Succeed to Request
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

- **Further description**

The Procedure of requesting for audio or video stream:

1. Client requests audio and video from device;

2. After receiving the request, the device sends 200 to the client and sends data to the client in about 1 second (the first frame is a metadata frame. The client runs the corresponding decoder to decode based on this frame, and the subsequent data is normal audio and video frame data)

3. After receiving the response, the client waits to start receiving audio and video data.

4. The client disconnects and the device releases resources.

NOTE:

1.The Connection field in the HTTP header must be Keep-Alive when requesting.

2. Each audio and video data consist of two parts: frame header + audio and video raw data, as shown in the figure below:

Frame header (24 Bytes)						Raw data
32bit	8bit	24bit	32bit	32bit	64bit	N bits
magic	type	resv	length	seqno	timestamp	rawdata

megic: magic number, it's fixed as 0x20150723;

Type: frame type

0: P frame; 1: I frame, 2: B frame; 3: A frame (audio frame) 0xa: metadata

AV_Meta_data (20 Bytes)													
Meta		Video data (8 Bytes)						Audio data (8 Bytes)					
4bit	28bit	16bit	16bit	4bit	6bit	2bit	20bit	16bit	16bit	4bit	6bit	2bit	20bit
meta	resv1	Video.width	Video.height	Video.codec	Video.fps	Video.profile	resv2	Audio.sample	Audio.bitrate	Audio.codec	Audio.bitwidth	Audio.channel	resv3

Resv: reserved

Length: Video or media data frame raw data length;

Seqno: Current frame sequence number, each session starts from 0;

Timestamp: Timestamp, invalid when type is media data frame (type = 0xa);

Rawdata: Audio and video or media data raw data.

The definition of metadata is like the below:

meta: Metadata mask, each bit represents whether the corresponding media data is valid (bit0: video bit1: audio)

resv1, resv2, resv3: reserved data bit

video.width: the width of the video transmitted

video.height: the height of the video transmitted

video.codec: video codec type（0：h264 1:h265）

video.fps: Frame rate of the video transmitted

video.profile: codec class of the video transmitted (0: base 1:main 2:high)

audio.sample: audio sampling rate of the audio transmitted.

audio.bitrate: bitrate of the audio transmitted. (not used)

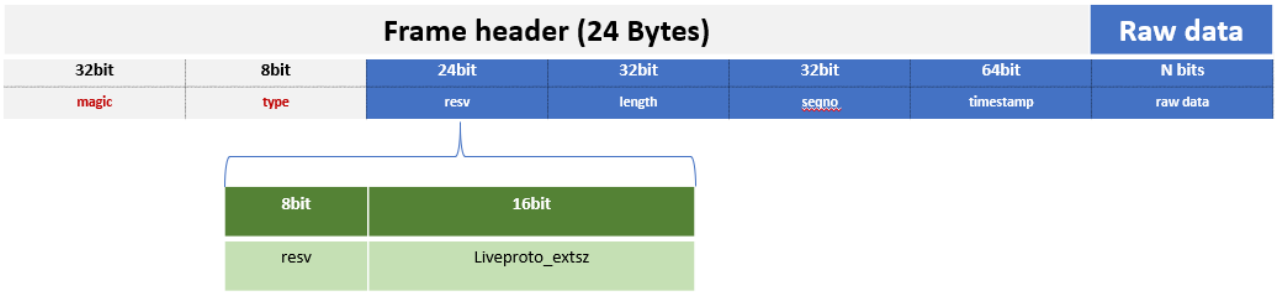
audio.codec: audio codec type of the audio transmitted（0：g711u 1:g711a）

audio.bitwidth: audio sampling bit width of the audio transmitted.

audio.channel: audio channel number of the audio transmitted（0: mono channel， 1:stereo）

Additional frame header definition when "liveproto=1" in the 1.2 API: /action/get?subject=devability

A

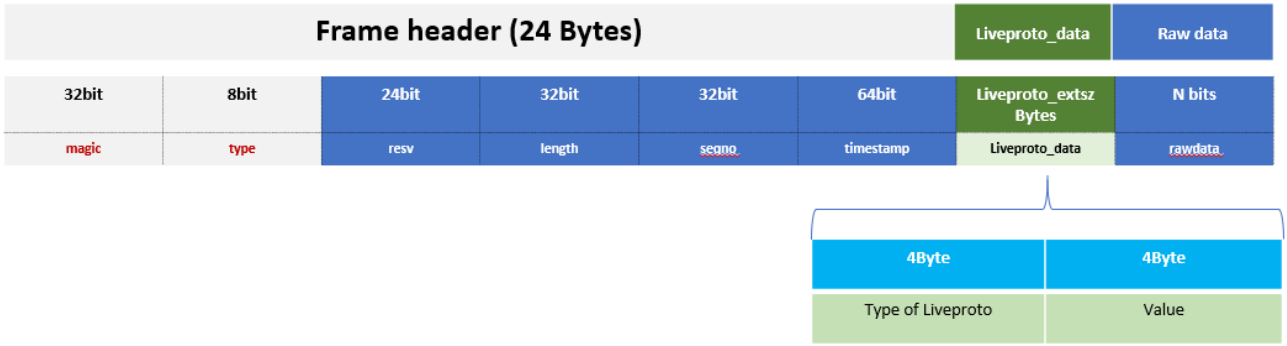


Frame header is valid only when “Liveproto = 1

" in the CGI: /action/get?subject=devability, and “Liveproto_extsz” is used to show how many bytes will be added as extended Liveproto data before “Raw data”;

Liveproto_extsz should be >=0; the length of “Liveproto_data” is Liveproto_extsz Bytes (for example when to use the API (6.7 API to get the shared data in camera buffer), it should be 8), the frame header will be changed to the below:

B



Type of Liveproto is 32bit value, each bit represents the type of extension;

When bit0 = 1, the value is the “Time difference between current frame's time and current camera's time", its unit is ms;

6.2 Request intercom

- **Description**

The interface to request the talkback or intercom.

- **Request URL**

```
/action/stream?subject=audiotalk
```

- **Request Body**

None.

■ **Response**

```
200: Succeed to Request

400: Error to Request.

403: No authorized.

500: Failed to Set.
```

■ **Further description**

codec: encoding type (0: g711u 1: g711a)

sample: sample rate

bitwidth: sample bit width

bitrate: bitrate in kps (unused)

channel: number of channels (0: mono, 1: stereo)

Audio intercom process

1. The client requests audio intercom from the device.

2. After the device responds successfully, the client sends audio frame data to the device about 1 second later (the first frame must be metadata, and the device needs to open the corresponding audio decoder according to this frame, and then the normal audio data).

3. The client disconnects and the device releases resources.

Note:

1. Currently only audio intercom with codec: g711u sample: 8000 bitwidth: 16 channel: 0 is supported.

2. The Connection field in the HTTP header must be Keep-Alive when requesting.

3. For the audio frame data format, please refer to-Request Audio and Video Stream data definition above.

6.3 Record Searching

■ **Description**

The interface to request the record searching.

■ **Request URL**

```
/action/stream?subject=recsearch
```

■ **Request Body**

Monthly Record searching:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <recsearch>
        <mode>0</mode>
        <chn>1</chn>
        <date>201507</date>
        <type>3</type>
        <stream>3</stream>
     </recsearch>
</request>
```

Daily record searching:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <recsearch>
        <mode>1</mode>
        <chn>1</chn>
        <date>20150724</date>
        <begin>0</begin>
        <end>3600</end>
        <type>3</type>
        <stream>3</stream>
     </recsearch>
</request>
```

■ **Response**

Monthly Record searching:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <recorddesc>
```

```
        <rdatemask>111</rdatemask>
      </recorddesc>
</response>
```

Daily record searching:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
        <recorddesc>
            <chn>0<chn>
            <begin>0</begin>
            <end>3600</end>
           <type>1</type>
           <stream>1<stream/>
           <path>/record/record1.avi<path/>
        </recorddesc>
        <recorddesc>
           <chn>0<chn>
           <begin>5000</begin>
            <end>6000</end>
           <type>2</type>
           <stream>2<stream/>
           <path>/record/record1.avi<path/>
        </recorddesc>
</request>
```

■ **Further description**

Month video search mode:

mode: Recording search mode (0: month 1: date).

chn: channel number, each bit represents a channel bit0 ~ bit31 represents channel 1 to channel 32, which is fixed to 1 for IPC.

date: Search for the specific month, format: YYYYMM.

type: search type, each bit represents a type (bit0: schedule recording bit1: alarm recording).

stream: The type of video stream. Each bit represents a type (bit0: main stream bit1: substream).

recorddesc.rdatemask: Recording date mask, each bit represents whether there is an eligible recording for a day in the month, up to 31 days. (bit0: the first day of the search month, bit1: the second day of the search

month, and so on), such as searching for videos in July 2015, returns rdatemask value equal to 15 (0xf), then 2015-07-01 Eligible videos existed until 2015-07-04, and no other videos existed on other dates.

Date recording search mode:

mode: video search mode (0: month mode 1: date mode)

chn: channel number, each bit represents a channel bit0 ~ bit31 represents channel 1 to channel 32, fixed to 1 for IPC

date: recording date, format: YYYYMMDD, when DD is 0, the entire month will be searched, and the search time parameter will be ignored

begin: Recording start time, in seconds

end: end time of recording, unit is second

type: recording type, each bit represents a type (bit0: timing recording bit1: alarm recording)

stream: the type of video stream, each bit represents a type (bit0: main stream bit1: substream)

recorddesc.chn: channel number, starting from 0, representing the first channel, fixed to 0 for IPC

recorddesc.begin: the recording start time, in seconds

recorddesc.end: end time of recording, unit is second

recorddesc.type: recording type, 1: timer recording 2: alarm recording

recorddesc.stream: Recording stream type, 0: main stream 1: sub stream

recorddesc.path: the path of the recording file

Note:

1. In the date recording search mode, if there are multiple eligible recording conditions, multiple recdesectc structure definitions will be returned.

2. The date recording search result reflects the recording timeline of the search date.

3. The monthly video search mode reflects the overall video status of the search month. If you need detailed video status, you need to use a specific date video search.

6.4 Playback and control

- **Description**

The interface to request the record playback and how to control the playback.

- **Request URL**

```
/action/stream?subject=playback
```

■ **Request Body**

Playback:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <playback>
        <chn>0</chn>
        <date>20150724</date>
        <begin>0</begin>
        <end>3600</end>
     </playback>
</request>
```

Control way:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <playctrl>
        <cmd>0</cmd>
        <args>3600</args>
     </playctrl>
</request>
```

■ **Response**

```
200: sucess
400：error for request
500：failed for request
```

■ **Further description**

chn: channel number, starting from 0, representing the first channel, fixed to 0 for IPC

cmd: playback control type (0: play 1: pause 2: time positioning 3: speed control)

args: playback control parameters (optional), explained as follows:

1) When playback control type is play or pause, it is meaningless

2) When the playback control type is time positioning, this parameter is the positioning time in seconds.

3) When the playback control type is speed control, this parameter is the speed value (0: Normal speed 1: 1 times fast playback 2: 2 times fast playback 3: 1 times slow playback 4: 2 times slow playback 5: frame playback)

Playback recording process (the process is similar to the real-time streaming process)

1) The client requests the video playback from the device

2) After receiving the request, the device sends 200 to the client and sends data to the client in about 1 second (the first frame is a metadata frame, and the client opens the corresponding decode according to this frame. (The decoder performs decoding, and the subsequent data is normal audio and video frame data)

3) After receiving the response, the client waits to start receiving audio and video data.

4) The client sends playback control (positioning, fast playback, slow playback, etc.) to the device (same socket)

5) Client disconnects, device releases resources.

Note:

1) During playback, if the encoded information (ie metadata) of the video or audio is inconsistent with the previous information, it will resend a new metadata frame, and then send the normal audio and video data frames. The client needs to compare this information and re-open the corresponding decoder to ensure normal decoding.

2) Playback positioning, only positioning the time position of I frame, and the first frame of video data returned after positioning is I frame

3) For the video and audio frame data format of the video, please refer to-Request Audio and Video Stream data definition.

6.5 Download the Archives

- **Description**

The interface is used to download the archives from the storage inside the cameras.

- **Request URL**

```
/action/stream?subject=recdownload
```

- **Request Body**

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <recdownload>
      <path>/record/record.avi</path>
     </recdownload>
</recdownload>
```

- **Response**

```
200: sucess
400：error for request
500：failed for request
```

- **Further description**

Path: the path of the archives.

Note:

Path: it's the value which is returned by the archives searching;

6.6 Request for video Raw data

- **Description**

The interface is used to request the video raw data from the camera.

- **Request URL**

```
/action/stream?subject=vrawdata&stream=0
```

- **Request Body**

```
Stream: stream type, 0: main stream 1: substream
```

- **Response**

```
200: sucess
400：error for request
```

500：failed for request

■ **Further description**

Request Rawdata Process

1) The client requests raw data from the device.
2) After receiving the request, the device sends 200 to the client and sends data to the client in about 1 second (the first frame is the attribute information frame, and the subsequent data is the normal Rawdata).
3) The client receives the response and waits to start receiving Rawdata
4) The client disconnects and the device releases resources.;

Note:

1) The Connection field in the HTTP header must be Keep-Alive when requesting
2) Each complete data packet consists of two parts: frame header + Rawdata, as shown in the figure below.

Frame header (24 Bytes)						data
32bit	8bit	24bit	32bit	32bit	64bit	N bits
magic	type	resv	length	seqno	timestamp	rawdata

megic: magic number, it's fixed as 0x20150723;

Type: frame type

0: video; 1: audio, 0xa: metadata

Resv: reserved

Length: Video or media data frame raw data length;

Seqno: Current frame sequence number, each session starts from 0;

Timestamp: Timestamp, invalid when type is media data frame (type = 0xa);

Data: Audio and video or media data raw data.

Video rawattr data is defined as follows:

Vrawdata (8 Bytes)			
4bit	28bit	16bit	16bit
rawfmt	resv1	width	height

Audio rawattr data is defined as follows:

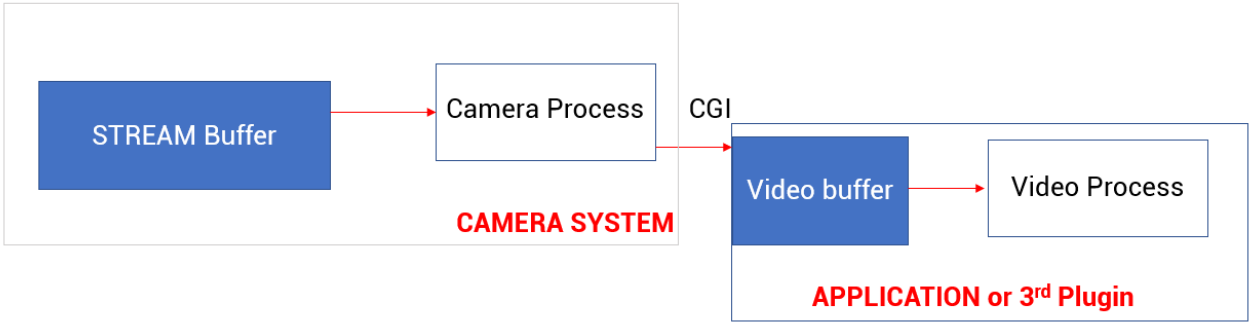
Arawdata (8 Bytes)			
4bit	28bit	16bit	16bit
rawfmt	resv1	sample	channel

6.7 API to get the shared data in camera buffer

■ **Description**

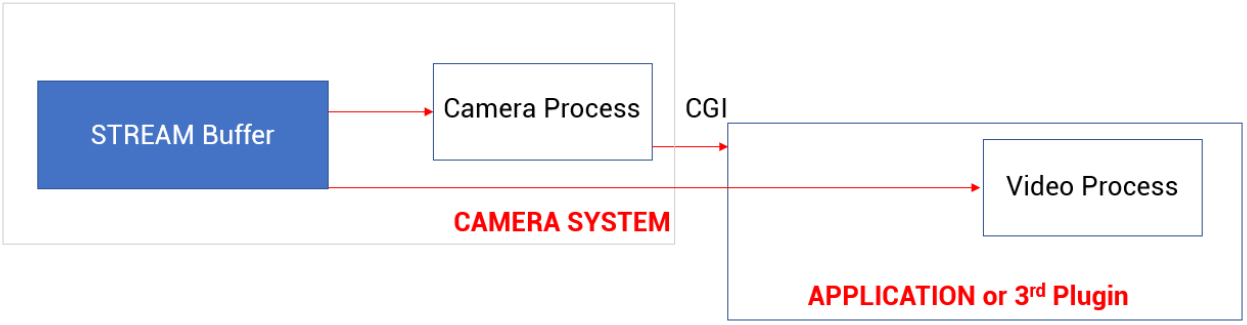
The application or 3rd part plugin can use the stream buffer which is existed in the camera, thus the application or 3rd plugin won't to have one more stream buffer.

The below is normal usage for the video stream:



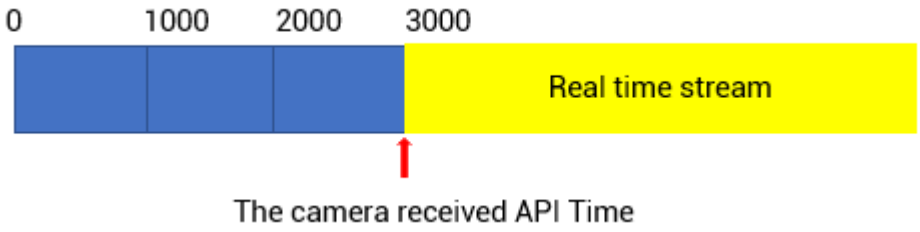
There are at least video stream bufferes, one is used for camera processes and another is used for applicaton or 3rd party plugin, in order to control the video stream for play or archiving etc.

Because of some reasons like limited RAM memory, or performance, the Application or 3rd party plugin can share the existed video stream buffer in camera side to save the memory as below.



The API is used to get the stream as the times before the live stream from the camera's video stream buffer

as below picture.



Request Delay = D, If D=1000, the camera sends the video frames from 2000ms; If D=2000, the camera sends the video frames from 1000ms; If D>=3000, the camera sends the video frames from 0ms;

- **Request URL**

```
action/stream?subject=liveplay&stream=0&forward=3000
```

- **Request Body**

```
NONE.
```

- **Response**

```
200:  success
400：error for request
500：failed for request
```

- **Further description**

stream: 0: mainstream, 1: substream;

forward: it's the “forward” value in buffer for request time, its unit is ms;

❑ Schedule Task part

7 Scheduled Task

7.1 Get and set the scheduled task

■ **Description**

The interface to get and set the scheduled task.

■ **Request URL**

GET:

```
/action/get?subject=schetask&type=0 [type: task type (0: snapshot 1: recording 2: ftp upload)]
```

SET:

```
/action/set?subject=schetask&type=0
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <schetask>
        <active>0</active>
        <schedule>
            <day>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
```

```
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
            </day>
            <day>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
            </day>
            ...
        </schedule>
    </schetask>
</response>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <schetask>
        <active>0</active>
        <schedule>
            <day>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
            </day>
            <day>
                <tsection>0-0</tsection>
                <tsection>0-0</tsection>
```

```
                    <tsection>0-0</tsection>
                    <tsection>0-0</tsection>
                    <tsection>0-0</tsection>
                    <tsection>0-0</tsection>
                </day>
                ...
            </schedule>
        </schetask>
</response>
```

SET:

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

active: work mode (0: Disable 1: 7*24 Hours 2: scheduled)

schedule: scheduled table, 1-week table, there is 6-time segments for each day. Unit: second.

Tsection: Time section, starting time - ending time, for example 0 clock - 1 clock is 0 - 3600.

7.2 Get and set the recording parameters

■ **Description**

The interface to get and set the recording parameters.

■ **Request URL**

GET:

```
/action/get?subject=record
```

SET:

```
/action/set?subject=record
```

■ **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <record>
        <stream>1</stream>
        <packsec>1</packsec>
        <presec>1</presec>
        <recycle>1</recycle>
        <path>0</path>
     </record>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <record>
        <stream>1</stream>
        <packsec>1</packsec>
        <presec>1</presec>
        <recycle>1</recycle>
        <path>0</path>
     </record>
</response>
```

SET:

```
200: Succeed to Request
400: Error to Request.
403: No authorized.
```

500: Failed to Set.

■ **Further description**

stream: recording stream type ()

packsec: recording packaging time, unit: second, valid data range: [180, 600]

presec: pre-recording time, unit: second, unit: second, valid data range: [0, 5]

recycle: Cycled recording index (0: Disable 1: Enable)

th: recording direction: 0: local recording; 1: NFS

❑ PTZ Operation part

8 PTZ Operation

8.1 PTZ Control

- **Description**

The interface to control the network PTZ cameras including Auto Focus motorized Lens camera Pan, Tilt and Zoom control.

- **Request URL**

```
/action/ptz?subject=ctrl
```

- **Request Body**

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <ptzcmd>
        <cmd>0</cmd>
        <focus>1</focus>
        <zoom>1</zoom>
        <move>
            <hoir>0</<hoir>
            <vert>1</<vert>
        </move>
        <preset>
            <index>1<index>
            <name>preset1<name>
        </preset>
         <cruise>
            <index>1<index>
            <cruisepoint>
                <preset>1</preset>
                 <speed>1</speed>
                 <second>1</second>
            </cruisepoint>
```

```
            <cruisepoint>
                <preset>2</preset>
                 <speed>2</speed>
                 <second>5</second>
            </cruisepoint>
                 ....
            <cruisepoint>
                <preset>8</preset>
                 <speed>2</speed>
                 <second>5</second>
            </cruisepoint>
        </cruise>
    </ptzcmd>
</request>
```

■ **Response**

```
200: Succeed to Set.
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

CMD: Command Type, CMD_PTZ_MOVE_HOR and CMD_PTZ_VERT can be used in combination.

```
CMD_PTZ_STOP            = 0x000,    //STOP;
CMD_PTZ_MOVE_HORI       =0x001,     // horizontal Moving;
CMD_PTZ_MOVE_VERT       =0x002,     //vertical Moving;
CMD_PTZ_FOCUS           =0x004,     //Focusing;
CMD_PTZ_ZOOM            =0x008,     //Zooming;
CMD_PTZ_RSET            =0x1000,    //Reset;
CMD_PTZ_GOTO_PREST      =0x1001,    //Call Preset;
CMD_PTZ_SET_PRESET      =0x1002,    //Set Preset;
CMD_PTZ_DEL_PRESET      =0x1003,    //Delete Preset;
CMD_PTZ_GOTO_HOME       =0x1004,    //Call HOME Preset;
CMD_PTZ_AUTOSCAN        =0x1005,    //Auto Scanning, it's invalid currently.
```

```
CMD_PTZ_IRIS                =0x1006,    //Auto Iris, it's invalid currently.
CMD_PTZ_SET_CRUISE          =0x1007,     //To set the cruise path;
CMD_PTZ_DEL_CRUISE          =0x1008,     //To delete the cruise path;
CMD_PTZ_CALL_CRUISE         =0x1009,     //Call the cruise path;

CMD_PTZ_EXT_ZMODE           =20000000, //User defined, Focus Mode
CMD_PTZ_EXT_INIT                       //User defined, Initialize
```

Focus:

Focus, it's valid when CMD_PTZ_FOCUS is taken effective.

0: Far, 1: Near;

Zoom:

Zooming, it's valid when CMS_PTZ_ZOOM is taken effective

0: wide 1: Tele

Move.hori:

Horizontal Moving, it is valid when CMD_PTZ_MOVE_HORI is taken effective

0: left 1: right

Move.vert:

Vertical moving. It's valid when CMD_PTZ_VERT is taken effective

0: up 1: down

Preset: Preset parameter, it's valid when CMD_PTZ_GOTO_PRESET, CMD_PTZ_SET_PRESET, CMD_PTZ_DEL_PRESET is taken effective;

Preset.index: it is the preset position index number, it is unique, the value range is [1,16]

Preset.name: it's the name of the preset, it cannot be empty;

Cruise: Cruise parameters, it's valid when CMD_PTZ_SET_CRUISE, CMD_PTZ_DEL_CRUISE, CMD_PTZ_CALL_CRUISE are taken effective;

Cruise.index: the index of the cruise path, it's fixed 1 as default;

Cruisepoint.preset: the presets of the cruise path, the value range is [1,16]

Cruisepoint.speed: the speed of moving to the next preset of the cruise, the value range is [1,5]. The bigger of the value, the quicker of the speed.

Cruisepoint.second: The stay time of each preset during cruise, the unit is second, the value range is [1,300]

Note:

Motorized lens cameras only support CMD_PTZ_FOCUS, CMD_PTZ_ZOOM, CMD_PTZ_EXT_INIT; CMD_PTZ_EXT_INIT is used to initiate the Auto Focus status.

8.2 Get and set the PTZ parameters

- **Description**

The interface to get and set the PTZ parameters.

- **Request URL**

GET:

```
/action/get?subject=ptz
```

SET:

```
/action/set?subject=ptz
```

- **Request Body**

GET: None

SET:

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <ptz>
        <speed>1</speed>
        <iris>1</iris>
        <light>1</light>
        <wiper>1</wiper>
        <zmode>1</zmode>
        <addr>1</addr>
        <protocol>1</protocol>
        <serial>
            <baudrate>2400</baudrate>
            <databit>8</databit>
```

```
            <stopbit>1</stopbit>
            <parity>0</parity>
            <flowctrl>0</flowctrl>
        </serial>
    </ptz>
</request>
```

■ **Response**

GET:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <ptz>
        <speed>1</speed>
        <iris>1</iris>
        <light>1</light>
        <wiper>1</wiper>
        <zmode>1</zmode>
        <addr>1</addr>
        <protocol>1</protocol>
        <serial>
            <baudrate>2400</baudrate>
            <databit>8</databit>
            <stopbit>1</stopbit>
            <parity>0</parity>
            <flowctrl>0</flowctrl>
        </serial>
        <preset>
            <pname>Preset0</pname>
            <pname>Preset1</pname>
            <pname>Preset2</pname>
                ...
            <pname>Preset127</pname>
        </preset>
        <cruise>
```

```
            <cruisepoint>
                <preset>1</preset>
                 <speed>1</speed>
                 <second>1</second>
            </cruisepoint>
            <cruisepoint>
                <preset>2</preset>
                 <speed>2</speed>
                 <second>5</second>
            </cruisepoint>
                 ....
            <cruisepoint>
                <preset>8</preset>
                 <speed>2</speed>
                 <second>5</second>
            </cruisepoint>
        </cruise>
    </ptz>
</response>
```

SET:

```
200: Succeed to Request
400: Error to Request.
403: No authorized.
500: Failed to Set.
```

■ **Further description**

speed: PTZ moving Speed, value rang is [0,5] , the bigger the value is, the quicker the speed is;

iris: Auto Iris, 0: Close, 1: Open (Option)

light: Light, 0: Close, 1: Open (Option)

wiper: Wiper, 0: Close, 1:Open (Option)

zmode: Focus Mode; (option)

protocol: type of the protocols, 1: PELCO-D, 2: PELCO-P;

addr: Address value, the value range is [1,255]

serial.baudrate: baudrate, the valid value is 2400/4800/9600/57600/19200/115200.

Serial.databit: date bit value, the valid value is 6/7/8.

Serial.stop: stop bit, valid value is 1/2

Serial.parity: the parity bit, 0: none, 1: odd, 2. Even

Serial.flowctrl: flow control bit, 0: disable 1: enable

Preset.name: the preset name, maximum 128 Presets can be set. If it's empty, it means the preset is invalid;

Cruise: Cruise path, maximum 3 cruise paths, each path supports maximum 32 presets.

Cruisepoint.preset: the preset of the cruise, the value range is [1,16]

Cruisepoint.speed: the speed of moving to the next preset of the cruise, the value range is [1,5]. The bigger of the value, the quicker of the speed.

Cruisepoint.second: The stay time of each preset during cruise, the unit is second, the value range is [1,300]

Note:

1. Preset can only be GET. If Preset wants to be set, please use /action/ptz?subject=ctrl to set and delete it.

2. Cruise can only be GET, if Cruise wants to be set, please use /action/ptz?subject=ctrl to set and delete it.

❑ HTTP URL of Stream and Snapshot part

9 HTTP URL of Snapshot and Stream

HTTP URL for the snapshot JPEG：

```
http://IP address:port/action/snap?cam=0&user=user&pwd=password;
```

HTTP Request for the URL of motion JPEG (1FPS):

```
http://IP address:port/action/stream?subject=mjpeg&user=user&pwd=password;
```

❑ Get the Log part

10 Get the LOG

HTTP URL for the request of the LOG from the camera：

```
http://IP address: port/action/log
```

❑ Smart Detection part (Specific MSTAR Solution)

11 Smart Detection APIs

11.1 NOTICE

This chapter describes the specifications of the CGI interface specialized in MSTAR solution-based IP cameras' smart detection features.

Smart detection on MSTAR solution supports: human detection, line crossing detection and intrusion detection.

11.2 Limitation for Smart Detection

❑ **Installation position**

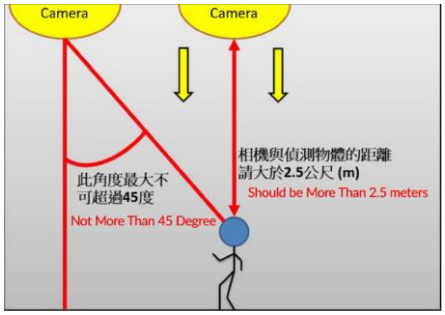
The camera should be installed in above of the monitoring objects

❑ **Installation angle**

The maximum angle with the detected object should not exceed 45 degrees.

❑ **Distance**

Distance from the detected object should be greater than 2.5 meters.



❑ **Application limitation**

1) Raindrops can affect judgment.

2) Too fast to detect.

3) The target object is too large (too small) to cause misjudgment (undetectable)

4) Shadows caused by light can cause misjudgments.

5) Luminance is not easy to detect

6) Reflections from the ground, wall or background can cause misjudgments.

7) Rapid changes in light are liable to be misjudged.

11.3 GET and SET Human Detection

11.3.1 Parameters Description

The related parameters are defined like the below:

enable: enable switch, 0: disable 1: enable

confidence: confidence value, range [0, 100]

scene: scene: 0: indoor 1: outdoor

border: border mark, 0: disabled 1: enabled

id: whether OSD displays id information [0: not displayed 1: displayed]

(HISILICON HI3516Cv500 Person detection only supports enable, confidence, border, showid)

areamask.row: number of rows in the detection area

areamask.col: number of columns in the detection area

areamask.mask: mask information of the detection area

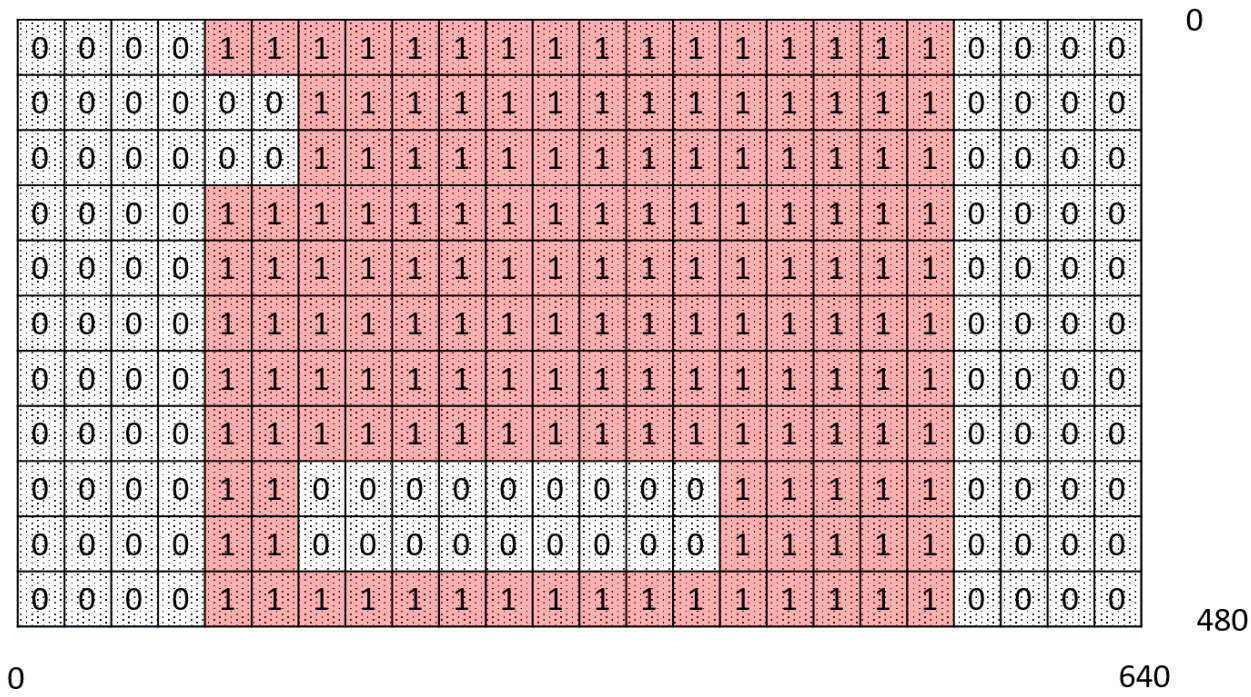
NOTE:

1) Detection Region: the detection region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid detection zone.

2) The value of area.mask identify if the macro zone is valid detection zone, 1: YES, 0: NO. Each bit represents each macro.

3) The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".

Example like the below:



The red colored zone is the valid detection zones, the macro zone was identified: 1;

For the above image, the value will be like the below:

areamask.row : 11

areamask.col: 24

areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0

11.3.2 GET Human Detection Parameters

Request URL	/action/get?subject=persondetect
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <persondetect ver="2.0"> <enable>0</enable> <confidence>50</confidence> <scene>0</scene> <border>1</border> <areamask> <row>18</row>

	<col>22</col> <mask>0,0,0,7000,7f8000,2000008,7e000,30200081,40020d80,4c20006,8000000,0,0</mask> </areamask> </persondetect> </response>

11.3.3 SET Human Detection Parameters

Request URL	/action/set?subject=persondetect
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <persondetect ver="2.0"> <enable>0</enable> <confidence>50</confidence> <scene>0</scene> <border>1</border> <areamask> <row>18</row> <col>22</col> <mask>0,0,0,7000,7f8000,2000008,7e000,30200081,40020d80,4c20006,8000000,0,0</mask> </areamask> </persondetect> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

11.4 GET and SET Line Crossing Parameters

11.4.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable the line crossing detection, 0: disable, 1: enable

line.begin: the beginning coordinate of the detected line, format: (x, y);

line.end: the ending coordinate of the detected line, format: (x, y)

sensitivity: the detection sensitivity value, its range is [1, 4]. The bigger of the value, the more sensitive.

direction: the crossing detection direction, 0: A to B; 1: B to A 2: A <->B;

scene: the application scenes, 0: indoor; 1: outdoor;

blink: enable blink or not when there is detection. 0: disable; 1: enable.

Show: enable or disable to show the detection line in the video. 0: disable, 1: enable;

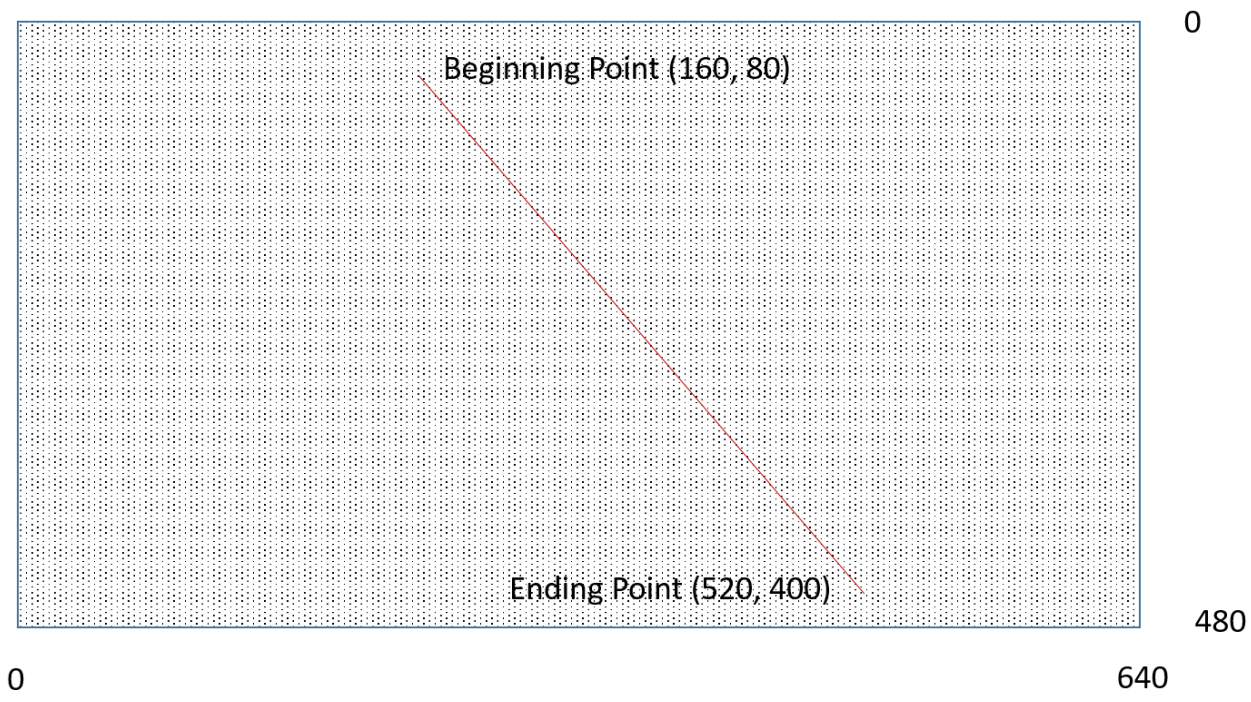
Note:

1) Enable: it can only be gotten, cannot be set. If to set this parameter, please use the API: 4.1 Get and Set Alarm Parameters

2) The coordinate value is the relative value which is percentage value of the crossing line.

3) For example, if the image of the zone resolution is 640 * 480, and the beginning coordinate was wanted to be set from (320, 240), thus the line.begin value should be (50, 50).

4) Below is the value of the line.begin and line.end calculated example.



On the above pictures shown, the line.begin value is (25,17), line.end value is (81,83)

11.4.2 GET Line Crossing Parameters

Request URL	/action/get?subject=crossline
Request Body	NONE

Response	<?xml version="1.0" encoding="utf-8"?> <response> <crossline> <enable>1</enable> <line> <begin>20,30</begin> <end>60,30</end> </line> <sensitivity>2</sensitivity> <direction>0</direction> <scene>0</scene> <blink>1</blink> <show>1</show> </crossline> </response>

11.4.3 SET Line Crossing Parameters

Request URL	/action/set?subject=vaglobal
Request Body	<?xml version="1.0" encoding="utf-8"?> <response> <crossline> <enable>1</enable> <line> <begin>20,30</begin> <end>60,30</end> </line> <sensitivity>2</sensitivity> <direction>0</direction> <scene>0</scene> <blink>1</blink> <show>1</show> </crossline> </response>

Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

11.5 GET and SET Intrusion Detection Parameters

11.5.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable intrusion detection, 0: disable, 1: enable

scene: the application scenes, 0: indoor; 1: outdoor;

sensitivity: the detection sensitivity value, its range is [0, 4]. The bigger of the value, the more sensitive.

direction: intrusion detection direction. 0: enter; 1: Leave; 2: both;

blink: enable blink or not when there is detection. 0: disable; 1: enable.

Show: enable or disable to show the detection frame. 0: disable, 1: enable;

areamask.row: the value of the raw of the macro zone in intrusion detection region.

areamask.col: the value of the column of the macro zone in intrusion detection region.

Areamask.mask: the mask value of the macro zone in the intrusion detection region.

polygon.point: The four vertex coordinates of the invading area are in turn the upper left, the upper right, the lower right, the lower left, and the percentage of coordinates in units.

NOTE:

Enable: it can only be gotten, cannot be set. If to set this parameter, please use the api: 4.1 Get and Set Alarm Parameters.

1) Intrusion Region: the intrusion region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid intrusion zone.

2) The value of area.mask identify if the macro zone is valid intrusion zone, 1: YES, 0: NO. Each bit represents each macro.

3) The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".

4) Intrusion area can be described in two ways, one is areamsk, the other is polygon. Different models may be described in different ways. Please take the actual way as the standard. For one type of intrusion detection, the region can only be described in one way, but not in both ways.

Example: Area mask:

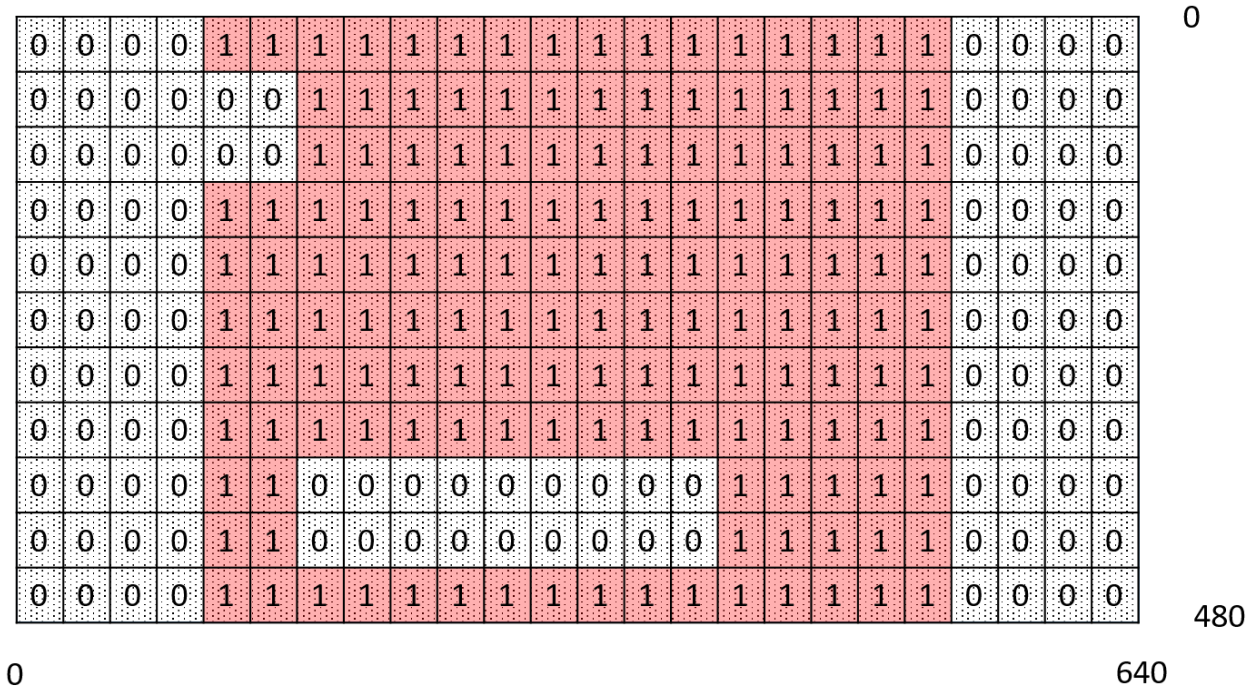
The red colored zone is the valid intrusion zones, the macro zone was identified: 1;

For the above image, the value will be like the below:

areamask.row : 11

areamask.col: 24

areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0



11.5.2 GET Intrusion Parameters

Request URL	/action/get?subject=intrusion
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <instrusion> <enable>1</enable> <scene>0</scene> <sensitivity>2</sensitivity> <direction>0</direction> <blink>1</blink> <show>1</show> <areamask> <row>16</row>

```
            <col>12</col>
            <mask>ffffffff,ffffffff,ffffffff</mask>
        </areamask>
        <polygon>
          <point>26,24</point>
          <point>60,33</point>
          <point>65,63</point>
           <point>42,69</point>
        </polygon>
    </instrusion>
</response>
```

11.5.3 SET Intrusion Parameters

Request URL	/action/set?subject=intrusion
Request Body	<?xml version="1.0" encoding="utf-8"?> <response> <instrusion> <enable>1</enable> <scene>0</scene> <sensitivity>2</sensitivity> <direction>0</direction> <blink>1</blink> <show>1</show> <areamask> <row>16</row> <col>12</col> <mask>ffffffff,ffffffff,ffffffff</mask> </areamask> <polygon> <point>26,24</point> <point>60,33</point> <point>65,63</point> <point>42,69</point> </polygon>

	</instrusion> </response>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

❑ Intelligent Video Analytics (IVA) part

12 IVA APIs

12.1 NOTICE

This chapter describes the specifications of the CGI interface specialized in HISILICON based IP cameras' intelligent video analytics (IVA) features.

Intelligent video analytics on HISILICON solution supports: people counting, crossing line, Intrusion, Human detection, object left and removed, Loitering, heatmap, wrong direction detection and face detection etc.

12.2 GET and SET Global Parameters for the feature.

12.2.1 Parameters Description

The related parameters are defined like the below:

width: the width of the image processed

height: the height of the image processed.

fps: the frame rate of the video processed, the valid value is (5,10,15)

bmin: the minimum size of the object detected, the format is (left, top, right, bottom), the unit is percentage.

bmax: the maximum size of the object detected, the format is (left, top, right, bottom), the unit is percentage.

Note:

1) both width and height value are read only parameters, it cannot be SET.

2) FPS value will impact the accuracy of the analytics result. The bigger of the value, the better accuracy, but it will bring the CPU load increasing, the performance will be impacted.

3) bmin should be smaller than bmax, the value will impact the sensitivity of the analytics, the reasonable value will take the better sensitivity on the reasonable CPU load.

12.2.2 GET Global Parameters for the feature

Request URL	/action/get?subject=vaglobal
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response>

	<vaglobal> <width>320</width> <height>180</height> <fps>5</fps> <bmin>0,0,5,5</bmin> <bmax>0,0,50,50</bmax> </vaglobal> </response>

12.2.3 SET Global Parameters for the feature

Request URL	/action/set?subject=vaglobal
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <vaglobal> <width>320</width> <height>180</height> <fps>10</fps> <bmin>0,0,5,5</bmin> <bmax>0,0,50,50</bmax> </vaglobal> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

12.3 GET and SET Human Detection and Face Detection Parameters

12.3.1 Parameters Description

The related parameters are defined like the below:

Person: enable or disable human/person detection, 0: disable, 1: enable

Face.active: enable or disable face detection, 0: disable, 1: enable

Face.confidence: the confidence value, value range: [0, 100]

Face.fsize: the size of the face, the value is (5,15,20);

Face.roi: the zone which can detect the face, the format is (left, top, right, bottom), the unit is percentage;

Note:

1) Human detection is now disabled.

2) Because of the limited performance of the camera, now the others features will be disabled when the face detection feature is enabled.

3) Starting face detection will take much higher CPU load;

4) The face size can be adjusted according to the distance of the camera with the face, generally the longer distance of the face, the smaller value of the face, but the CPU load will be higher.

5) To set the face detection zone (ROI) can lower the CPU load much. Generally, the size of the zone is not bigger than 60% of the image. The bigger size of the zone, the much higher CPU load.

12.3.2 GET Human and Face Detection Parameters

Request URL	/action/get?subject=human
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <human> <person>1</person> <face> <active>1</active> <confidence>50</confidence> <fsize>1</fsize> <roi>10,10,50,30</roi> </face> </human> </response>

12.3.3 SET Human and Face Detection Parameters

Request URL	/action/set?subject=human
Request Body	<?xml version="1.0" encoding="utf-8"?> <request>

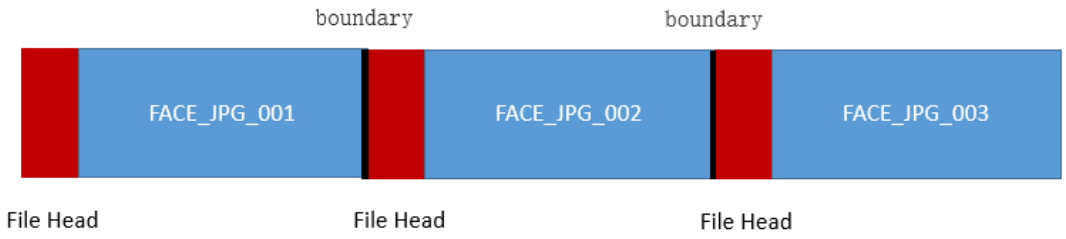
	<human> <person>1</person> <face> <active>1</active> <confidence>50</confidence> <fsize>1</fsize> <roi>10,10,50,30</roi> </face> </human> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

12.4 GET the Snapshot of the detected Face

12.4.1 Parameters Description

The standard Motion JPEG Stream will be responded when ask the HTTP request.

These images sending is using the standard MJPEG-Streamer technology, and there is the images stream sending from the camera's definition:



The boundary stream between the images: brovotechmjpegstreamboundary which is the string to show SOI (start of Image) or EOI (end of image)

File Head Definition:

File Type: image/jpeg

File-Length:

File-Name: face_20160826T134457_[46_50_82_86].jpg

20160826T134457: the image snapshotted time: YYYYMMDDThhmmss

[46_50_82_86]: the line and columns of the image in the full picture: lef_top_right_bottom

Here is the example response information if you ask the request:

HTTP/1.0 200 ok

Server: Brovotech/2.0.0

Connection: Keep-Alive

Content-Type:multipart/x-mixed-replace;boundary=------brovotectmjpegstreamboundary

--------------------------brovotectmjpegstreamboundary----------

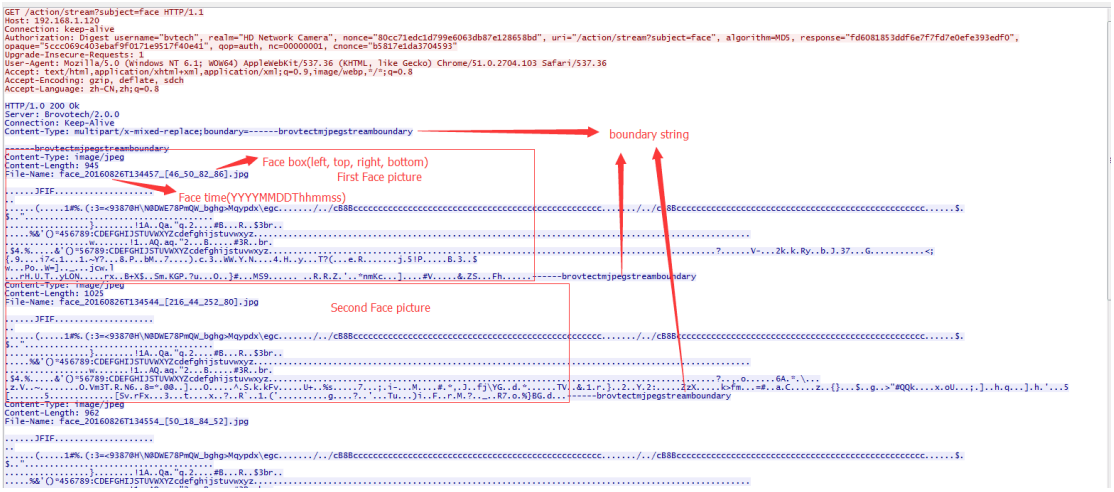
Content-Type: image/jpeg

Content-Length: 945

File-Name: face_20160826T134457_[46_50_82_86].jpg

……JFIF……..

The first face picture



The user can use the HTTP CGI to get all JPEG images to the same directory.

12.4.2 GET Snapshot images stream of the detected Face

Request URL	/action/stream?subject=face
Request Body	None.
Response	200: Succeed in Request 400: Error of Request 403: No Right to Request 500: Error in Server

12.5 GET and SET People Counting Parameters

12.5.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable the people counting, 0: disable, 1: enable

ftp_enable: enable or disable FTP uploading the people counting records fle (DBF format).

ftp_trytime: the qty of the trying to upload to the FTP server;

ftpsendtime: the specified FTP upload time, the FTP upload will be executed as this specified time.

line.begin: the beginning coordinate of the boundary of the zone, format: (x, y);

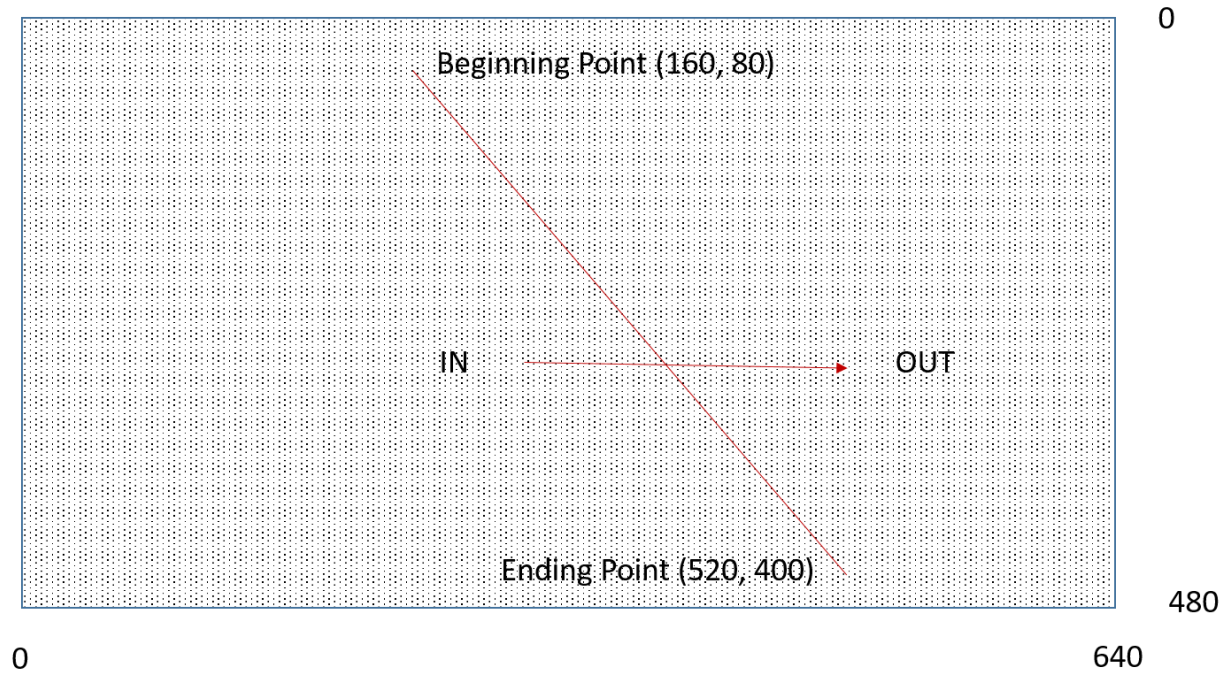
line.end: the ending coordinate of the boundary of the zone, format: (x, y)

Note:

the coordinate value is the relative value which is percentage value of the zone size which need people counting.

For example, if the image of the zone resolution is 640 * 480, and the beginning coordinate was wanted to be set from (320, 240), thus the line.begin value should be (50, 50).

Below is the value of the line.begin and line.end calculated example.



On the above pictures shown, the line.begin value is (25,17), line.end value is (81,83)

12.5.2 GET Parameters of People Counting

Request URL	/action/get?subject=counter
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <counter> <enable>1</enable> <ftp_enable>1</ftp_enable> <ftp_trytime>3</ftp_trytime> <ftp_sendtime>48900</ftp_sendtime> <line> <begin>20,30</begin> <end>60,30</end> </line> </counter> </response>

12.5.3 SET Parameters of People Counting

Request URL	/action/set?subject=counter
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <counter> <enable>1</enable> <ftp_enable>1</ftp_enable> <ftp_trytime>3</ftp_trytime> <ftp_sendtime>48900</ftp_sendtime> <line> <begin>20,30</begin> <end>60,30</end> </line> </counter> </request>

Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

12.5.4 Inquiry the People Counting Information

Request URL	/action/peoplecount?subject=peoplecount_info&start=0000-00-00&end=9999-99-99
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <peoplecount> <peoplecountfile>DB19990101.dbf</peoplecountfile> <snedtime>11:59:00</snedtime> <upstatus>1</upstatus> </peoplecount> <peoplecount> <peoplecountfile>DB19990102.dbf</peoplecountfile> <sendtime>11:59:00</sendtime> <upstatus>1</upstatus> </peoplecount> </response>
	peoplecounrfile: the statistic file name of the people counting sendtime:peoplecoun: the uploading time of the file. upstatus:peoplecoun: the uploading status of the file: 0: success; -1: failed; 1: not upload.

12.5.5 Download the Statistic File of the People Counting

Request URL	/action/peoplecount?subject=peoplecount_file&name=DB19990101.dbf
Request Body	NONE
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set
Note	The parameter “name” is the returned value of one item in peoplecount list from the statistic file of peoplecountfile.

	The CGI returned is the statistic DBF file.
	The API is used to download the statistic file per day from the camera to the computer manually.

12.5.6 Upload the Statistic File of the People Counting

Request URL	/action/peoplecount?subject=peoplecount_ftp&name=DB19990101.dbf
Request Body	NONE
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set
	The parameter "name" is the returned value of one item in peoplecount list from the statistic file of peoplecountfile. A return value of 200 indicates that the upload was successful.
	The API is used to upload the statistic file per day from the camera to the specified FTP server manually or by schedule.

12.6 GET and SET Line Crossing Parameters

12.6.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable the line crossing detection, 0: disable, 1: enable

line.begin: the beginning coordinate of the detected line, format: (x, y);

line.end: the ending coordinate of the detected line, format: (x, y)

sensitivity: the detection sensitivity value, its range is [1, 4]. The bigger of the value, the more sensitive.

direction: the crossing detection direction, 0: A to B; 1: B to A 2: A <->B;

scene: the application scenes, 0: indoor; 1: outdoor;

blink: enable blink or not when there is detection. 0: disable; 1: enable.

Show: enable or disable to show the detection line in the video. 0: disable, 1: enable;

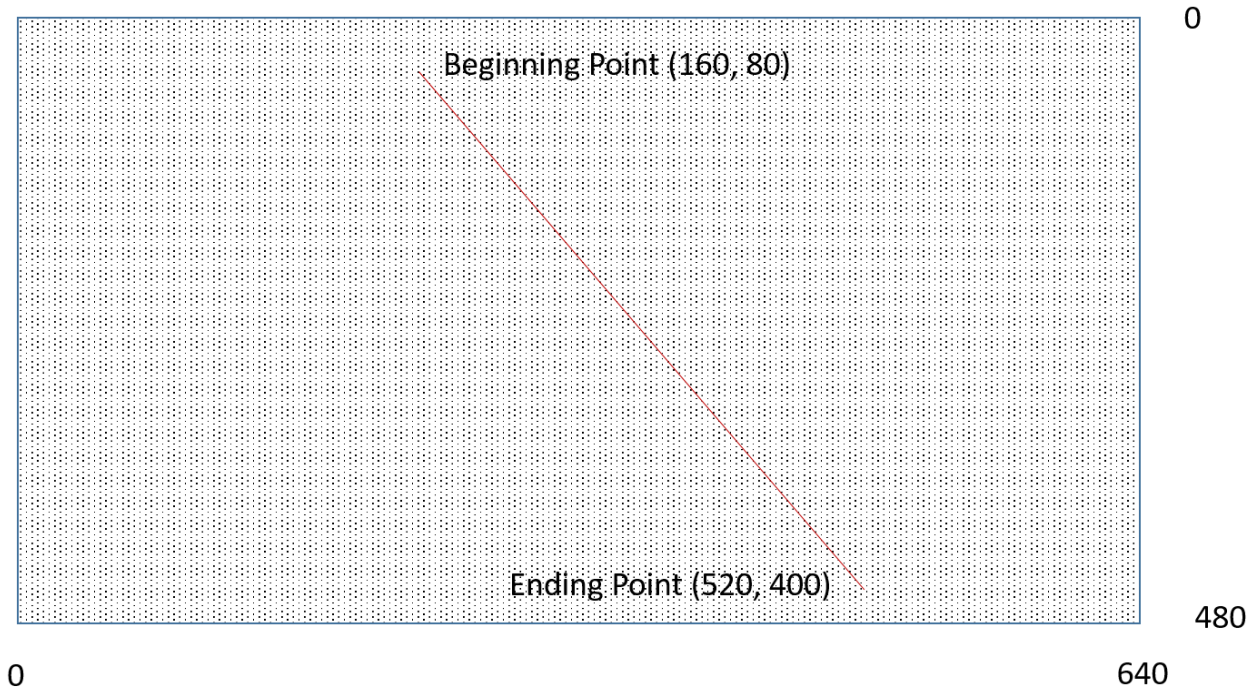
Note:

1) Enable: it can only be gotten, cannot be set. If to set this parameter, please use the API: 4.1 Get and Set Alarm Parameters

2) The coordinate value is the relative value which is percentage value of the crossing line.

3) For example, if the image of the zone resolution is 640 * 480, and the beginning coordinate was wanted to be set from (320, 240), thus the line.begin value should be (50, 50).

4) Below is the value of the line.begin and line.end calculated example.



On the above pictures shown, the line.begin value is (25,17), line.end value is (81,83)

12.6.2 GET Line Crossing Parameters

Request URL	/action/get?subject=crossline
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <crossline> <enable>1</enable> <line> <begin>20,30</begin> <end>60,30</end> </line> <sensitivity>2</sensitivity>

	<direction>0</direction> <scene>0</scene> <blink>1</blink> <show>1</show> </crossline> </response>

12.6.3 SET Line Crossing Parameters

Request URL	/action/set?subject=vaglobal
Request Body	<?xml version="1.0" encoding="utf-8"?> <response> <crossline> <enable>1</enable> <line> <begin>20,30</begin> <end>60,30</end> </line> <sensitivity>2</sensitivity> <direction>0</direction> <scene>0</scene> <blink>1</blink> <show>1</show> </crossline> </response>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

12.7 GET and SET Intrusion Detection Parameters

12.7.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable intrusion detection, 0: disable, 1: enable

scene: the application scenes, 0: indoor; 1: outdoor;

sensitivity: the detection sensitivity value, its range is [0, 4]. The bigger of the value, the more sensitive.

direction: intrusion detection direction. 0: enter; 1: Leave; 2: both;

blink: enable blink or not when there is detection. 0: disable; 1: enable.

Show: enable or disable to show the detection frame. 0: disable, 1: enable;

areamask.row: the value of the raw of the macro zone in intrusion detection region.

areamask.col: the value of the column of the macro zone in intrusion detection region.

Areamask.mask: the mask value of the macro zone in the intrusion detection region.

polygon.point: The four vertex coordinates of the invading area are in turn the upper left, the upper right, the lower right, the lower left, and the percentage of coordinates in units.

NOTE:

1) Enable: it can only be gotten, cannot be set. If to set this parameter, please use the api: 4.1 Get and Set Alarm Parameters.

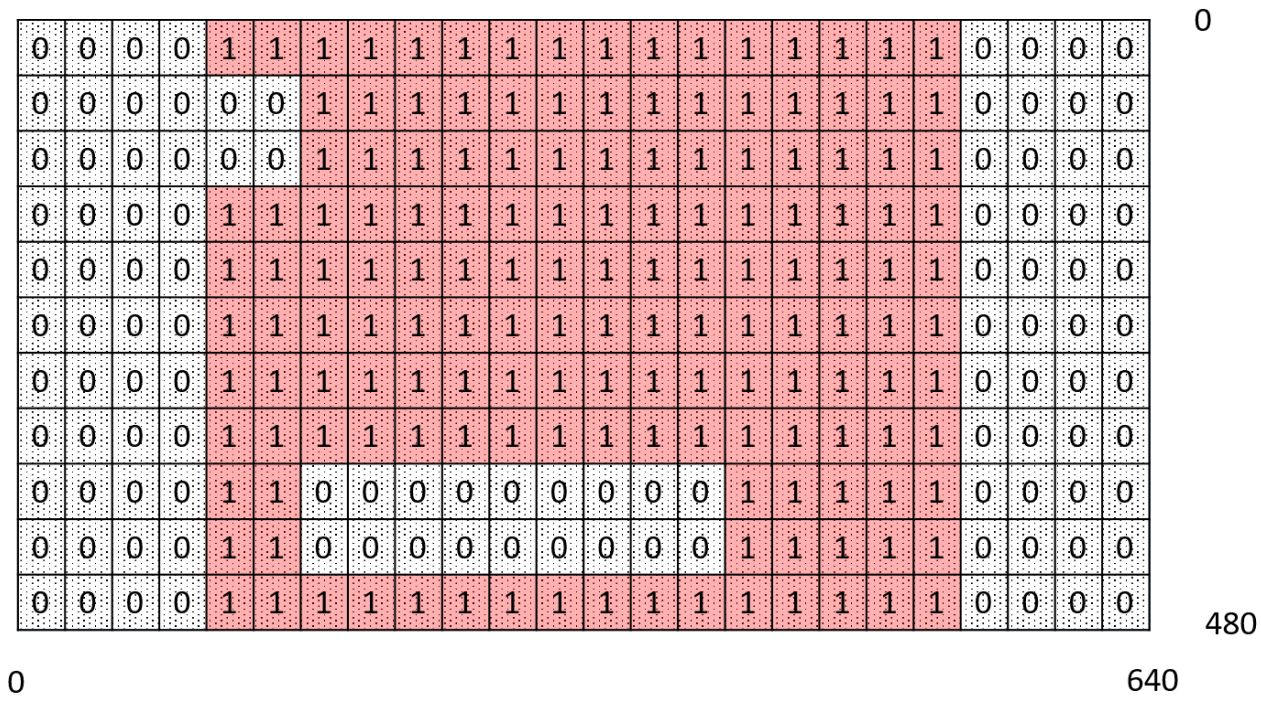
2) Intrusion Region: the intrusion region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid intrusion zone.

3) The value of area.mask identify if the macro zone is valid intrusion zone, 1: YES, 0: NO. Each bit represents each macro.

4) The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".

5) Intrusion area can be described in two ways, one is areamsk, the other is polygon. Different models may be described in different ways. Please take the actual way as the standard. For one type of intrusion detection, the region can only be described in one way, but not in both ways.

Example: Area mask:
The red colored zone is the valid intrusion zones, the macro zone was identified: 1;
For the above image, the value will be like the below:



areamask.row : 11

areamask.col: 24

areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0

12.7.2 GET Intrusion Parameters

Request URL	/action/get?subject=intrusion
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <instrusion> <enable>1</enable> <scene>0</scene> <sensitivity>2</sensitivity> <direction>0</direction> <blink>1</blink> <show>1</show> <areamask> <row>16</row>

```
            <col>12</col>
            <mask>ffffffff,ffffffff,ffffffff</mask>
        </areamask>
        <polygon>
          <point>26,24</point>
          <point>60,33</point>
          <point>65,63</point>
           <point>42,69</point>
        </polygon>
    </instrusion>
</response>
```

12.7.3 SET Intrusion Parameters

Request URL	/action/set?subject=intrusion
Request Body	<?xml version="1.0" encoding="utf-8"?> <response> <instrusion> <enable>1</enable> <scene>0</scene> <sensitivity>2</sensitivity> <direction>0</direction> <blink>1</blink> <show>1</show> <areamask> <row>16</row> <col>12</col> <mask>ffffffff,ffffffff,ffffffff</mask> </areamask> <polygon> <point>26,24</point> <point>60,33</point> <point>65,63</point> <point>42,69</point> </polygon>

	</instrusion> </response>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

12.8 GET and SET Object Left and Removed Detection

12.8.1 Parameters Description

The related parameters are defined like the below:

left: enable or disable object left detection, 0: disable, 1: enable

removed: enable or disable object removed detection, 0: disable, 1: enable

areamask.row: the value of the raw of the macro zone in detection region.

areamask.col: the value of the column of the macro zone in detection region.

areamask.mask: the mask value of the macro zone in the detection region.

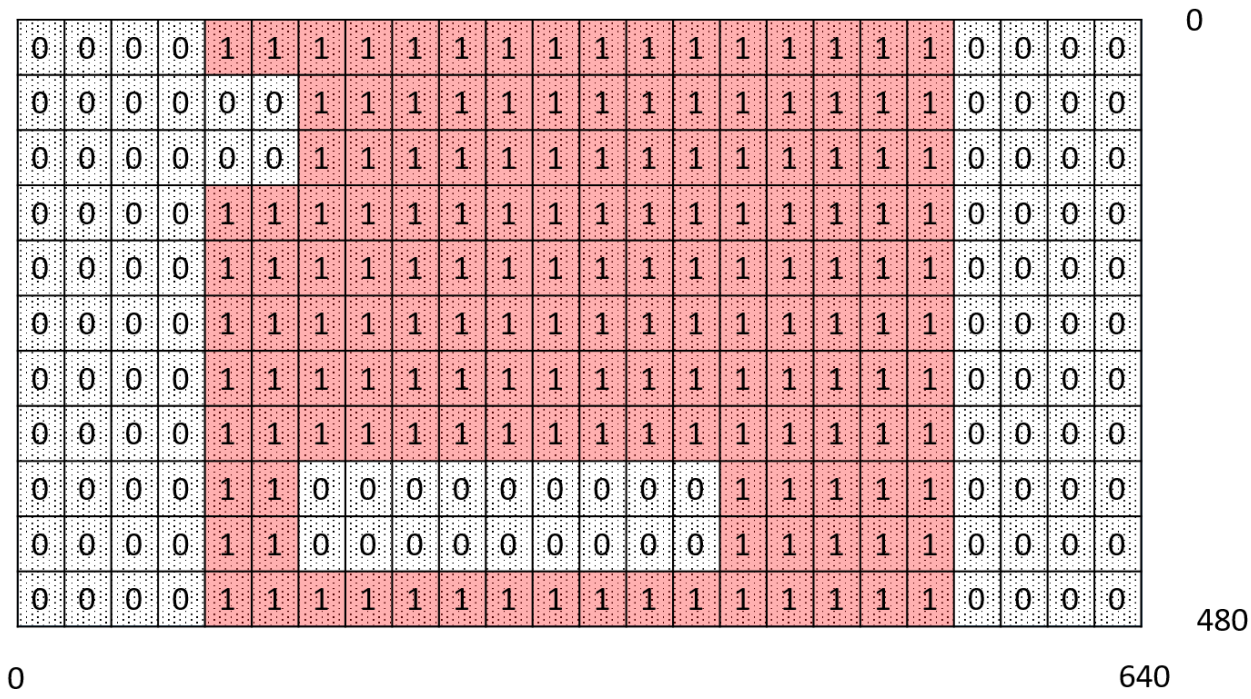
NOTE:

1) Detection Region: the detection region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid detection zone.

2) The value of area.mask identify if the macro zone is valid detection zone, 1: YES, 0: NO. Each bit represents each macro.

3) The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".

4) Example like the below:



The red colored zone is the valid detection zones, the macro zone was identified: 1;

For the above image, the value will be like the below:

areamask.row : 11

areamask.col: 24

areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0

12.8.2 GET Object Left and Removed Detection

Request URL	/action/get?subject=leftremoved
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <leftremoved> <left>1</left> <removed>1</removed> <areamask> <row>11</row> <col>24</col> <mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask> </areamask>

	</leftremoved> </response>

12.8.3 SET Object Left and Removed Detection

Request URL	/action/set?subject=leftremoved
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <leftremoved> <left>1</left> <removed>1</removed> <areamask> <row>11</row> <col>24</col> <mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask> </areamask> </leftremoved> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

12.9 GET and SET Loitering Detection

12.9.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable loitering detection, 0: disable, 1: enable

second: the minimum loitering time of the object stayed in the detection zone. Unit: second, the valid value: (5,10,15)

areamask.row: the value of the raw of the macro zone in detection region.

areamask.col: the value of the column of the macro zone in detection region.

areamask.mask: the mask value of the macro zone in the detection region.

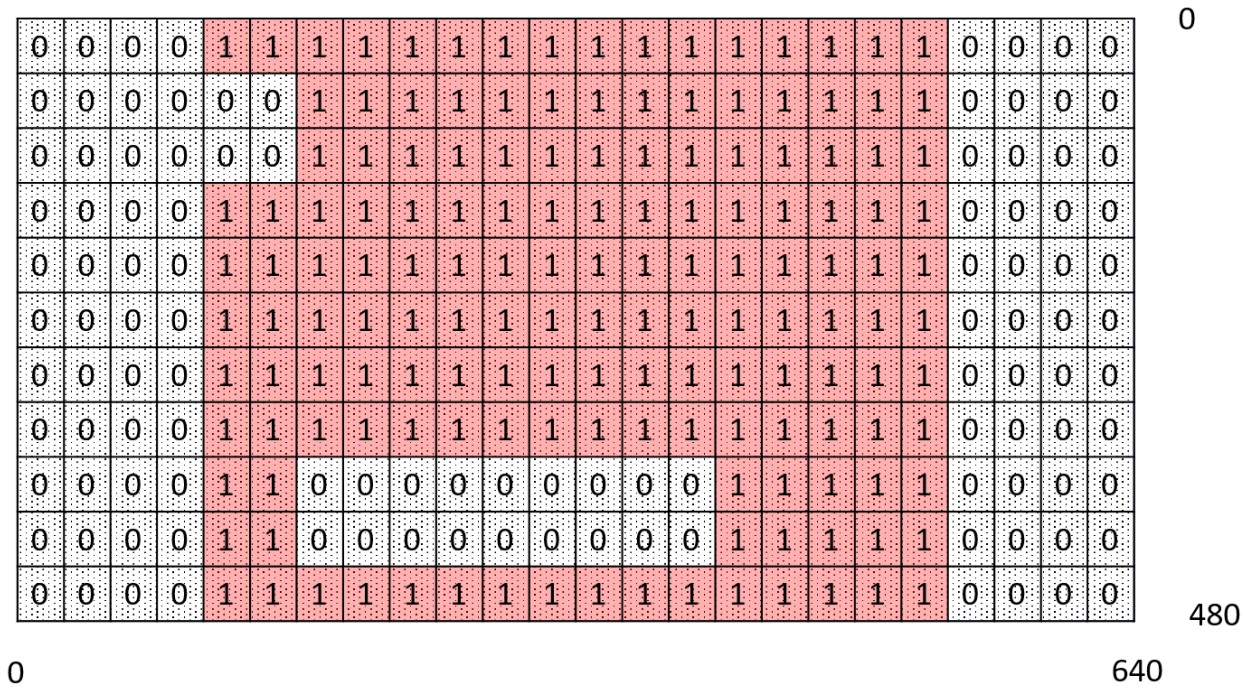
NOTE:

1) Detection Region: the detection region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid detection zone.

2) The value of area.mask identify if the macro zone is valid detection zone, 1: YES, 0: NO. Each bit represents each macro.

3) The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".

4) Example like the below:



The red colored zone is the valid detection zones, the macro zone was identified: 1;

For the above image, the value will be like the below:

areamask.row : 11

areamask.col: 24

areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0

5) enable, can only be obtained, cannot be set, if enabled or disabled, please use the corresponding event switch of the alarm event parameter

12.9.2 GET Loitering Detection

Request URL	/action/get?subject=loitering
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response>

	<loitering> <enable>1</enable> <second>5</second> <areamask> <row>11</row> <col>24</col> <mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask> </areamask> </loitering> </response>

12.9.3 SET Loitering Detection

Request URL	/action/set?subject=loitering
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <loitering> <enable>1</enable> <second>5</second> <areamask> <row>11</row> <col>24</col> <mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask> </areamask> </loitering> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

12.10 GET and SET Heatmap Detection

12.10.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable heatmap detection, 0: disable, 1: enable

second: the minimum time of the object stayed in the detection zone. Unit: second, the valid value: (5,10,15)

areamask.row: the value of the raw of the macro zone in detection region.

areamask.col: the value of the column of the macro zone in detection region.

areamask.mask: the mask value of the macro zone in the detection region.

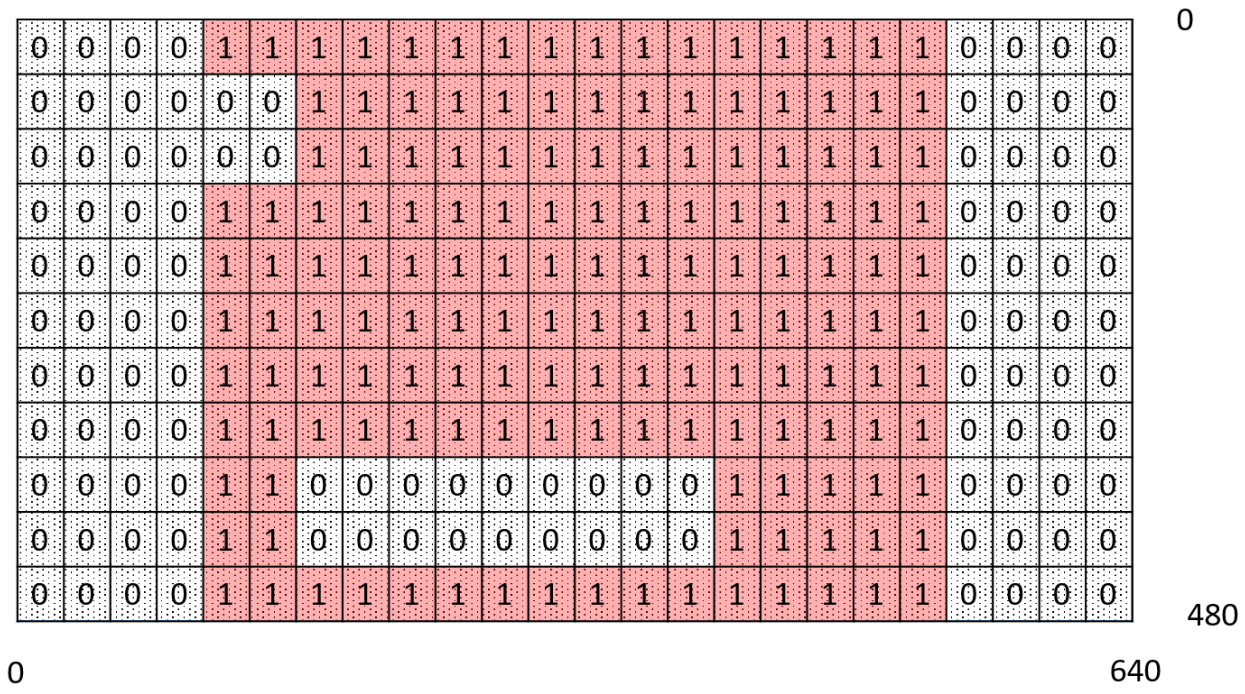
NOTE:

1) Detection Region: the detection region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid detection zone.

2) The value of area.mask identify if the macro zone is valid detection zone, 1: YES, 0: NO. Each bit represents each macro.

3) The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".

4) Example like the below:



The red colored zone is the valid detection zones, the macro zone was identified: 1;

For the above image, the value will be like the below:

areamask.row : 11

areamask.col: 24

areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0

12.10.2 GET Heatmap Detection

Request URL	/action/get?subject=heatmap
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <heatmap> <enable>1</enable> <second>5</second> <areamask> <row>11</row> <col>24</col> <mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask> </areamask> </heatmap> </response>

12.10.3 SET Heatmap Detection

Request URL	/action/set?subject=heatmap
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <heatmap> <enable>1</enable> <second>5</second> <areamask> <row>11</row> <col>24</col> <mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask> </areamask> </heatmap> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set

	500: Failed to Set

12.11 GET and SET Wrong Direction Detection Parameters

12.11.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable the wrong direction detection, 0: disable, 1: enable

line.begin: the beginning coordinate of the direction, format: (x, y);

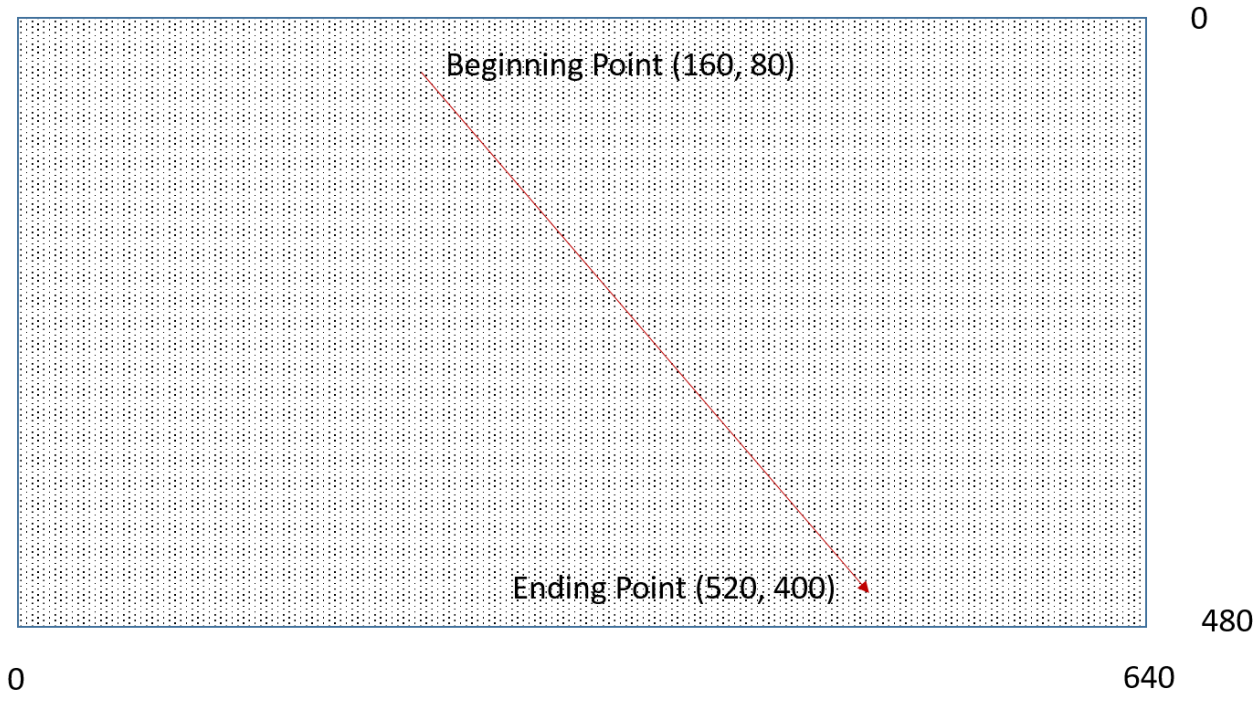
line.end: the ending coordinate of the direction, format: (x, y)

Note:

1) The coordinate value is the relative value which is percentage value of the zone size which need people counting.

2) For example, if the image of the zone resolution is 640 * 480, and the beginning coordinate was wanted to be set from (320, 240), thus the line.begin value should be (50, 50).

3) Below is the value of the line.begin and line.end calculated example.



On the above pictures shown, the line.begin value is (25,17), line.end value is (81,83).

12.11.2 GET the Wrong Direction Detection Parameters

Request URL	/action/get?subject=wrongdir
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <wrongdir> <enable>1</enable> <line> <begin>25,17</begin> <end>81,83</end> </line> </wrongdir> </response>

12.11.3 SET the Wrong Direction Detection Parameters

Request URL	/action/set?subject=wrongdir
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <wrongdir> <enable>1</enable> <line> <begin>25,17</begin> <end>85,83</end> </line> </wrongdir> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

12.12 The interface to reset the statistic result

12.12.1 Parameters Description

The interface is used to make the reset the statistic result of the people counting and heatmap analytics.

The related parameters are defined like the below:

counter: RESET the counting result of the people counting, 0: NO, 1: YES

heatmap: RESET the statistics result of the heatmap counting, 0: NO, 1: YES

12.12.2 The interface to manage IVA

Request URL	/action/set?subject=vactrl
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <request> <vactrl> <reset> <counter>1</counter> <heatmap>1</heatmap> </reset> </vactrl> </request>

12.13 GET the Result of IVA

12.13.1 Parameters Description

The related parameters are defined like the below:

lseqno: the sequence number of the latest triggered event.

vaevent.id: ID number of the triggered event

vaevent.etype: the type of the triggered event: 1, Object Left; 2, Object Removed; 3, Crossing Line; 4, Intrusion; 5, Loitering; 6, Wrong Direction Way

vaevent.time: the triggered time of the event, 20160708T162004 means it's the time: Year 2016, Month: 07, Day: 08, Hour: 16, Minute: 20, Second: 04

counter.in: the quantity of the object IN;

counter.out: the quantity of the object OUT;

heatmap.count: the statistics result of the heatmap.

Note:

1) If there are many events which are happened at the same time, thus several "vaevent" will be returned, and all of the "lseqno" are valid.

2) If there is no any new event triggered, there will be no any "vaevent" and "lseqno" returned any more.

3) Upon getting the events, the "last" value should be set in the request URL. It means that the current "last" value is the value of the last returned and valid "lseqno". If it's the first time to get, the value of "last" can be set "-1";

12.13.2 GET the Analytics Result

Request URL	/action/get?subject=varesult&last=-1
Request Body	last: the sequence number of the current event got, if it's -1, it means that it's to get all of the current event list.
Response	<?xml version="1.0" encoding="utf-8"?> <request> <varesult> <lseqno>1</lseqno> <vaevent> <id>1</id> <etype>0</etype> <time>20160204T120802</time> </vaevent> <vaevent> <id></id> <etype></etype> <time>20160204T120802</time> </vaevent> ..... <counter> <in>10</in> <out>10</out> </counter>

	<heatmap> <count>10</count> </heatmap> </varesult> </request>

❑ AI Functions APIs

13 AI Functions APIs

13.1 GET the Snapshot of the detected Face

13.1.1 Parameters Description

The interface is the same as the features mentioned in 12.4 of IVA funtions.

The standard Motion JPEG Stream will be responded when ask the HTTP request.

These images sending is using the standard MJPEG-Streamer technology, and there is the images stream sending from the camera's definition:

The boundary stream between the images: brovotechmjpegstreamboundary which is the string to show SOI (start of Image) or EOI (end of image)

File Head Definition:

File Type: image/jpeg

File-Length:

File-Name: face_20160826T134457_[46_50_82_86].jpg

20160826T134457: the image snapshotted time: YYYYMMDDThhmmss

[46_50_82_86]: the line and columns of the image in the full picture: lef_top_right_bottom

Here is the example response information if you ask the request:

HTTP/1.0 200 ok

Server: Brovotech/2.0.0

Connection: Keep-Alive

Content-Type:multipart/x-mixed-replace;boundary=------brovotectmjpegstreamboundary

--------------------------brovotectmjpegstreamboundary----------

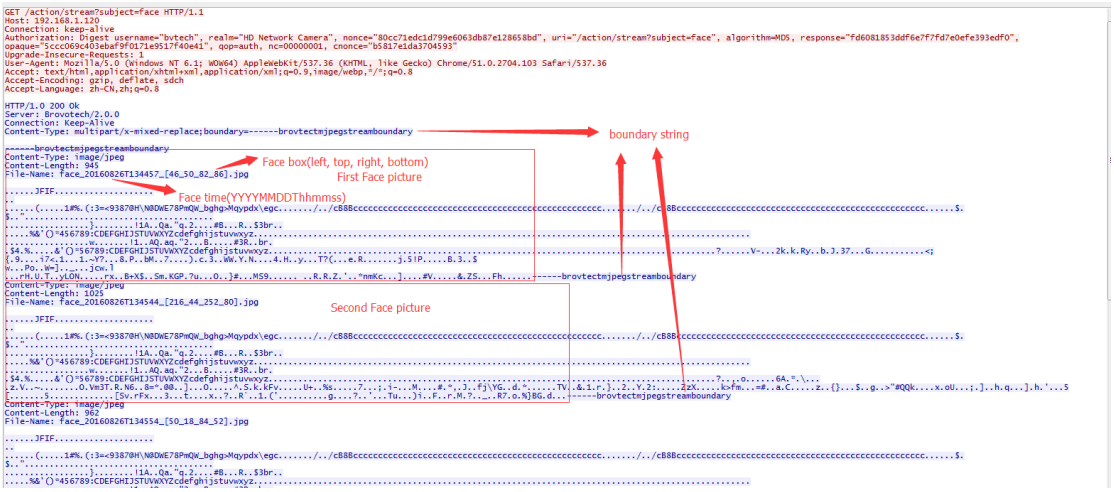
Content-Type: image/jpeg

Content-Length:945

File-Name: face_20160826T134457_[46_50_82_86].jpg

……JFIF……..

The first face picture



The user can use the HTTP CGI to get all JPEG images to the same directory.

13.1.2 GET Snapshot images stream of the detected Face

Request URL	/action/stream?subject=face
Request Body	None.
Response	200: Succeed in Request 400: Error of Request 403: No Right to Request 500: Error in Server

13.2 The interface to get the detected face pictures

13.2.1 Parameters Description

When it returns 200 successfully, the message body data is binary data of the face captured picture, the format is jpg.

Note:

1. When the return is successful, the Fseqno field is returned in the HTTP message header. It is required for the next request. This value must be filled in last to indicate that the next value of the current sequence number is obtained.

When last = -1, it means to get the latest face image

2. When the return is successful, the FName field is returned in the HTTP message header in the format of face_YYYYMMDDThhmmss.ms_ [left_top_right_bottom] _id_confidence.jpg

among them:

YYYYMMDDDThhmmss.ms is year, month, day, T hours, minutes, seconds. Milliseconds

left_top_right_booom is the coordinate information of the face, top left / bottom right

id: Face ID number

confidence: the confidence of the face.

13.2.2 The interface to get the detected face pictures

Request URL	/action/face?subject=facepic&last=-1
Request Body	NONE
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

13.3 GET and SET Face Detection Parameters

13.3.1 Parameters Description

The related parameters are defined like the below:

enable: enable switch, 0: disable 1: enable

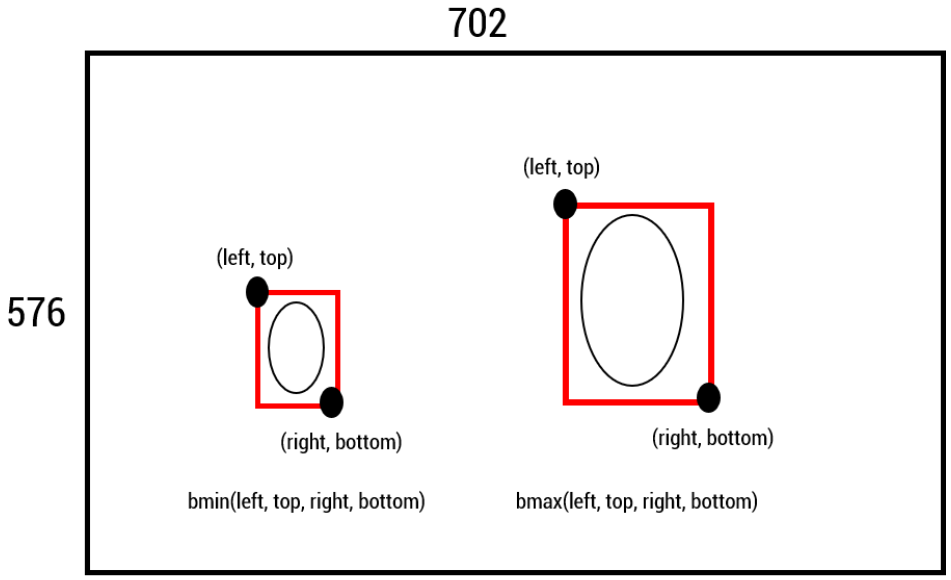
confidence: confidence value, range [0, 100]

pquality: the quality of the captured face image, range [0, 2], poor, medium, good

pmode: Face capture picture mode, 0: small picture 1: large picture

bmin: Detects the minimum size of a face, in the format (left, top, right, bottom), with the unit being a percentage

bmax: detect the maximum size of a face, in the format (left, top, right, bottom), the unit is percentageExplanation about bmin and bmax parameters is like below:



The video for settings
As example, the video used resolution is D1 (702*576)

In CGI, parameters (left,top,right,bottom) is the percentage of the pixels of the video resolution.

As example if the CMS/Web/Nvr etc. devices are using the D1(702*576) video for parameter settings.

If the bmin in camera is (5,5,10,10), and the value in pixel should be (702*5%,576*5%,702*10%,576*10%) (35,28, 70, 58). So, in pixel in left-top point is (35,28) and right-bottom point is (70,58). thus if these devices is using “pixels” to get the settings from the camera and then do the following settings in its side, they should use this calculation internal as this way.
By contraries, if the CMS/Web/NVR want to set bin or bmax parameters to the camera by CGI, and at the same time they are using “pixel” internally, thus the parameters in CGI should be calculated as the contrary calculations , at the same time the calculated percentage value should not be floating point number.

As example if the left-top and right-bottom points are (35,28) and (70,58), the video is D1 resolution, the parameters in CGI for the camera should be

(35/702*100, 28/576*100, 70/702*100, 58/576*100) = (5,5,10,10)

mode: Face capture mode, 0: Best face interval time, 1: Best face stay

update: Face capture update frequency, range [250, 4000], unit ms, effective when mode is 0

zoom: face size, magnification, range [1, 10]

ratio: Whether the size of the face snapshot is maintained in the ratio, 0: no, 1 yes

keep: minimum retention time for face capture, [1, 10], unit: frame, effective when mode is 1

delay: maximum delay time for face capture, [5, 10], unit: frame, effective when mode is 1

frevent: face recognition alarm mode, 0: recognition success alarm, 1: recognition failure alarm

similiar: Face recognition similarity threshold, range [50, 100], the greater the similarity, the higher the recognition accuracy

maskconfd: the confidence value for masking detection, [50, 100], the more confidence, the more accurate accuracy for mask wearing detection, but the more low sensitivity.

show.boder: to shown the box of the face detected. 0: disable, 1: enable.

Show.id: to show the ID of the face detected. 0: disable, 1: enable.

Show.name: to show the name of the face recognized, 0: disable, 1: enable;

Show.similar: to show the similarity of the face recognized, 0: disable, 1: enable;

Show.birthday: to show the birthday of the face recognized, 0: disable, 1: enable;

Show.sex: to show the gender of the face recognized, 0: disable, 1: enable;

Show.number: to show the number ID of the face recognized, 0: disable, 1: enable;

Show.custom: to show the customized content of the face recognized, 0: disable, 1: enable;

Show.mask: to show the frame if the face has no mask, 0: disable, 1: enable.

Note:
1. The enable switch can only be obtained but cannot be set. If it is enabled or disabled, use the corresponding event switch of the alarm event parameter.
2. The mode/update/keep/delay/frevent/smiliar/show.name/show.similar/show.birthday/show.sex/show.number /show.custom parameter is only valid in the HISILICON HI3516DV300 face recognition software.
3. The mode/update/keep/delay parameter is valid in the HISILICON HI3516CV500 face detection software, and the frevent/smiliar/show.name/show.similar/parameter is invalid.
4. The mode/update/keep/delay/frevent/smiliar/show.name/show.similar/show.birthday/show.sex /show.number /show.custom parameter is invalid on the HISILICON HI351Av100 / 3516Dv100 / 351616Av200 / 3519V101 platforms.

13.3.2 GET Face Detection Parameters

Request URL	/action/get?subject=facedetect
Request Body	None
Response	<?xml version="1.0" encoding="utf-8"?> <response> <facedetect ver="2.0"> <enable>0</enable> <confidence>50</confidence> <pquality>0</pquality> <pmode>1</pmode> <bmin>0,0,5,5</bmin> <bmax>0,0,50,50</bmax> <update>500</update>

	<zoom>1</zoom> <ratio>1</ratio> <keep>1</keep> <delay>1</delay> <similiar>1</similiar> <frevent>1</frevent> <maskconfd>1</maskconfd> <show> <border>1</border> <id>1</id> <name>1</name> <similiar>1</similiar> <number>1</number> <sex>1</sex> <birthday>1</birthday> <custom>1</custom> <mask>1</mask> </show> </facedetect> </response>

13.3.3 SET Face Detection Parameters

Request URL	/action/set?subject=facedetect
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <facedetect ver="2.0"> <enable>0</enable> <confidence>50</confidence> <pquality>0</pquality> <pmode>1</pmode> <bmin>0,0,5,5</bmin> <bmax>0,0,50,50</bmax> <update>500</update> <zoom>1</zoom>

	<ratio>1</ratio> <keep>1</keep> <delay>1</delay> <similiar>1</similiar> <frevent>1</frevent> <maskconfd>1</maskconfd> <show> <border>1</border> <id>1</id> <name>1</name> <similiar>1</similiar> <number>1</number> <sex>1</sex> <birthday>1</birthday> <custom>1</custom> <mask>1</mask> </show> </facedetect> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

13.4 The interface to get the result of the face comparison

13.4.1 Parameters Description

When 200 is returned successfully, the message body data is binary data of the comparison picture in the face database, and the format is jpg.

Note:
1. id: the ID number of the face, which can be obtained from the /action/ face? subject = facepic interface.
2. When 200 is returned successfully, the FName field is returned in the HTTP message header, which needs base64 decryption. After decryption, the format is FR#regid#name#number#sex#birthday#custom#
among them:
regid: registration information ID for face
name: Face name

number: Face ID number
sex: sex, M: male, F: female
birthday: date of birth, format YYYYMMDD
custom: custom information

3. When 200 is returned successfully, the Simliar field is returned in the HTTP message header, indicating the similarity of this face contrast.

13.4.2 The interface to get the result of the face comparison

Request URL	/action/face?subject=facereco&id=1
Request Body	NONE
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

13.5 The interface to manage the face registration

13.5.1 Parameters Description

The face registration actions include the face adding, the face modify, the face deletion, getting the face registration information and searching the face registration information APIs.

The description of the **FR HTTP request body** which will be used in the below APIs.

1. When requested, the HTTP message body data is binary data of the face contrast picture, and the format is jpg.
2. When requesting, the FName field needs to be added to the HTTP message header. Base64 encryption is required. The format before encryption is "FR#regid#name#number#sex#birthday#custom", # should be among them:
   regid: fixed at 0 when added
   name: Face name
   number: Face ID number
   sex (**gender**): M: male, F: female
   birthday: date of birth, format YYYYMMDD
   custom: custom information

13.5.2 The interface to add the face registration

Request URL	/action/face?subject=facereginfo&do=add
Request Body	FR HTTP request body data
Response	200: Succeed in Set 400: Error of Request

	403: No Right to Set 500: Failed to Set
NOTE	**When 200 returns successfully, the Regid field returned in the HTTP message header indicates the registration ID number after the face was successfully added.** **Note:** **1. Before use, you must first call the / action / face? Subject = facereginfo & do = begin interface;** **2. After use, if there is no subsequent addition / deletion / change operation, the /action/face? Subject = facereginfo & do = end interface must be called;**

13.5.3 The interface to modify the face registration

Request URL	/action/face?subject=facereginfo&do=modify
Request Body	HTTP request body data
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set
NOTE	If no face picture information is uploaded, it indicates that only the name / number and other information have been modified Note: 1. regid, must exist and be legal. 2. Before use, you must first call the /action/face? Subject = facereginfo & do = begin interface. 3. After use, if there is no subsequent addition / deletion / change operation, the / action/face? Subject = facereginfo & do = end interface must be called..

13.5.4 The interface to delete the face registration

Request URL	/action/face?subject=facereginfo&do=del
Request Body	HTTP request body data
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set
NOTE	Note: 1. regid, must exist and be legal.

	2. Before use, you must first call the /action/face? Subject = facereginfo & do = begin interface. 3. After use, if there is no subsequent addition / deletion / change operation, the /action/face? Subject = facereginfo & do = end interface must be called.

13.5.5 The interface to get the face registration

Request URL	/action/face?subject=facereginfo&do=get&regid=1
Request Body	NONE
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set
NOTE	When 200 is returned successfully, the message body data is binary data of the comparison picture in the face database, and the format is jpg. Note: 1. id: Face registration information ID. The face registration information ID can be obtained through a search interface first. 2. When 200 is returned successfully, the FName field is returned in the HTTP message header, which needs base64 decryption. After decryption, the format is FR # regid # name # number # sex # birthday # #bw#custom among them: regid: registration information ID for face name: Face name number: Face ID, M: Male, F: Female sex: sex birthday: date of birth, format YYYYMMDD bw: 0: black 1: white custom: custom information

13.5.6 The interface to search the face registration

Request URL	/action/face?subject=facereginfo&do=search
Request Body	<?xml version="1.0" encoding="utf-8"?> <request>

	<freginfosearch> <name>tester</name> <number>54321</number> <birthday>19000101-20190627</birthday> <sex>0</sex> <custom>0</custom> </freginfosearch> </request>
Response	<?xml version="1.0" encoding="utf-8"?> <response> <regid>1</regid> <regid>2</regid> <regid>3</regid> … <regid>1000</regid> </response>
NOTE	name: the name of the person searching for a face, when empty, it means searching all. number: Search for the ID number of the face. When empty, it means to search all. birthday: Search for the birthday of the face, format: YYYYMMDD-YYYYMMDD, when 0-0, it means search all. sex: search the sex of the face, 0: all, 1: male 2: female. custome: Search for custom information of faces. When empty, it means search all. Note: 1. When searching, all face registration information IDs with full membership conditions will be returned. After getting the information ID, get the specific face registration information through the /action/face? Subject = facereginfo & do = get & regid = xx interface

13.6 The interface to update the face database

13.6.1 Parameters Description

/action/face?subject=facereginfo&do=begin is to start to update the face database, when it is used, the face recognition function won't be valid. The API should be used with the API: */action/face?subject=facereginfo&do=end.*

/action/face?subject=facereginfo&do=end is to stop to update the face database, when it is used, the face recognition function will be recovered to be valid. The API should be used with the API: */action/face?subject=facereginfo&do=begin*

Note

The data will be truly updated into the database until execute API: */action/face?subject=facereginfo&do=end.*

13.6.2 The interface to start to update the face database

Request URL	/action/face?subject=facereginfo&do=begin
Request Body	NONE
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

13.6.3 The interface to stop to the update the face database

Request URL	/action/face?subject=facereginfo&do=end
Request Body	NONE
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

13.7 GET and SET Human Detection

13.7.1 Parameters Description

The related parameters are defined like the below:

enable: enable switch, 0: disable 1: enable

confidence: confidence value, range [0, 100]

scene: scene: 0: indoor 1: outdoor

border: border mark, 0: disabled 1: enabled

id: whether OSD displays id information [0: not displayed 1: displayed]

(HISILICON HI3516Cv500 Person detection only supports enable, confidence, border, showid)

areamask.row: number of rows in the detection area

areamask.col: number of columns in the detection area

areamask.mask: mask information of the detection area

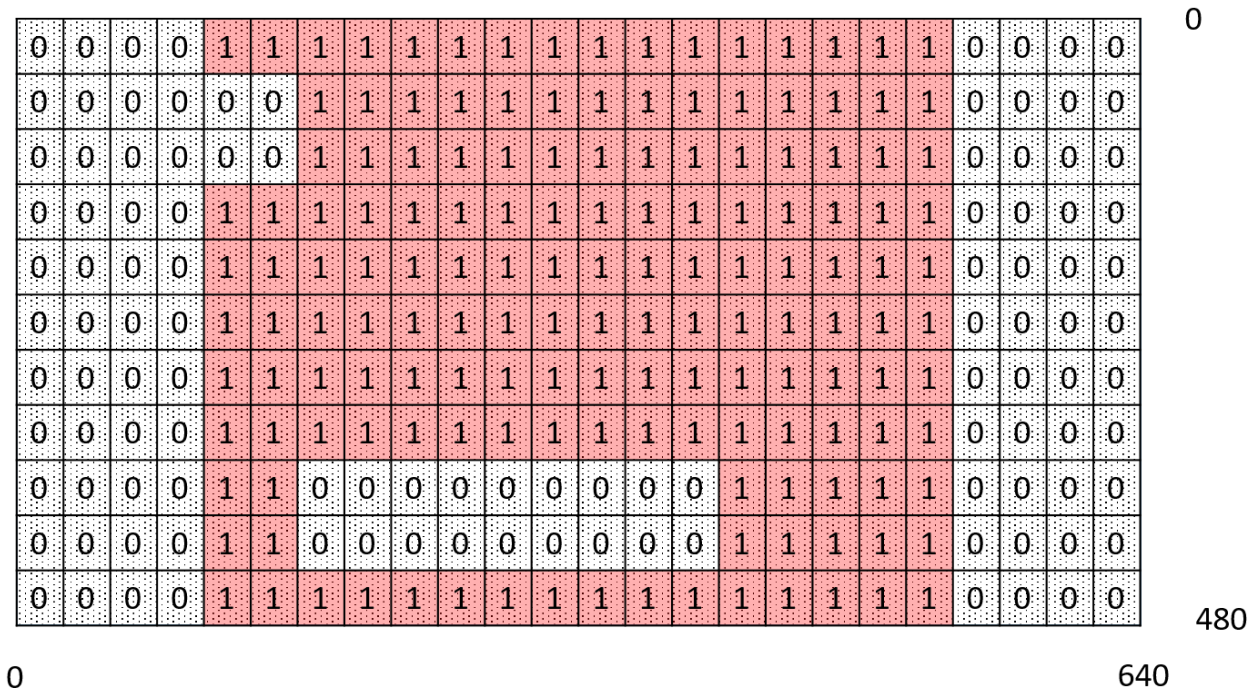
NOTE:

4) Detection Region: the detection region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid detection zone.

5) The value of area.mask identify if the macro zone is valid detection zone, 1: YES, 0: NO. Each bit represents each macro.

6) The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".

Example like the below:



The red colored zone is the valid detection zones, the macro zone was identified: 1;

For the above image, the value will be like the below:

areamask.row : 11

areamask.col: 24

areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0

13.7.2 GET Human Detection Parameters

Request URL	/action/get?subject=persondetect
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <persondetect ver="2.0">

	<enable>0</enable> <confidence>50</confidence> <scene>0</scene> <border>1</border> <areamask> <row>18</row> <col>22</col> <mask>0,0,0,7000,7f8000,2000008,7e000,30200081,40020d80,4c20006,8000000,0,0</mask> </areamask> </persondetect> </response>

13.7.3 SET Human Detection Parameters

Request URL	/action/set?subject=persondetect
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <persondetect ver="2.0"> <enable>0</enable> <confidence>50</confidence> <scene>0</scene> <border>1</border> <areamask> <row>18</row> <col>22</col> <mask>0,0,0,7000,7f8000,2000008,7e000,30200081,40020d80,4c20006,8000000,0,0</mask> </areamask> </persondetect> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

13.8 GET and SET People Counting Parameters

13.8.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable the people counting, 0: disable, 1: enable

ftp_enable: enable or disable FTP uploading the people counting records fle (DBF format).

ftp_trytime: the qty of the trying to upload to the FTP server;

ftpsendtime: the specified FTP upload time, the FTP upload will be executed as this specified time.

line.begin: the beginning coordinate of the boundary of the zone, format: (x, y);

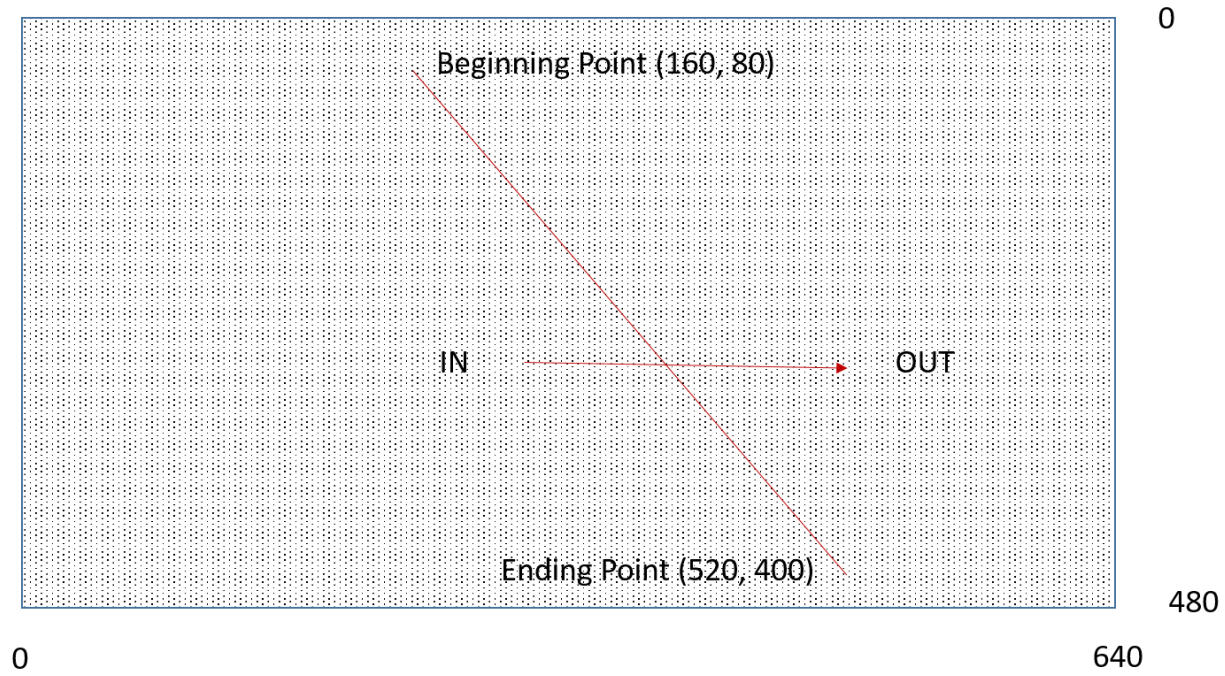
line.end: the ending coordinate of the boundary of the zone, format: (x, y)

Note:

the coordinate value is the relative value which is percentage value of the zone size which need people counting.

For example, if the image of the zone resolution is 640 * 480, and the beginning coordinate was wanted to be set from (320, 240), thus the line.begin value should be (50, 50).

Below is the value of the line.begin and line.end calculated example.



On the above pictures shown, the line.begin value is (25,17), line.end value is (81,83)

13.8.2 GET Parameters of People Counting

Request URL	/action/get?subject=counter
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <counter> <enable>1</enable> <ftp_enable>1</ftp_enable> <ftp_trytime>3</ftp_trytime> <ftp_sendtime>48900</ftp_sendtime> <line> <begin>20,30</begin> <end>60,30</end> </line> </counter> </response>

13.8.3 SET Parameters of People Counting

Request URL	/action/set?subject=counter
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <counter> <enable>1</enable> <ftp_enable>1</ftp_enable> <ftp_trytime>3</ftp_trytime> <ftp_sendtime>48900</ftp_sendtime> <line> <begin>20,30</begin> <end>60,30</end> </line> </counter> </request>

Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

13.8.4 Inquiry the People Counting Information

Request URL	/action/peoplecount?subject=peoplecount_info&start=0000-00-00&end=9999-99-99
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <peoplecount> <peoplecountfile>DB19990101.dbf</peoplecountfile> <snedtime>11:59:00</snedtime> <upstatus>1</upstatus> </peoplecount> <peoplecount> <peoplecountfile>DB19990102.dbf</peoplecountfile> <sendtime>11:59:00</sendtime> <upstatus>1</upstatus> </peoplecount> </response>
	peoplecounrfile: the statistic file name of the people counting sendtime:peoplecoun: the uploading time of the file. upstatus:peoplecoun: the uploading status of the file: 0: success; -1: failed; 1: not upload.

13.8.5 Download the Statistic File of the People Counting

Request URL	/action/peoplecount?subject=peoplecount_file&name=DB19990101.dbf
Request Body	NONE
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set
Note	The parameter “name” is the returned value of one item in peoplecount list from the statistic file of peoplecountfile.

	The CGI returned is the statistic DBF file.
	The API is used to download the statistic file per day from the camera to the computer manually.

13.8.6 Upload the Statistic File of the People Counting

Request URL	/action/peoplecount?subject=peoplecount_ftp&name=DB19990101.dbf
Request Body	NONE
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set
	The parameter "name" is the returned value of one item in peoplecount list from the statistic file of peoplecountfile. A return value of 200 indicates that the upload was successful.
	The API is used to upload the statistic file per day from the camera to the specified FTP server manually or by schedule.

13.9 GET and SET Line Crossing Parameters

13.9.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable the line crossing detection, 0: disable, 1: enable

line.begin: the beginning coordinate of the detected line, format: (x, y);

line.end: the ending coordinate of the detected line, format: (x, y)

sensitivity: the detection sensitivity value, its range is [1, 4]. The bigger of the value, the more sensitive.

direction: the crossing detection direction, 0: A to B; 1: B to A 2: A <->B;

scene: the application scenes, 0: indoor; 1: outdoor;

blink: enable blink or not when there is detection. 0: disable; 1: enable.

Show: enable or disable to show the detection line in the video. 0: disable, 1: enable;

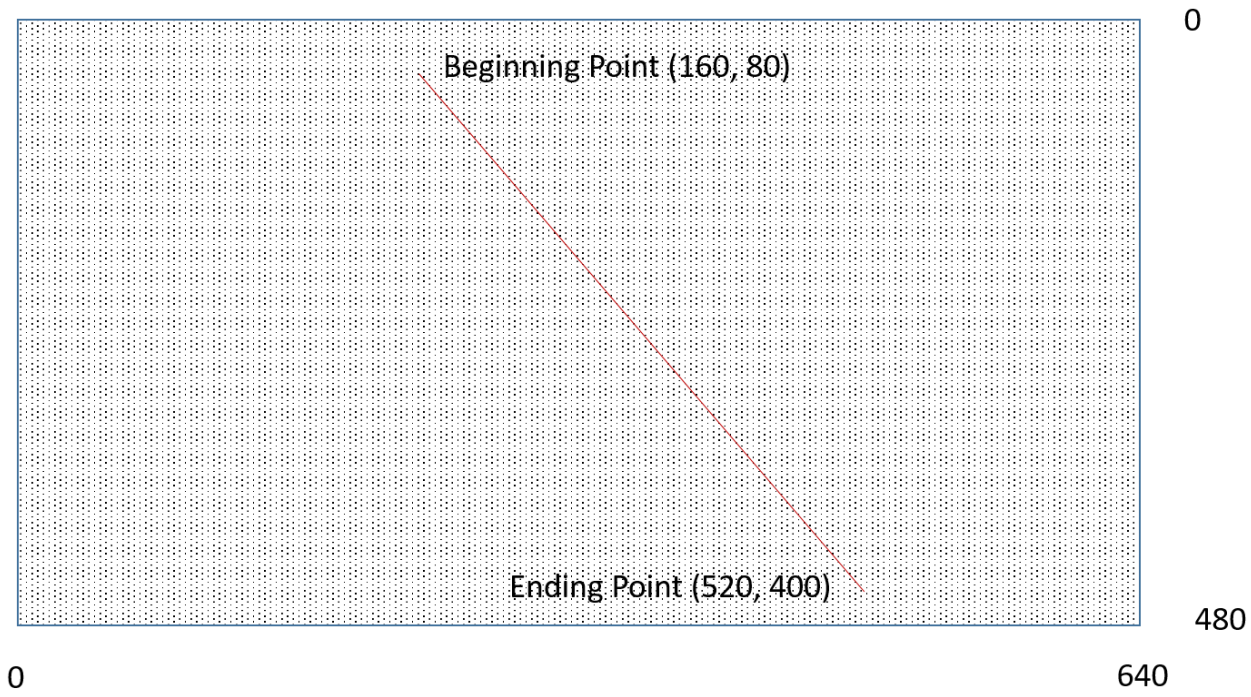
Note:

5) Enable: it can only be gotten, cannot be set. If to set this parameter, please use the API: 4.1 Get and Set Alarm Parameters

6) The coordinate value is the relative value which is percentage value of the crossing line.

7) For example, if the image of the zone resolution is 640 * 480, and the beginning coordinate was wanted to be set from (320, 240), thus the line.begin value should be (50, 50).

8) Below is the value of the line.begin and line.end calculated example.



On the above pictures shown, the line.begin value is (25,17), line.end value is (81,83)

13.9.2 GET Line Crossing Parameters

Request URL	/action/get?subject=crossline
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <crossline> <enable>1</enable> <line> <begin>20,30</begin> <end>60,30</end> </line> <sensitivity>2</sensitivity> <direction>0</direction>

	<scene>0</scene> <blink>1</blink> <show>1</show> </crossline> </response>

13.9.3 SET Line Crossing Parameters

Request URL	/action/set?subject=vaglobal
Request Body	<?xml version="1.0" encoding="utf-8"?> <response> <crossline> <enable>1</enable> <line> <begin>20,30</begin> <end>60,30</end> </line> <sensitivity>2</sensitivity> <direction>0</direction> <scene>0</scene> <blink>1</blink> <show>1</show> </crossline> </response>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

13.10 GET and SET Intrusion Detection Parameters

13.10.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable intrusion detection, 0: disable, 1: enable

scene: the application scenes, 0: indoor; 1: outdoor;

sensitivity: the detection sensitivity value, its range is [0, 4]. The bigger of the value, the more sensitive.

direction: intrusion detection direction. 0: enter; 1: Leave; 2: both;

blink: enable blink or not when there is detection. 0: disable; 1: enable.

Show: enable or disable to show the detection frame. 0: disable, 1: enable;

areamask.row: the value of the raw of the macro zone in intrusion detection region.

areamask.col: the value of the column of the macro zone in intrusion detection region.

Areamask.mask: the mask value of the macro zone in the intrusion detection region.

polygon.point: The four vertex coordinates of the invading area are in turn the upper left, the upper right, the lower right, the lower left, and the percentage of coordinates in units.

NOTE:

Enable: it can only be gotten, cannot be set. If to set this parameter, please use the api: 4.1 Get and Set Alarm Parameters.

5) Intrusion Region: the intrusion region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid intrusion zone.

6) The value of area.mask identify if the macro zone is valid intrusion zone, 1: YES, 0: NO. Each bit represents each macro.

7) The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".

8) Intrusion area can be described in two ways, one is areamsk, the other is polygon. Different models may be described in different ways. Please take the actual way as the standard. For one type of intrusion detection, the region can only be described in one way, but not in both ways.

Example: Area mask:

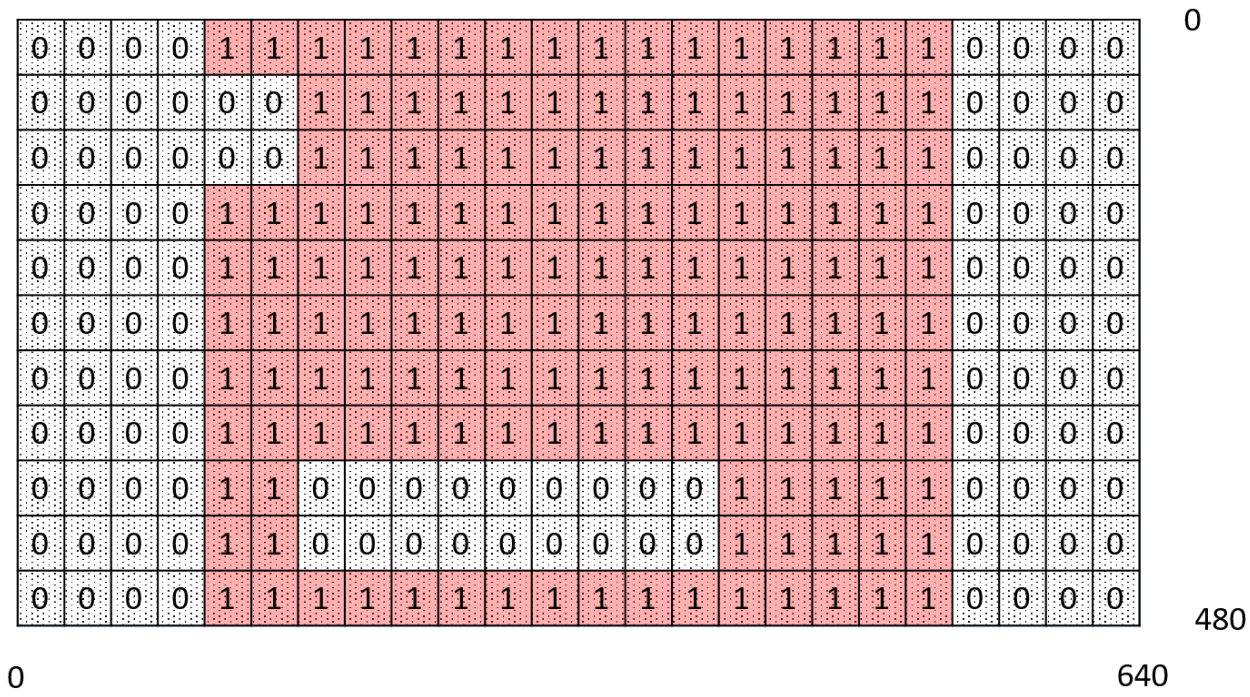
The red colored zone is the valid intrusion zones, the macro zone was identified: 1;

For the above image, the value will be like the below:

areamask.row : 11

areamask.col: 24

areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0



13.10.2 GET Intrusion Parameters

Request URL	/action/get?subject=intrusion
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <instrusion> <enable>1</enable> <scene>0</scene> <sensitivity>2</sensitivity> <direction>0</direction> <blink>1</blink> <show>1</show> <areamask> <row>16</row> <col>12</col> <mask>ffffffff,ffffffff,ffffffff</mask> </areamask> <polygon>

	<point>26,24</point> <point>60,33</point> <point>65,63</point> <point>42,69</point> </polygon> </instrusion> </response>

13.10.3 SET Intrusion Parameters

Request URL	/action/set?subject=intrusion
Request Body	<?xml version="1.0" encoding="utf-8"?> <response> <instrusion> <enable>1</enable> <scene>0</scene> <sensitivity>2</sensitivity> <direction>0</direction> <blink>1</blink> <show>1</show> <areamask> <row>16</row> <col>12</col> <mask>ffffffff,ffffffff,ffffffff</mask> </areamask> <polygon> <point>26,24</point> <point>60,33</point> <point>65,63</point> <point>42,69</point> </polygon> </instrusion> </response>
Response	200: Succeed in Set 400: Error of Request

	403: No Right to Set 500: Failed to Set

13.11 GET and SET Loitering Detection

13.11.1 Parameters Description

The related parameters are defined like the below:

enable: enable or disable loitering detection, 0: disable, 1: enable

second: the minimum loitering time of the object stayed in the detection zone. Unit: second, the valid value: (5,10,15)

areamask.row: the value of the raw of the macro zone in detection region.

areamask.col: the value of the column of the macro zone in detection region.

areamask.mask: the mask value of the macro zone in the detection region.

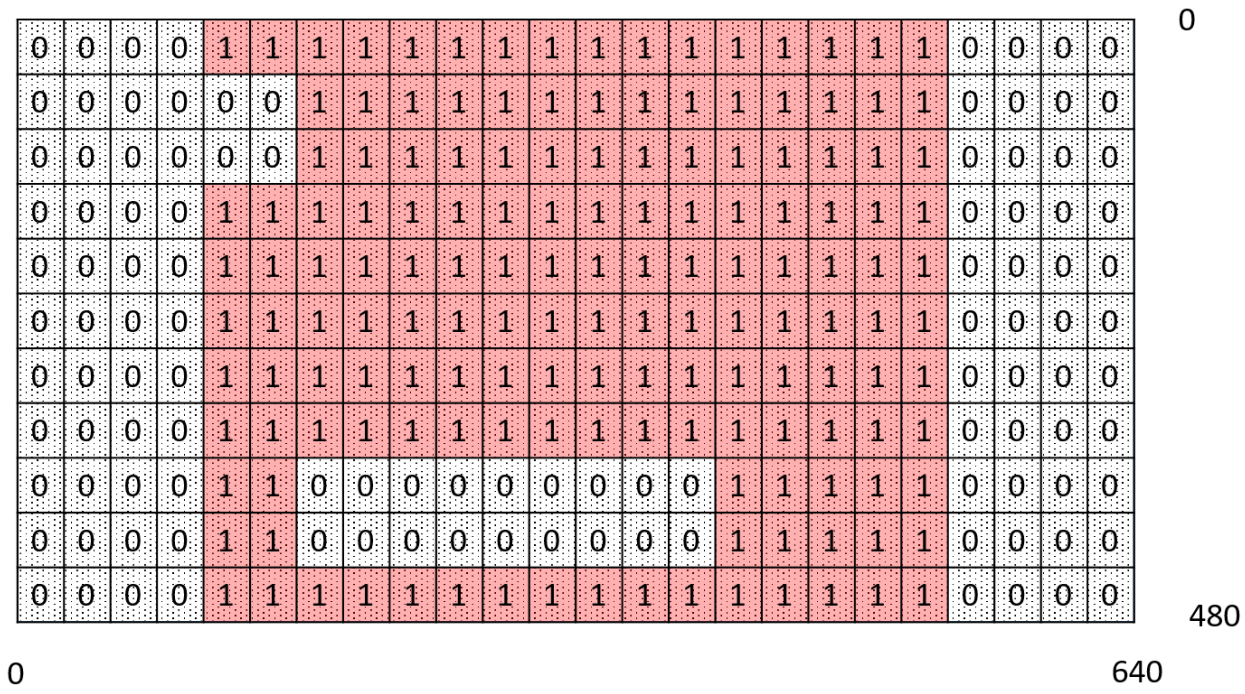
NOTE:

5) Detection Region: the detection region is full size of the image which can be divided into several (row * col) detection macro zones, each macro zone can be set separately to be valid or invalid detection zone.

6) The value of area.mask identify if the macro zone is valid detection zone, 1: YES, 0: NO. Each bit represents each macro.

7) The format of areamask.mask is several groups which was built by every 4 bytes which is shown by hexadecimal strings. If less than 4 bytes, 0 will be alignment. Each group should be separated by ",".

8) Example like the below:



The red colored zone is the valid detection zones, the macro zone was identified: 1;

For the above image, the value will be like the below:

areamask.row : 11

areamask.col: 24

areamask.mask: 0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0

5) enable, can only be obtained, cannot be set, if enabled or disabled, please use the corresponding event switch of the alarm event parameter

13.11.2 GET Loitering Detection

Request URL	/action/get?subject=loitering
Request Body	NONE
Response	<?xml version="1.0" encoding="utf-8"?> <response> <loitering> <enable>1</enable> <second>5</second> <areamask> <row>11</row> <col>24</col>

	<mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask> </areamask> </loitering> </response>

13.11.3 SET Loitering Detection

Request URL	/action/set?subject=loitering
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <loitering> <enable>1</enable> <second>5</second> <areamask> <row>11</row> <col>24</col> <mask>0ffff0,03fff0,03fff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0ffff0,0c01f0,0c01f0,0ffff0</mask> </areamask> </loitering> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

13.12 The interface to authorize the license to the device

13.12.1 Parameters Description

/action/download?file=vacertificate is used for downloading the device information certificates which will be used for applying the license files.

/action/uupload?file=valicense is used to upload the applied license files.

/action/upload?file=frmodel is used for uploading and refreshing the model file related with face recognition.

13.12.2 The interface to download the device' certificate file

Request URL	/action/download?file=vacertificate
Request Body	NONE
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

13.12.3 The interface to upload the license file to the device

Request URL	/action/upload?file=valicense
Request Body	The authorized license files
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

13.12.4 The interface to upload the model file of FR to the device

Request URL	/action/upload?file=frmodel
Request Body	The model file of the face recognitiion
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

❑ ANPR part

14 ANPR/LPR APIs

14.1 The ANPR/LPR APIs

14.1.1 The interface to upload the LPR model file

Request URL	/action/upload?file=lprmodel
Request Body	The mode file
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

14.1.2 The interface to get and set the parameters of the LPR

Request URL	/action/get?subject=lpr
Request Body	None
Response	<?xml version="1.0" encoding="utf-8"?> <response> <lpr ver="2.0"> <lprenable>1</lprenable> <lprsnapenable>1</lprsnapenable> <lprevent>1</lprevent> <lprrealtime>1</lprrealtime> <lprleft>5,5,40,90</lprleft> <lprright>50,5,90,90</lprright> </lpr> </response>
Request URL	/action/set?subject=lpr
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <lpr> <lprenable>1</lprvenable> <lprsnapenable>1</lprsnapenable>

	<lprevent>1</lprvevent> <lprrealtime>1</lprrealtime> <lprleft>5,5,40,90</lprleft> <lprright>50,5,90,90</lprright> </lpr> </request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

Note:

Lprenable: to enable LPR. 1: enable, 0: disable

lprsnapenable: to enable the snapshot. 1: enable 0: disable

lprevent: the working of the White or Black lit. 0: Black list; 1: White list.

Lprrealtime: to enable the realtime analysis of the license plate detection. 1: enable; 0: disable.

lprleft: the lleft area detection coordinates.

Lprright: the right area detection coordinates.

The coordinates are defined as the format: (left, top, right, bottom), it's the percentage.

14.1.3 The interface to get the snapshot of the plate with the recognized information

Request URL	/action/lpr?subject=lprinfo&do=pic&last=-1
Request Body	NONE
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

Note:

If 200 is returned, the returned data is the captured license plate picture.

The head of the returned data includes the following segment:

Fseqno: Each time an event is fetched, the last value is set in the request url, the current request last value is the last valid Fseqno value returned, and can be set to -1 on the first request.

There are 2 options for the license detection modes, IV mode and JIC mode, as default, we used IV mode.

When the license plate is IV mode, FName: lpr_ license plate number [string]_YYYYMMDDTHHMMSS.jpg

When the license plate is JIC mode, FName : lpr_ license plate number [string]_YYYYMMDDTHHMMSS_ body color [string]_speed [integer]_direction [integer].jpg

14.1.4 The interface to get synchronous frame picture

Request URL	/action/lpr?subject=lprinfo&do=big&No=5
Request Body	None
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

Note:

When 200 is successful, the returned data is the frame picture when making the snapshot of the license plate.

“No” is the “last” value in the returned detected plate info.

14.1.5 The interface to get the snapshot of the full screen

Request URL	/action/lpr?subject=lprinfo&do=full&last=-1
Request Body	None
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

Note:

When 200 is successful, the returned data is the full frame of the license plate.

There are special fields in the return header field:

Fseqno: Each time an event is fetched, the last value is set in the request url, the current

request last value is the last valid Fseqno value returned, and can be set to -1 on the first request.

14.1.6 The interface to download the White and Black list of the license

Request URL	/action/lpr?subject=lprbwlist&do=download
Request Body	None
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

Note:

The license plate black and white list file will be downloaded.

14.1.7 The interface to upload the White and Black list of the license

Request URL	/action/upload?file=lprbwlist
Request Body	None
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

Note:

Upload a black and white list file used for license plate recognition

Black and white list file description:

1.@ initial behavior comment line

2. The format of each line is the license plate number n* space 0/1;

0 means blacklist 1 means whitelist

14.1.8 The interface to add the license to the white and black list

Request URL	/action/lpr?subject=lprbwlist&do=add&plate=403&attr=0&starttime=2019-01-01T00:00:00&endtime=2020-01-01T00:00:00
Request Body	None
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

Note:

The API is used to add a license plate information to the black and white list

plate: plate number

attr: the attribution of the list, 0: Black list, 1: white list.

starttime: effective start time of the added black and white list.

endtime: effective end time of the added black and white list.

14.1.9 The interface to delete the license to the white and black list

Request URL	/action/lpr?subject=lprbwlist&do=del&plate=403&attr=0
Request Body	None
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

Note:

The API is used to delete a license plate information to the black and white list

plate: plate number

attr: the attribution of the list, 0: Black list, 1: white list.

14.1.10 The interface to get the total number of the White and Blact list of the license

Request URL	/action/lpr?subject=lprbwlist&do=num&mode=0
Request Body	None
Response	<?xml version="1.0" encoding="utf-8"?> <response> <LprNoNum>0</LprNoNum> </response>

Note:

the mode: 0: blacklist, 1: whitelist, 2: two types

LprNoNum in the return value indicates the number of corresponding type lists.

14.1.11 The interface to get the White and Black list

Request URL	/action/lpr?subject=lprbwlist&do=get&start=1&end=100&mode=0
Request Body	None
Response	<?xml version="1.0" encoding="utf-8"?> <response> <lprbwlist> <lprNo>400</lprNo> <lprstart>40000101T000000</lprstart> <lprend>40990101T000000</lprend> <lprGroup>0</lprGroup> </lprbwlist> <lprbwlist> <lprNo>401</lprNo> <lprstart>40000101T000000</lprstart> <lprend>40990101T000000</lprend> <lprGroup>0</lprGroup> </lprbwlist> <lprbwlist> <lprNo>402</lprNo> <lprstart>40000101T000000</lprstart>

	<lprend>40990101T000000</lprend> <lprGroup>0</lprGroup> </lprbwlist> </response>

Note:

The API is used to get the black and white lists of license plates.

Mode: 0: whitelist, 1: blacklist, 2: two types

Start: the starting position, end refers to the ending position.

lprNo: license plate number

Lprtm: time when the black and white list was added

(When mode is 2, the returned ivbwlist will additionally contain lprGroup: indicating black and white attributes)

14.1.12 The interface to get the country list

Request URL	/action/lpr?subject=lprcountry&do=list
Request Body	None
Response	<?xml version="1.0" encoding="utf-8"?> <response> <lprcountry> <lprcountrystr>IND</lprcountrystr> <lprcountrystr>USA</lprcountrystr> <lprcountrystr>TWN</lprcountrystr> <lprcountrystr>RUS</lprcountrystr> <lprcountrystr>CHN</lprcountrystr> <lprcountrystr>ZAF</lprcountrystr> <lprcountrystr>TUR</lprcountrystr> <lprcountrystr>EUR</lprcountrystr> <lprcountrystr>BLR</lprcountrystr> <lprcountrystr>CAN</lprcountrystr> <lprcountrystr>AUS</lprcountrystr> <lprcountrystr>LKA</lprcountrystr>

	<lprcountrystr>KOR</lprcountrystr> <lprcountrystr>BRA</lprcountrystr> <lprcountrystr>GBR</lprcountrystr> <lprcountrystr>DEU</lprcountrystr> <lprcountrystr>MEX</lprcountrystr> <lprcountrystr>VNM</lprcountrystr> <lprcountrystr>FRA</lprcountrystr> </lprcountry> </response>

Note:

The current supported all countries returned.

{
IND
USA
TWN
RUS
CHN
ZAF
TUR
EUR
BLR
CAN
AUS
LKA
KOR
BRA
GBR
DEU
MEX
VNM
FRA
}

The meaning of the above:

IND India,

USA USA,

TWN Taiwan,

RUS Russia,

CHN China,

ZAF Africa,

TUR Turkey,

EUR Europe

BLR Belarus

CAN Canada,

AUS Australia,

LKA Srilanka,

KOR SouthKorea,

BRA Brazil,

GBR GreatBritain

DEU Germany,

MEX Mexico,

VNM Vietnam,

FRA France,

14.1.13 The interface to get the country

Request URL	/action/lpr?subject=lprcountry&do=get
Request Body	None
Response	<?xml version="1.0" encoding="utf-8"?> <response> <lprcountry>USA</lprcountry> </response>

14.1.14 The interface to set the working country

Request URL	/action/lpr?subject=lprcountry&do=set&lprcountry=USA
Request Body	None
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

14.1.15 The interface to search the archived LPR list

Request URL	/action/lpr?subject=lprinfo&do=search&start=2019-12-05T00:00:00& end=2019-12-05T23:59:59&speed=0&plate=&direction=0&startpos=0&endpos=10
Request Body	None
Response	<?xml version="1.0" encoding="utf-8"?> <response> <lprrecord> <lprtm>2019-12-05T16:35:24</lprtm> <index>0</index> <plate>745WHN</plate> <speed>0</speed> <direction>0</direction> <color></color> </lprrecord> <lprrecord> <lprtm>2019-12-05T16:35:25</lprtm> <index>0</index> <plate>9451B</plate> <speed>0</speed> <direction>0</direction> <color></color> </lprrecord> </response>

Note:

The explanation for the return values:

Searching conditions:

start: searching by starting time

end: searching by ending time

(Current only one day searching is supported, so the searching can be executed in cycle if to search several days. The reason is the quantity of the searched plate is very huge per day.)

speed: searching by car's speed. Value range:1-500

plate: searching by car's plate number, the maximum is 16 characters.

direction: searching by car's driving direction, Value: 1 or 2;

startpos: the beginning point of the returned result. It's started from 0.

endpos: the ending point of the returned result

（the range of the "startpos to endpos" is the same as the WB list searched, as example if ti's 5 pcs items, startpos=0 endpos=4）

The time range is must, speed, plate and direction are options, if not filled, speed=0, direction=0, and plate is empty. URL is speed=0&plate=&direction=0

Explanation for the returned result:

lprtm: the detection time for the plate

index: the index to get the detailed picture

plate: plate number

speed: speed of the car

direction: the driving direction of the car

color: the car's color

14.1.16 The interface to search the qty of the archived LPR list

Request URL	/action/lpr?subject=lprinfo&do=hitnum&start=2019-12-09T00:00:00&end=2019-12-09T23:59:59& speed=0&plate=&direction=0
Request Body	None
Response	<?xml version="1.0" encoding="utf-8"?> <response> <lprrecord_hitnum> <hitnum>176</hitnum> </lprrecord_hitnum> </response>

Note:

The explanation for the return values:

Hitnum: Return the number of qualified license plates

14.1.17 The interface to search the archived plate picture

Request URL	/action/lpr?subject=lprinfo&do=pull&time=2019-12-05T16:35:28&index=0

Request Body	None
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

Note:

The explanation for the return values:

time: Time of license plate recognition returned from search list

index: License plate index bit returned from search list

14.1.18 The interface to get and set DBF file of the captured info

Request URL	/action/get?subject=lpr_dbf
Request Body	None
Response	<?xml version="1.0" encoding="utf-8"?> <response> <lpr_dbf ver="2.0"> <enable>1</enable> <sendtime>59960</sendtime> <trytime>5</trytime> </lpr_dbf> </response>

Request URL	/action/set?subject=lpr_dbf
Request Body	<?xml version="1.0" encoding="utf-8"?> <request> <lpr_dbf ver="2.0"> <enable>1</enable> <sendtime>61040</sendtime> <trytime>5</trytime> </lpr_dbf>

	</request>
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set

enable: if enable the DBF upload;

sendtime：the upload time, the second unit of the time of the day.
trytime: the maximum upload times for trying.

14.1.19 The interface to get the DBF file list

Request URL	/action/lpr?subject=lprdbf&do=get&start=0001-00-00&end=2222-00-00
Request Body	None
Response	<?xml version="1.0" encoding="utf-8"?> <response> <lpr_dbf> <lpr_dbf_file>DB20200617.dbf</lpr_dbf_file> <sendtime>17:31:20</sendtime> <upstatus>1</upstatus> </lpr_dbf> </response>

Lpr_dfb_file: the file name.

sendtime：the time of the file uploading.
upstatus: the uploaded status, 0: not uploaded, 1: uploaded successfully, -1: failed to upload.

14.1.20 The interface to download the DBF file list

Request URL	/action/lpr?subject=lprdbf&do=download&name=DB20200617.dbf
Request Body	None
Response	200: Succeed in Set 400: Error of Request

	403: No Right to Set 500: Failed to Set When returned 200: success, the HTTP response body is the DBF file.

14.1.21 The interface to upload the DBF file list

Request URL	/action/lpr?subject=lprdbf&do=upload&name=DB20200617.dbf
Request Body	None
Response	200: Succeed in Set 400: Error of Request 403: No Right to Set 500: Failed to Set When returned 200, the name is the returned file name.

----------**END**----------