API of HTTP Protocol Specification

V2.76

2019-07-25

Cybersecurity Recommendations

Mandatory actions to be taken towards cybersecurity

1. Change Passwords and Use Strong Passwords:

The number one reason systems get "hacked" is due to having weak or default passwords. It is recommended to change default passwords immediately and choose a strong password whenever possible. A strong password should be made up of at least 8 characters and a combination of special characters, numbers, and upper and lower case letters.

2. Update Firmware

As is standard procedure in the tech-industry, we recommend keeping NVR, DVR, and IP camera firmware up-to-date to ensure the system is current with the latest security patches and fixes.

"Nice to have" recommendations to improve your network security

1. Change Passwords Regularly

Regularly change the credentials to your devices to help ensure that only authorized users are able to access the system.

2. Change Default HTTP and TCP Ports:

● Change default HTTP and TCP ports for systems. These are the two ports used to communicate and to view video feeds remotely.

● These ports can be changed to any set of numbers between 1025-65535. Avoiding the default ports reduces the risk of outsiders being able to guess which ports you are using.

3. Enable HTTPS/SSL:

Set up an SSL Certificate to enable HTTPS. This will encrypt all communication between your devices and recorder.

4. Enable IP Filter:

Enabling your IP filter will prevent everyone, except those with specified IP addresses, from accessing the system.

5. Change ONVIF Password:

On older IP Camera firmware, the ONVIF password does not change when you change the system's credentials. You will need to either update the camera's firmware to the latest revision or manually change the ONVIF password.

6. Forward Only Ports You Need:

● Only forward the HTTP and TCP ports that you need to use. Do not forward a huge range of numbers to the device. Do not DMZ the device's IP address.

● You do not need to forward any ports for individual cameras if they are all connected to a recorder on site; just the NVR is needed.

7. Limit Features of Guest Accounts:

If your system is set up for multiple users, ensure that each user only has rights to features and functions they need to use to perform their job.

8. UPnP:

● UPnP will automatically try to forward ports in your router or modem. Normally this would be a good thing. However, if your system automatically forwards the ports and you leave the credentials defaulted, you may end up with unwanted visitors.

● If you manually forwarded the HTTP and TCP ports in your router/modem, this feature should be turned off regardless. Disabling UPnP is recommended when the function is not used in real applications.

9. SNMP:

Disable SNMP if you are not using it. If you are using SNMP, you should do so only temporarily, for tracing and testing purposes only.

10. Multicast:

Multicast is used to share video streams between two recorders. Currently there are no known issues involving Multicast, but if you are not using this feature, deactivation can enhance your network security.

11. Check the Log:

If you suspect that someone has gained unauthorized access to your system, you can check the system log. The system log will show you which IP addresses were used to login to your system and what was accessed.

12. Physically Lock Down the Device:

Ideally, you want to prevent any unauthorized physical access to your system. The best way to achieve this is to install the recorder in a lockbox, locking server rack, or in a room that is behind a lock and key.

13. Connect IP Cameras to the PoE Ports on the Back of an NVR:

Cameras connected to the PoE ports on the back of an NVR are isolated from the outside world and cannot be accessed directly.

14. Isolate NVR and IP Camera Network

The network your NVR and IP camera resides on should not be the same network as your public computer network. This will prevent any visitors or unwanted guests from getting access to the same network the security system needs in order to function properly.

Purpose

Welcome to use API of HTTP protocol specification.

Reader

- API software development engineers
- Project managers
- Product managers

Safety Instructions

The following categorized signal words with defined meaning might appear in the Manual.

Signal Words	Meaning
DANGER	Indicates a high potential hazard which, if not avoided, will result in death or serious injury.
WARNING	Indicates a medium or low potential hazard which, if not avoided, could result in slight or moderate injury.
CAUTION	Indicates a potential risk which, if not avoided, could result in property damage, data loss, lower performance, or unpredictable result.
TIPS	Provides methods to help you solve a problem or save you time.
NOTE	Provides additional information as the emphasis and supplement to the text.

Revision History

No.	Version	Revision Content	Release Time
1	V2.40	Based on the old version.	August 8, 2018
2	V2.60	Add find media file with TrafficCar info.	September 12, 2018
3	V2.61	Add the "16 AI APIs " chapter.	October 24, 2018
4	V2.62	Add find media files with NonMotor info	November 06, 2018
5	V2.63	Add close door and access control custom password	November 27, 2018
6	V2.64	Add access control and video talk settings.	December 12, 2018

No.	Version	Revision Content	Release Time
7	V2.65	Add intelligent caps and resource usage info	January 17, 2019
8	V2.66	Add people heat map, realtime trace, history trace info	February 13, 2019
9	V2.67	Add subscribe heat map raw data	April 18, 2019
10	V2.68	Add get traffic snap event info record	May 15, 2019
11	V2.69	Add export traffic blacklist/redlist record	May 20, 2019
12	V2.70	Add privacy masking	May 29, 2019
13	V2.71	Add get the max and min temperature values	June 3, 2019
14	V2.72	Add smart motion detection config	June 27, 2019
15	V2.73	Add ecrypted download media file	July 16, 2019
16	V2.74	Adjust Camera Image, Exposure, Backlight, White Balance, Day-Night, Zoom and Focus, Lighting, Video in Options config.	July 25, 2019

Privacy Protection Notice

As the device user or data controller, you might collect personal data of other such as face, fingerprints, car plate number, Email address, phone number, GPS and so on. You need to be in compliance with the local privacy protection laws and regulations to protect the legitimate rights and interests of other people by implementing measures include but not limited to: providing clear and visible identification to inform data subject the existence of surveillance area and providing related contact.

About the Manual

- The Manual is for reference only. If there is inconsistency between the Manual and the actual product, the actual product shall prevail.
- We are not liable for any loss caused by the operations that do not comply with the Manual.
- The Manual would be updated according to the latest laws and regulations of related regions. For detailed information, see the paper manual, CD-ROM, QR code or our official website. If there is inconsistency between paper manual and the electronic version, the electronic version shall prevail.
- All the designs and software are subject to change without prior written notice. The product updates might cause some differences between the actual product and the Manual. Please contact the customer service for the latest program and supplementary documentation.
- There still might be deviation in technical data, functions and operations description, or errors in print. If there is any doubt or dispute, please refer to our final explanation.
- Upgrade the reader software or try other mainstream reader software if the Manual (in PDF format) cannot be opened.
- All trademarks, registered trademarks and the company names in the Manual are the properties of their respective owners.
- Please visit our website, contact the supplier or customer service if there is any problem occurred when using the device.

- If there is any uncertainty or controversy, please refer to our final explanation.

Table of Contents

1 Overview

This document specifies the HTTP-based application programming interface of video products.

The HTTP-based interface provides the functionality for requesting snapshot and media stream, controlling camera functions (for example, PTZ and focus), and getting and setting internal parameter values.

The video products serve as a server. The client sends requests to server, and then server handles requests and returns resources accordingly.

2 References

[1].RFC 2616 Hypertext Transfer Protocol-HTTP/1.1

[2].RFC 2396 Uniform Resource Identifiers (URI): Generic Syntax and Semantics

[3].RFC 2617 HTTP Authentication: Basic and Digest Access Authentication

[4].RFC 3986: Uniform Resource Identifiers (URI) Generic Syntax

3 Definitions

3.1 Abbreviations

The following abbreviations are used throughout this document.

API	Application Programming Interface. In the document, it especially presents application programming interface of video products.

3.2 Syntax Convention

- In URL syntax and in descriptions of API parameters, text in italic within angle brackets denotes content that should be replaced with either a value or a string. When replacing the text string, the angle brackets must also be replaced. For example, <*server*> in URL syntax is replaced with the IP number of server, e.g., 192.168.1.108.
- String shown in bold face denotes a brief explanatory note of the string close to it.
- Name-value pair in square brackets denotes content that is optional. For example, "http ://<*server*>/cgi-bin/snapshot.cgi[?channel=1]" can be like this "http ://<*server*>/cgi-bin/snapshot.cgi".
- The API syntax must follow the standard of URI. (RFC 3986: Uniform Resource Identifiers (URI) Generic Syntax); that is, spaces and other reserved characters (e.g, ":", "/", "?", "@", ";", "=", "+", ",", "$", "&") within a name-value pair should be replaced with %< ASCII hex>. For example, the blank should be replaced with %20.
- To describe the range of a variable, we use some symbols such as "[]" and "{}". For example, "[0-100]" denotes an integer not less than 0 and not larger than 100. "{0, 1, 2, 3}" denotes the valid value of an integer among 0, 1, 2 and 3.
- "[]" following a string denotes an array. The index is an integer and starts from 0. For example, "Snap[*channel*] " may be "Snap[0] ".
- The variable may be different types: string, integer, bool or float. Integer is 32 bits. The range of bool is "true" and "false."
- "R/O"in parameters means this parameter is required or not, "R"means required, "O"means optional.

3.3 API Request Format

This section defines the syntax and semantics for APIs.

<**protocol**> ://<**server**><**abs_path**> [?**query**]

protocol: URL scheme for the particular request. The http and https protocols are both supported in this specification. So "http", as most of the APIs' default protocol except several RTSP APIs, can be replaced by "https".

server: Server could be "**hostname[: port]** ". The **hostname** can be IP address or the fully qualified domain name of an IP device. The **port** is the port number of **server** listening for TCP connections. If the port is not given, the default port is assumed. For HTTP, the default port is 80. For HTTPS, the default port is 443.

abs_path: The Request-URI for the resources is abs_path. The abs_path in this specification is most often of the form "/cgi-bin/*.cgi".

query: The query field is a string of information to be interpreted by the resource. It consists of resource-related parameters. And it must be listed in name-value pair syntax (p1=v1&p2=v2&…&pn=vn).

For example: http://192.168.1.108/cgi-bin/snapshot.cgi?channel=1

3.4 Server Response Format

The server uses the standard HTTP status codes.

Return:

HTTP/1.1 <HTTP code> <HTTP text>\r\n

With the following HTTP code and meanings

Table 3-1

HTTP code	HTTP text	Description
200	OK	The request has succeeded. The requested resource will be returned in the HTTP text.
400	Bad Request	The request had bad syntax or was inherently impossible to be satisfied.
401	Unauthorized	The request requires user authentication or the authorization has been refused.
403	Forbidden	The user does not have the right to access the service.
404	Not Found	The server has not found anything matching the request.
500	Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request.
501	Not Implemented	The server has not implemented the service.

If the HTTP code is 200, means the API execute success, and the response data in http body (maybe multipart) can be a multiline key=value data, or a json object, or just a line with a word "OK".

Example: success respose with multiline key=value

HTTP/1.1 200 OK Server: xxx Content-Length: <*length*> status.Focus=0.5 status.Zoom=0.5 …

Example: success respose with a word "OK"

HTTP/1.1 200 OK Server: xxx Content-Length: <*length*> OK

If the HTTP code is not 200, means the API execute failed, and the response data in http body maybe empty, or just two line, first line is a word "Error"to indicate error happened, the second line contain error detail.

Example: request does not fit with syntax.

```
HTTP/1.1 404 Not Found
Server: xxxx
```

Example: Request spells wrong.

```
HTTP/1.1 400 Bad Request
Server: xxx
Content-Length: <length>

Error
Bad Request!
```

Example: If the request fits with syntax but an error occurs while the server handles it, the response would like this:

```
HTTP/1.1 500 Internal Server Error
Server: xxx
Content-Length: <length>

Error
Internal Server Error!
```

3.5 Authentication

Video products support either basic authentication or digest authentication, see RFC 2617 for detail. If the http request sent by client does not provide valid "Authorization" header information, video products would return HTTP status code 401 and some information for authentication, then client should calculate authentication information according RFC 2617, and sent request again with authentication information using “Authorization” header. Video products return the required resource only if authorization information correct.

For example:

1. When basic authentication fails, response is:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Basic realm="XXXXXX"
```

The client encodes the username and password with base64, and then sends it to server. A valid Authorization like this:

```
Authorization: Basic VXZVXZ
```

2. When digest authentication fails, response is:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest realm="DH_00408CA5EA04",
nonce="000562fdY631973ef04f77a3ede7c1832ff48720ef95ad", stale=FALSE, qop="auth"
```

The client calculates the digest authorization using information like username, password, nonce, HTTP method and URI with MD5, and then sends it to server.

For example:

```
Authorization: Digest username="admin", realm="DH_00408CA5EA04", nc=00000001,
cnonce="0a4f113b", qop="auth",
nonce="000562fdY631973ef04f77a3ede7c1832ff48720ef95ad",
uri="/cgi-bin/magicBox.cgi?action=getLanguageCaps",
response="65002de02df697e946b750590b44f8bf"
```

4 General APIs

The requests specified in this section are supported by all video products.

4.1 APIs of RTSP

4.1.1 Get Real-Time Stream

Table 4-1

Syntax	rtsp://<***username***>:<***password***>@<***ip***>:<***port***>/cam/realmonitor?channel=<***ChannelNo***>&subtype=<***typeNo***>
Description	Get real-time media stream.
Example	We request the extra stream 1 of channel 1, the URL is: rtsp://admin:admin@10.7.6.67:554/cam/realmonitor?channel=1&subtype=1
Success Return	media stream data
Comment	<***username***>: a valid user name. <***password***>: user's password. < ***ip***>: the IP address of the video product. <***port*** >: the default port is 554. It can be omitted. <***ChannelNo***>: integer, video channel index which starts from 1. <***typeNo***>: the stream type. The <*typeNo*> of main stream is 0, extra stream 1 is 1, extra stream 2 is 2. The extra stream counts can be obtained in "4.5.1 Get Max Extra Stream Numbers". If the stream does not exist or not enabled, response will be error. The IP Camera supports both TCP and UDP transmission forms. It also supplies basic authentication and digest authentication ways. The authentication process is similar with "3.5 Authentication".

4.1.2 Get Playback Stream

Table 4-2

Syntax	rtsp://<***username***>:<***password***>@<***ip***>:<***port***>/cam/playback?channel=<***ChannelNo***>&starttime=<***starttime***>&endtime=<***endtime***>
Description	Get playback media stream.
Example	rtsp://admin:admin@10.44.200.8:554/cam/playback?channel=1&starttime=2012_09_15_12_37_05&endtime=2012_09_15_18_34_14
Success Return	media stream data
Comment	It's similar with "4.1.1 Get Real-Time Stream". Except there are parameters "starttime" and "endtime".

4.1.3 Get File Stream

Table 4-3

Syntax	rtsp://<***username***>:<***password***>@<***ip***>:<***port***>/<***filename***>
Description	Get specific file stream.
Example	rtsp://admin:admin@10.44.200.8:554**//mnt**/sd/2015-09-16/001/dav/20/20.32.08-20.32.28[M][0@0][0].dav
Success Return	media stream data
Comment	It's similar with "4.1.1 Get Real-Time Stream". ***filename***: absolute path.

4.2 Get MJPG Stream

Table 4-4

Syntax	http://<*server*>/cgi-bin/mjpg/video.cgi[?channel=<***ChannelNo***>&subtype=<***typeNo***>]
Method	GET
Description	Get a video stream encoded by mjpg.
Example	To get a video stream of channel 1, main stream, the URL can be http://192.168.1.108/cgi-bin/mjpg/video.cgi or http://192.168.1.108/cgi-bin/mjpg/video.cgi?channel=1&subtype=0
Success Return	Video stream encoded by MJPG. For example: HTTP Code: 200 OK Content-Type: multipart/x-mixed-replace; boundary=<boundary> Body: --<boundary> Content-Type: image/jpeg Content-Length:<image size> <JPEG image data> --<boundary>
Comment	***ChannelNo***: integer, video channel index which starts from 1, default 1 if not specified. ***typeNo***: the stream type, default 0 if not specified. It can be the following value: 0-Main Stream 1-Extra Stream 1 2-Extra Stream 2

4.3 Audio

4.3.1 Get Audio Input Channel Numbers

Table 4-5

Syntax	http://<*server*>/cgi-bin/devAudioInput.cgi?action=getCollect
Method	GET
Description	Get audio input channel number.
Example	http://192.168.1.108/cgi-bin/devAudioInput.cgi?action=getCollect
Success Return	result=1

Comment	Above response means there are 2 audio input channels.

4.3.2 Get Audio Output Channel Numbers

Table 4-6

Syntax	http://<*server*>/cgi-bin/devAudioOutput.cgi?action=getCollect
Method	GET
Description	Get audio output channel number.
Example	http://192.168.1.108/cgi-bin/devAudioOutput.cgi?action=getCollect
Success Return	result=1
Comment	Above response means there are 2 audio output channels.

4.3.3 Post Audio Stream

Table 4-7

Syntax	http://<*server*>/cgi-bin/audio.cgi?action=postAudio&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	POST
Description	Post audio.
Example	Example for single part The URL of transmit a single part, channel 1 audio stream(encoded with G.711 A-law) is: http://192.168.1.108/cgi-bin/audio.cgi?action=postAudio&httptype=singlepart&channel=1 example: POST /cgi-bin/audio.cgi?action=postAudio&httptype=singlepart&channel=1 HTTP/1.1 Content-Type: Audio/G.711A Content-Length: 9999999 <Audio data> <Audio data> Example for multipart The URL of transmit a multipart, channel 1 audio stream(encoded with G.711 A-law) is: http://192.168.1.108/cgi-bin/audio.cgi?action=postAudio&httptype=multipart&channel=1 example: POST /cgi-bin/audio.cgi?action=postAudio&httptype=multipart&channel=1 HTTP/1.1 Content-Type: multipart/x-mixed-replace; boundary=<boundary> --<boundary> Content-Type: Audio/G.711A Content-Length: 800 <Audio data> --<boundary>

Success Return	OK
Comment	Parameters in URL: The paramName and paramValue are in the table below.

Appendix A: Parameters in URL

ParamName	ParamValue type	Description
httptype	string	singlepart: HTTP content is a continuous flow of audio packets multipart: HTTP content type is multipart/x-mixed-replace, and each audio packet ends with a boundary string
channel	integer	The audio channel index which starts from 1.

Appendix B: Audio Encode Type

MIME	Description
Audio/PCM	PCM
Audio/ADPCM	ADPCM
Audio/G.711A	G.711 A Law
Audio/G.711Mu	G.711 Mu Law
Audio/G.726	G.726
Audio/G.729	G.729
Audio/MPEG2	MPEG2
Audio/AMR	AMR
Audio/AAC	AAC

4.3.4 Get Audio Stream

Table 4-8

Syntax	http://<*server*>/cgi-bin/audio.cgi?action=getAudio&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Get audio.
Example	Example for single part The URL of Request a single part, channel 1 audio stream(encoded with G.711 A-law) is: http://192.168.1.108/cgi-bin/audio.cgi?action=getAudio&httptype=singlepart&channel=1 If the request was successful, the server returns a continuous flow of audio packets. The content type is only set at the beginning of the connection. Return: HTTP Code: 200 OK Content-Type: Audio/G.711A Body: <Audio data> <Audio data>

	Example for multipart The URL of Request a multipart, channel 1 audio stream(encoded with G.711 A-law) is: http://192.168.1.108/cgi-bin/audio.cgi?action=getAudio&httptype=multipart&channel=1 If the request was successful, the server returns a continuous flow of audio packets. The content type is "multipart/x-mixed-replace" and each audio packet ends with a boundary string. Return: HTTP Code: 200 OK Content-Type: multipart/x-mixed-replace; boundary=<boundary> --<boundary> Content-Type: Audio/G.711A Content-Length: 800 <Audio data> --<boundary>
Success Return	OK
Comment	Parameters in URL: The paramName and paramValue are in the table below.

Appendix:

ParamName	ParamValue type	Description
httptype	string	singlepart: HTTP content is a continuous flow of audio packets multipart: HTTP content type is multipart/x-mixed-replace, and each audio packet ends with a boundary string
channel	integer	The audio channel index which starts from 1.

4.4 Snapshot

4.4.1 Snap

- Get snap config

Table 4-9

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=Snap
Method	GET
Description	Get snap config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Snap
Success Return	table.Snap[0].HolidayEnable=false table.Snap[0].TimeSection[0][0]=6 00:00:00-23:59:59 table.Snap[0].TimeSection[0][1]=0 00:00:00-23:59:59 table.Snap[0].TimeSection[0][2]=0 00:00:00-23:59:59 table.Snap[0].TimeSection[0][3]=0 00:00:00-23:59:59 table.Snap[0].TimeSection[0][4]=0 00:00:00-23:59:59 table.Snap[0].TimeSection[0][5]=0 00:00:00-23:59:59 table.Snap[0].TimeSection[1][0]=6 00:00:00-23:59:59

	table.Snap[0].TimeSection[1][1]=0 00:00:00-23:59:59 table.Snap[0].TimeSection[1][2]=0 00:00:00-23:59:59 table.Snap[0].TimeSection[1][3]=0 00:00:00-23:59:59 table.Snap[0].TimeSection[1][4]=0 00:00:00-23:59:59 table.Snap[0].TimeSection[1][5]=0 00:00:00-23:59:59 …
Comment	Response format: table. Snap[***channelNo***].TimeSection[***weekday***][***configNo***]=1 00:00:00-23:59:59 ***channelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***weekday***: range is [0—6] (Sunday — Saturday). ***configNo***: the index of time section config. There are many time sections each day.

- Set snap config

Table 4-10

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set snap config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Snap[0].TimeSection[0][0]=1%2012:00:00-18:00:00
Success Return	OK
Comment	In table below, ***ch***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***wd***: week day index, range is [0—6] (Sunday — Saturday). ***ts***: time section index, starts from 0.

Appendix:

ParamName	ParamValue type	Description
Snap[***ch***].TimeSection[***wd***][***ts***]	string	wd (week day) range is [0—6] (Sunday—Saturday) ts (time section) range is [0—23], it's time section table index. Format: mask hh:mm:ss-hh:mm:ss Mask: [0—65535], hh: [0—24], mm: [0—59], ss: [0—59] Mask indicates record type by bits: Bit0: regular snapshot Bit1: motion detection snapshot Bit2: alarm snapshot Bit3: card snapshot

4.4.2 Get a Snapshot

Table 4-11

Syntax	http://<*server*>/cgi-bin/ snapshot.cgi[?channel=<***ChannelNo***>]
Method	GET
Description	Get a snapshot of a video channel.
Example	To get a snapshot of video channel 1, the URL can be http://192.168.1.108/cgi-bin/snapshot.cgi or http://192.168.1.108/cgi-bin/snapshot.cgi?channel=1
Success Return	Image of jpg format.
Comment	***ChannelNo***: integer, video channel index which starts from 1, default 1 if not specified.

4.4.3 Subscribe to Snapshot

Table 4-12

Syntax	http://<*server*>/cgi-bin/snapManager.cgi?action=attachFileProc&Flags[0]=Event&Events=[<***eventCode***>,<***eventCode***>...][&channel=<***ChannelNo***>][&heartbeat=<***Heartbeat***>]
Method	GET
Description	Subscribe pictures when event of code ***eventCode*** happens.
Example	http://192.168.1.108/cgi-bin/snapManager.cgi?action=attachFileProc&Flags[0]=Event&Events=[VideoMotion%2CVideoLoss]&heartbeat=5
Success Return	--<*boundary*>\r\n Content-Type: text/plain\r\n Content-Length: <data length>\r\n Events[0].Code=TrafficJunction Events[0].CountInGroup=1 Events[0].IndexInGroup=1 Events[0].Lane=1 Events[0].Data.PTS= 42949485818.0 Events[0].TrafficCar.PlateNumber=Z A12345 Events[0].TrafficCar. DeviceAddress=Hangzhou …… Events[1].Code=TrafficJunction …… --<*boundary*> Content-Type: image/jpeg Content-Length:<image size> <JPEG image data> --<*boundary*> Content-Type: text/plain Content-Length:<data length> Heartbeat --<*boundary*>

Comment	***ChannelNo***: integer, video channel index which starts from 1, default 1 if not specified. ***Heartbeat****:* integer,range is [1,60],unit is second.If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client,the heartbeat meaage is the string "Heartbeat". ***eventCode***: it can be any one of the standard codes defined in DHIIF. ***eventCode*** includes: VideoMotion: motion detection event VideoLoss: video loss detection event VideoBlind: video blind detection event. AlarmLocal: alarm detection event. TrafficJunction: vehicle passing event TrafficRetrograde: retrograde event TrafficParking: illegal parking event TrafficJam: traffic congestion event TrafficThrow: throw something out of the car event TrafficPedestrain: pedestrain enter traffic lane(driverway) event TrafficParkingSpaceParking: a vehicle in the parking space event TrafficParkingSpaceNoParking: no vehicle in the parking space event TrafficParkingSpaceOverLine: a vehicle over the parking line event TrafficManualSnap: manual snapshot event TrafficSuspiciousCar: when the vehicle is in the blacklist, it wiil produce this event. FaceRecognition: face recognition event. Also, the ***eventCode*** can be ***All*** ***All*** means all kinds of the eventcode. Parameters in Response: ***GroupID***: string, the ID of the snapshot event ***CountInGroup***: integer, the number of the snapshots ***IndexInGroup***: integer, the index of the snapshots. For example, the CountInGroup is 3, the IndexInGroup is 1,it means there are 3 pictures in this snapshot event, and this is the first picture.

4.5 Video Attributes

4.5.1 Get Max Extra Stream Numbers

Table 4-13

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getProductDefinition&name=MaxExtraStream
Method	GET
Description	Get max extra stream count.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getProductDefinition&name=MaxExtraStream
Success Return	table.***MaxExtraStream***=1
Comment	***MaxExtraStream***: max extra stream numbers. It can be 1, 2 or 3.

4.5.2 Get Encode Capability

Table 4-14

Syntax	http://<*server*>/cgi-bin/encode.cgi?action=getCaps
Method	GET
Description	Get encode capabilities.
Example	http://192.168.1.108/cgi-bin/encode.cgi?action=getCaps
Success Return	caps.PlaybackCompressSplitNumList[0]=1 caps.PlaybackCompressSplitNumList[1]=2 caps.PlaybackCompressSplitNumList[2]=4 caps.PreviewMode=SplitSnap caps.VideoEncodeDevices[0].CoverAreaPercent=100 caps.VideoEncodeDevices[0].CoverCount=4 caps.VideoEncodeDevices[0].LadenBitrate=162201600 caps.VideoEncodeDevices[0].MaxCIFPFrameSize=40 caps.VideoEncodeDevices[0].MaxExtraStream=1 caps.VideoEncodeDevices[0].MinCIFPFrameSize=7 caps.VideoEncodeDevices[0].RecordIndividualResolution=true caps.VideoEncodeDevices[0].SupportIndividualResolution=true caps.VideoEncodeDevices[0].TitleCount=4
Comment	—

4.5.3 Get Encode Config Capability

Table 4-15

Syntax	http://<*server*>/cgi-bin/encode.cgi?action=getConfigCaps[&channel=<***ChannelNo***>&<***paramName***>=<***paramValue***>&<***paramName***>=<***paramValue***>...]
Method	GET
Description	Get encode config capabilities.
Example	http://192.168.1.108/cgi-bin/encode.cgi?action=getConfigCaps&channel=1&Encode[0].MainFormat[0].Video.Width=1920&Encode[0].MainFormat[0].Video.Height=1080
Success Return	***headMain***.Video.BitRateOptions=448,2560 ***headMain***.Video.CompressionTypes=H.264,MJPG ***headMain***.Video.FPSMax=25 ***headMain***.Video.ResolutionTypes=2048 x 1536,1080,SXGA, 1280 x 960,720,D1,CIF ***headExtra***.Video.BitRateOptions=80,448 ***headExtra***.Video.CompressionTypes=H.264,MJPG ***headExtra***.Video.FPSMax=25 ***headExtra***.Video.ResolutionTypes=D1,CIF ***headSnap***.Video.CompressionTypes=H.264,MJPG ***headSnap***.Video.ResolutionTypes=2048 x 1536,1080,SXGA, 1280 x 960,720,D1,CIF

Comment	Parameters in URL: ***ChannelNo***: integer,video channel index which starts from 1.it is a optional parameter, if it is not present in the URL, the value of the channelno will be regarded as 1. The description of <***paramName***> and <***paramValue***> is the same as **4.5.5**(Set encode config). Parameters in Response: ***headMain***= caps[***Channel***].MainFormat[***RecordType***] ***headExtra*** = caps[***Channel***].ExtraFormat[***ExtraStream***] ***headSnap*** = caps[***Channel***].SnapFormat[***SnapType***] Channel: integer, array index starts from 0, which means video channel (equals to video channel index ChannelNo -1, and so 0 means channel 1). ***RecordType***: 0 = regular record 1 = motion detection record 2 = alarm record ***ExtraStream***: 0 = extra stream 1 1 = extra stream 2 2 = extra stream 3 ***SnapType***: 0 = regular snapshot 1 = motion detection snapshot 2 = alarm snapshot

Appendix A: Encode Config Capabilities

Field in response	Value range	Description
BitRateOptions	string	Before comma is minimum bit rate. (kbps), after comma is maximum bit rate.(kbps) BitRateOptions=80,448 80 is the minimum bitrates, 448 is maximum.
CompressionTypes	string	To video, it contains all supported video compression types, separated by comma. Range is {MPEG4, MPEG2, MPEG1, MJPG, H.263, H.264, H.265}. To audio, it contains all supported audio compression types, separated by comma. Range is {PCM, ADPCM, G.711A, G.711Mu, G.726, G.729, MPEG2, AMR}
FPSMax	integer	Maximum FPS.
ResolutionTypes	string	It contains all supported video resolutions. Range is in below Resolution list.

Appendix B: Video Resolution

Fixed Resolution Name	Size in PAL	Size in NTSC
"D1"	704 x 576	704 x 480
"HD1"	352 x 576	352 x 480
"BCIF"/"2CIF"	704 x 288	704 x 240

Fixed Resolution Name	Size in PAL	Size in NTSC
"CIF"	352 x 288	352 x 240
"QCIF"	176 x 144	176 x 120
"NHD"	640 x 360	—
"VGA"	640 x 480	—
"QVGA"	320 x 240	—
"SVCD"	480 x 480	—
"QQVGA"	160 x 128	—
"SVGA"	800 x 592	—
"SVGA1"	800 x 600	—
"WVGA"	800 x 480	—
"FWVGA"	854 x 480	—
"DVGA"	960 x 640	—
"XVGA"	1024 x 768	—
"WXGA"	1280 x 800	—
"WXGA2"	1280 x 768	—
"WXGA3"	1280 x 854	—
"WXGA4"	1366 x 768	—
"SXGA"	1280 x 1024	—
"SXGA+"	1400 x 1050	—
"WSXGA"	1600 x 1024	—
"UXGA"	1600 x 1200	—
"WUXGA"	1920 x 1200	—
"ND1"	240 x 192	—
"720P"	1280 x 720	—
"1080P"	1920 x 1080	—
"QFHD"	3840 x 2160	—
"1_3M", "1280x960"	1280 x 960 (1.3 Mega Pixels)	—
"2_5M", "1872x1408"	1872 x 1408 (2.5 Mega Pixels)	—
"5M", "3744x1408"	3744 x 1408 (5 Mega Pixels)	—
"3M", "2048x1536"	2048 x 1536 (3 Mega Pixels)	—
"5_0M", "2432x2048"	2432 x 2048 (5 Mega Pixels)	—
"1_2M", "1216x1024"	1216 x 1024 (1.2 Mega Pixels)	—
"1408x1024"	1408 x 1024 (1.5 Mega Pixels)	—
"3296x2472"	3296 x 2472 (8 Mega Pixels)	—
"5_1M", "2560x1920"	2560 x 1920 (5 Mega Pixels)	—
"960H",	960 x 576	960 x 480
"DV720P"	960 x 720	—
"2560x1600"	2560 x 1600 (4 Mega Pixels)	—
"2336x1752"	2336 x 1752 (4 Mega Pixels)	—
"2592x2048"	2592 x 2048	—
"2448x2048"	2448 x 2048	—
"1920x1440"	1920x1440	—
"2752x2208"	2752x2208	—
"3840x2160"	3840x2160	—
"4096x2160"	4096x2160	—

Fixed Resolution Name	Size in PAL	Size in NTSC
"3072x2048"	3072x2048	—

Appendix C: Audio Compression Type

Field in response	Value range	Description
CompressionTypes	string	It contains all supported audio compression types, separated by comma. Range is {PCM, ADPCM, G.711A, G.711Mu, G.726, G.729, MPEG2, AMR}

4.5.4 Encode of Media

- Get encode config

Table 4-16

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=Encode
Method	GET
Description	Get video encode config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Encode
Success Return	table.Encode[0].MainFormat[0].Audio.Bitrate=64 table.Encode[0].MainFormat[0].Audio.Channels[0]=0 table.Encode[0].MainFormat[0].Audio.Compression=G.711A table.Encode[0].MainFormat[0].Audio.Depth=16 table.Encode[0].MainFormat[0].Audio.Frequency=8000 table.Encode[0].MainFormat[0].Audio.Mode=0 table.Encode[0].MainFormat[0].Audio.Pack=DHAV table.Encode[0].MainFormat[0].AudioEnable=true table.Encode[0].MainFormat[0].Video.resolution=1920x1080 table.Encode[0].MainFormat[0].Video.BitRate=4096 table.Encode[0].MainFormat[0].Video.BitRateControl=CBR table.Encode[0].MainFormat[0].Video.Compression=H.264 table.Encode[0].MainFormat[0].Video.CustomResolutionName=1080P table.Encode[0].MainFormat[0].Video.FPS=18 table.Encode[0].MainFormat[0].Video.GOP=36 table.Encode[0].MainFormat[0].Video.Height=1080 table.Encode[0].MainFormat[0].Video.Pack=DHAV table.Encode[0].MainFormat[0].Video.Profile=High table.Encode[0].MainFormat[0].Video.Quality=4 table.Encode[0].MainFormat[0].Video.QualityRange=6 table.Encode[0].MainFormat[0].Video.SVCTLayer=1 table.Encode[0].MainFormat[0].Video.Width=1920 table.Encode[0].MainFormat[0].VideoEnable=true table.Encode[0].MainFormat[1].Audio.Bitrate=64 table.Encode[0].MainFormat[1].Audio.Channels[0]=0 table.Encode[0].MainFormat[1].Audio.Compression=G.711A table.Encode[0].MainFormat[1].Audio.Depth=16 table.Encode[0].MainFormat[1].Audio.Frequency=8000 table.Encode[0].MainFormat[1].Audio.Mode=0

	table.Encode[0].MainFormat[1].Audio.Pack=DHAV table.Encode[0].MainFormat[1].AudioEnable=true table.Encode[0].MainFormat[1].Video.resolution=1920x1080 table.Encode[0].MainFormat[1].Video.BitRate=4096 table.Encode[0].MainFormat[1].Video.BitRateControl=CBR table.Encode[0].MainFormat[1].Video.Compression=H.264 table.Encode[0].MainFormat[1].Video.CustomResolutionName=1080P table.Encode[0].MainFormat[1].Video.FPS=18 table.Encode[0].MainFormat[1].Video.GOP=36 table.Encode[0].MainFormat[1].Video.Height=1080 table.Encode[0].MainFormat[1].Video.Pack=DHAV table.Encode[0].MainFormat[1].Video.Profile=High table.Encode[0].MainFormat[1].Video.Quality=4 table.Encode[0].MainFormat[1].Video.QualityRange=6 table.Encode[0].MainFormat[1].Video.SVCTLayer=1 table.Encode[0].MainFormat[1].Video.Width=1920 table.Encode[0].MainFormat[1].VideoEnable=true table.Encode[0].MainFormat[2].Audio.Bitrate=64 table.Encode[0].MainFormat[2].Audio.Channels[0]=0 table.Encode[0].MainFormat[2].Audio.Compression=G.711A table.Encode[0].MainFormat[2].Audio.Depth=16 table.Encode[0].MainFormat[2].Audio.Frequency=8000 table.Encode[0].MainFormat[2].Audio.Mode=0 table.Encode[0].MainFormat[2].Audio.Pack=DHAV table.Encode[0].MainFormat[2].AudioEnable=true table.Encode[0].MainFormat[2].Video.resolution=1920x1080 table.Encode[0].MainFormat[2].Video.BitRate=4096 table.Encode[0].MainFormat[2].Video.BitRateControl=CBR table.Encode[0].MainFormat[2].Video.Compression=H.264 table.Encode[0].MainFormat[2].Video.CustomResolutionName=1080P table.Encode[0].MainFormat[2].Video.FPS=18 table.Encode[0].MainFormat[2].Video.GOP=36 table.Encode[0].MainFormat[2].Video.Height=1080 table.Encode[0].MainFormat[2].Video.Pack=DHAV table.Encode[0].MainFormat[2].Video.Profile=High table.Encode[0].MainFormat[2].Video.Quality=4 table.Encode[0].MainFormat[2].Video.QualityRange=6 table.Encode[0].MainFormat[2].Video.SVCTLayer=1 table.Encode[0].MainFormat[2].Video.Width=1920 table.Encode[0].MainFormat[2].VideoEnable=true table.Encode[0].MainFormat[3].Audio.Bitrate=64 table.Encode[0].MainFormat[3].Audio.Channels[0]=0 table.Encode[0].MainFormat[3].Audio.Compression=G.711A table.Encode[0].MainFormat[3].Audio.Depth=16 table.Encode[0].MainFormat[3].Audio.Frequency=8000 table.Encode[0].MainFormat[3].Audio.Mode=0 table.Encode[0].MainFormat[3].Audio.Pack=DHAV

	table.Encode[0].MainFormat[3].AudioEnable=true table.Encode[0].MainFormat[3].Video.resolution=704x576 table.Encode[0].MainFormat[3].Video.BitRate=2048 table.Encode[0].MainFormat[3].Video.BitRateControl=VBR table.Encode[0].MainFormat[3].Video.Compression=H.264 table.Encode[0].MainFormat[3].Video.FPS=25 table.Encode[0].MainFormat[3].Video.GOP=50 table.Encode[0].MainFormat[3].Video.Height=576 table.Encode[0].MainFormat[3].Video.Pack=DHAV table.Encode[0].MainFormat[3].Video.Profile=Main table.Encode[0].MainFormat[3].Video.Quality=4 table.Encode[0].MainFormat[3].Video.QualityRange=6 table.Encode[0].MainFormat[3].Video.SVCTLayer=1 table.Encode[0].MainFormat[3].Video.Width=704 table.Encode[0].MainFormat[3].VideoEnable=true table.Encode[0].ExtraFormat[0].Audio.Bitrate=64 … table.Encode[0].SnapFormat[0].Audio.Bitrate=64 …
Comment	Parameters in Response: The format of the config is *head.configItems.* The *head* can be: *headMain*= table. Encode[***Channel***].MainFormat[***Type***] *headSnap* = table. Encode[***Channel***].SnapFormat[***Type***] *headExtra* =table. Encode[***Channel***].ExtraFormat[***ExtraStream***] *Channel*: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). *Type*: 0 = regular encode 1 = motion detection encode 2 = alarm encode 3= emergency encode *ExtraStream*: 0 = extra stream 1 1 = extra stream 2 2 = extra stream 3 The *configItems are list as bellow.*

- Set encode config

Table 4-17

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>...]
Method	GET
Description	Set encode config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Encode[1].MainFormat[*0*].Video.Compression=MPEG4&
Success	OK

Return	
Comment	Parameters in URL: In table below, ***head***=Encode[***Channel***].MainFormat[***RecordType***] (or) Encode[***Channel***].ExtraFormat[***ExtraStream***] ***Channel***: integer, array index starts from 0, which means video channel(equals to video channel index -1, and so 0 means channel 1). ***RecordType***: 0 = regular record 1 = motion detection record 2 = alarm record ***ExtraStream***: 0 = extra stream 1 1 = extra stream 2 2 = extra stream 3

Appendix A: Video Encode Config

ParamName	ParamValue type	Description
head.Video.BitRate	integer	Unit is Kbps Range depends on capability in GetVideoConfigCaps
head.Video.BitRateControl	string	Range is {CBR, VBR} CBR: constant bitrates VBR: variable bitrates
head.Video.Compression	String	Range is {MPEG4, MPEG2, MPEG1, MJPG, H.263, H.264, H.265} Depends on capacity in GetVideoConfigCaps
head.Video.FPS	float	Range is [0.2—30]. Frames per second. <1.0: several seconds/frame, FPS=0.3333: 3 seconds per frame. >1.0: several frames/second. FPS=3: 3 frames per second.
head.Video.GOP	integer	Range is [1—100]. Group of picture, it's the interval of I Frame, Example: GOP=50, means there is one I frame every 49 P or B frames
head.Video.Height	integer	Video height
head.Video.Width	integer	Video Width
head.Video.Profile	String	Range is { Baseline, Main , Extended , High } Only when video compression is H.264, it's effective.
head.Video.Quality	integer	Range is [1—6]. Image Quality, available when Video.BitRateControl=VBR 1: worst quality 6: best quality
head.VideoEnable	bool	True: enable video

Appendix B: Audio Encode Config

ParamName	ParamValue type	Description
head.Audio.Bitrate	integer	Unit is kbps Range depends on capacity in **GetAudioConfigCaps**
head.Audio.Compression	string	Range depends on capacity in **GetAudioConfigCaps**
head.Audio.Depth	integer	Audio sampling depth
head.Audio.Frequency	integer	Audio sampling frequency
head.Audio.Mode	integer	Range is {0,1,2,3,4,5,6,7} Audio encode mode. 0: 4.75kbps, 1: 5.15 kbps, 2: 5.9 kbps, 3: 6.7 kbps, 4: 7.4 kbps, 5: 7.95 kbps, 6: 10.2 kbps, 7: 12.2 kbps,
head.AudioEnable	bool	Enable/Disable audio

4.5.5 Encode of Region Interested

- Get encode config of region interested

Table 4-18

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoEncodeROI
Method	GET
Description	Get video encode config of region interested.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoEncodeROI
Success Return	***head***.DynamicTrack=false
Comment	Parameters in Response: ***head***=table.VideoEncodeROI[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

- Set encode config of region interested

Table 4-19

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set video encode config of region interested.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoEncodeROI[0].DynamicTrack=true
Success	OK

Return	
Comment	Parameters in URL: ***paramName*** and ***paramValue*** are as table below. In table below, ***head*** = VideoEncodeROI[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel(equals to video channel index -1, and so 0 means channel 1).

Appendix:

ParamName	ParamValue type	Description
head. DynamicTrack	bool	Enable/Disable

4.5.6 Channel Title

- Get channel title

Table 4-20

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=ChannelTitle
Method	GET
Description	Get the title of the video channel.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=ChannelTitle
Success Return	table.ChannelTitle[***Channel***].Name=CAM1|123
Comment	Parameters in Response: ***Channel***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***Name***: string, which means title content. character '|' means newlines.

- Set channel title

Table 4-21

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>
Method	GET
Description	Set the title of the channel.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&ChannelTitle[1].Name= test1|test2
Success Return	OK
Comment	If VideoWidget[*Channel*].ChannelTitle.EncodeBlend is true, this title is blended to the video frames. Please refer to SetVideoWidgetConfig. Parameters in URL: Channel Name Format: ChannelTitle[***Channel***].Name ***Channel***: integer, array index starts from 0, which means video channel(equals to video channel index -1, and so 0 means channel 1). Name: string, which means title content. Character '|' means newlines.

	Limit: Support 2 lines at most.

4.5.7 Get Video Input Channels Device Supported

Table 4-22

Syntax	http://<*server*>/cgi-bin/devVideoInput.cgi?action=getCollect
Method	GET
Description	Get the video input channel numbers.
Example	http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getCollect
Success Return	result=1
Comment	—

4.5.8 Get Video Output Channels Device Supported

Table 4-23

Syntax	http://<*server*>/cgi-bin/devVideoOutput.cgi?action=getCollect
Method	GET
Description	Get the video output channel numbers.
Example	http://192.168.1.108/cgi-bin/devVideoOutput.cgi?action=getCollect
Success Return	result=2
Comment	—

4.5.9 Get Max Remote Input Channels

Table 4-24

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getProductDefinition&name=MaxRemoteInputChannels
Method	GET
Description	Get max remote input channels.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getProductDefinition&name=MaxRemoteInputChannels
Success Return	table.MaxRemoteInputChannels=16
Comment	MaxRemoteInputChannels: max remote input channels.

4.5.10 Video Standard

- Get video standard

Table 4-25

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoStandard
Method	GET
Description	Get video standard config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoStandar

	d
Success Return	table.VideoStandard=PAL
Comment	—

- Set video standard

Table 4-26

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&VideoStandard=<***paramValue***>
Method	GET
Description	Set video standard config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoStandard=PAL
Success Return	OK
Comment	VideoStandard: string, range is {PAL, NTSC} Video Standard.

4.5.11 Video Widget

- Get video widget config

Table 4-27

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoWidget
Method	GET
Description	Video widget config contains Channel Title, User-defined Title, Covers and Time Title parameters, defines the background color, front color and positions of channel title and time title, User-defined title and defines the regions which are not visible (cover).
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoWidget
Success Return	***head***.BackColor[0]=0 ***head***.BackColor[1]=0 ***head***.BackColor[2]=0 ***head***.BackColor[3]=128 ***head***.EncodeBlend=true ***head***.FrontColor[0]=255 ***head***.FrontColor[1]=255 ***head***.FrontColor[2]=255 ***head***.FrontColor[3]=0 ***head***.Rect[0]=0 ***head***.Rect[1]=8191 ***head***.Rect[2]=0 ***head***.Rect[3]=8191 …
Comment	Parameters in Response: ***head***=table.VideoWidget[***Channel***].ChannelTitle (or) table.VideoWidget[***Channel***].Covers[***CoReg***] (or) table.VideoWidget[***Channel***].TimeTitle table.VideoWidget[***Channel***].UserDefinedTitle[***Index***] ***Channel***: integer, array index starts from 0, which means video channel (equals to

	video channel index -1, and so 0 means channel 1). ***CoReg***: Cover Region, Covers is an array which sustains multi- Cover regions 0 = region 1 1 = region 2 2 = region 3 3 = region 4 ***Index***: integer, User-defined title index, now only index 0 is valid, others are reserved.

- Set video widget config

Table 4-28

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set video widget config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoWidget[1].Covers[0].BackColor[0]=128&VideoWidget[1].Covers[0].BackColor[1]=128&VideoWidget[1].Covers[0].BackColor[2]=128&VideoWidget[1].Covers[0].BackColor[3]=0
Success Return	OK
Comment	Parameters in URL: In table below, ***headChannelTitle*** = VideoWidget[***Channel***].ChannelTitle ***headCover*** = VideoWidget[***Channel***].Covers[***CoReg***] ***headTimeTitle*** = VideoWidget[***Channel***].TimeTitle ***headUserDefinedTitle*** = VideoWidget[***Channel***].UserDefinedTitle[***Index***] ***Channel***: integer, array index starts from 0, which means video channel(equals to video channel index -1, and so 0 means channel 1). ***CoReg***: Cover region index. Covers is an array which contains multiple cover regions 0 = region 1 1 = region 2 2 = region 3 3 = region 4 ***Index***: integer, User-defined title index, now only index 0 is valid, others are reserved.

Appendix:

ParamName	ParamValue type	Description
headCover.BackColor[0] *headCover*.BackColor[1] *headCover*.BackColor[2] *headCover*.BackColor[3]	integer	Range is [0—255]. BackColor[0]:red value BackColor[1]:green value BackColor[2]:blue value BackColor[3]: alpha value
headCover.EncodeBlend	bool	false - widget blend is disabled.

ParamName	ParamValue type	Description
headCover.FrontColor[0] *headCover*.FrontColor[1] *headCover*.FrontColor[2] *headCover*.FrontColor[3]	integer	Range is [0—255]. FrontColor[0]:red value FrontColor[1]:green value FrontColor[2]:blue value FrontColor[3]: alpha value
headCover.Rect[0] *headCover*.Rect[1] *headCover*.Rect[2] *headCover*.Rect[3]	integer	Range is [0—8191]. Rect[0]: top left corner x coordinate (left) Rect[1]: top left corner y coordinate (top) Rect[2]: bottom right x coordinate (right) Rect[3]: bottom right y coordinate (bottom)
headChannelTitle.BackColor[0] *headChannelTitle*.BackColor[1] *headChannelTitle*.BackColor[2] *headChannelTitle*.BackColor[3]	integer	Range is the same with *headCover.*
headChannelTitle.EncodeBlend	bool	
headChannelTitle.FrontColor[0] *headChannelTitle*.FrontColor[1] *headChannelTitle*.FrontColor[2] *headChannelTitle*.FrontColor[3]	integer	
headChannelTitle.Rect[0] *headChannelTitle*.Rect[1] *headChannelTitle*.Rect[2] *headChannelTitle*.Rect[3]	integer	Only use the value of (left, top),the value of (right, bottom) is the same as (left, top) Rect[0], Rect[1] are used, and Rect[2] must be same with Rect[0], Rect[3] must be same with Rect[1].
headUserDefinedTitle.Text	string	Title content. Character '|' means newlines. Limit: Support 2 lines at most.
headUserDefinedTitle.TextAlign	integer	Range is the same with headChannelTitle These are configs about User-defined title.
headUserDefinedTitle.BackColor[0] *headUserDefinedTitle.BackColor[1]* *headUserDefinedTitle.BackColor[2]* *headUserDefinedTitle.BackColor[3]*	integer	
headUserDefinedTitle.EncodeBlend	bool	
headUserDefinedTitle.FrontColor[0] *headUserDefinedTitle.FrontColor[1]* *headUserDefinedTitle.FrontColor[2]* *headUserDefinedTitle.FrontColor[3]*	integer	
headUserDefinedTitle.Rect[0] *headUserDefinedTitle.Rect[1]* *headUserDefinedTitle.Rect[2]* *headUserDefinedTitle.Rect[3]*	integer	
headTimeTitle.BackColor[0] *headTimeTitle.BackColor[1]* *headTimeTitle.BackColor[2]* *headTimeTitle.BackColor[3]*	integer	Range is the same with *headChannelTitle.* These are configs about time title.

ParamName	ParamValue type	Description
headTimeTitle.EncodeBlend	bool	
headTimeTitle.FrontColor[0] *headTimeTitle*.FrontColor[1] *headTimeTitle*.FrontColor[2] *headTimeTitle*.FrontColor[3]	integer	
headTimeTitle.Rect[0] *headTimeTitle*.Rect[1] *headTimeTitle*.Rect[2] *headTimeTitle*.Rect[3]	integer	
*headTimeTitle.*ShowWeek	bool	True: Display week within the time title.

4.5.12 Get Video Input Capability

Table 4-29

Syntax	http://<*server*>/cgi-bin/devVideoInput.cgi?action=getCaps&channel=<***ChannelNo***>
Method	GET
Description	Get video input capabilities.
Example	http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getCaps&channel=1
Success Return	caps. AutoSyncPhase=false caps. Backlight=2 caps. BrightnessCompensation=true caps. ChipID=0 caps. CoverCount=4 caps. CoverType=1 caps. CustomManualExposure=true caps. DayNightColor=true caps. DayNightColorIO=0 caps. DoubleExposure=0 caps. DownScaling=false caps. EEModeRange=100 caps. ElectricFocus=false caps. Exposure=16 caps. ExposureMode=31 caps. ExternalSyncInput=0 caps. FishEye=false caps. FlashAdjust=false caps. Flip=true caps. FormatCount=5 caps. Gain=true caps. GainAuto=true caps. Gamma=true caps. GammaModeRange=100 caps. GlareInhibition=1 caps. HorizontalBinning=0 caps. IRCUT=true caps. ImageEnhancement.LevelRange[0]=0

	caps. ImageEnhancement.LevelRange[1]=100 caps. ImageEnhancement.Support=true caps. InfraRed=true caps. Iris=true caps. IrisAuto=true caps. LadenBitrate=972000 caps. LimitedAutoExposure=true caps. MaxExposureTime=300 caps. MaxExposureTime1=0 caps. MaxHeight=1080 caps. MaxMultiProfile=3 caps. MaxWidth=1920 caps. MeteringRegionCount=0 caps. MinExposureTime=1 caps. MinExposureTime1=0 caps. Mirror=true caps. MultiOptions=false caps. NightOptions=true caps. ReferenceLevel=false caps. Rotate90=true caps. SetColor=true caps. SignalFormats=Inside caps. SignalType[0]=VGA caps. SnapshotExposure=false caps. SupportProfile=false caps. SupportWhiteLevel=true caps. SupportWriteLevel=false caps. SyncChipChannels=false caps. SyncFocus=0 caps. TitleCount=4 caps. TridimDenoise=2 caps. TridimDenoiseDetails=0 caps. UTC=0 caps. UpScaling=false caps. Version=0 caps. VerticalBinning=0 caps. VideoInDenoise.2D.LevelRange[0]=0 caps. VideoInDenoise.2D.LevelRange[1]=100 caps. VideoInDenoise.2D.Support=true caps. VideoInDenoise.3D.3DAutoType.ModRange[0]=0 caps. VideoInDenoise.3D.3DAutoType.ModRange[1]=100 caps. VideoInDenoise.3D.Support=true caps. VideoInDenoise.Support=true caps. WhiteBalance=3 caps. WideDynamicRange=1
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1.

	Parameters in Response: see table below

Appendix:

Field in response	Value type	Description
Backlight	bool	True: support backlight
ChipID	String	ID of chips in this channel
CoverCount	integer	The maximum cover region count.
CoverType	integer	0: don't support cover 1: support realtime cover 2: support non-realtime cover
CustomManualExposure	bool,	true: support use defined manual exposure time
DayNightColor	bool	true: support color alternate between day and night.
DownScaling	bool	true: support down scaling, binning mode not included.
Exposure	integer	Exposure grade. 0 — don't support exposure control.
ExternalSyncInput	bool	true: support HD signal external synchronization.
FlashAdjust	bool	true: support flash adjust
Flip	bool	true: support picture flip.
Gain	bool	true: support gain control.
GainAuto	bool	true: support auto gain.
HorizontalBinning	integer	Horizontal/Vertical pixel binning mask, 1 — support 2 pixel binning, 2 — support 3 pixel binning 4 — support 4 pixel binning … 2^n — support n+2 pixel binning
VerticalBinning	integer	
InfraRed	bool	true: support Infra compensation
Iris	bool	true: support Iris adjust
IrisAuto	bool	true: support auto Iris adjust
LadenBitrate	integer	Unit is Kbps. Maximum value of video stream bitrates, 16bpp, not in binning mode.
LimitedAutoExposure	bool	true: support auto exposure with time limit.
MaxHeight	integer	Maximum video height
MaxWidth	integer	Maximum video width
Mirror	bool	true: support picture mirror.
NightOptions	bool	true: support night options.
ReferenceLevel	bool	true: support reference level.
Rotate90	bool	true: support clockwise/anticlockwise 90° rotate
SetColor	bool	true: support color set.

Field in response	Value type	Description
SignalFormats	string	It's a string contains supported video input signal formats for this channel. Signal formats are separated by comma. Range is {Inside, BT656, 720p, 1080p, 1080i, 1080sF, 1_3M} Inside — inside input. 1_3M — 1280*960
SyncChipChannels	bool	True: channels in same chip should be synchronized. Synchronized means video resolution of these channels should be the same.
TitleCount	integer	Maximum count of blending titles.
UpScaling	bool	true: support up scaling.
WhiteBalance	integer	Range is {0, 1, 2, 3} 0: don't support white balance. 1: support auto white balance 2: support auto and pre defined white balance. 3: support auto, pre defined and user defined white balance

4.5.13 Get Coordinates of Current Window

Table 4-30

Syntax	http://<*server*>/cgi-bin/devVideoInput.cgi?action=getCurrentWindow&channel=<***ChannelNo***>
Method	GET
Description	Get the coordinates of the current window.
Example	http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getCurrentWindow&channel=1
Success Return	***rect***[0]=500 ***rect***[1]=500 ***rect***[2]=5000 ***rect***[3]=5000
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. Parameters in Response: ***rect***[n] : relative coordinates, range is 0-8192.{0,0,0,0} top-left, {8192,0,0,0} top-right, {0,8192,0,0} bottom-left, {8192,8192,0,0} bottom-right

4.5.14 Set Coordinates of Current Window

Table 4-31

Syntax	http://<*server*>/cgi-bin/devVideoInput.cgi?action=setCurrentWindow&channel=<***ChannelNo***>&rect[0]=<***rect0***>&rect[1]=<***rect1***>&rect[2]=<***rect2***>&rect[3]=<***rect3***>
Method	GET
Description	Set the coordinates of the current window.
Example	http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=setCurrentWindow&channel=1&

	rect[0]=0&rect[1]=0&rect[2]=5000&rect[3]=5000
Success Return	OK
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. ***rect0*** & ***rect1*** & ***rect2*** & ***rect3***: relative coordinates, range is 0-8192.{0,0,0,0} top-left, {8192,0,0,0} top-right, {0,8192,0,0} bottom-left, {8192,8192,0,0} bottom-right

4.5.15 Video Out

- Get video out config

Table 4-32

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoOut
Method	GET
Description	Get video out config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoOut
Success Return	***head***.Margin[0]=0 ***head***.Margin[1]=0 ***head***.Margin[2]=0 ***head***.Margin[3]=0 ***head***.Color.Brightness=50 ***head***.Color. Contrast =50 ***head***.Color. Satuation =50 ***head***.Color. Hue =50 ***head***.Mode. Width =800 ***head***.Mode. Height=600 ***head***.Mode. BPP =16 ***head***.Mode. Format ="Auto" ***head***.Mode. RefreshRate =60 …
Comment	Parameters in Response: ***head*** = table.VideoOut[***ChannelNo***]. ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

- Set video out config

Table 4-33

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set video out config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&table.VideoOut[1].Color.Brightness=50
Success Return	OK
Comment	Parameters in URL: In table below,

	head = table.VideoOut[***ChannelNo***]. ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

Appendix:

ParamName	ParamValue type	Description
head. Margin[0] ***head***. Margin[1] ***head***. Margin[2] ***head***. Margin[3]	integer	Margin.
head. Color.Brightness	integer	Brightness.
head. Color.Contrast =50	integer	Contrast.
head.Color.Satuation =50	integer	Satuation.
head. Color.Hue =50	integer	Hue.
head. Mode.Width =800 ***head***. Mode.Height=600	integer	Resolution.
head. Mode.BPP =16	integer	Bits per pixel.
head.Mode.Format="Auto"	string	The range is {"Auto", "TV", "VGA", "DVI"}.
head.Mode.RefreshRate=60	integer	Refresh rate.

4.5.16 Smart Code

- Get Smart Code config

Table 4-34

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=SmartEncode
Method	GET
Description	Get smart code config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=SmartEncode
Success Return	head.Enbale=false head.Extra[0]=false
Comment	Parameters in Response: ***head*** = table.SmartEncode[***ChannelNo***]. ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

- Set Smart Code config

Table 4-35

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set smart code config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SmartEncode[0].Enbale=true
Success Return	OK
Comment	Parameters in URL:

	In table below, ***head*** = SmartEncode[***ChannelNo***]. ***ChannelNo***: integer, array index starts from 0, which means video channel(equals to video channel index -1, and so 0 means channel 1).

Appendix:

ParamName	ParamValue type	Description
head.Enbale	bool	Open or close the smart code of the main stream.
head.Extra[0]	bool	Open or close the smart code of the extra stream.

4.5.17 Get Decoder Caps

URL	http://<*server*>/cgi-bin/DevVideoDec.cgi?action=getCaps		
Method	GET		
Description	Get capability of decoder		
[Request Params] (None)			
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
Video	object	R	The video caps
+StreamType	array<string>	R	The stream types that the device supports, ex : h264, h265 etc.
+LadenBitrate	int	R	The maximum decoding capability that the device supports (w * h * FPS * 16)/1024, uint: kbps, ex : 7680*4320*30*8/1024 = 7776000
[Example]			
Request	GET http://192.168.1.108/cgi-bin/DevVideoDec.cgi?action=getCaps		
Response	Video.StreamType[0]=h264 Video.StreamType[1]=h265 Video.LadenBitrate=7776000		

4.5.18 [Config] PrivacyMasking

- Get privacy masking config

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=PrivacyMasking		
Method	GET		
Description	Get privacy masking setting		
[Request Params] (None)			
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
table	object	R	Config info table.
+PrivacyMasking	Array<Array<object>>	R	PrivacyMasking config, each channel has several masking region
++Enable	bool	R	Enable/Disable

++Postion	Array<double>	O	Size is 3 Postion[0] is horizontal angle, range is [-1—1] Postion[1] is vertical angel, range is [-1—1] Postion[2] is zoom, range is [0—1]
++ShapType	string	O	Shape type,can be: “Rect”, “Polygon”
++Rect	Array<int>	O	It is valid if ShapType is Rect , top left and bottom right point, each point has x and y value, oordinate remap to 0 — 8192.
++Polygon	Array<Array<int>>	O	It is valid if ShapType is Polygon the first array is point list, minimum item is 3, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
++Color	Array<int>	O	Mask color, RGBA color, 4 integer range is 0-255
++Mosaic	int	O	Mosaic type, Range is {0, 8, 16, 24, 32} 0: no mosaic 8: mosaic size 8*8 16: mosaic size 16*16 24: mosaic size 24*24 32: mosaic size 32*32
++ViewAngle	double	O	View angle, range is [0.0, 360.0], unit: degree
[Example]			
Request	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=PrivacyMasking		
Response	table.PrivacyMasking[0][0].Enable=true table.PrivacyMasking[0][0].Postion[0]=0.0 table.PrivacyMasking[0][0].Postion[1]=0.0 table.PrivacyMasking[0][0].Postion[2]=1.0 table.PrivacyMasking[0][0].ShapType=Rect table.PrivacyMasking[0][0].Rect[0]=0 table.PrivacyMasking[0][0].Rect[1]=0 table.PrivacyMasking[0][0].Rect[2]=50 table.PrivacyMasking[0][0].Rect[3]=100 table.PrivacyMasking[0][0].Color[0]=128 table.PrivacyMasking[0][0].Color[1]=128 table.PrivacyMasking[0][0].Color[2]=128 table.PrivacyMasking[0][0].Color[3]=255 table.PrivacyMasking[0][0].Mosaic=8 table.PrivacyMasking[0][0].ViewAngle=30.0 table.PrivacyMasking[0][1].Enable=true …		

- Set privacy masking config

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set privacy masking setting		
[Request Params] (key=value format)			
Name	Type	R/O	Param Description

PrivacyMasking	Array<Array<object>>	R	privacy masking config, each channel has several masking region
+Enable	bool	R	Enable/Disable
+Postion	Array<double>	O	Size is 3 Postion[0] is horizontal angle, range is [-1—1] Postion[1] is vertical angel, range is [-1—1] Postion[2] is zoom, range is [0—1]
+ShapType	string	O	Shape type,can be: “Rect”, “Polygon”
+Rect	Array<int>	O	It is valid if ShapType is Rect , top left and bottom right point, each point has x and y value, oordinate remap to 0 — 8192.
+Polygon	Array<Array<int>>	O	It is valid if ShapType is Polygon the first array is point list, minimum item is 3, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
+Color	Array<int>	O	Mask color, RGBA color, 4 integer range is 0-255
+Mosaic	int	O	Mosaic type, Range is {0, 8, 16, 24, 32} 0: no mosaic 8: mosaic size 8*8 16: mosaic size 16*16 24: mosaic size 24*24 32: mosaic size 32*32
+ViewAngle	double	O	View angle, range is [0.0, 360.0], unit: degree
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&PrivacyMasking[0][0].Enable=true&PrivacyMasking[0][0].Postion[0]=0.0&PrivacyMasking[0][0].Postion[1]=0.0&PrivacyMasking[0][0].Postion[2]=1.0&PrivacyMasking[0][0].ShapType=Rect&PrivacyMasking[0][0].Rect[0]=0&PrivacyMasking[0][0].Rect[1]=0&PrivacyMasking[0][0].Rect[2]=50&PrivacyMasking[0][0].Rect[3]=100&PrivacyMasking[0][0].Mosaic=8&PrivacyMasking[0][0].ViewAngle=30.0		
Response	OK		

4.5.19 Get Privacy Masking

Syntax	http://<*server*>/cgi-bin/PrivacyMasking.cgi?action=getPrivacyMasking		
Method	GET		
Description	Get privacy masking		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
offset	int	R	Offset in the result record set, range is [0, Total – 1].
limit	int	R	Count of result to get.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description

total	int	R	Count of result
PrivacyMasking	Array<object >	R	PrivacyMasking config
+Index	int	R	Index of PrivacyMasking which starts from 0
+Enable	int	R	1: enable; 0: disable
+ShapeType	string	O	Shape type,can be: “Rect”, “Polygon”
+Rect	Array<int>	O	It is valid if ShapType is Rect , top left and bottom right point, each point has x and y value, oordinate remap to 0 — 8192.
+Polygon	Array<Array<int>>	O	It is valid if ShapType is Polygon the first array is point list, minimum item is 3, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
+Color	Array<int>	O	Mask color, RGBA color, 4 integer range is 0-255
+Mosaic	int	O	Mosaic type, Range is {0, 8, 16, 24, 32} 0: no mosaic 8: mosaic size 8*8 16: mosaic size 16*16 24: mosaic size 24*24 32: mosaic size 32*32
[Example]			
Request	http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=getPrivacyMasking&channel=1&offset=2&limit =5		
Response	total=5 PrivacyMasking[0].Index=0 PrivacyMasking[0].Enable=1 PrivacyMasking[0].ShapType=Polygon PrivacyMasking[0].Polygon[0][0]= 0 PrivacyMasking[0].Polygon[0][1]=0 PrivacyMasking[0].Polygon[1][0]=128 PrivacyMasking[0].Polygon[1][1]=128 PrivacyMasking[0].Polygon[2][0]=256 PrivacyMasking[0].Polygon[2][1]=200 … PrivacyMasking[0].Color[0]=128, PrivacyMasking[0].Color[1]=128, PrivacyMasking[0].Color[2]=128, PrivacyMasking[0].Color[3]=255, PrivacyMasking[0].Mosaic=8, PrivacyMasking[1].Index =1, PrivacyMasking[1].Enable=1, …		

4.5.20 Set Privacy Masking

Syntax	http://<*server*>/cgi-bin/PrivacyMasking.cgi?action=setPrivacyMasking

Method	GET		
Description	Set privacy masking		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
PrivacyMasking	object	R	Privacy masking config
+Index	int	R	Index of PrivacyMasking which starts from 0
+Enable	int	R	1: enable; 0: disable
+ShapType	string	O	Shape type,can be: “Rect”, “Polygon”
+Rect	Array<int>	O	It is valid if ShapType is Rect , top left and bottom right point, each point has x and y value, oordinate remap to 0 — 8192.
+Polygon	Array<Array<int>>	O	It is valid if ShapType is Polygon the first array is point list, minimum item is 3, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
+Color	Array<int>		Mask color, RGBA color, 4 integer range is 0-255
+Mosaic	int		Mosaic type, Range is {0, 8, 16, 24, 32} 0: no mosaic 8: mosaic size 8*8 16: mosaic size 16*16 24: mosaic size 24*24 32: mosaic size 32*32
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=setPrivacyMasking&channel=2&PrivacyMasking.Index=3&PrivacyMasking.Enable=true&PrivacyMasking.ShapType=Rect&PrivacyMasking.Rect[0]=0&PrivacyMasking.Rect[1]=0&PrivacyMasking.Rect[2]=50&PrivacyMasking.Rect[3]=100&PrivacyMasking.Mosaic=8		
Response	OK		

4.5.21 Goto Privacy Masking

Syntax	http://<*server*>/cgi-bin/PrivacyMasking.cgi?action=gotoPrivacyMasking		
Method	GET		
Description	Goto privacy masking		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
index	int	R	Index of PrivacyMasking which starts from 0
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=gotoPrivacyMasking&channel=2&index=3		
Response	OK		

4.5.22 Delete Privacy Masking

Syntax	http://<*server*>/cgi-bin/PrivacyMasking.cgi?action=deletePrivacyMasking		
Method	GET		
Description	Delete privacy masking		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
index	int	R	Index of PrivacyMasking which starts from 0
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=deletePrivacyMasking&channel=2&index=3		
Response	OK		

4.5.23 Clear Privacy Masking

Syntax	http://<*server*>/cgi-bin/PrivacyMasking.cgi?action=clearPrivacyMasking		
Method	GET		
Description	Clear privacy masking		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=clearPrivacyMasking&channel=2		
Response	OK		

4.5.24 Get Privacy Masking Rect

Syntax	http://<*server*>/cgi-bin/PrivacyMasking.cgi?action=getRealRect		
Method	GET		
Description	Get privacy masking rect		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	video channel index which starts from 1
index	int	R	Index of PrivacyMasking which starts from 0
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
Rect	Array<int>	R	Rect, top left and bottom right point, each point has x and y value, oordinate remap to 0 — 8192.
[Example]			

Request	http://192.168.1.108/cgi-bin/PrivacyMasking.cgi?action=getRealRect&channel=2&index=1
Response	Rect[0]=0 Rect[1]=0 Rect[2]=50 Rect[3]=100

4.5.25 [Config] SmartMotionDetect

- Get smart motion detection setting

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=SmartMotionDetect		
Method	GET		
Description	Get smart motion detection setting		
[Request Params] (None)			
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
table	object	R	Config info table.
+SmartMotionDetect	Array<object>	R	PrivacyMasking config, each channel has one config object
++Enable	bool	R	Enable/Disable
++Sensitivity	string	R	Detection sensitivity, can be: "Low", "Middle", "High"
++ObjectTypes	object	R	Detection object type
+++Human	bool	R	Whether detect motion of human
+++Vehicle	bool	R	Whether detect motion of vehicle
[Example]			
Request	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=SmartMotionDetect		
Response	table.SmartMotionDetect[0].Enable=true table.SmartMotionDetect[0].Sensitivity=Middle table.SmartMotionDetect[0].ObjectTypes.Human=true table.SmartMotionDetect[0].ObjectTypes.Vejhicle=true …		

- Set smart motion detection setting

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set smart motion detection setting		
[Request Params] (key=value format)			
Name	Type	R/O	Param Description
SmartMotionDetect	Array<object>	R	Smart motion detection config, each channel has one config object
+Enable	bool	R	Enable/Disable
+Sensitivity	string	R	Detection sensitivity, can be: "Low", "Middle", "High"
+ObjectTypes	object	R	Detection object type

++Human	bool	R	Whether detect motion of human
++Vehicle	bool	R	Whether detect motion of vehicle
[Response Params] (OK)			
[Example]			
Request	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&SmartMotionDetect[0].Enable=true&SmartMotionDetect[0].Sensitivity=Middle&SmartMotionDetect[0].ObjectTypes.Human=true&SmartMotionDetect[0].ObjectTypes.Vejhicle=true		
Response	OK		

4.6 System

4.6.1 General

- Get general config

Table 4-36

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=General
Method	GET
Description	Get general config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=General
Success Return	table.General.MachineName=Test001 table.General.LocalNo=8 table.General.MachineAddress="binjiangqv jiangnandadao weiyelu" table.General.MachineGroup="jiaojing yidui
Comment	—

- Set general config

Table 4-37

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set general config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&General.MachineName=MyIPC
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
General.MachineName	string	Device name or serial number.
General. LocalNo	integer	Local number for remote controller
General. MachineAddress	string	Address machine places in
General. MachineGroup	string	Group machine belongs to

4.6.2 Get Current Time

Table 4-38

Syntax	http://<*server*>/cgi-bin/global.cgi?action=getCurrentTime
Method	GET
Description	Get current time.
Example	http://192.168.1.108/cgi-bin/global.cgi?action=getCurrentTime
Success Return	result = 2011-7-3 21:02:32
Comment	The time format is "Y-M-D H-m-S". It's not be effected by Locales. TimeFormat in SetLocalesConfig.

4.6.3 Set Current Time

Table 4-39

Syntax	http://<*server*>/cgi-bin/global.cgi?action=setCurrentTime&time=2011-7-3%2021:02:32
Method	GET
Description	Set current time.
Example	http://192.168.1.108/cgi-bin/global.cgi?action=setCurrentTime&time=2016-01-01%2021:02:32
Success Return	OK
Comment	The time format is "Y-M-D H-m-S". It's not be effected by Locales. TimeFormat in SetLocalesConfig.

4.6.4 Locales

- Get locales config

Table 4-40

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=Locales
Method	GET
Description	Get locales config
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Locales
Success Return	table.Locales.DSTEnable=false table.Locales.DSTEnd.Day=1 table.Locales.DSTEnd.Hour=0 table.Locales.DSTEnd.Minute=0 table.Locales.DSTEnd.Month=1 table.Locales.DSTEnd.Week=2 table.Locales.DSTEnd.Year=2011 table.Locales.DSTStart.Day=0 table.Locales.DSTStart.Hour=0 table.Locales.DSTStart.Minute=0 table.Locales.DSTStart.Month=1 table.Locales.DSTStart.Week=1

	table.Locales.DSTStart.Year=2011 table.Locales.TimeFormat=yyyy-MM-dd HH:mm:ss
Comment	—

- Set locales config

Table 4-41

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set Locales config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Locales.DSTEnable=false
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
Locales.DSTEnable	bool	Enable/Disable DST (daylight saving time)
Locales.DSTEnd.Day	integer	Range is [0—6] or [1—31] [0—6]: week day, 0 = Sunday, 6 = Saturday [1—31]: month day If Locales.DSTEnd.Week is 0, use month day, otherwise, use week day.
Locales.DSTEnd.Hour	integer	Range is [0—23]
Locales.DSTEnd.Minute	integer	Range is [0—59]
Locales.DSTEnd.Month	integer	Range is [1—12]
Locales.DSTEnd.Week	Integer	Range is {1, 2, 3, 4, -1, 0}. 0 = Use month day [1, 2, 3, 4, -1]: use week day. 1 = first week, 2 = second, 3 = third, 4 = fourth, -1 = last.
Locales.DSTEnd.Year	Integer	Range is [2000-2038]
Locales.DSTStart.Day	Integer	Range is the same with items in Locales.DSTEnd. Locales.DSTStart table and Locales.DSTEnd table define the time range of DST.
Locales.DSTStart.Hour		
Locales.DSTStart.Minute		
Locales.DSTStart.Month		
Locales.DSTStart.Week		
Locales.DSTStart.Year		

ParamName	ParamValue type	Description
Locales.TimeFormat	string	Defines time format displayed in video time title. String form is: *year-month-day hour*:mm:ss. Position of *year*, *month* and *day* can be exchanged. Range of *year* is {yy, yyyy} yy = year without century, yyyy = year with century. Range of *month* is {M, MM, MMMM} M = 1 for January, MM = 01 for January, MMMM = Jan for January Range of *day* is {d, dd} d = 1 for first day, dd = 01 for first day Range of *hour* is {H, HH, h, hh} H = 1 for 1:00, HH = 01 for 1:00, range is 0-23 h = 1 for 1:00, hh = 01 for 1:00, time range is 1-12 Example: yyyy-MM-dd HH:mm:ss or MM-dd-yyyy HH:mm:ss or dd-M-yy hh:mm:ss

4.6.5 Get Language Capability

Table 4-42

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getLanguageCaps
Method	GET
Description	Get the list of supported languages.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getLanguageCaps
Success Return	Languages=SimpChinese,English,French
Comment	response is a string contains languages with comma separated. Languages include {English, SimpChinese, TradChinese, Italian, Spanish, Japanese, Russian, French, German}

4.6.6 Language

- Get language config

Table 4-43

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=Language
Method	GET
Description	Get system language config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Language
Success Return	table.Language=SimpChinese
Comment	—

- Set language config

Table 4-44

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set system language config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Language=SimpChinese
Success Return	OK
Comment	NOTE: After changing language setting, system will automatically reboot!

Appendix:

ParamName	ParamValue type	Description
Language	string	The language range is get from interface in **GetLanguageCaps**

4.6.7 Get Client Access Filter

Table 4-45

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=AccessFilter
Method	GET
Description	Get access filter config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=AccessFilter
Success Return	table.AccessFilter.Enable=false table.AccessFilter.TrustList[0]=10.6.10.23 table.AccessFilter.TrustList[1]=10.6.10.62 table.AccessFilter.**Type**=TrustList
Comment	Parameters in Response: **Type**: Range is {TrustList, BannedList}. If Trustlist is supported, Bannedlist is not supported. More information in Appendix.

Appendix:

ParamName	ParamValue type	Description
AccessFilter.BannedList[***index***]	string	Banned IP address list
AccessFilter.TrustList[***index***]	string	Trusted IP address list
AccessFilter.Enable	bool	Enable/Disable access filter function
AccessFilter.Type	string	Range is {TrustList, BannedList}, TrustList: Trust list is used, banned list is not used. BannedList: Banned list is used, trust list is not used.

4.6.8 Auto Maintain

- Get auto maintain config

Table 4-46

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=AutoMaintain
Method	GET
Description	Get Auto Maintain config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=AutoMaintain
Success Return	table.AutoMaintain. AutoRebootDay=3 table.AutoMaintain. AutoRebootHour=0 table.AutoMaintain. AutoRebootMinute=0 table.AutoMaintain. AutoShutdownDay=1 table.AutoMaintain. AutoShutdownHour=0 table.AutoMaintain. AutoShutdownMinute=0 table.AutoMaintain. AutoStartUpDay=1 table.AutoMaintain. AutoStartUpHour=2 table.AutoMaintain. AutoStartUpMinute=0
Comment	—

- Set auto maintain config

Table 4-47

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set auto maintain config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AutoMaintain.AutoRebootDay=7
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
AutoMaintain. AutoRebootDay	integer	Range is [-1—7]. Auto restart day. -1 = never auto restart 0—6 = Sunday—Saturday 7 = restart every day
AutoMaintain. AutoRebootHour	integer	Range is [0—23]. Auto restart hour
AutoMaintain. AutoRebootMinute	integer	Range is [0—59]. Auto restart minute
AutoMaintain. AutoShutdownDay	integer	auto reboot time Range is same with AutoOpenDay, AutoOpenHour and AutoOpenMinute.
AutoMaintain. AutoShutdownHour		

ParamName	ParamValue type	Description
AutoMaintain. AutoShutdownMinute		
AutoMaintain. AutoStartUpDay	integer	Auto shutdown time. Range is same with AutoOpenDay, AutoOpenHour, and AutoOpenMinute.
AutoMaintain. AutoStartUpHour		
AutoMaintain. AutoStartUpMinute		

4.6.9 Holiday Management

- Get holiday config

Table 4-48

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=Holiday
Method	GET
Description	Get holiday config for record or snap.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Holiday
Success Return	table.Holiday.MonthMask[0]=3 table.Holiday.MonthMask[1]=0 table.Holiday.MonthMask[2]=0 table.Holiday.MonthMask[3]=0 table.Holiday.MonthMask[4]=0 table.Holiday.MonthMask[5]=0 table.Holiday.MonthMask[6]=0 table.Holiday.MonthMask[7]=0 table.Holiday.MonthMask[8]=0 table.Holiday.MonthMask[9]=1610612739 table.Holiday.MonthMask[10]=0 table.Holiday.MonthMask[11]=0
Comment	—

- Set holiday config

Table 4-49

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set holiday config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Holiday.MonthMask[0]=3
Success Return	OK
Comment	Parameters in URL: In table below, ***monthIndex*** presents the index of a month. 0 presents January, 1 presents February, 11 presents December.

Appendix:

ParamName	ParamValue type	Description
Holiday.MonthMask[***monthIndex***]	integer	It is the mask of a month.Every bit present a day. For example, 0x0001 presents the first day of a month is holiday. 0x0002 presents the second day of a month is holiday, 0x0003 presents the first day and second day of a month is holiday.

4.6.10 Get Device Type

Table 4-50

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getDeviceType
Method	GET
Description	Get the device type displaying which is not the real type.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getDeviceType
Success Return	type=DVR
Comment	—

4.6.11 Get Hardware Version

Table 4-51

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getHardwareVersion
Method	GET
Description	Get the device hardware version.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getHardwareVersion
Success Return	version=1.00
Comment	—

4.6.12 Get Serial Number of Device

Table 4-52

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getSerialNo
Method	GET
Description	Get the device serial number.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getSerialNo
Success Return	sn=YZC0GZ05100020
Comment	—

4.6.13 Get Machine Name

Table 4-53

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getMachineName
Method	GET

Description	Get the device machine name.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getMachineName
Success Return	name=YZC0GZ05100020
Comment	—

4.6.14 Get System Information

Table 4-54

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getSystemInfo
Method	GET
Description	Get the system information.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getSystemInfo
Success Return	serialNumber= PA1FQ15900207 deviceType=27 processor= ST7108
Comment	—

4.6.15 Get Vendor Information

Table 4-55

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getVendor
Method	GET
Description	Get the vendor information.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getVendor
Success Return	vendor=TTT
Comment	—

4.6.16 Get Software Information

Table 4-56

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getSoftwareVersion
Method	GET
Description	Get the software information.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getSoftwareVersion
Success Return	version=2.212.0000.0.R,build:2013-11-14
Comment	—

4.6.17 Get Version of Onvif

Table 4-57

Syntax	http://<*server*>/cgi-bin/IntervideoManager.cgi?action=getVersion&Name=Onvif
Method	GET
Description	Get onvif version.

Example	http://192.168.1.108/cgi-bin/IntervideoManager.cgi?action=getVersion&Name=Onvif
Success Return	version=2.4.2
Comment	—

4.6.18 Get Version of HTTP API

Table 4-58

Syntax	http://<*server*>/cgi-bin/IntervideoManager.cgi?action=getVersion&Name=CGI
Method	GET
Description	Get CGI version.
Example	http://192.168.1.108/cgi-bin/IntervideoManager.cgi?action=getVersion&Name=CGI
Success Return	version=2.0.0
Comment	—

4.6.19 Get Device Class

Table 4-59

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getDeviceClass
Method	GET
Description	Get the device class.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getDeviceClass
Success Return	class=HDVR
Comment	—

4.6.20 Onvif Service Authorization

- Get config of Onvif service authorization

Table 4-60

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=UserGlobal
Method	GET
Description	Get user global config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=UserGlobal
Success Return	table.UserGlobal.OnvifLoginCheck=false
Comment	If "OnvifLoginCheck" is false, you can get Onvif service directly; if true, you should enter your ID/username and password.

- Set config of Onvif service authorization

Table 4-61

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&UserGlobal.OnvifLoginCheck=<***flag***>
Method	GET
Description	Enable onvif login check or not.

Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&UserGlobal.OnvifLoginCheck=true
Success Return	OK
Comment	Parameters in URL: ***flag*** : range is {true, false}.

4.6.21 Backup of Config

Table 4-62

Syntax	http://<*server*>/cgi-bin/Config.backup?action=All
Method	GET
Description	Download all the settings of a device as a file named Config.Backup by default.
Example	http://192.168.1.108/cgi-bin/Config.backup?action=All

Success Return	HTTP/1.1 200 OK
	CONTENT-LENGTH: 743087
	CONNECTION: close
	Content-type: application/binarytet-stream; charset=utf-8
	
	{
	"ATM" : {
	"DataSource" : "RS232",
	"DisplayPostion" : "lefttop",
	"EncodeBlend" : true,
	"PreviewBlend" : true,
	"ProtocolAbility" : ["POS"],
	"ProtocolName" : "ATM\/POS",
	"RecordChannels" : [0, 1, 2, 3]
	}
	……
	}

Comment	—

4.6.22 Restore the Config

Table 4-63

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=restore&names[0]=<***xxx***>&names[1]=<***yyy***>[&…]
Method	GET
Description	Restore config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=restore&names[0]=UPnP
Success Return	OK
Comment	Parameters in URL: ***xxx*** and ***yyy*** is config name which need to be restore

4.6.23 Restore except the Config

Table 4-64

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=restoreExcept&names[0]=<***xxx***>&names[1]=<***yyy***>[&…]
Method	GET
Description	Restore all config except several.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=restoreExcept&names[0]=UPnP
Success Return	OK
Comment	Parameters in URL: All the config file but ***xxx*** and ***yyy*** will be restored.

4.6.24 Reboot

Table 4-65

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=reboot[&delay=<*paramValue*>]
Method	GET
Description	Reboot the device.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=reboot
Success Return	OK
Comment	If successful, response OK. If fail, response Error.

4.6.25 Shutdown

Table 4-66

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=shutdown
Method	GET
Description	Shutdown the device.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=shutdown
Success Return	OK
Comment	If successful, response OK. If fail, response Error.

4.6.26 FactoryReset

Table 4-67

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=resetSystemEx[&type=<***type***>]
Method	GET
Description	Reset the configuration for the device to the factory default.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=resetSystemEx&type=0
Success Return	OK
Comment	Parameters in URL: ***Type***: integer, it can be 0 or1.

	0 means all parameters are set to their factory default value; 1 means all parameters are set to their factory default value expect the specific parameters; The specific parameters of different device types are different. But it always contains the network settings and user settings. After resetting, the device is reachable on the same ip address as used before the reset. When the ***Type*** parameter is not present in the URL, the default value of the ***Type*** is 0.

4.6.27 Get Tracing Code of Device

Table 4-68

Syntax	http://<*server*>/cgi-bin/magicBox.cgi?action=getTracingCode
Method	GET
Description	Get the tracing code of the device.
Example	http://192.168.1.108/cgi-bin/magicBox.cgi?action=getTracingCode
Success Return	**tc**=0001...1101
Comment	Only system default admin can get it. **tc** is a 64bit binary string.

4.7 Network

4.7.1 Get Network Interfaces

Table 4-69

Syntax	http://<*server*>/cgi-bin/netApp.cgi?action=getInterfaces
Method	GET
Description	Get all of the system network interfaces.
Example	http://192.168.1.108/cgi-bin/netApp.cgi?action=getInterfaces
Success Return	netInterface[0].**Name**=eth0 netInterface[0].**Type**=Normal netInterface[0].**Valid**=true …
Comment	result item value: **Name**: network interface name. "eth0" - wired network interface "eth2" - wireless network interface "3G" - 3G network interface **Type**: "Normal" — wired network "Wireless" — wireless network "Auto", "TD-SCDMA", "WCDMA", "CDMA1x", "EDGE", "EVDO" — 3G network types. **Valid**: network interface is valid if netInterface[n].Valid is true.

4.7.2 Network Basic Config

- Get network config

Table 4-70

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=Network
Method	GET
Description	Get network basic config. The basic config contains basic network parameters (default interface, domain name, host name), and configuration of each network interface.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Network
Success Return	table.Network.DefaultInterface=eth0 table.Network.Domain=ttt table.Network.Hostname=badak table.Network.***interface***.DefaultGateway=10.7.0.1 table.Network.***interface***.DhcpEnable=false table.Network.***interface***.DnsServers[0]=221.123.33.228 table.Network.***interface***.DnsServers[1]=221.12.1.228 table.Network.***interface***.IPAddress=10.7.2.3 table.Network.***interface***.MTU=1500 table.Network.***interface***.PhysicalAddress=00:10:5c:f2:1c:b4 table.Network.***interface***.SubnetMask=255.255.0.0
Comment	***interface*** in response is network interface name, such as eth0, eth2…

- Set network config

Table 4-71

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set network basic config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&NetWork.Domain=ttt&NetWork.eth0.DhcpEnable=true
Success Return	OK
Comment	***interface*** in below ParamName is network interface name, such as eth0 and eth2

Appendix:

ParamName	ParamValue type	Description
Network. DefaultInterface	string	Set default network interface when multiple interfaces exist. Range of interfaces is depends on **GetInterfaces**.
Network. Domain	string	Domain name.
Network. Hostname	string	Hostname and Domain compose a network address.
Network. ***interface***.DefaultGateway	string	IP address.
Network.	bool	Enable/Disable DHCP.

ParamName	ParamValue type	Description
interface.DhcpEnable		
Network. ***interface***.DnsServers[0]	string	IP address of first DNS server.
Network. ***interface***.DnsServers[1]	string	IP address of second DNS server.
Network. ***interface***.IPAddress	string	Interface IP address.
Network. ***interface***.MTU	integer	Interface MTU.
Network. ***interface***.PhysicalAddress	string	MAC address of interface. HEX string in the form of: xx:xx:xx:xx:xx:xx. Range of x is [0-9, a-f, A-F] Example: 00:10:5c:f2:1c:b4 00:10:5C:F2:1C:B5
Network. ***interface***.SubnetMask	string	Network mask string: In the form of x.x.x.x, range of x is [0-255] Example: 255.255.255.0

4.7.3 PPPoE

- Get PPPoE config

Table 4-72

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=PPPoE
Method	GET
Description	Get PPPoE config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=PPPoE
Success Return	table.PPPoE.Enable=false table.PPPoE.Password=123456 table.PPPoE.UserName=123456
Comment	—

- **Set PPPoE config**

Table 4-73

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set PPPoE config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&PPPoE.UserName=user1&PPPoE.Password=123456
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
PPPoE. Enable	bool	Enable/Disable PPPoE.
PPPoE. UserName	string	PPPoE user name.
PPPoE. Password	string	PPPoE user password.

4.7.4 DDNS

- Get DDNS config

Table 4-74

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=DDNS
Method	GET
Description	Get DDNS config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=DDNS
Success Return	table.DDNS[***index***].Address=www.ttt.com table.DDNS[***index***].Enable=true table.DDNS[***index***].HostName=www.ttt.com table.DDNS[***index***].KeepAlive=10 table.DDNS[***index***].Password=none table.DDNS[***index***].Port=5050 table.DDNS[***index***].Protocol= Quick DDNS table.DDNS[***index***].UserName=user1 table.DDNS[***index***].DefaultHostName.Enable=false table.DDNS[***index***].DefaultHostName.HostName=9002A9D77133.quickddns.com
Comment	***index*** in response is the DDNS protocol table index, start from 0. The meaning of parameters can refer to SetDDNSConfig chapter.

- Set DDNS config

Table 4-75

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set DDNS config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&DDNS[0].Address=www.ttt.com&DDNS[0].Enable=true
Success Return	OK
Comment	***index*** in below ParamName is the DDNS protocol table index, start from 0.

Appendix:

ParamName	ParamValue type	Description
DDNS[***index***].Address	string	DDNS server IP address or name.
DDNS[***index***].Enable	bool	Multiple DDNS hostname can be configured, but Only one hostname can be enabled, others should be disabled.
DDNS[***index***].HostName	String	Hostname of this device.

ParamName	ParamValue type	Description
DDNS[***index***].KeepAlive	integer	Range is [1—65535]. Unit is minutes.
DDNS[***index***].Password	string	DDNS user password
DDNS[***index***].Port	integer	Range is [1—65535]. Port of DDNS server
DDNS[***index***].Protocol	string	DDNS protocol type. Range is {"NO-IP DDNS", "Dyndns DDNS", "Private DDNS", "DHDDNS", "QUICK DDNS"}.
DDNS[***index***].UserName	string	DDNS user name
DDNS[***index***].DefaultHostName.Enable	bool	Only protocol is in range {"Private DDNS", "DHDDNS", "QUICK DDNS"}, it effects. true : use the DefaultHostName.HostName false: use the HostName
DDNS[***index***].DefaultHostName.HostName	string	The default hostname. It cannot be modified.

4.7.5 Email

- Get email config

Table 4-76

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=Email
Method	GET
Description	Get Email config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Email
Success Return	table.Email.Address=www.ttt.com table.Email.Anonymous=true table.Email.AttachEnable=true table.Email.AttachmentEnable=true table.Email.Enable=true table.Email.HealthReport.Enable=false table.Email.HealthReport.Interval=61 table.Email.Password=123456 table.Email.Port=26 table.Email.Receivers[0]=x@tttt.com table.Email.Receivers[1]=y@ttt.com table.Email.Receivers[2]=z@ttt.com table.Email.SendAddress=x@ttt.com table.Email.SslEnable=false table.Email.Title=DVRMessage table.Email.UserName=anonymity
Comment	—

- Set email config

Table 4-77

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramV***

	alue>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set Email config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Email.Address=mail.ttt.com&Email.Anonymous=false
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
Email. Address	string	SMTP server IP address or name.
Email. Anonymous	bool	Enable/Disable anonymous email.
Email. AttachEnable	bool	Enable/Disable email attachment
Email. AttachmentEnable	bool	Enable/Disable email attachment
Email. Enable	bool	Enable/Disable email function
Email. HealthReport.Enable	bool	Enable/Disable report device status by email.
Email. HealthReport.Interval	integer	Range is [30-1440]. Unit is minutes
Email. Password	string	User password of email account.
Email. Port	integer	Range is [1-65535]
Email. Receivers[0]	string	Email addresses of 3 receivers.
Email. Receivers[1]	string	
Email. Receivers[2]	string	
Email. SendAddress	string	Sender email address.
Email. SslEnable	bool	True: enable SSL email.
Email. Title	string	Title of email.
Email. UserName	string	User name of email account.

4.7.6 WLan

- Get WLan config

Table 4-78

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=WLan
Method	GET
Description	Get Wlan config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=WLan
Success Return	table.WLan.eth2.Enable=true table.WLan.eth2.Encryption=off table.WLan.eth2.KeyFlag=false table.WLan.eth2.KeyID=0 table.WLan.eth2.KeyType=Hex table.WLan.eth2.Keys[0]=password1 table.WLan.eth2.Keys[1]=password2

	table.WLan.eth2.Keys[2]=password3 table.WLan.eth2.Keys[3]=password4 table.WLan.eth2.LinkMode=Auto table.WLan.eth2.SSID=ttt
Comment	—

- Set WLan config

Table 4-79

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set WLan config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&WLan.eth2.Enable=true&WLan.eth2.KeyType=Hex
Success Return	OK
Comment	In below ParamName, ***interface*** is name of wireless interface.

Appendix:

ParamName	ParamValue type	Description
WLan.***interface***.Enable	bool	True: Enable WLan on this interface.
WLan.***interface***.Encryption	string	Range is {Off, On, WEP64Bits, WEP128Bits, WPA-PSK-TKIP, WPA-PSK-CCMP} Encryption mode.
WLan.***interface***.KeyFlag	bool	true: key is configured.
WLan.***interface***.KeyID	integer	Range is [0—3] Indicates which key is used. 0: WLan.*interface*.Keys[0] is used.
WLan.***interface***.KeyType	string	Range is {Hex, ASCII]
WLan.***interface***.Keys[0]	string	For ASCII key type: 64bits encryption key length is 5, 128bits encryption key length is 13, consists of [0—9, a—z, A—Z] For HEX key type: 64bits encryption key length is 10, 128bits encryption key length is 26, consists of [0—9, a—z, A—Z]
WLan.***interface***.Keys[1]	string	
WLan.***interface***.Keys[2]	string	
WLan.***interface***.Keys[3]	string	
WLan.***interface***.LinkMode	string	Range is {Auto, Ad-hoc, and Infrastructure}. Auto – select suitable mode automatically. Ad-hoc – Device with wireless network adapter can connect to each other without Access Point. Infrastructure – Integrate wire and wireless LAN together to share network resource, access point is need in this mode.
WLan.***interface***.SSID	string	SSID.

4.7.7 Scan Wlan Devices

Table 4-80

Syntax	http://<*server*>/cgi-bin/wlan.cgi?action=scanWlanDevices&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Search Wi-Fi device information.
Example	http://192.168.1.108/cgi-bin/wlan.cgi?action=scanWlanDevices&SSID=xia_yuguo 13098 Internet
Success Return	found=1 wlanDevice[0].ApConnected=0 wlanDevice[0].ApMaxBitRate=54000000 wlanDevice[0].ApNetWorkType=255 wlanDevice[0].AuthMode=7 wlanDevice[0].BSSID=28:2c:b2:5c:de:36 wlanDevice[0].EncrAlgr=3 wlanDevice[0].LinkMode=0 wlanDevice[0].LinkQuality=31 wlanDevice[0].RSSIQuality=0 wlanDevice[0].SSID=xia_yuguo 13098 Internet
Comment	—

Appendix:

ParamName	ParamValue type	Description
SSID	string	Specified SSID, if not include any SSID, all Wi-Fi information will be searched and displayed.

4.7.8 UPnP

- Get UPnP config

Table 4-81

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=UPnP
Method	GET
Description	Get UPnP config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=UPnP
Success Return	table.UPnP.Enable=true table.UPnP.MapTable[***index***].Enable=true table.UPnP.MapTable[***index***].InnerPort=80 table.UPnP.MapTable[***index***].OuterPort=8080 table.UPnP.MapTable[***index***].Protocol=TCP table.UPnP.MapTable[***index***].ServiceName=HTTP
Comment	***index*** in response is the UPNP map table index, start from 0.

- Set UPnP config

Table 4-82

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set UPnP config.

Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&UPnP.Enable=true& UPnP.MapTable[0].Protocol=TCP
Success Return	OK
Comment	index in below ParamName is UPNP map table index, range is [0-255]

Appendix:

ParamName	ParamValue type	Description
UPnP.Enable	bool	Enable/Disable UPNP feature.
UPnP.MapTable[***index***].Enable	bool	Enable/Disable this UPNP map.
UPnP.MapTable[***index***].InnerPort	integer	Range is [1—65535]. Inner port number
UPnP.MapTable[***index***].OuterPort	integer	Range is [1—65535]. Outer port number.
UPnP.MapTable[***index***].Protocol	string	Range is {TCP, UDP]
UPnP.MapTable[***index***].ServiceName	string	User defined UPnP service name.

4.7.9 Get UPnP Status

Table 4-83

Syntax	http://<*server*>/cgi-bin/netApp.cgi?action=getUPnPStatus
Method	GET
Description	Get UPnP Status.
Example	http://192.168.1.108/cgi-bin/netApp.cgi?action=getUPnPStatus
Success Return	status.InnerAddress=0.0.0.0 status.OuterAddress=0.0.0.0 status.PortMapStatus[0]=Failed status.PortMapStatus[1]=Failed status.PortMapStatus[2]=Failed status.PortMapStatus[3]=Failed status.Status=Unknown status.Working=false
Comment	—

4.7.10 NTP

- Get NTP config

Table 4-84

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=NTP
Method	GET
Description	Get NTP config
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=NTP
Success Return	table.NTP.Address=clock.isc.org table.NTP.Enable=false table.NTP.Port=38 table.NTP.TimeZone=9

	table.NTP.UpdatePeriod=31
Comment	—

- Set NTP config

Table 4-85

Syntax	http://<***server***>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set NTP config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&NTP.Address=time.ttt.com&NTP.Enable=true
Success Return	OK
Comment	—

4.7.11 RTSP

- Get RTSP config

Table 4-86

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=RTSP
Method	GET
Description	Get RTSP config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=RTSP
Success Return	table.RTSP.Enable=true table.RTSP.Port=554 table.RTSP.RTP.EndPort=40000 table.RTSP.RTP.StartPort=20000
Comment	—

- Set RTSP config

Table 4-87

Syntax	http://<***server***>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set RTSP config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&RTSP.Enable=true&RTSP.Port=554
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
RTSP.Enable	bool	Enable/Disable RTSP.
RTSP.Port	integer	RTSP port.
RTSP.RTP.StartPort	integer	RTP start port.
RTSP.RTP.EndPort	integer	RTP end port.

4.7.12 Alarm Server

- Get alarm server config

Table 4-88

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=AlarmServer
Method	GET
Description	Get alarm server config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=AlarmServer
Success Return	table.AlarmServer.Address=10.7.8.9 table.AlarmServer.Enable=false table.AlarmServer.Password= table.AlarmServer.Port=8888 table.AlarmServer.Protocol=ttt table.AlarmServer.ReportTime=02:00:00 table.AlarmServer.ReportWeekDay=2 table.AlarmServer.UserName=admin
Comment	—

- Set alarm server config

Table 4-89

Syntax	http://<***server***>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set Alarm Server config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AlarmServer.Address=as.ttt.com&AlarmServer.Enable=false
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
AlarmServer. Address	string	Alarm server IP address or name.
AlarmServer. Enable	bool	Enable/Disable Alarm server.
AlarmServer. Port	integer	Range is [1—65535]. Port of Alarm server.

4.8 Motion Detection

4.8.1 Motion Detection Settings

- Get motion detect config

Table 4-90

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=MotionDetect
Method	GET

Description	Motion detect config of a video channel contains Enable, MotionDetectWindow and EventHandler.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=MotionDetect
Success Return	table.MotionDetect[0].Enable=false table.MotionDetect[0].EventHandler.AlarmOut=1 table.MotionDetect[0].EventHandler.AlarmOutChannels[0]=0 table.MotionDetect[0].EventHandler.AlarmOutEnable=true table.MotionDetect[0].EventHandler.AlarmOutLatch=10 table.MotionDetect[0].EventHandler.BeepEnable=false table.MotionDetect[0].EventHandler.Dejitter=5 table.MotionDetect[0].EventHandler.Delay=0 table.MotionDetect[0].EventHandler.ExAlarmOut=1 table.MotionDetect[0].EventHandler.ExAlarmOutChannels[0]=0 table.MotionDetect[0].EventHandler.ExAlarmOutEnable=false table.MotionDetect[0].EventHandler.FlashEnable=false table.MotionDetect[0].EventHandler.FlashLatch=10 table.MotionDetect[0].EventHandler.LogEnable=true table.MotionDetect[0].EventHandler.MailEnable=false table.MotionDetect[0].EventHandler.Matrix=1 table.MotionDetect[0].EventHandler.MatrixChannels[0]=0 table.MotionDetect[0].EventHandler.MatrixEnable=false table.MotionDetect[0].EventHandler.MessageEnable=false table.MotionDetect[0].EventHandler.PtzLink[0][0]=None table.MotionDetect[0].EventHandler.PtzLink[0][1]=1 table.MotionDetect[0].EventHandler.PtzLinkEnable=false table.MotionDetect[0].EventHandler.Record=1 table.MotionDetect[0].EventHandler.RecordChannels[0]=0 table.MotionDetect[0].EventHandler.RecordEnable=true table.MotionDetect[0].EventHandler.RecordLatch=10 table.MotionDetect[0].EventHandler.Snapshot=1 table.MotionDetect[0].EventHandler.SnapshotChannels[0]=0 table.MotionDetect[0].EventHandler.SnapshotEnable=false table.MotionDetect[0].EventHandler.TimeSection[0][0]=1 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[0][1]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[0][2]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[0][3]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[0][4]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[0][5]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[1][0]=1 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[1][1]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[1][2]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[1][3]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[1][4]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[1][5]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[2][0]=1 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[2][1]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[2][2]=0 00:00:00-23:59:59

	table.MotionDetect[0].EventHandler.TimeSection[2][3]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[2][4]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[2][5]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[3][0]=1 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[3][1]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[3][2]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[3][3]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[3][4]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[3][5]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[4][0]=1 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[4][1]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[4][2]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[4][3]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[4][4]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[4][5]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[5][0]=1 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[5][1]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[5][2]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[5][3]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[5][4]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[5][5]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[6][0]=1 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[6][1]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[6][2]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[6][3]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[6][4]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TimeSection[6][5]=0 00:00:00-23:59:59 table.MotionDetect[0].EventHandler.TipEnable=false table.MotionDetect[0].EventHandler.Tour=1 table.MotionDetect[0].EventHandler.TourChannels[0]=0 table.MotionDetect[0].EventHandler.TourEnable=false table.MotionDetect[0].EventHandler.Voice.AudioFileName= table.MotionDetect[0].EventHandler.VoiceEnable=false table.MotionDetect[0].MotionDetectWindow[0].Id=0 table.MotionDetect[0].MotionDetectWindow[0].Name=Region1 table.MotionDetect[0].MotionDetectWindow[0].Region[0]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[1]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[2]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[3]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[4]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[5]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[6]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[7]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[8]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[9]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[10]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[11]=4194303

	table.MotionDetect[0].MotionDetectWindow[0].Region[12]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[13]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[14]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[15]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[16]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Region[17]=4194303 table.MotionDetect[0].MotionDetectWindow[0].Sensitive=60 table.MotionDetect[0].MotionDetectWindow[0].Threshold=5 table.MotionDetect[0].MotionDetectWindow[0].Window[0]=0 table.MotionDetect[0].MotionDetectWindow[0].Window[1]=0 table.MotionDetect[0].MotionDetectWindow[0].Window[2]=8191 table.MotionDetect[0].MotionDetectWindow[0].Window[3]=8191 table.MotionDetect[0].MotionDetectWindow[1].Id=1 table.MotionDetect[0].MotionDetectWindow[1].Name=Region2 table.MotionDetect[0].MotionDetectWindow[1].Region[0]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[1]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[2]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[3]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[4]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[5]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[6]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[7]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[8]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[9]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[10]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[11]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[12]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[13]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[14]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[15]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[16]=0 table.MotionDetect[0].MotionDetectWindow[1].Region[17]=0 table.MotionDetect[0].MotionDetectWindow[1].Sensitive=60 table.MotionDetect[0].MotionDetectWindow[1].Threshold=5 table.MotionDetect[0].MotionDetectWindow[1].Window[0]=0 table.MotionDetect[0].MotionDetectWindow[1].Window[1]=0 table.MotionDetect[0].MotionDetectWindow[1].Window[2]=0 table.MotionDetect[0].MotionDetectWindow[1].Window[3]=0 table.MotionDetect[0].MotionDetectWindow[2].Id=2 table.MotionDetect[0].MotionDetectWindow[2].Name=Region3 table.MotionDetect[0].MotionDetectWindow[2].Region[0]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[1]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[2]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[3]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[4]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[5]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[6]=0

	table.MotionDetect[0].MotionDetectWindow[2].Region[7]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[8]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[9]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[10]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[11]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[12]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[13]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[14]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[15]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[16]=0 table.MotionDetect[0].MotionDetectWindow[2].Region[17]=0 table.MotionDetect[0].MotionDetectWindow[2].Sensitive=60 table.MotionDetect[0].MotionDetectWindow[2].Threshold=5 table.MotionDetect[0].MotionDetectWindow[2].Window[0]=0 table.MotionDetect[0].MotionDetectWindow[2].Window[1]=0 table.MotionDetect[0].MotionDetectWindow[2].Window[2]=0 table.MotionDetect[0].MotionDetectWindow[2].Window[3]=0 table.MotionDetect[0].MotionDetectWindow[3].Id=3 table.MotionDetect[0].MotionDetectWindow[3].Name=Region4 table.MotionDetect[0].MotionDetectWindow[3].Region[0]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[1]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[2]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[3]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[4]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[5]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[6]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[7]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[8]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[9]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[10]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[11]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[12]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[13]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[14]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[15]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[16]=0 table.MotionDetect[0].MotionDetectWindow[3].Region[17]=0 table.MotionDetect[0].MotionDetectWindow[3].Sensitive=60 table.MotionDetect[0].MotionDetectWindow[3].Threshold=5 table.MotionDetect[0].MotionDetectWindow[3].Window[0]=0 table.MotionDetect[0].MotionDetectWindow[3].Window[1]=0 table.MotionDetect[0].MotionDetectWindow[3].Window[2]=0 table.MotionDetect[0].MotionDetectWindow[3].Window[3]=0 table.MotionDetect[0].OsdTwinkleEnable=false table.MotionDetect[0].PirMotionLevel=3
Comment	—

- Set motion detect config

Table 4-91

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set motion detection config.
Example	Enable motion detection: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&MotionDetect[0].Enable=true Set motion detection regions: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&MotionDetect[0].MotionDetectWindow[0].Region[0]=1&MotionDetect[0].MotionDetectWindow[0].Region[1]=1&MotionDetect[0].MotionDetectWindow[0].Region[2]=1&MotionDetect[0].MotionDetectWindow[0].Region[3]=1&MotionDetect[0].MotionDetectWindow[0].Region[4]=1&MotionDetect[0].DetectVersion=V3.0
Success Return	OK
Comment	In table below, ***head*** = MotionDetect[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel(equals to video channel index -1, and so 0 means channel 1). ***LineNum*** Index of region, region is divided into lines and each line has several blocks, a line is described by a 32 bit integer, a bit for a block. 0=Line 1 1=Line 2 … ***WinNum*** Index of detect window, there are 4 detect windows at present. Each window is divided into 18 lines and 22 blocks per line. **Notice**: When setting "MotionDetect[***ChannelNo***].MotionDetectWindow[***WinNum***].Region", you need to contain the parameter "MotionDetect[***ChannelNo***].DetectVersion=V3.0" along.

Appendix:

ParamName	ParamValue type	Description
head. Enable	bool	Enable/Disable motion detect feature in a channel.
head. EventHandler	—	Setting of EventHandler is described in **SetEventHandler.**
head.MotionDetectWindow [***WinNum***].Id	integer	It is the Id of a detect window.
head.MotionDetectWindow [***WinNum***].Name	string	It is the name of a detect window.

ParamName	ParamValue type	Description
head.MotionDetectWindow [***WinNum***].Sensitive	integer	Range is [0—100]. Sensitivity of motion detection. It presents more sensitive if the value is larger.
*head.*MotionDetectWindow [***WinNum***].Threshold	integer	Range is [0—100]. It presents the threshold value when trigger motion detect.
*head.*MotionDetectWindow [***WinNum***].Region[***LineNum***]	integer	It is similar with head.Region [LineNum]. Currently, a region is divided into 18 lines and 22 blocks per line. A bit describes a block in the line. Bit = 1: motion in this block is monitored. Example: MotionDetect [0].Region [0] = 4194303 (0x3FFFFF): the 22 blocks in channel 0 line 0 is monitored. MotionDetect [0].Region [1] =0: the 22 blocks in channel 0 line 1 is not monitored. MotionDetect [0].Region [17] = 3: the left two blocks in the last line of channel 0 is monitored.

4.9 Event

4.9.1 Event Handler

- Get event handler config

Table 4-92

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=<***handlerName***>
Method	GET
Description	Get event handler settings.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Alarm[0].EventHandler
Success Return	***handlerName***.EventHandler.AlarmOutChannels[0]=1 ***handlerName***.EventHandler.AlarmOutChannels[1]=1 … ***handlerName***.EventHandler.AlarmOutEnable=false ***handlerName***.EventHandler.AlarmOutLatch=10 ***handlerName***.EventHandler.BeepEnable=true ***handlerName***.EventHandler.Dejitter=0 ***handlerName***.EventHandler.Delay=30 ***handlerName***.EventHandler.LogEnable=true ***handlerName***.EventHandler.MailEnable=true ***handlerName***.EventHandler.PtzLink[0][0]=None ***handlerName***.EventHandler.PtzLink[0][1]=0 ***handlerName***.EventHandler.PtzLink[1][0]=None ***handlerName***.EventHandler.PtzLink[1][1]=0 …

	handlerName.EventHandler.PtzLinkEnable=false ***handlerName***.EventHandler.RecordChannels[0]=1 ***handlerName***.EventHandler.RecordChannels[1]=1 … ***handlerName***.EventHandler.RecordEnable=true ***handlerName***.EventHandler.RecordLatch=10 ***handlerName***.EventHandler.SnapshotChannels[0]=1 ***handlerName***.EventHandler.SnapshotChannels[1]=1 … ***handlerName***.EventHandler.SnapshotEnable=false ***handlerName***.EventHandler.SnapshotPeriod=3 ***handlerName***.EventHandler.SnapshotTimes=0 ***handlerName***.EventHandler.TimeSection[0][0]=1 01:00:00-24:00:00 ***handlerName***.EventHandler.TimeSection[0][1]=1 01:00:00-24:00:00… … ***handlerName***.EventHandler.TimeSection[6][5]=1 01:00:00-24:00:00 ***handlerName***.EventHandler.TipEnable=true ***handlerName***.EventHandler. ExAlarmOutEnable=true ***handlerName***.ExAlarmOutChannels[0] =2 ***handlerName***.ExAlarmOutChannels[1]=3 …
Comment	Parameters in URL: ***handlerName*** can be one of below four formats: ***Alarm***[ChannelNo].EventHandler ***MotionDetect*** [ChannelNo]. EventHandler ***BlindDetect*** [ChannelNo]. EventHandler ***LossDetect*** [ChannelNo]. EventHandler

- Set event handler config

Table 4-93

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Modify event handler settings.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Alarm[0].EventHandler.AlarmOutChannels[0]=1&Alarm[0].EventHandler.AlarmOutEnable=true
Success Return	OK
Comment	In below paramName, Meaning of ***handlerName*** is the same with GetEventHandler.

Appendix:

paramName	**paramValue type**	**Description**

paramName	paramValue type	Description
handlerName.EventHandler.AlarmOutChannels[***ch***]	integer	***ch*** is alarm out channel index which starts from 0. Range is {0, 1}, 0 — do not output alarm at alarm out channel *ch* 1 — output alarm at alarm out channel *ch*
handlerName.EventHandler.AlarmOutEnable	bool	Enable/Disable alarm out function.
handlerName.EventHandler.AlarmOutLatch	Integer	Range is [10—300]. Unit is seconds, indicates the time to output alarm after input alarm is cleared.
handlerName.EventHandler.BeepEnable	bool	Enable/Disable beep.
handlerName.EventHandler.Dejitter	integer	Range is [0—255]. Alarm signal dejitter seconds. Alarm signal change during this period is ignored.
handlerName.EventHandler.Delay	integer	Range is [0—300]. Delay seconds before setting take effect.
handlerName.EventHandler.LogEnable	bool	Enable/Disable log for alarm.
handlerName.EventHandler.MailEnable	bool	Enable/Disable mail send for alarm.
handlerName.EventHandler.PtzLink[*ch*][0]	string	Range is {None, Preset, Tour, Pattern} This is PTZ action linked with events. *ch* is PTZ channel index which starts from 0.
handlerName.EventHandler.PtzLink[*ch*][1]	integer	This is the parameter of PtzLink[*ch*][0], If PtzLink[*ch*]][0] is Preset: this is preset point. Tour: this is tour path number. Pattern: this is pattern number.
handlerName.EventHandler.PtzLinkEnable	Bool	Enable/Disable PTZ link.
handlerName.EventHandler.RecordChannels[*ch*]	Integer	Range is {0, 1} 0 – do not record on video channel *ch* 1 – record. on video channel *ch*

paramName	paramValue type	Description
handlerName.EventHandler.RecordEnable	bool	Enable/Disable record function.
handlerName.EventHandler.RecordLatch	integer	Range is [10—300]. Unit is seconds, indicates the time to record after input alarm is cleared.
handlerName.EventHandler.SnapshotChannels[*ch*]	integer	Range is {0, 1} 0 — do not snapshot on video channel *ch* 1 — snapshot on video channel *ch*
handlerName.EventHandler.SnapshotEnable	bool	Enable/Disable snapshot function.
handlerName.EventHandler.SnapshotPeriod	integer	Range is [0—255]. Frames between snapshots. 0 means continuously snapshot for every frame.
handlerName.EventHandler.SnapshotTimes	integer	Range is [0—65535] Snapshot times before stop, 0 means don't stop snapshot.
handlerName.EventHandler.TimeSection[*wd*][*ts*]	String	It's an effective time period for eventHanlder everyday. *wd* (week day) range is [0—6] (Sunday—Saturday) *ts* (time section) range is [0-23], it's index of time section table. Format: mask hh:mm:ss-hh:mm:ss Mask: {0,1}, hh: [0—24], mm: [00—59], ss: [00—59] Mask 0: this time section is not used. Mask 1: this time section is used. Example: TimeSection[1][0]=1 12:00:00-18:00:00 Means EventHandler is effective between 12:00:00 and 18:00:00 at Monday.

paramName	paramValue type	Description
handlerName.EventHandler.TipEnable	bool	Enable/Disable local message box tip.
handlerName.EventHandler.ExAlarmOutEnable	bool	Enable/Disable extend alarm out ability
handlerName.ExAlarmOutChannels[*ch*]	integer	extend alarm out channels

4.9.2 Alarm Event

- Get alarm config

Table 4-94

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=Alarm
Method	GET
Description	Get alarm config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Alarm
Success Return	table.Alarm[0].Enable=false table.Alarm[0].EventHandler….(output of EventHandler is described in GetEventHandler) table.Alarm[0].Name=Door1 table.Alarm[0].SensorType=NC table.Alarm[1]…. …
Comment	—

- Set alarm config

Table 4-95

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set alarm config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Alarm[0].Enable=true
Success Return	OK
Comment	In below ParamName, input is external alarm input channel which starts from 0. EventHandler defines parameter of relevant actions when alarm or event happens. It's also used in following sections about events.

Appendix:

ParamName	ParamValue type	Description
Alarm[***input***].Enable	bool	Enable/Disable alarm from a input channel
Alarm[***input***].EventHandler	—	Setting of EventHandler is described in **SetEventHandler.**
Alarm[***input***].Name	string	Name of alarm input channel.
Alarm[***input***].SensorType	string	Range is {NC, NO]. NC: normal close NO: normal open

4.9.3 Alarm Out

- Get alarm out config

Table 4-96

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=AlarmOut
Method	GET
Description	Get alarm out config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=AlarmOut
Success Return	table.AlarmOut[***alarmOutChannel***].Mode=0 table.AlarmOut[***alarmOutChannel***].Name=Beep
Comment	Parameters in Response: ***alarmOutChannel***: the alarm out channel index which starts from 0.

- Set alarm out config

Table 4-97

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set alarm out config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&AlarmOut[0].Mode=0&AlarmOut[0].Name=port1
Success Return	OK
Comment	***port*** in below ParamName is alarm out port index, start form 0.

Appendix:

ParamName	ParamValue type	Description
AlarmOut[***port***].Mode	integer	Range is {0, 1, 2} 0: automatically alarm 1: force alarm 2: close alarm
AlarmOut[***port***].Name	string	Alarm out port name.

4.9.4 Get Alarm Input Channels

Table 4-98

Syntax	http://<*server*>/cgi-bin/alarm.cgi?action=getInSlots
Method	GET
Description	Get alarm input channel number.
Example	http://192.168.1.108/cgi-bin/alarm.cgi?action=getInSlots
Success Return	result=2
Comment	—

4.9.5 Get Alarm Output Channels

Table 4-99

Syntax	http://<*server*>/cgi-bin/alarm.cgi?action=getOutSlots
Method	GET
Description	Get alarm output channel number.
Example	http://192.168.1.108/cgi-bin/alarm.cgi?action=getOutSlots
Success Return	result=1
Comment	—

4.9.6 Get States of Alarm Input Channels

Table 4-100

Syntax	http://<*server*>/cgi-bin/alarm.cgi?action=getInState
Method	GET
Description	Get alarm input state for all channels.
Example	http://192.168.1.108/cgi-bin/alarm.cgi?action=getInState
Success Return	result=3
Comment	A bit in the response result indicates a channel alarm states, result 3 means alarm channel 1 and channel 2 have alarm now.

4.9.7 Get States of Alarm Output Channels

Table 4-101

Syntax	http://<*server*>/cgi-bin/alarm.cgi?action=getOutState
Method	GET
Description	Get alarm output state for all channels.
Example	http://192.168.1.108/cgi-bin/alarm.cgi?action=getOutState
Success Return	result=0
Comment	A bit in the response result indicates a channel, result 1 means alarm is present.

4.9.8 Video Blind Event

- Get video blind detect config

Table 4-102

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=BlindDetect
Method	GET
Description	Get blind detect config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=BlindDetect
Success Return	***head***. Enable=false ***head***. EventHandler= (output of EventHandler is described in GetEventHandler) ***head***. Level=3

Comment	Parameters in Response: ***head***= table.BlindDetect[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

- Set video blind detect config

Table 4-103

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set blind detect config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&BlindDetect[0].Enable=true
Success Return	OK
Comment	Parameters in URL: In table below, ***head***= BlindDetect[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

Appendix:

ParamName	ParamValue type	Description
head. Enable	bool	Enable/Disable blind detect feature.
head. EventHandler	—	Setting of EventHandler is described in **SetEventHandler.**
head. Level	integer	Range is [1—6]. Sensitivity of blind detection. 1: lowest sensitivity. 6: highest sensitivity.

4.9.9 Video Loss Event

- Get video loss detect config

Table 4-104

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=LossDetect
Method	GET
Description	Get video loss detect config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=LossDetect
Success Return	***head***. Enable=false ***head***. EventHandler= (output of EventHandler is described in GetEventHandler)
Comment	Parameters in Response: ***head***=table.LossDetect [***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

- Set video loss detect config

Table 4-105

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set video loss detection config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&LossDetect[0].Enable=true
Success Return	OK
Comment	Parameters in URL: In table below, ***head***= LossDetect [***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel(equals to video channel index -1, and so 0 means channel 1).

Appendix:

ParamName	ParamValue type	Description
head. Enable	bool	Enable/Disable loss detect feature.
head. EventHandler	—	Setting of EventHandler is described in **SetEventHandler.**

4.9.10 Login Failure Event

- Get login failure event config

Table 4-106

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=LoginFailureAlarm
Method	GET
Description	Get login failure alarm config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=LoginFailureAlarm
Success Return	***head***. Enable=false ***head***. EventHandler= (output of EventHandler is described in GetEventHandler)
Comment	Parameters in Response: ***head***= table.LoginFailureAlarm

- Set login failure alarm config

Table 4-107

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set login failure alarm config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&LoginFailureAlarm.Enable=true
Success Return	OK
Comment	Parameters in URL:

	In table below, ***head***= LoginFailureAlarm

Appendix:

ParamName	ParamValue type	Description
head. Enable	bool	Enable/Disable to notify LoginFailure event. Now this event can be linked with send email and alarm out. The max try login times can be configured in chapter **SetGeneralConfig**.
head. EventHandler	—	Setting of EventHandler is described in **SetEventHandler**.

4.9.11 Storage Not Exist Event

- Get storage not exist event config

Table 4-108

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=StorageNotExist
Method	GET
Description	Get storage not exist event config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=StorageNotExist
Success Return	StorageNotExist.Enable=false StorageNotExist.EventHandler= (output of EventHandler is described in GetEventHandler)
Comment	—

- Set storage not exist event config

Table 4-109

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set storage not exist event config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&StorageNotExist.Enable=true
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
StorageNotExist.Enable	bool	Enable/Disable loss detect feature.
StorageNotExist.EventHandler	—	Setting of EventHandler is described in **SetEventHandler**.

4.9.12 Storage Access Failure Event

- Get storage access failure event config

Table 4-110

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=StorageFailure
Method	GET
Description	Get storage failure event config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=StorageFailure
Success Return	StorageFailure.Enable=false StorageFailure.EventHandler= (output of EventHandler is described in GetEventHandler)
Comment	—

- Set storage access failure event config

Table 4-111

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set storage failure event config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&StorageFailure.Enable=true
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
StorageFailure.Enable	bool	Enable/Disable loss detect feature.
StorageFailure.EventHandler	—	Setting of EventHandler is described in **SetEventHandler**.

4.9.13 Storage Low Space Event

- Get storage low space event config

Table 4-112

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=StorageLowSpace
Method	GET
Description	Get storage low space event config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=StorageLowSpace
Success Return	StorageLowSpace.Enable=false StorageLowSpace.EventHandler= (output of EventHandler is described in GetEventHandler)
Comment	—

- Set storage low space event config

Table 4-113

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set storage low space event config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&StorageLowSpace.Enable=true
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
StorageLowSpace.Enable	bool	Enable/Disable loss detect feature.
StorageLowSpace.EventHandler	—	Setting of EventHandler is described in **SetEventHandler**.

4.9.14 Net Abort Event

- Get net abort event config

Table 4-114

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=NetAbort
Method	GET
Description	Get net abort event config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=NetAbort
Success Return	NetAbort.Enable=false NetAbort.EventHandler= (output of EventHandler is described in GetEventHandler)
Comment	—

- Set net abort event config

Table 4-115

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set net abort event config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&NetAbort.Enable=true
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
NetAbort.Enable	bool	Enable/Disable loss detect feature.

NetAbort.EventHandler	—	Setting of EventHandler is described in **SetEventHandler**.

4.9.15 IP Conflict Event

- Get IP conflict event config

Table 4-116

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=IPConflict
Method	GET
Description	Get IP conflict event config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=IPConflict
Success Return	IPConflict.Enable=false IPConflict.EventHandler= (output of EventHandler is described in GetEventHandler)
Comment	—

- Set IP conflict event config

Table 4-117

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set IP Conflict event config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&IPConflict.Enable=true
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
IPConflict.Enable	bool	Enable/Disable loss detect feature.
IPConflict.EventHandler	—	Setting of EventHandler is described in **SetEventHandler**.

4.9.16 Get Channels Event Happened

Table 4-118

Syntax	http://<*server*>/cgi-bin/eventManager.cgi?action=getEventIndexes&code=<***eventCode***>
Method	GET
Description	Get channels indexes that event of code eventCode happens. Not all events support this command. Do not recommend to use it, use Attach command instead.
Example	http://192.168.1.108/cgi-bin/eventManager.cgi?action=getEventIndexes&code=AlarmLocal
Success Return	channels[0]=0 channels[1]=2

	channels[2]=3 … (This response means event happened on channel 0, channel 2 and channel 3 while video channel index starts from 0)
Comment	Parameters in URL: eventCode includes: VideoMotion: motion detection event VideoLoss: video loss detection event VideoBlind: video blind detection event. AlarmLocal: alarm detection event. StorageNotExist: storage not exist event. StorageFailure: storage failure event. StorageLowSpace: storage low space event. AlarmOutput: alarm output event.

4.9.17 Subscribe to Event Message

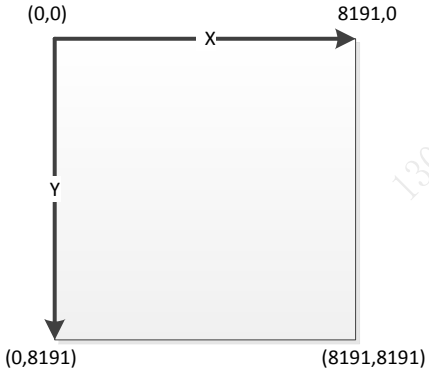
Table 4-119

Syntax	http://<*server*>/cgi-bin/eventManager.cgi?action=attach&codes=[<***eventCode***>,<***event Code***>,…][&*keepalive* = 20][&heartbeat=<***Heartbeat***>]
Method	GET
Description	Subscribe to messages that event of code eventCode happens.
Example	http://192.168.1.108/cgi-bin/eventManager.cgi?action=attach&codes=[AlarmLocal%2CVideoMotion%2CVideoLoss%2CVideoBlind] http://192.168.1.108/cgi-bin/eventManager.cgi?action=attach&codes=[All]&heartbeat=5
Success Return	HTTP Code: 200 OK\r\n Cache-Control: no-cache\r\n Pragma: no-cache\r\n Expires: Thu, 01 Dec 2099 16:00:00 GMT\r\n Connection: close\r\n Content-Type: multipart/x-mixed-replace; boundary=<*boundary*>\r\n Body: --<*boundary*>\r\n Content-Type: text/plain\r\n Content-Length: <*data length*>\r\n <*eventInfo*>\r\n\r\n --<*boundary*>\r\n Content-Type: text/plain\r\n Content-Length: <*data length*>\r\n <*eventInfo*>\r\n\r\n For example: HTTP Code: 200 OK\r\n Cache-Control: no-cache\r\n Pragma: no-cache\r\n Expires: Thu, 01 Dec 2099 16:00:00 GMT\r\n

	Connection: close\r\n Content-Type: multipart/x-mixed-replace; boundary=myboundary\r\n\r\n Body: --myboundary\r\n Content-Type: text/plain\r\n Content-Length: 39\r\n Code=VideoMotion;action=Start;index=0\r\n\r\n --myboundary\r\n Content-Type: text/plain\r\n Content-Length: 38\r\n Code=VideoBlind;action=Start;index=0\r\n\r\n --myboundary\r\n Content-Type: text/plain\r\n Content-Length: 9\r\n Heartbeat\r\n\r\n --myboundary\r\n …
Comment	***eventCode*** can be any one of the standard codes defined in DHIIF, or "**All**". **All** means all kinds of the eventcode. ***eventcode*** include: VideoMotion: motion detection event SmartMotionHuman: human smart motion detection SmartMotionVehicle：Vehicle smart motion detection VideoLoss: video loss detection event VideoBlind: video blind detection event. AlarmLocal: alarm detection event. CrossLineDetection: tripwire event CrossRegionDetection: intrusion event LeftDetection: abandoned object detection TakenAwayDetection: missing object detection VideoAbnormalDetection: scene change event FaceDetection: face detect event AudioMutation: intensity change AudioAnomaly: input abnormal VideoUnFocus: defocus detect event WanderDetection: loitering detection event RioterDetection: People Gathering event ParkingDetection: parking detection event MoveDetection: fast moving event StorageNotExist: storage not exist event. StorageFailure: storage failure event. StorageLowSpace: storage low space event. AlarmOutput: alarm output event. MDResult: motion detection data reporting event. The motion detect window contains 18 rows and 22 columns. The event info contains motion detect data with mask of every row. HeatImagingTemper: temperature alarm event

	CrowdDetection: crowd density overrun event FireWarning: fire warning event FireWarningInfo: fire warning specific data info In the example, you can see most event info is like "Code=***eventcode***; action=Start; index=0", but for some specific events, they will contain an another parameter named "data", the event info is like "Code=***eventcode***; action=Start; index=0; data=***datainfo***", the ***datainfo***'s fomat is JSON(JavaScript Object Notation). The detail information about the specific events and ***datainfo*** are listed in the appendix below this table. ***keepalive****:* If this param exist, the client must send any data to device by this connection in cycle. The keepalive is in range of [1,60] second. For example: The keeplive data can be the string "keep alive". Notice: the keepalive parameter is not recommended to use. It is recommended to use the Heartbeat parameter. ***Heartbeat****:* integer, range is [1,60],unit is second.If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client,the heartbeat message are "Heartbeat". **Note**: Heartbeat message must be sent before heartbeat timeout.

Appendix A: Coordinate system



4.9.18 Get Capability of Event Management

Table 4-120

Syntax	http://<*server*>/cgi-bin/eventManager.cgi?action=getCaps
Method	GET
Description	Get event manager capabilities.
Example	http://192.168.1.108/cgi-bin/eventManager.cgi?action=getCaps
Success Return	caps.AlarmOutEnable=true caps.BeepEnable=true caps.DejitterEnable=true caps.MMSEnable=true caps.MailEnable=true caps.MonitorTourEnable=true caps.PtzLinkEnable=true

	caps.RecordEnable=true caps.SnapshotEnable=true caps.TimeSectionEnable=true caps.TipEnable=true
Comment	—

4.9.19 NetAlarm

Table 4-121

Syntax	http://<*server*>/cgi-bin/netAlarm.cgi?action=setState&channel=<***ChannelNo***>&alarm=<***Alarm***>[&name=<***Name***>&trigger=<***Trigger***>&desc=<***Desc***>]
Method	GET
Description	Set the netalarm state.
Example	http://192.168.1.108/cgi-bin/netAlarm.cgi?action=setState&channel=1&alarm=true&name=somke&trigger=SmokingSensor&desc=Zone8
Success Return	OK
Comment	Parameters in URL: ***ChannelNo***: integer, alarm channel index which starts from 1 ***Alarm***: bool, the state of the alarm, the value can be true and false, true means the alarm is start. ***Name***: string, the name of the alarm, optional parameter, the length of this value should less than 32. ***Trigger***:string, the trigger of the alarm,for example, it can be Infrared, WaterSensor or Manual and so on. Optional parameter, the length of this value should less than 32. ***Desc***: string, the description of the alarm, optional parameter, the length of this value should less than 128.

4.9.20 GetSupportedEvents

Table 4-122

Syntax	http://<*server*>/cgi-bin/eventManager.cgi?action=getExposureEvents
Method	GET
Description	Get the event list which the device supports.
Example	http://192.168.1.108/cgi-bin/eventManager.cgi?action=getExposureEvents
Success Return	events[0]=VideoMotion events[1]=AlarmLocal events[2]=FaceDetection events[3]=VideoMotion …
Comment	—

4.10 PTZ

4.10.1 PTZ Config

- Get PTZ config

Table 4-123

Syntax	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=Ptz
Method	GET
Description	Get PTZ config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Ptz
Success Return	table.Ptz[***port***].Address=8 table.Ptz[***port***].Attribute[0]=115200 table.Ptz[***port***].Attribute[1]=8 table.Ptz[***port***].Attribute[2]=Even table.Ptz[***port***].Attribute[3]=1 table.Ptz[***port***].Homing[0]=0 table.Ptz[***port***].Homing[1]=30 table.Ptz[***port***].NumberInMatrixs=0 table.Ptz[port].ProtocolName=NONE
Comment	Parameters in Response: ***port*** is PTZ port index, start form 0.

- Set PTZ config

Table 4-124

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set PTZ config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Ptz[0].Address=192.168.0.1&Ptz[0].Attribute[0]=9600
Success Return	OK
Comment	***port*** in below ParamName is PTZ port index, start form 0.

Appendix:

ParamName	ParamValue type	Description
Ptz[***port***].Address	integer	Range is [0—255]. Device address, if there are more than one device connected to this port, distinguish them by this address.
Ptz[***port***].Attribute[0]	integer	The baud rate. Range is {1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200}.
Ptz[***port***].Attribute[1]	integer	Range is {4, 5, 6, 7, 8}. Data bit.
Ptz[***port***].Attribute[2]	string	Range is {Even, Mark, None, Odd, Space}.

ParamName	ParamValue type	Description
		Parity verification mode.
Ptz[***port***].Attribute[3]	float	Range is {1, 1.5, 2}. Stop bit.
Ptz[***port***].Homing[0]	integer	Range is {-1,0—255} -1: homing is disabled. [0—255]: preset point number
Ptz[***port***].Homing[1]	integer	Range is [0—65535]. No operation timeout, unit is seconds. After no operation timeout, PTZ go to preset point set in Ptz[port].Homing[0].
Ptz[***port***].ProtocolName	string	PTZ protocol name depends on PTZ capability. Refer to **GetProtocolList** to get the protocol list.

4.10.2 Get PTZ Protocol List

Table 4-125

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=getProtocolList[&channel=<***ChannelNo***>]
Method	GET
Description	Get the protocol list that PTZ can support. Unsupported now.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=getProtocolList&channel=1
Success Return	info.RS[0]=Pelco info.RS[1]=DH-SD1 info.Coaxial[0]=HD-CVI info.Coaxial[1]=HD-CVI2.0
Comment	Response contains all support PTZ protocols of the server. ***ChannelNo***: integer, video channel index which starts from 1.

4.10.3 Get PTZ Capability of Current Protocol

Table 4-126

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=getCurrentProtocolCaps[&channel=<***ChannelNo***>]
Method	GET
Description	Get PTZ channel protocol capabilities.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=getCurrentProtocolCaps&channel=1
Success Return	caps.**AlarmLen**=0 caps.**AuxMax**=8 caps.**AuxMin**=1 caps.**CamAddrMax**=255 caps.**CamAddrMin**=1 caps.**Flip**=false caps.**Focus**=false caps.**Interval**=200 caps.**Iris**=false caps.**Menu**=false caps.**MonAddrMax**=255

	caps.**MonAddrMin**=0 caps.**Name**=DH-SD1 caps.**Pan**=false caps.**PanSpeedMax**=255 caps.**PanSpeedMin**=1 caps.**PatternMax**=5 caps.**PatternMin**=1 caps.**PresetMax**=80 caps.**PresetMin**=1 caps.**Tile**=false caps.**TileSpeedMax**=255 caps.**TileSpeedMin**=1 caps.**TourMax**=7 caps.**TourMin**=0 caps.**Type**=1 caps.**Zoom**=false
Comment	Parameters in URL: ***ChannelNo***: PTZ channel index which starts from 1.

Appendix:

Field in response	Description
AlarmLen	Alarm length in protocol.
AuxMax	Maximum/Minimum number for auxiliary functions.
AuxMin	
CamAddrMax	Maximum/Minimum channel address.
CamAddrMin	
Flip	True or false, support picture flip or not.
Focus	True or false, support focus or not.
Iris	True or false, support Iris adjusts or not.
Menu	True or false, support internal menu of the PTZ or not.
MonAddrMax	Maximum/Minimum monitor address.
MonAddrMin	
Name	Name of the operation protocol.
Pan	True or false, support pan or not.
PanSpeedMax	Maximum/Minimum pan speed.
PanSpeedMin	
PatternMax	Maximum/Minimum pattern path number.
PatternMin	
PresetMax	Maximum/Minimum preset point number.
PresetMin	
Tile	True or false, support tilt or not.
Zoom	True or false, support zoom or not.
TileSpeedMax	Maximum/Minimum tile speed.
TileSpeedMin	
TourMax	Maximum/Minimum tour path number.
TourMin	

Field in response	Description
Type	Type of PTZ protocol.

4.10.4 Get PTZ Status

Table 4-127

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=getStatus[&channel=<***ChannelNo***>]
Method	GET
Description	Get PTZ status.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=getStatus&channel=1
Success Return	status.UTC=6538920 status.MoveStatus=Idle status.ZoomStatus=Idle status.PresetID=10 status.Position=120,12,2
Comment	***ChannelNo***: integer, video channel index which starts from 1.

4.10.5 PTZ Control

- Basic movement

Table 4-128

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=<***action***>&channel=<***ch***>&code=<***code***>&arg1=<***arg1***>&arg2=<***arg2***>&arg3=<***arg3***>
Method	GET
Description	PTZ basic movement.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Up&arg1=0&arg2=1&arg3=0 http://192.168.1.108/cgi-bin/ptz.cgi?action=stop&channel=1&code=Up&arg1=0&arg2=1&arg3=0
Success Return	OK
Comment	Parameters in URL: ***action*** is PTZ control command, it can be start or stop. ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***code*** is PTZ operation, and ***arg1***, ***arg2***, ***arg3*** are arguments of operation. ***code*** and ***argN*** values are listed in table below.
Comment	The default exposure mode of PTZ is auto-exposure, aperture adjustment (Aperture larger 、Aperture smaller) would take effects temporarily, PTZ will auto—adjust the aperture after the auto—exposure recovery time (15 mins) The default focus mode of PTZ is semi—auto, PTZ control or other operations will trigger the auto focus There is no zoom times parameter for ZoomWide and ZoomTele command, if you need to change zoom times, see "**错误!未找到引用源。 错误!未找到引用源。**".

Appendix:

Code	Code description	arg1	arg2	arg3

Code	Code description	arg1	arg2	arg3
Up	Tile up	0	Vertical speed, range is [1—8]	0
Down	Tile down	0	Vertical speed, range is [1—8]	0
Left	Pan left	0	Horizontal speed, range is [1—8]	0
Right	Pan right	0	Horizontal speed, range is [1—8]	0
LeftUp	Pan left and tile up	Vertical speed, range is [1—8]	Horizontal speed, range is [1—8]	0
RightUp	Pan right and tile up	Vertical speed, range is [1—8]	Horizontal speed, range is [1—8]	0
LeftDown	Pan left and tile down	Vertical speed, range is [1—8]	Horizontal speed, range is [1—8]	0
RightDown	Pan right and tile down	Vertical speed, range is [1—8]	Horizontal speed, range is [1—8]	0
ZoomWide	Zoom out	0	0	0
ZoomTele	Zoom in	0	0	0
FocusNear	Focus near	0	0	0
FocusFar	Focus far	0	0	0
IrisLarge	Aperture larger	0	0	0
IrisSmall	Aperture smaller	0	0	0

- Start the continuous movement

Table 4-129

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=Continuously&arg1=<***arg1***>&arg2=<***arg2***>&arg3=<***arg3***>&arg4=<***arg4***>
Method	GET
Description	Start the continuous movement.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Continuously&arg1=5&arg2=5&arg3=5&arg4=60
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is horizontal speed, range is [-8—8]； ***arg2*** is vertical speed, range is [-8—8]； ***arg3*** is zoom speed, range is [-100—100]； ***arg4*** is timeout value，the measurement unit is second, it's an integer greater than 0，the PTZ will stop moving automatically if timeout value reached and no"stop"command received. The maximum timeout value is 3600 seconds.

- Stop the continuous movement

Table 4-130

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=stop&channel=<ch>&code=Continuously&arg1=0&arg2=0&arg3=0&arg4=0

Method	GET
Description	Immediately stop the continuous movement.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=stop&channel=1&code=Continuously&arg1=0&arg2=0&arg3=0&arg4=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n].

- 3D positioning

Table 4-131

Syntax	http://<*server*>/cgi-bin/ptzBase.cgi?action=moveDirectly&channel=<***ChannelNo***>&startPoint[0]=<***startX***>&startPoint[1]=<***startY***>&endPoint[0]=<***endX***>&endPoint[1]=<***endY***>
Method	GET
Description	Three-dimensional orientation. Move to the rectangle with screen coordinate [startX, startY], [endX, endY] .
Example	http://192.168.1.108/cgi-bin/ptzBase.cgi?action=moveDirectly&channel=1&startPoint[0]=7253&startPoint[1]=2275&endPoint[0]=7893&endPoint[1]=3034
Success Return	OK
Comment	***ChannelNo***: integer, video channel index which starts from 1. ***startX***, ***startY***, ***endX***, ***endY***: relative coordinates, range is 0-8192. The two points [startX, startY] and [endX, endY] makes the destination rectangle.

- PTZ relative movement

Table 4-132

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=moveRelatively&channel=<***ch***>&arg1=<***arg1***>&arg2=<***arg2***>&arg3=<***arg3***>
Method	GET
Description	PTZ relative movement.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=moveRelatively&channel=1&arg1=1&arg2=1&arg3=1
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is relative horizontal coordinate ***arg2*** is relative vertical coordinate ***arg3*** is relative zoom coordinate The normalization of above three values is[-1,1]

- PTZ accurate positioning

Table 4-133

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=moveAbsolutely&channel=<***ch***>&arg1=<***arg1***>&arg2=<***arg2***>&arg3=<***arg3***>
Method	GET
Description	Accurate positioning.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=moveAbsolutely&channel=1&arg1=-0.8&arg2=

	0.3&arg3=0.5
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is absolute horizontal coordinate, the normalization value is [-1, 1]. ***arg2*** is absolute vertical coordinate, the normalization value is [-1, 1]. ***arg3*** is absolute zoom coordinate, the normalization value is [0, 1].

4.10.6 Preset

- Get preset information

Table 4-134

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=getPresets[&channel=<***ChannelNo***>]
Method	GET
Description	Get preset of PTZ control.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=getPresets&channel=1
Success Return	presets[0].Index=1 presets[0].Name=Preset 1 presets[0].Type=0 …
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. The size of presets-array is the number of presets.

- Go to preset

Table 4-135

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=GotoPreset&arg1=0&arg2=<***index***>&arg3=0
Method	GET
Description	Go to preset.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=GotoPreset&arg1=0&arg2=1&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg2*** is number of preset

- Set preset

Table 4-136

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=SetPreset&arg1=0&arg2=<***index***>&arg3=0
Method	GET
Description	Set preset.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=SetPreset&arg1=0&arg2=2&arg3=0

Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg2*** is preset number

- Set the name for specified preset

Table 4-137

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=setPreset&channel=<***ch***>&arg1=<***index***>&arg2=<***name***>
Method	GET
Description	Set the name for specified preset.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=setPreset&channel=1&arg1=2&arg2=2
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is preset number ***arg2*** is preset name, could be letter,number or Chinese

- Delete specified preset

Table 4-138

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=ClearPreset&arg1=0&arg2=<***index***>&arg3=0
Method	GET
Description	Delete the specified preset.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=ClearPreset&arg1=0&arg2=1&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg2*** is preset number

4.10.7 Tour

- Start the tour

Table 4-139

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=StartTour&arg1=<***index***>&arg2=0&arg3=0
Method	GET
Description	Start the tour.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=StartTour&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL:

	ch is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is the number of tour route

- Stop the tour

Table 4-140

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=StopTour&arg1=<***index***>&arg2=0&arg3=0
Method	GET
Description	Stop the tour.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=StopTour&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is the number of tour route

- Add tour group

Table 4-141

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=setTour&channel=<***ch***>&arg1=<***index***>&arg2=<***name***>
Method	GET
Description	Add tour group, including set the tour route and name. Frequently used with add tour preset.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=setTour&channel=1&arg1=1&arg2=1
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is the number of tour route ***arg2*** is tour name

- Delete tour group

Table 4-142

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=ClearTour&arg1=<***index***>&arg2=0&arg3=0
Method	GET
Description	Delete the tour group with specified number.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=ClearTour&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is the number of tour group

- Add tour preset

Table 4-143

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=AddTour&arg1=<***index 1***>&arg2=<***index2***>&arg3=0
Method	GET
Description	Add preset to tour group.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=AddTour&arg1=1&arg2=2&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is the number of tour route ***arg2*** is the number of preset

- Delete tour preset

Table 4-144

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=DelTour&arg1=<***index1***>&arg2=<***index2***>&arg3=0
Method	GET
Description	Delete the specified preset from a particular tour group.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=DelTour&arg1=1&arg2=2&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is the number of tour route ***arg2*** is the number of preset

4.10.8 Scan

- Set left boundary

Table 4-145

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=SetLeftLimit&arg1=<***index***>&arg2=0&arg3=0
Method	GET
Description	Set left boundary.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=SetLeftLimit&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is the scan number

- Set right boundary

Table 4-146

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=SetRightLimit&arg1=<***in***

	dex>&arg2=0&arg3=0
Method	GET
Description	Set right boundary.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=SetRightLimit&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is the scan number

- Start scan

Table 4-147

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=AutoScanOn&arg1=<***index***>&arg2=0&arg3=0
Method	GET
Description	Start scanning.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=AutoScanOn&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1—n]. ***arg1*** is scan number

- Stop scan

Table 4-148

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=AutoScanOff&arg1=<***index***>&arg2=0&arg3=0
Method	GET
Description	Stop scanning.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=AutoScanOff&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is scan number

4.10.9 Pattern

- Start pattern record

Table 4-149

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=SetPatternBegin&arg1=<***index***>&arg2=0&arg3=0
Method	GET
Description	Start pattern record.

Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=SetPatternBegin&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is pattern number

- Stop pattern record

Table 4-150

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=SetPatternEnd&arg1=<***index***>&arg2=0&arg3=0
Method	GET
Description	Stop pattern record.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=SetPatternEnd&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is pattern number

- Start pattern

Table 4-151

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=StartPattern&arg1=<***index***>&arg2=0&arg3=0
Method	GET
Description	Start pattern.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=StartPattern&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***arg1*** is pattern number

- Stop pattern

Table 4-152

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=StopPattern&arg1=<***index***>&arg2=0&arg3=0
Method	GET
Description	Stop pattern.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=StopPattern&arg1=1&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n].

	arg1 is pattern number

4.10.10 Pan

- Start pan

Table 4-153

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=AutoPanOn&arg1=0&arg2=0&arg3=0
Method	GET
Description	Start pan.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=AutoPanOn&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n].

- Stop pan

Table 4-154

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=AutoPanOff&arg1=0&arg2=0&arg3=0
Method	GET
Description	Stop pan.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=AutoPanOff&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n].

4.10.11 PTZ Auto Movement

- Get PTZ auto movement configuration

Table 4-155

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=PtzAutoMovement
Method	GET
Description	Get PTZ auto movement configuration.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=PtzAutoMovement
Success Return	table.PtzAutoMovement[***port***][***Task***].Enable = true table.PtzAutoMovement[***port***][***Task***].TimeSection[week][section] = 0 00:00:00-23:59:59 … table.PtzAutoMovement[***port***][***Task***].Fuction = None table.PtzAutoMovement[***port***][***Task***].ScanId = 0 table.PtzAutoMovement[***port***][***Task***].PresetId = 0

	table.PtzAutoMovement[***port***][***Task***].PatternId = 0 table.PtzAutoMovement[***port***][***Task***].TourId = 0 table.PtzAutoMovement[***port***][***Task***].AutoHoming.Time = 30 table.PtzAutoMovement[***port***][***Task***].SnapshotEnable = false table.PtzAutoMovement[***port***][***Task***].SnapshotDelayTime = 30
Comment	Parameters in Response: ***port*** is PTZ port index, start from 0. ***Task*** is the number of task, start from 0. ***week***: from 1 to 7. ***section***: time section, from 0 to 5.

- Set PTZ auto movement

Table 4-156

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set PTZ auto movement.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&PtzAutoMovement[0][0].Function=Preset&PtzAutoMovement[0][0].PresetId=1
Success Return	OK
Comment	Parameters in URL: In table below, ***head***=PtzAutoMovement[port][task] ***port*** is PTZ port index, start from 0. ***task*** is the number of task, start from 0. ***week***: from 1 to 7. ***section***: time section, from 0 to 5.

Appendix:

ParamName	ParamValue type	Description
head. Enable	bool	Enable/Disable PtzAutoMovement
head. TimeSection	timeSchedule	timeSchedule[***week***][***section***]=1 10:00:00-11:00:00 …
head. Fuction	string	Range is {Scan, Preset, Pattern, Tour}.
head. ScanId	integer	Scan Id, start from 1
head. PresetId	integer	Preset Id, start from 1
head. PatternId	integer	Pattern Id, start from 1
head. TourId	integer	Tour Id, start from 1
head. AutoHoming.Time	integer	Recover time, unit is second.
head. SnapshotEnable	bool	Enable/Disable Snap, when "Fuction" is "Preset".
head. SnapshotDelayTime	integer	Delay time of snap, when "Fuction" is "Preset".

4.10.12 PTZ Restart

Table 4-157

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=Restart&arg1=0&arg2=0&arg3=0
Method	GET
Description	PTZ restart.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Restart&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n].

4.10.13 PTZ Reset

Table 4-158

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=Reset&arg1=0&arg2=0&arg3=0
Method	GET
Description	PTZ reset.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Reset&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n].

4.10.14 OSD Menu

- Enter the menu

Table 4-159

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=Menu&arg1=0&arg2=0&arg3=0
Method	GET
Description	Enter the menu.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Menu&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n].

- Exit the menu

Table 4-160

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=Exit&arg1=0&arg2=0&arg3=0
Method	GET
Description	Exit the menu.

Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Exit&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n].

- Confirm

Table 4-161

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=Enter&arg1=0&arg2=0&arg3=0
Method	GET
Description	Confirm the menu.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=Enter&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n].

- Start the basic operation of menu

Table 4-162

Syntax	http://<*server*>/cgi-bin/ptz.cgi?action=start&channel=<***ch***>&code=<***code***>&arg1=<***arg1***>&arg2=<***arg2***>&arg3=<***arg3***>
Method	GET
Description	Start the basic operation of menu.
Example	http://192.168.1.108/cgi-bin/ptz.cgi?action=start&channel=1&code=MenuUp&arg1=0&arg2=0&arg3=0
Success Return	OK
Comment	Parameters in URL: ***ch*** is PTZ channel index which starts from 1. Range is [1 — n]. ***code*** is PTZ operation, and arg1, arg2, arg3 are arguments of operation. ***code*** and argN values are listed in table below.

Appendix:

Code	**Code description**	**arg1**	**arg2**	**arg3**
MenuUp	MenuUp	0	0	0
MenuDown	MenuDown	0	0	0
MenuLeft	MenuLeft	0	0	0
MenuRight	MenuRight	0	0	0

4.11 Record

4.11.1 Get Capability of Recording

Table 4-163

Syntax	http://<*server*>/cgi-bin/recordManager.cgi?action=getCaps
Method	GET
Description	Get record manager capabilities.
Example	http://192.168.1.108/cgi-bin/recordManager.cgi?action=getCaps
Success Return	caps.MaxPreRecordTime=30 caps.PacketLengthRange[0]=1 caps.PacketLengthRange[1]=60 caps.PacketSizeRange[0]=131072 caps.PacketSizeRange[1]=2097152 caps.SupportExtraRecordMode=true caps.SupportHoliday=true caps.SupportPacketType[0]=Time caps.SupportPacketType[1]=Size caps.SupportResumeTransmit=false
Comment	—

4.11.2 Record Config

- Get record config

Table 4-164

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=Record
Method	GET
Description	Get record config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Record
Success Return	table.Record[***channel***].PreRecord=6 table.Record[***channel***].TimeSection[***weekday***][0]=1 00:00:00-24:00:00 table.Record[***channel***].TimeSection[***weekday***][1]=0 02:00:00-24:00:00 table.Record[***channel***].TimeSection[***weekday***][2]=0 03:00:00-24:00:00 table.Record[***channel***].TimeSection[***weekday***][3]=0 04:00:00-24:00:00 table.Record[***channel***].TimeSection[***weekday***][4]=0 05:00:00-24:00:00 table.Record[***channel***].TimeSection[***weekday***][5]=0 06:00:00-24:00:00
Comment	Parameters in Response: ***channel***: integer, array index starts from 0, which means video channel(equals to video channel index -1, and so 0 means channel 1). ***weekday***: range is [0-6] (Sunday - Saturday). Record config contains pre record time and record time sections of every day.

- Set record config

Table 4-165

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]

Method	GET
Description	Set record config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Record[0].TimeSection[0][0]=6 00:00:00-23:59:59 Set record time to every Sunday all day. Record type is motion detection and alarm. In this example, "6 00:00:00-23:59:59" means motion detection and alarm record all day (6 = 4 & 2, alarm is 4, motion detection is 2.).
Success Return	OK
Comment	Parameters in URL: In table below, ***ch***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***wd***: week day index ***ts***: time section index

Appendix:

ParamName	ParamValue type	Description
Record[***ch***].PreRecord	integer	Range is [0—300]. Prerecord seconds, 0 means no prerecord. ch (Channel number) starts form 0
Record[***ch***].TimeSection[***wd***][***ts***]	string	*wd* (week day) range is [0—6] (Sunday - Saturday) *ts* (time section) range is [0—23], time section table index. Format: mask hh:mm:ss-hh:mm:ss Mask: [0—65535], hh: [0—24], mm: [0—59], ss: [0—59] Mask indicates record type by bits: Bit0: regular record Bit1: motion detection record Bit2: alarm record Bit3: card record

4.11.3 Record Mode

- Get record mode config

Table 4-166

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=RecordMode
Method	GET
Description	Get record mode config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=RecordMode
Success Return	table.RecordMode[***Channel***].Mode=0
Comment	Parameters in Response:

	Channel: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

- Set record mode config

Table 4-167

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>
Method	GET
Description	Set record mode config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&RecordMode[0].Mode=0
Success Return	OK
Comment	Parameters in URL: In table below, ***Channel***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

Appendix:

ParamName	ParamValue type	Description
RecordMode[***Channel***].Mode	integer	Range is {0, 1, 2 }. 0: automatically record 1: manually record 2: stop record.

4.11.4 Media Global

- Get media global config

Table 4-168

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=MediaGlobal
Method	GET
Description	Get media global config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=MediaGlobal
Success Return	table.MediaGlobal.SnapFormatAs=MainFormat
Comment	—

- Set media global config

Table 4-169

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>
Method	GET
Description	Set media global config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&MediaGlobal.SnapFormatAs=MainFormat
Success Return	OK

Comment	—

Appendix:

ParamName	ParamValue type	Description
MediaGlobal.SnapFormatAs	string	The range is {"MainFormat", "ExtraFormat"}.

4.11.5 Find Media Files

Step 1 Create a media files finder.

Table 4-170

Syntax	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=factory.create
Method	GET
Description	Create a media file finder.
Example	http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=factory.create
Success Return	result=08137
Comment	The ***result*** is the finder's ***objectId***, needed by the following API.

Step 2 Start to find media files satisfied the conditions with the finder.

Table 4-171

Syntax	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findFile&object=<***objectId***>&condition.Channel=<***ChannelNo***>&condition.StartTime=<***start***>&condition.EndTime=<***end***>[&condition.Dirs[0]=<***dir***>&condition.Types[0]=<***type***>&condition.Flags[0]=<***flags***>&condition.Events[0]=<***event***>&condition.VideoStream=<***stream***>]
Method	GET
Description	Check if there are files that satisfy all the conditions.
Example	Find a file in channel 1, in directory "/mnt/dvr/sda0",event type is "AlarmLocal" or "VideoMotion", file type is "dav", and time between 2014-1-1 12:00:00 and 2015-1-10 12:00:00 , URL is: http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.Dirs[0] =/mnt/dvr/sda0&condition.Types[0]=dav&condition.Events[0]=AlarmLocal&condition.Events[1]=VideoMotion&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.VideoStream=Main
Success Return	OK

Comment	Start to find a file with the above condition. If files exist, return OK, else return Error. Parameters in URL: ***objecteId***: The object Id is the finder created before. You must create a finder before finding files. ***ChannelNo***: integer, video channel index which starts from 1. ***start / end***: the start/end time when recording. ***dir*** : in which directories you want to find the file. It is an array. The index starts from 0. The range of dir is {"/mnt/dvr/sda0", "/mnt/dvr/sda1"}. This condition can be omitted. If omitted, find files in all the directories. ***type*** : which types of the file you want to find. It is an array. The index starts from 0. The range of type is {"dav", "jpg", "mp4"}. If omitted, find files with all the types. ***flags*** : which flags of the file you want to find. It is an array. The index starts from 0. The range of flag is {"Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout"}. If omitted, find files with all the flags. ***event*** : by which event the record file is triggered. It is an array. The index starts from 0. The range of event is {"AlarmLocal", "VideoMotion", "VideoLoss", "VideoBlind", "Traffic*"}. This condition can be omitted. If omitted, find files of all the events. ***stream*** : which video stream type you want to find. The range of stream is {"Main", "Extra1", "Extra2", "Extra3"}. If omitted, find files with all the stream types.

Step 3 Get the media file information found by the finder.

Table 4-172

Syntax	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=<***objectId***>&count=<***fileCount***>
Method	GET
Description	Find the next files no more than ***fileCount***.
Example	http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100
Success Return	***found***=1 items[0].***Channel***=1 items[0].***StartTime***=2011-1-1 12:00:00 items[0].***EndTime***=2011-1-1 13:00:00 items[0].***Type***=dav items[0].***Events[0]***=AlarmLocal items[0].***VideoStream***=Main items[0].***FilePath***=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].***Length***=790 items[0].***Duration***=3600
Comment	The maximum value of ***fileCount*** is 100.

Appendix:

Field in Response	**Description**
found	Count of found file, found is 0 if no file is found.
Channel	Internal video channel index starts from 0, equals to API findFile input condition.Channel -1.
StartTime	Start time.

Field in Response	Description
EndTime	End time.
Type	File type.
Events	Event type.
VideoStream	Video stream type.
FilePath	File path.
CutLength	File length that cut between StartTime and EndTime.
Length	File length of the whole file.
Duration	Duration time.

Step 4 Close the finder.

Table 4-173

Syntax	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=close&object=<objectId>
Method	GET
Description	Stop finding.
Example	http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=close&object=08137
Success Return	OK
Comment	—

Step 5 Destroy the finder.

Table 4-174

Syntax	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=destroy&object=<objectId>
Method	GET
Description	Destroy the media file finder.
Example	http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=destroy&object=08137
Success Return	OK
Comment	—

4.11.6 Find media files with FaceDetection info

Step 1 Create a media files finder.
This API is the same as the API in "4.11.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and FaceDetection condition with the finder.

Table 4-175

URL	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Description	Start to find media files satisfied the common conditions and FaceDetection conditions with the finder.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
condition	object	R	The search conditions

+Channel	int	R	The video channel to search, video channel index start from 1, use -1 to search all video channel.
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Types	array<string>	O	Search file types, should be "jpg"
+Flags	array<string>	O	Search flags, can be : "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout" should include “Event”
+Events	array<string>	R	Search event list, must be one string : "**FaceDetection**"
+Dirs	array<string>	O	Search directory list, if omit, search all. Each dir path max string length is 259.
+DB	object	R	DB filter object
++FaceDetectionRecordFilter	object	R	DB filter for FaceDetection
+++ImageType	string	O	Picture type, if omit, means search all type. "GlobalSence" : means big picture of the full sence, "Small" : means small picture of the people face.
+++Sex	string	O	Sex, it can be "Man", "Woman", if omit, search all
+++Age	array<int>	O	Age range, ex：[25, 40]
+++Glasses	int	O	Glasses Status, 0: all, 1: not wear, 2: wear
+++Mask	int	O	Mask Status, 0: all, 1: not wearing mask, 2: wearing mask
+++Beard	int	O	Beard Status, 0: all, 1: no beard, 2: has beard
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.Events[0]=FaceDetection&condition.DB.FaceDetectionRecordFilter.ImageType=GlobalSence&condition.DB.FaceDetectionRecordFilter.Sex=Man&condition.DB.FaceDetectionRecordFilter.Age[0]=25&condition.DB.FaceDetectionRecordFilter.Age[1]=40&condition.DB.FaceDetectionRecordFilter.Glasses=1		
Response	OK		

Step 3 Get the media file information found by the finder

Table 4-176

URL	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findNextFile		
Method	GET		
Description	Get the media file information found by the finder.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
count	int	R	Record num to get, should less than 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
found	int	R	Record num that found.

items	array<object>	R	Result record items.
+Channel	int	R	The video channel index start from 0,
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Type	string	R	File type, should be: "jpg"
+Events	array<string>	R	Events with this record
+FilePath	string	R	File path, string max length is 259
+CutLength	int	R	File length that cut between start time and end time
+Length	int	O	File length of the whole file
+SummaryNew	array<object>	R	DB record object
++Key	string	R	DB record name, should be "**FaceDetectionRecord**"
++Value	object	R	DB record value
+++ImageType	string	O	Picture type. It can be: "GlobalSence" , "Small"
+++TimeStamp	object	O	Picture timestamp
++++UTC	int	O	UTC seconds
++++UTCMS	int	O	UTC miliseconds
+++Sex	string	O	Sex, it can be "Man", "Woman", "Unknown"
+++Age	int	O	Age
+++Glasses	int	O	Glasses Status, 0: unknown, 1: not wear, 2: wear
+++Mask	int	O	Mask Status, 0: unknown, 1: not wearing mask, 2: wearing mask
+++Beard	int	O	Beard Status, 0: unknown, 1: no beard, 2: has beard

[Example]

Request	GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100
Response	found=100 items[0].Channel=1 items[0].StartTime=2011-1-1 12:00:00 items[0].EndTime=2011-1-1 13:00:00 items[0].Type=jpg items[0].Events[0]=FaceDetection items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].CutLength=79000 items[0].SummaryNew[0].Key=FaceDetectionRecord items[0].SummaryNew[0].Value.ImageType=GlobalSence items[0].SummaryNew[0].Value.TimeStamp.UTC=134652732 items[0].SummaryNew[0].Value.TimeStamp.UTCMS=134 items[0].SummaryNew[0].Value.Sex=Man items[0].SummaryNew[0].Value.Age=30 items[0].SummaryNew[0].Value.Glasses=1 items[0].SummaryNew[0].Value.Mask=2 items[0].SummaryNew[0].Value.Beard=1 …

Step 4 Close the finder.

This API is the same as the API in "4.11.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.11.5 Find Media Files".

4.11.7 Find media files with FaceRecognition info

Step 1 Create a media files finder.

This API is the same as the API in "4.11.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and FaceRecognition conditions with the finder.

Table 4-177

URL	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Description	Start to find media files satisfied the common conditions and FaceRecognition conditions with the finder.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
condition	object	R	The search conditions
+Channel	int	R	The video channel to search, video channel index start from 1, use -1 to search all video channel.
+StartTime	string	O	Start time to search, ex: "2010-05-25 12:05:00" Ignored, use StartTime in DB param
+EndTime	string	O	End time to search, ex: "2010-06-25 12:05:00" Ignored, use EndTime in DB param
+Types	array<string>	O	Search file types, should be "jpg"
+Flags	array<string>	O	Search flags. It can be: "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout" should include “Event”.
+Events	array<string>	R	Search event list, must be one string: "FaceRecognition"
+Dirs	array<string>	O	Search directory list, if omit, search all. Each dir path max string length is 259.
+DB	object	R	DB filter object
++FaceRecognitionRecordFilter	object	R	DB filter for FaceRecognition
+++MachineAddress	string	O	Machine address
+++StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+++EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+++Person	object	O	Search condition of history person and the similar face group person candidates
++++Name	string	O	Person's name, max string length is 15
++++Sex	string	O	Sex, can be "Male", "Female", if omit, search all
++++Birthday	string	O	Birthday, max string length is 11, ex: 1990-5-1
++++Country	string	O	Country, ISO 3166, string length should be 2
++++Province	string	O	Province, max string length is 63

++++City	string	O	City, max string length is 63
++++Certificate Type	string	O	Certificate Type. It can be: "IC", "Passport", "Military", "Unknown"
++++ID	string	O	Person ID of CertificateType, max string length is 31
++++GroupID	string	O	The identity of the Face Group that this Person in. max string length is 63
++++Age	array<int>	O	Age range, ex：[25, 40]
++++Glasses	int	O	Glasses Status, 0: all, 1: not wear, 2: wear
++++Mask	int	O	Mask Status, 0: all, 1: not wearing mask, 2: wearing mask
++++Beard	int	O	Beard Status, 0: all, 1: no beard, 2: has beard
+++GroupID	array<string>	O	GroupID list
+++SimilaryRange	array<int>	O	Similary Range, ex: [40, 100]
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.Events[0]=FaceRecognition&condition.DB.FaceRecognitionRecordFilter.RegType=RecSuccess&condition.DB.FaceRecognitionRecordFilter.StartTime=2014-1-1%2012:00:00&condition.DB.FaceRecognitionRecordFilter.EndTime=2015-1-10%2012:00:00&condition.DB.FaceRecognitionRecordFilter.Person.Sex=Male&condition.DB.FaceRecognitionRecordFilter.Person.Country=CN&condition.DB.FaceRecognitionRecordFilter.Person.Age[0]=25&condition.DB.FaceRecognitionRecordFilter.Person.Age[1]=40&condition.DB.FaceRecognitionRecordFilter.Person.Glasses=1&condition.DB.FaceRecognitionRecordFilter.GroupID[0]=10001&condition.DB.FaceRecognitionRecordFilter.GroupID[1]=10003&condition.DB.FaceRecognitionRecordFilter.GroupID[2]=10005&condition.DB.FaceRecognitionRecordFilter.SimilaryRange[0]=40&condition.DB.FaceRecognitionRecordFilter.SimilaryRange[1]=100		
Response	OK		

Step 3 Get the media file information found by the finder

Table 4-178

URL	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findNextFile		
Method	GET		
Description	Get the media file information found by the finder		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
count	int	R	Record num to get, should less than 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
found	int	R	Record num that found.
items	array<object>	R	Result record items.
+Channel	int	R	The video channel index start from 0,
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"

+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Type	string	R	File type, should be: "jpg"
+Events	array<string>	R	Events with this record
+FilePath	string	R	File path, string max length is 259
+CutLength	int	R	File length that cut between start time and end time
+Length	int	O	File length of the whole file
+SummaryNew	array<object>	R	DB record object
++Key	string	R	DB record name, should be "**FaceRecognitionRecord**"
++Value	object	R	DB record value
+++RecResult	int	O	Recognition result, 0 means recognition failed, no candidates. 1 means recognition success, has candidates.
+++MachineAddress	string	O	Machine address, string max length is 259
+++IsGlobalScene	bool	O	Is the global scene picture or not
+++ImageInfo	object	O	Big picture info
++++Length	int	O	The length of the picture
++++FilePath	string	O	The file path of the picture, max string length is 259
+++Object	object	O	The target face info
++++Sex	string	O	Sex, it can be "Man", "Woman", "Unknown"
++++Age	int	O	Age
++++Glasses	int	O	Glasses Status, 0: unknown, 1: not wear, 2: wear
++++Eye	int	O	Eye status, 0: not detected, 1: close eye, 2: open eye
++++Mouth	int	O	Mouth status, 0: not detected, 1: close mouth, 2: open mouth
++++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask
++++Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard
+++Candidates	array<object>	O	The info of candidates face from face group.
++++Similarity	int	O	Similarity
++++Person	object	O	person info
+++++Name	string	O	Person Name, max string length is 63
+++++Sex	string	O	Sex, it can be "Male", "Female", "Unknown"
+++++Birthday	string	O	The person's birthday, ex: "1980-01-01"
+++++Country	string	O	Country name, length must be 2, value should be according to ISO3166
+++++Province	string	O	Province name, max string length is 63
+++++City	string	O	City name, max string length is 63
+++++CertificateType	string	O	Certificate Type. It can be: "IC', "Passport", "Military", "Unknown"
+++++ID	string	O	Person ID of CertificateType, max string length is 31
+++++Feature State	int	O	Feature State, 0:Unknown, 1:Failed, 2:OK
[Example]			
Request	GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100		

Response	found=100 items[0].Channel=1 items[0].StartTime=2011-1-1 12:00:00 items[0].EndTime=2011-1-1 13:00:00 items[0].Type=jpg items[0].Events[0]=FaceRecognition items[0].FilePath =/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].Length =790 items[0].SummaryNew[0].Key=FaceRecognitionRecord items[0].SummaryNew[0].Value.RecResult=1 items[0].SummaryNew[0].Value.MachineAddress=BinJiang items[0].SummaryNew[0].Value.IsGlobalScene=true items[0].SummaryNew[0].Value.ImageInfo.Length=123 items[0].SummaryNew[0].Value.ImageInfo.FilePath=/tmp/1.jpg items[0].SummaryNew[0].Value.Object.Sex=Man items[0].SummaryNew[0].Value.Object.Age=40 items[0].SummaryNew[0].Value.Object.Glasses=1 items[0].SummaryNew[0].Value.Object.Eye=2 items[0].SummaryNew[0].Value.Object.Mouth=1 items[0].SummaryNew[0].Value.Candidates[0].Similarity=50 items[0].SummaryNew[0].Value.Candidates[0].Person.Name=ZhangSan items[0].SummaryNew[0].Value.Candidates[0].Person.Birthday=1980-01-01 items[0].SummaryNew[0].Value.Candidates[0].Person.Sex=Male items[0].SummaryNew[0].Value.Candidates[0].Person.Country=CN items[0].SummaryNew[0].Value.Candidates[0].Person.Province=ZheJiang items[0].SummaryNew[0].Value.Candidates[0].Person.City=HangZhou items[0].SummaryNew[0].Value.Candidates[0].Person.CertificateType=IC items[0].SummaryNew[0].Value.Candidates[0].Person.ID=1234567890 items[0].SummaryNew[0].Value.Candidates[0].Person.FeatureState=0 …

Step 4 Close the finder.

This API is the same as the API in "4.11.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.11.5 Find Media Files".

4.11.8 Find media files with HumanTrait info

Step 1 Create a media files finder.

This API is the same as the API in "4.11.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and HumanTrait conditions with the finder

Table 4-179

URL	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findFile
Method	GET
Description	Start to find media files satisfied the common conditions and HumanTrait conditions with the finder.

[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
condition	object	R	The search conditions
+Channel	int	R	The video channel to search, video channel index start from 1, use -1 to search all video channel.
+StartTime	string	O	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	O	End time to search, ex: "2010-06-25 12:05:00"
+Types	array<string>	O	Search file types, should be "jpg"
+Flags	array<string>	O	Search flags. It can be: "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout" should include “Event”
+Events	array<string>	R	Search event list, must be one string: "HumanTrait"
+Dirs	array<string>	O	Search directory list, if omit, search all. Each dir path max string length is 259.
+DB	object	R	DB filter object
++HumanTrait RecordFilter	object	R	DB filter for HumanTrait
+++HumanAttributes	object	R	Human attributes
++++CoatColor	array<string>	O	Coat color, refer to CoatColor in 16.1.3.5 [Event] HumanTrait for available values, max array size is 4
++++CoatType	array<int>	O	Coat type, refer to CoatType in 16.1.3.5 [Event] HumanTrait for available values, max array size is 2
++++Trousers Color	array<string>	O	Trousers color, refer to TrousersColor in 16.1.3.5 [Event] HumanTrait for available values, max array size is 4
++++TrousersType	array<int>	O	Trousers type, refer to TrousersType in 16.1.3.5 [Event] HumanTrait for available values, max array size is 2
++++HasHat	int	O	Has hat or not, 0: all, 1: not has hat, 2: has hat
++++HasBag	int	O	Has bag or not, 0: all, 1: not has bag, 2: has bag
++++Sex	string	O	Sex, can be "Man", "Woman", if omit, search all
++++Age	array<int>	O	Age range, ex：[25, 40]
++++HairStyle	int	O	Hair style, 0: all, 1: long hair, 2: short hair, 3: ponytail, 4: updo, 5: hiddened
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.Events[0]=HumanTrait&condition.DB.HumanTraitRecordFilter.HumanAttributes.CoatColor[0]=White&condition.DB.HumanTraitRecordFilter.HumanAttributes.CoatColor[1]=Yellow&condition.DB.HumanTraitRecordFilter.HumanAttributes.CoatType=1&condition.DB.HumanTraitRecordFilter.HumanAttributes.HasHat=2&condition.DB.HumanTraitRecordFilter.HumanAttributes.Sex=Man&condition.DB.HumanTraitRecordFilter.HumanAttributes.Age[0]=30&condition.DB.HumanTraitRecordFilter.HumanAttributes.Age[1]=50&condition.DB.HumanTraitRecordFilter.HumanAttribute		

	s.HairStyle=1
Response	OK

Step 3 Get the media file information found by the finder

Table 4-180

URL	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findNextFile		
Method	GET		
Description	Get the media file information found by the finder		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
count	int	R	Record num to get, should less than 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
found	int	R	Record num that found.
items	array<object>	R	Result record items.
+Channel	int	R	The video channel index start from 0,
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Type	string	R	File type, should be: "jpg"
+Events	array<string>	R	Events with this record
+FilePath	string	R	File path, string max length is 259
+CutLength	int	R	File length that cut between start time and end time
+Length	int	O	File length of the whole file
+SummaryNew	object	R	Face info
++Key	string	R	DB record name, should be "**HumanTraitRecord**"
++Value	object	R	DB record value
+++HumanAttributes	object	O	Human attributes
++++CoatColor	string	O	Coat color, refer to CoatColor in 16.1.3.5 [Event] HumanTrait for available values,
++++CoatType	int	O	Coat type, refer to CoatType in 16.1.3.5 [Event] HumanTrait for available values,
++++TrousersColor	string	O	Trousers color, refer to TrousersColor in 16.1.3.5 [Event] HumanTrait for available values,
++++TrousersType	int	O	Trousers type, refer to TrousersType in 16.1.3.5 [Event] HumanTrait for available values,
++++HasHat	int	O	Has hat or not, 0: unknown, 1: not has hat, 2: has hat
++++HasBag	int	O	Has bag or not, 0: unknown, 1: not has bag, 2: has bag
++++Sex	string	O	Sex, can be "Man", "Woman", "Unknown"
++++Age	int	O	Age
++++HairStyle	int	O	Hair style, 0: unknown, 1: long hair, 2: short hair, 3: ponytail, 4: updo, 5: hiddened

++++HasUmbrella	int	O	Has umbrella or not, 0: unknown, 1: not has umbrella, 2: has umbrella
++++Bag	int	O	Bag type, 0: unknown, 1: handbag, 2: shoulder bag, 3: knapsack, 4: draw-bar box
++++Cap	int	O	Cap style, 0: unknown, 1: normal cap, 2: helmet
+++FaceAttributes	object	O	Face attributes
++++Sex	string	O	Sex, can be "Man", "Woman", "Unknown"
++++Age	int	O	Age
++++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask
++++Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard
++++Glass	Int	O	Glasses Status, 0: unknown, 1: not wearing, 2: normal Glasses, 3: sun glasses, 4: black frame glasses
++++Emotion	string	O	Emotion info. It can be: "Unknown", "Smile", "Anger", "Sadness", "Disgust", "Fear", "Surprise", "Neutral", "Laugh", "Happy", "Confused", "Scream", "Lookaside"
+++FacePath	string	O	Face picture path, max string length is 259
+++FaceScenePath	string	O	Face scene picture path, max string length is 259
+++HumanPath	string	O	Human picture path, max string length is 259
+++HumanScenePath	string	O	Human scene picture path, max string length is 259
[Example]			
Request	GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100		

Response	found=100 items[0].Channel=1 items[0].StartTime=2011-1-1 12:00:00 items[0].EndTime=2011-1-1 13:00:00 items[0].Type=jpg items[0].Events[0]=HumanTrait items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].Length=790 items[0].SummaryNew.Key=HumanTraitRecord items[0].SummaryNew.Value.HumanAttributes.CoatColor=White items[0].SummaryNew.Value.HumanAttributes.CoatType=1 items[0].SummaryNew.Value.HumanAttributes.TrousersColor=Black items[0].SummaryNew.Value.HumanAttributes.TrousersType=1 items[0].SummaryNew.Value.HumanAttributes.HasHat=2 items[0].SummaryNew.Value.HumanAttributes.HasBag=1 items[0].SummaryNew.Value.HumanAttributes.Sex=Man items[0].SummaryNew.Value.HumanAttributes.Age=30 items[0].SummaryNew.Value.HumanAttributes.HairStyle=2 items[0].SummaryNew.Value.HumanAttributes.HasUmbrella=1 items[0].SummaryNew.Value.HumanAttributes.Bag=0 items[0].SummaryNew.Value.HumanAttributes.Cap=2 items[0].SummaryNew.Value.FaceAttributes.Sex=Man items[0].SummaryNew.Value.FaceAttributes.Age=35 items[0].SummaryNew.Value.FaceAttributes.Mask=0 items[0].SummaryNew.Value.FaceAttributes.Beard=1 items[0].SummaryNew.Value.FaceAttributes.Glass=2 items[0].SummaryNew.Value.FaceAttributes.Emotion=Smile items[0].SummaryNew.Value.FacePath=/mnt/2010/8/11/dav/15:40:50.jpg items[0].SummaryNew.Value.FaceScenePath=/mnt/2010/8/11/dav/15:40:51.jpg …

Step 4 Close the finder.

This API is the same as the API in "4.11.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.11.5 Find Media Files".

4.11.9 Find media files with TrafficCar info

Step 1 Create a media files finder.

This API is the same as the API in "4.11.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and TrafficCar conditions with the finder

Table 4-181

URL	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findFile
Method	GET
Description	Start to find media files satisfied the common conditions and TrafficCar conditions with the finder.

	Note: some request params can use compare condition, it's value is an array, first item is a string of compare condition type, and the following items are compare values. Compare condition type can be: "==": means equal, followed with one param, if param type is string, then param value can contain some “*” to match any string. "||": means equal one of the params, followed with one or more params. "<>": means inside range, followed by two integer param, "><": means outside range, followed by two integer param,

[Request Params] (key=value format at URL)

Name	Type	R/O	Param Description
object	int	R	The finder object id
condition	object	R	The search conditions
+Channel	int	R	The video channel to search, video channel index start from 1, use -1 to search all video channel.
+StartTime	string	O	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	O	End time to search, ex: "2010-06-25 12:05:00"
+Types	array<string>	O	Search file types
+Flags	array<string>	O	Search flags. It can be: "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout"
+Events	array<string>	O	Search event list Ignored, use Event under DB param.
+Dirs	array<string>	O	Search directory list, if omit, search all. Each dir path max string length is 259.
+DB	object	R	DB object
++TrafficCar	object	R	DB filter for TrafficCar
+++PlateNumber	<compare condition>	O	PlateNumber condition, use compare condition format, ex: ["==", "*888"]
+++Event	<compare condition>	O	Event condition, ex: ["||", "TrafficGate", "Alarm*"]
+++Speed	<compare condition>	O	Speed condition, ex: ["<>", 40, 80]
+++PlateType	<compare condition>	O	Plate type condition, use compare condition format, ex: ["==", "Armed"],
+++PlateColor	<compare condition>	O	Plate color condition, use compare condition format, ex: ["==", "Blue"],
+++VehicleColor	<compare condition>	O	Vehicle color condition, use compare condition format, ex: ["==", "White"],

[Response Params] (OK)

[Example]

Request	GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.Flags[0]=Event&condition.DB.TrafficCar.PlateNumber[0]=%3d%3d&condition.DB.TrafficCar.PlateNumber[1]=%2a888&condition.DB.TrafficCar.Speed[0]=%3c%3e&condition.DB.TrafficCar.Speed[1]=40&condition.DB.TrafficCar.Speed[2]=80&condition.DB.TrafficCar.VehicleColor[0]=%3d%3d&condition.DB.TrafficCar.VehicleColor[1]=White

Response	OK

Step 3 Get the media file information found by the finder

Table 4-182

URL	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findNextFile		
Method	GET		
Description	Get the media file information found by the finder		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
count	int	R	Record num to get, should less than 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
found	int	R	Record num that found.
items	array<object>	R	Result record items.
+Channel	int	R	The video channel index start from 0,
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Type	string	R	File type, should be: "jpg"
+Events	array<string>	R	Events with this record
+FilePath	string	R	File path, string max length is 259
+CutLength	int	R	File length that cut between start time and end time
+Length	int	O	File length of the whole file
+Summary	object	R	DB record info
++TrafficCar	object	R	TrafficCar record info
+++PlateNumber	string	R	Car plate number
+++PlateType	string	O	Plate type,
+++PlateColor	string	O	Plate color, ex: "Yellow", "Blue", … etc
+++VehicleColor	string	O	Vehicle color, ex: "Yellow", "Blue", … etc
+++Country	string	O	Country info. max string length is 19
+++Speed	int	O	Vehicle speed, unit is km/hour
+++Event	string	O	The event info, ex: “TrafficJunction”
[Example]			
Request	GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100		

Response	found=100 items[0].Channel=1 items[0].StartTime=2011-1-1 12:00:00 items[0].EndTime=2011-1-1 13:00:00 items[0].Type=jpg items[0].Events[0]=TrafficJunction items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].Length=790 items[0].Summary.TrafficCar.PlateNumber=A08888 items[0].Summary.TrafficCar.PlateType=Unknown items[0].Summary.TrafficCar.PlateColor=Blue items[0].Summary.TrafficCar.VehicleColor=White items[0].Summary.TrafficCar.Country=China items[0].Summary.TrafficCar.Speed=70 items[0].Summary.TrafficCar.Event=TrafficJunction …

Step 4 Close the finder.

This API is the same as the API in "4.11.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.11.5 Find Media Files".

4.11.10 Find media files with IVS info

Step 1 Create a media files finder.

This API is the same as the API in "4.11.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and IVS conditions with the finder

Table 4-183

URL	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Description	Start to find media files satisfied the common conditions and IVS conditions with the finder.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
condition	object	R	The search conditions
+Channel	int	R	The video channel to search, video channel index start from 1, use -1 to search all video channel.
+StartTime	string	O	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	O	End time to search, ex: "2010-06-25 12:05:00"
+Types	array<string>	O	Search file types
+Flags	array<string>	O	Search flags. It can be: "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout"
+Events	array<string>	O	Search event list,
+Dirs	array<string>	O	Search directory list, if omit, search all. Each dir path max string length is 259.

+DB	object	R	DB object
++IVS	object	R	DB filter for IVS
+++Rule	string	O	IVS rule condition. It can be: "CrossLineDetection" , "CrossRegionDetection" , "LeftDetection" , "WanderDetection" "MoveDetection" , "RioterDetection" , "CrossFenceDetection" , "TakenAwayDetection" , "PasteDetection" , "Preservation" , "StayDetection" , "TailDetection"
+++Action	string	O	IVS action. It can be: "Appear" , "Disappear" , "Inside" , "Cross"
+++ObjectType	array<string>	O	IVS object type, item in array can be: "Unknown" , "Human" , "Vehicle" , "NonMotor"
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.DB.IVS.Rule=CrossLineDetection&condition.DB.IVS.Action=Cross&condition.DB.IVS.ObjectType[0]=Human&condition.DB.IVS.ObjectType[1]=NonMotor		
Response	OK		

Step 3 Get the media file information found by the finder

Table 4-184

URL	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findNextFile		
Method	GET		
Description	Get the media file information found by the finder		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
count	int	R	Record num to get, should less than 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
found	int	R	Record num that found.
items	array<object>	R	Result record items.
+Channel	int	R	The video channel index start from 0,
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Type	string	R	File type, should be: "jpg"
+Events	array<string>	R	Events with this record
+FilePath	string	R	File path, string max length is 259
+CutLength	int	R	File length that cut between start time and end time
+Length	int	O	File length of the whole file
+Summary	object	R	DB record info
++IVS	object	R	IVS record info
+++Rule	string	R	IVS rule, see above for valid value
+++Action	string	O	IVS action, see above for valid value

+++ObjectType	string	O	IVS object type, see above for valid value
[Example]			
Request	GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100		
Response	found=100 items[0].Channel=1 items[0].StartTime=2011-1-1 12:00:00 items[0].EndTime=2011-1-1 13:00:00 items[0].Type=jpg items[0].Events[0]=CrossLineDetection items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].Length=790 items[0].Summary.IVS.Rule=CrossLineDetection items[0].Summary.IVS.Action=Cross items[0].Summary.IVS.ObjectType=Human …		

Step 4 Close the finder.

This API is the same as the API in "4.11.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.11.5 Find Media Files".

4.11.11 Find media files with NonMotor info

Step 1 Create a media files finder.

This API is the same as the API in "4.11.5 Find Media Files".

Step 2 Start to find media files satisfied the common conditions and NonMotor conditions with the finder

Table 4-185

URL	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findFile		
Method	GET		
Description	Start to find media files satisfied the common conditions and NonMotor conditions with the finder.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
condition	object	R	The search conditions
+Channel	int	R	The video channel to search, video channel index start from 1, use -1 to search all video channel.
+StartTime	string	O	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	O	End time to search, ex: "2010-06-25 12:05:00"
+Types	array<string>	O	Search file types
+Flags	array<string>	O	Search flags. It can be: "Timing", "Manual", "Marker", "Event", "Mosaic", "Cutout"
+Events	array<string>	O	Search event list,
+Dirs	array<string>	O	Search directory list, if omit, search all. Each dir path max string length is 259.

+DB	object	R	DB object
++NonMotorRecordFilter	object	R	DB filter for NonMotor
+++NumOfCycling	int	O	Number of people that cycling, value can be 1 ~ 3
+++Color	string	O	NonMotor color, can be "White" "Orange" "Pink" "Black" "Red" "Yellow" "Gray" "Blue" "Green" "Purple" "Brown" "Sliver" "Darkviolet" "Maroon" "Dimgray" "Whitesmoke" "Darkorange" "Mistyrose" "Tomato" "Olive" "Gold" "Darkolivegreen" "Chartreuse" "Greenyellow" "Forestgreen" "Seagreen" "Deepskyblue" "Cyan" "Other"
+++Category	string	O	NonMotor type, can be : "Non-Motor" "Bicycle" "Tricycle" "Motorcycle" "DualTriWheelMotorcycle" "LightMotorcycle" "EmbassyMotorcycle" "MarginalMotorcycle" "AreaoutMotorcycle" "ForeignMotorcycle" "TrialMotorcycle" "CoachMotorcycle"
+++Helmet	int	O	Helmet status, 0 : unknown, 1 : without helmet, 2 : with helmet
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findFile&object=08137&condition.Channel=1&condition.StartTime=2014-1-1%2012:00:00&condition.EndTime=2015-1-10%2012:00:00&condition.Types[0]=jpg&condition.DB.NonMotorRecordFilter.NumOfCycling=2&condition.DB.NonMotorRecordFilter.Color=White&condition.DB.NonMotorRecordFilter.Category=Bicycle&condition.DB.NonMotorRecordFilter.Helmet=1		
Response	OK		

Step 3 Get the media file information found by the finder

Table 4-186

URL	http://<*server*>/cgi-bin/mediaFileFind.cgi?action=findNextFile		
Method	GET		
Description	Get the media file information found by the finder		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
object	int	R	The finder object id
count	int	R	Record num to get, should less than 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
found	int	R	Record num that found.
items	array<object>	R	Result record items.
+Channel	int	R	The video channel index start from 0,
+StartTime	string	R	Start time to search, ex: "2010-05-25 12:05:00"
+EndTime	string	R	End time to search, ex: "2010-06-25 12:05:00"
+Type	string	R	File type, should be: "jpg"
+Events	array<string>	R	Events with this record
+FilePath	string	R	File path, string max length is 259
+CutLength	int	R	File length that cut between start time and end time
+Length	int	O	File length of the whole file

+SummaryNew	object	R	NonMotor info
++Key	string	R	DB record name, should be "**NonMotorRecordFilter**"
++Value	object	R	DB record value
+++NumOfCycling	int	O	Number of people that cycling, value can be 1 ~ 3
+++Color	string	O	NonMotor color, can be "White" "Orange" "Pink" "Black" "Red" "Yellow" "Gray" "Blue" "Green" "Purple" "Brown" "Sliver" "Darkviolet" "Maroon" "Dimgray" "Whitesmoke" "Darkorange" "Mistyrose" "Tomato" "Olive" "Gold" "Darkolivegreen" "Chartreuse" "Greenyellow" "Forestgreen" "Seagreen" "Deepskyblue" "Cyan" "Other"
+++Category	string	O	NonMotor type, can be : "Non-Motor" "Bicycle" "Tricycle" "Motorcycle" "DualTriWheelMotorcycle" "LightMotorcycle" "EmbassyMotorcycle" "MarginalMotorcycle" "AreaoutMotorcycle" "ForeignMotorcycle" "TrialMotorcycle" "CoachMotorcycle"
+++Helmet	int	O	Helmet status, 0 : unknown, 1 : without helmet, 2 : with helmet
[Example]			
Request	GET http://192.168.1.108/cgi-bin/mediaFileFind.cgi?action=findNextFile&object=08137&count=100		
Response	found=100 items[0].Channel=1 items[0].StartTime=2011-1-1 12:00:00 items[0].EndTime=2011-1-1 13:00:00 items[0].Type=jpg items[0].Events[0]=NonMotorDetect items[0].FilePath=/mnt/dvr/sda0/2010/8/11/dav/15:40:50.jpg items[0].Length=790 items[0].SummaryNew.Key=NonMotorRecordFilter items[0].SummaryNew.Value.NumOfCycling=2 items[0].SummaryNew.Value.Color=White items[0].SummaryNew.Value.Category=Bicycle items[0].SummaryNew.Value.Helmet=1 …		

Step 4 Close the finder.

This API is the same as the API in "4.11.5 Find Media Files".

Step 5 Destroy the finder.

This API is the same as the API in "4.11.5 Find Media Files".

4.11.12 Download Media File with the File Name

Table 4-187

Syntax	http://<*server*>/cgi-bin/RPC_Loadfile/<***Filename***>
Method	GET
Description	Download a file by filename. To get filename by chapter file finding.
Example	http://192.168.1.108/cgi-bin/RPC_Loadfile/mnt/sd/2015-01-08/001/dav/19/19.57.12-19.58.25[M][0@0][0].dav

Success Return	HTTP Code: 200 OK Content-Type: Application/octet-stream Content-Length: <fileLength> Body: <data> <data>
Comment	Parameters in URL: ***Filename***: name of media files which would be downloaded.

4.11.13 Download Media File between Times

Table 4-188

Syntax	http://<*server*>/cgi-bin/loadfile.cgi?action=startLoad&channel=<***ChannelNo***>&startTime=<***starttime***>&endTime=<***endtime***>[&subtype=<***typeNo***>]
Method	GET
Description	Download the media data between start time and end time.
Example	http://192.168.1.108/cgi-bin/loadfile.cgi?action=startLoad&channel=1&startTime=2012-10-8%2013:00:01&endTime=2012-10-8%2014:00:01&subtype=0
Success Return	HTTP Code: 200 OK Content-Type: Application/octet-stream Content-Length:<fileLength> Body: <data> <data>
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. ***typeNo***: the stream type, default 0 if not specified. 0-Main Stream 1-Extra Stream 1 2-Extra Stream 2 ***starttime*** & ***endtime***: video start time and end time. Time format: yyyy-mm-dd hh:mm:ss

4.11.14 Encrypted Download Media File with the File Name

Table 4-189

Syntax	http://<*server*>/cgi-bin/RecordStreamInterleaved.cgi?action=attachStream&path=<***File name***>[&password=<**password**>]
Method	GET
Description	Encrypted Download a file by filename. To get filename by chapter file finding.
Example	http://172.29.2.241/cgi-bin/RecordStreamInterleaved.cgi?action=attachStream&path=/mnt/sd/2019-07-01/001/dav/12/12.36.16-12.36.26[F][0@0][0].dav

Success Return	HTTP Code: 200 OK Content-Type: Application/octet-stream Content-Length: <fileLength> Body: <data> <data>
Comment	Parameters in URL: ***Filename***: name of media files which would be downloaded. ***password:*** password for encrypting media file data. if not set password, then use the preset password

4.12 User management

4.12.1 Get Information of a Particular User

Table 4-190

Syntax	http://<*server*>/cgi-bin/userManager.cgi?action=getUserInfo&name=<***userName***>
Method	GET
Description	Get user information with name ***userName***.
Example	http://192.168.1.108/cgi-bin/userManager.cgi?action=getUserInfo&name=admin
Success Return	user.Name=admin user.Memo=admin 's account user.Group=admin user.Reserved=true user.Sharable=true user. AuthList=<***authList***>
Comment	—

4.12.2 Get Information of All Users

Table 4-191

Syntax	http://<*server*>/cgi-bin/userManager.cgi?action=getUserInfoAll
Method	GET
Description	Get information of all users.
Example	http://192.168.1.108/cgi-bin/userManager.cgi?action=getUserInfoAll
Success Return	users[0].Group=admin users[0].Id=1 users[0].Memo=admin 's account users[0].Name=admin users[0].Reserved=true users[0].Sharable=true users[0]. AuthList=<authList> users[1].Group=admin …
Comment	—

4.12.3 Get Information of All Active Users

Table 4-192

Syntax	http://<*server*>/cgi-bin/userManager.cgi?action=getActiveUserInfoAll
Method	GET
Description	Get active users.
Example	http://192.168.1.108/cgi-bin/userManager.cgi?action=getActiveUserInfoAll
Success Return	users[0].name=admin users[0].ip=10.43.2.16 users[0].group=admin users[0].clienttype=web3.0 users[0].logintime=2011-11-08 09:51:03
Comment	—

4.12.4 Get Information of a Particular Group

Table 4-193

Syntax	http://<*server*>/cgi-bin/userManager.cgi?action=getGroupInfo&name=<***groupName***>
Method	GET
Description	Get group setting with name ***groupName***.
Example	http://192.168.1.108/cgi-bin/userManager.cgi?action=getGroupInfo&name=admin
Success Return	group.Name=admin group.Memo=administrator group group. AuthorityList=<authList>
Comment	Parameters in URL: The device has one or two default user groups: "admin"or "admin"and "user". The "admin"group has all the authorities of operating the device. The "user"group only has monitoring and replaying authorities. ***groupName***: name of the group. If the group named groupName does not exist, the device returns Error.

4.12.5 Get Information of All Groups

Table 4-194

Syntax	http://<*server*>/cgi-bin/userManager.cgi?action=getGroupInfoAll
Method	GET
Description	Get information of all groups.
Example	http://192.168.1.108/cgi-bin/userManager.cgi?action=getGroupInfoAll
Success Return	group[0].Name=admin group[0].Memo=administrator group group[0]. AuthorityList=<*authList*> group[1].Name=user group[1].Memo=user group group[1]. AuthorityList=<*authList*> group[2]….
Comment	—

4.12.6 Add a New User

Table 4-195

Syntax	http://<*server*>/cgi-bin/userManager.cgi?action=addUser&user.Name=<***userName***>&user.Password=<***userPassword***>&user.Group=<***userGroup***>&user.Sharable=<***userSharable***>[&user.Memo=<***userMemo***>&user.Reserved=<***userReserved***>]
Method	GET
Description	Add a user.
Example	http://192.168.1.108/cgi-bin/userManager.cgi?action=addUser&user.Name=George&user.Password=123456&user.Group=user&user.Sharable=true&user.Reserved=false
Success Return	OK
Comment	Parameters in URL: ***userGroup***: string, the range is "admin"and "user". In different group, the user has different authorities. ***userSharable***: bool, true means allow multi-point login. ***userReserved****:* bool, true means this user can't be deleted.

4.12.7 Delete a User

Table 4-196

Syntax	http://<*server*>/cgi-bin/userManager.cgi?action=deleteUser&name=<***userName***>
Method	GET
Description	Delete user with name ***username***.
Example	http://192.168.1.108/cgi-bin/userManager.cgi?action=deleteUser&name=George
Success Return	OK
Comment	—

4.12.8 Modify User Information

Table 4-197

Syntax	http://<*server*>/cgi-bin/userManager.cgi?action=modifyUser&name=<*UserName*>&user.Memo=<*userMemo*>&user.Group=<*userGroup*>&user.Reserved=<*userReserved*>&user.Sharable=<*userSharable*>
Method	GET
Description	Modify user info.
Example	http://192.168.1.108/cgi-bin/userManager.cgi?action=modifyUser&name=George&user.Group=admin
Success Return	OK
Comment	User is identified by <UserName>, other params are the same with AddUser.

4.12.9 Modify User's Password

Table 4-198

Syntax	http://<*server*>/cgi-bin/userManager.cgi?action=modifyPassword&name=<***username***>&pwd=<***newPwd***>&pwdOld=<***oldPwd***>
Method	GET
Description	Modify user password.
Example	http://192.168.1.108/cgi-bin/userManager.cgi?action=modifyPassword&name=George&pwd=abcdef&pwdOld=123456
Success Return	OK
Comment	Old password ***oldPwd*** should be supplied, new password is ***newPwd***.

4.13 Log

4.13.1 Find Logs

1. Whether or not found logs satisfied the conditions

Table 4-199

Syntax	http://<*server*>/cgi-bin/log.cgi?action=startFind&condition.StartTime=<***start***>&condition.EndTime=<***end***>[& condition.Type=<***type***>]
Method	GET
Description	Start to find log.
Example	Find log between 2011-1-1 12:00:00 and 2011-1-10 12:00:00, URL is: http://192.168.1.108/cgi-bin/log.cgi?action=startFind&condition.StartTime=2011-1-1%2012:00:00&condition.EndTime=2011-1-10%2012:00:00
Success Return	token=1
Comment	Parameters in URL: ***start/end***: the start/end time of log. Format is: yyyy-mm-dd hh:mm:ss. In response, there is a token for further log finding process. If token is greater than 0, logs are found; otherwise no logs are found. ***Type***: log type. The range is { "System", "Config", "Event", "Storage", "Account", "Data", "File", "CourseRecord" }.

2. Get the particular number of logs

Table 4-200

Syntax	http://<*server*>/cgi-bin/log.cgi?action=doFind&token=<***TokenValue***>&count=<***logCount***>
Method	GET
Description	Find log with token TokenValue and count logCount.
Example	http://192.168.1.108/cgi-bin/log.cgi?action=doFind&token=1&count=100
Success Return	***found***=2 items[0].***RecNo***=789 items[0].***Time***=2011-05-20 11:59:10 items[0].***Type***=ClearLog items[0].***User***=admin items[1].***Detail***.Compression=H.264->MJPG items[1].***Detail***.Data=Encode items[1].***RecNo***=790

	items[1].***Time***=2011-05-20 11:59:21 items[1].***Type***=SaveConfig items[1].***User***=System …
Comment	Parameters in URL: The ***TokenValue*** is got by startFind in the above section, and ***logCount*** is the count of logs for this query. The maximum value of ***logCount*** is 100.

Appendix:

Field in Response	**Description**
found	Count of found log, found is 0 if no log is found.
User	User name.
Type	Log type.
Time	Time of this log.
RecNo	Log number.
Detail	Log details.

3. Stop query logs

Table 4-201

Syntax	http://<*server*>/cgi-bin/log.cgi?action=stopFind&token=<***TokenValue***>
Method	GET
Description	Stop query log by token ***TokenValue***.
Example	http://192.168.1.108/cgi-bin/log.cgi?action=stopFind&token=1
Success Return	OK
Comment	Parameters in URL: The ***TokenValue*** is got by startFind in above section

4.13.2 Clear All the Logs

Table 4-202

Syntax	http://<*server*>/cgi-bin/log.cgi?action=clear
Method	GET
Description	Clear all the logs.
Example	http://192.168.1.108/cgi-bin/log.cgi?action=clear
Success Return	OK
Comment	—

4.13.3 Backup Logs

Table 4-203

Syntax	http://<*server*>/cgi-bin/Log.backup?action=All&condition.StartTime=<***startTime***>&condition.EndTime=<***endTime***>

Method	GET
Description	Download the log information between the start time and the end time as a file named Log. Backup default.
Example	http://192.168.1.108/cgi-bin/Log.backup?action=All&condition.StartTime=2014-8-25%2000:02:32&condition.EndTime=2020-8-25%2001:02:32
Success Return	HTTP/1.1 200 OK CONTENT-LENGTH: 743087 CONNECTION: close Content-type: application/binarytet-stream; charset=utf-8 &w_User: default &Time: 2014-09-01 15:20:45 &Type: VideoLoss &Content: EventType: VideoLoss channel: <8> StartTime: 2014-09-01 15:20:45 …
Comment	Parameters in URL: ***startTime/endTime***: the start/end time when log info built. 24 hour Format, as: yyyy-mm-dd hh:mm:ss. For example: 2014-8-25 00:02:32 2020-8-25 01:02:32

4.14 Upgrader

4.14.1 Strat to Upgrade

Table 4-204

Syntax	http://<*server*>/cgi-bin/upgrader.cgi?action=uploadFirmware
Method	POST
Description	Use this message to upload the firmware, and when the device receiving all the data successfully, it will start to upgrade the device, and then use the getState method to get the state.

Example	POST /cgi-bin/upgrader.cgi?action=uploadFirmware HTTP/1.1 Host: 192.168.1.108 Connection: keep-alive Content-Type: multipart/form-data; boundary=--------------------------8655433224198 Content-Length: xxxxxxxxx ----------------------------8655433224198 Content-Disposition:form-data;name="upgrade"; filename= "xxxxxx.bin" Content-Type: application/octet-stream Firmware data…. ----------------------------8655433224198--
Success Return	OK
Comment	—

4.14.2 Get Upgrade State

Table 4-205

Syntax	http://<*server*>/cgi-bin/upgrader.cgi?action=getState
Method	GET
Description	Use this message to upload the firmware, and when the device receiving all the data successfully, it will start to upgrade the device, and then use the getState method to get the state.
Example	http://192.168.1.108/cgi-bin/upgrader.cgi?action=getState
Success Return	state.***State***=Upgrading state.***Progress***=45
Comment	Parameters in Response: ***State***: the state of the upgrade, it can be Preparing, Downloading, DownloadFailed, Upgrading, Invalid, Failed, Succeeded, Cancelled, and NotEnoughMemory. ***Progress***: the progress of the upgrade.

4.14.3 Set upgrader url

Table 4-206

Syntax	**http://<*server*>/cgi-bin/upgrader.cgi?action=updateFirmwareByUrl**
Method	**GET**
Description	**Set upgrader url**
Example	**http://192.168.1.108//cgi-bin/upgrader.cgi?action=updateFirmwareByUrl&Url=https://aaa/bbb/ccc/license.bin**
Success Return	**OK**
Comment	**url with Upgrade packet address**

4.15 Wiper

4.15.1 Move Continuously

Table 4-207

Syntax	http://<*server*>/cgi-bin/rainBrush.cgi?action=moveContinuously&interval=<***Second***>[&channel=<***ChannelNo***>]
Method	GET
Description	Control the wiper to move continuously.
Example	http://192.168.1.108/cgi-bin/rainBrush.cgi?action=moveContinuously&interval=5
Success Return	OK
Comment	***Second***: integer, rain brush movement time interval which start from 1. ***ChannelNo***: integer, video channel index which starts from 1, default 1 if not specified.

4.15.2 Stop Move

Table 4-208

Syntax	http://<*server*>/cgi-bin/rainBrush.cgi?action=stopMove[&channel=<***ChannelNo***>]
Method	GET
Description	Control the wiper to stop moving.
Example	http://192.168.1.108/cgi-bin/rainBrush.cgi?action=stopMove
Success Return	OK
Comment	***ChannelNo***: integer, video channel index which starts from 1, default 1 if not specified.

4.15.3 Move Once

Table 4-209

Syntax	http://<*server*>/cgi-bin/rainBrush.cgi?action=moveOnce[&channel=<***ChannelNo***>]
Method	GET
Description	Control the wiper to move once.
Example	http://192.168.1.108/cgi-bin/rainBrush.cgi?action=moveOnce
Success Return	OK
Comment	***ChannelNo***: integer, video channel index which starts from 1, default 1 if not specified.

5.1 Image

5.1.1 Brightness, Contrast and Saturation

- Get brightness, contrast and saturation

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoColor
Method	GET
Description	Get brightness, contrast and saturation
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoColor
Success Return	***head***.*Brightness=50* ***head***.*Contrast=50* ***head***.*Saturation=50*
Comment	Parameters in URL: paramName and paramValue are as table below. In table below, ***head*** = table.VideoInSharpness [***ChannelNo***] [***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set brightness, contrast and saturation

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	SET
Description	Set brightness, contrast and saturation
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoColor[0][0].Brightness=50
Success Return	OK
Comment	Parameters in URL: paramName and paramValue are as table below. In table below, ***head*** = table.VideoInSharpness [***ChannelNo***] [***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

Appendix:

ParamName	ParamValue type	Description
head. Brightness	integer	Brightness, range is [0—100]
head. Contrast	integer	Contrast, range is [0—100]
head. Saturation	integer	Saturation, range is [0—100]

5.1.2 Sharpness

- Get sharpness

Table 5-1

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInSharpness
Method	GET
Description	Get sharpness
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoInSharpness
Success Return	head.Level=4 head.Sharpness=8
Comment	Parameters in URL: paramName and paramValue are as table below. In table below, ***head*** = table.VideoInSharpness [***ChannelNo***] [***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set sharpness

Table 5-2

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	SET
Description	Set sharpness
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInSharpness[0][0].Level=10&VideoInSharpness[0][0].Mode=1&VideoInSharpness[0][0].Sharpness=0
Success Return	OK
Comment	Parameters in URL: paramName and paramValue are as table below. In table below, ***head*** = VideoInSharpness [***ChannelNo***] [***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

Appendix:

ParamName	ParamValue type	Description
head. Sharpness	integer	Range is 0—100
head. Level	integer	Range is 0—100

5.1.3 Flip, Mirror and Rotate90

- Get flip, mirror and Rotate90

Table 5-3

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoImageControl
Method	GET
Description	Get flip, mirror and Rotate90
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoImageControl
Success Return	head.Flip=true head.Mirror=false head.Rotate90=0
Comment	Parameters in Response: ***head*** = table.VideoImageControl [***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set flip, mirror and Rotate90

Table 5-4

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	SET
Description	Set flip, mirror and Rotate90
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoImageControl[0].Flip=true
Success Return	OK
Comment	Parameters in URL: ***head*** = VideoImageControl [***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

Appendix:

ParamName	ParamValue type	Description
head. Flip	bool	true: enable video flip function false: disable video flip function
head. Mirror	bool	true: enable video mirror function false: disable video mirror function

ParamName	ParamValue type	Description
head. Rotate90	integer	Range is {0,1,2} Video rotation: 0: No rotate 1: clockwise rotate 90° 2: anticlockwise rotate 90°

5.2 Exposure

5.2.1 Exposure Config

- Get exposure

Table 5-5

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInExposure
Method	GET
Description	Get exposure
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoInExposure
Success Return	head.AntiFlicker=0 head.Gain=50 head.GainMax=50 head.GainMin=0 head.Iris=50 head.IrisAuto=false head.Mode=0 head.Value1=40 head.Value2=40
Comment	Parameters in Response: ***head*** = table. VideoInExposure [***ChannelNo***][***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set exposure

Table 5-6

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	SET
Description	Set exposure
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInExposure[0][0].Iris=50
Success Return	OK
Comment	Parameters in URL: ***head*** = VideoInExposure [***ChannelNo***] [***ConfigNo***]

	ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

Appendix:

ParamName	ParamValue type	Description
head. AntiFlicker	integer	Range is {0,1,2} AntiFlicker mode: 0: Outdoor 1: 50 Hz AntiFlicker 2: 60 Hz AntiFlicker
head. Gain	integer	Range is [0—100] If GainAuto is true, it's upper limit of auto gain, else it's the fixed gain adjust value.
head. GainMax	integer	Range is 0–100; the value must be greater than GainMin.
head. GainMin	integer	Range is 0–100; the value must be smaller than GainMax.
head. Iris	integer	Manual Iris setting. Range is 0–100.
head. IrisAuto	bool	Enable Iris automatically. true: IrisAuto **false: No IrisAuto**
head. Mode	integer	0: "Auto" by default 1: Low noise 2: Anti-smear 4: Manual (**range**) 5: **Aperture priority** 6: Manual (fixed) 7: **Gain priority** 8: **Shutter priority** 9: Flash light matching mode
head. Value1	float	Range is [0-1000], unit is millisecond If ExposureSpeed is 0(AutoExposure enable), it's lower limit of AutoExposure time, otherwise it's time of manualExposure
head. Value2	float	Range is [0-1000], unit is millisecond Upper limit of AutoExposure time, should be bigger than ExposureValue1

5.3 Backlight

5.3.1 Backlight Config

- Get backlight

Table 5-7

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInBacklight
Method	GET
Description	Get backlight
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoInBacklight
Success Return	head.GlareInhibition=50 head.Mode=Off head.WideDynamicRange=50
Comment	Parameters in Response: ***head*** = table. VideoInBacklight [***ChannelNo***][***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set backlight

Table 5-8

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	SET
Description	Set backlight
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInBacklight[0][0].GlareInhibition=50
Success Return	OK
Comment	Parameters in URL: ***head*** = VideoInBacklight [***ChannelNo***] [***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

Appendix:

ParamName	ParamValue type	Description
head. GlareInhibition	integer	Range is 1-100.
head. Mode	char[32]	Off: Switched off Backlight: Backlight compensation GlareInhibition: HLC WideDynamic: WDR

ParamName	ParamValue type	Description
		SSA: Scene adaptation
head. WideDynamicRange	integer	Range is 1–100.

5.4 White Balance

5.4.1 White Balance Config

- Get white balance

Table 5-9

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInWhiteBalance
Method	GET
Description	Get white balance
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoInWhiteBalance
Success Return	head.GainBlue=50 head.GainRed=50 head.Mode=Auto
Comment	Parameters in Response: ***head*** = table. VideoInWhiteBalance [***ChannelNo***][***ColorConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set white balance

Table 5-10

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	SET
Description	Set white balance
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInWhiteBalance[0][0].GainBlue=50
Success Return	OK
Comment	Parameters in URL: ***head*** = VideoInWhiteBalance [***ChannelNo***] [***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

Appendix:

ParamName	ParamValue type	Description

ParamName	ParamValue type	Description
head. GainBlue	integer	Range is [0—100] Gain for blue value, Value is effective when WhiteBalance is "Custom."
head. GainRed	integer	Range is [0—100] Gain for red value, Value is effective when WhiteBalance is "Custom."
head. Mode	char[][32]	Auto; Indoor; Outdoor; ATW; Manual; Sodium; Natural; StreetLamp; ManualDatum

5.5 Day-Night

5.5.1 Day-Night Config

- **Get day-night**

Table 5-11

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInDayNight
Method	GET
Description	Get day-night
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoInDayNight
Success Return	head.Delay=10 head.Mode=Brightness head.Sensitivity=2 head.Type=Mechanism
Comment	***head*** = table.VideoInDayNight[***ChannelNo***][***ConfigNo***]: ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- **Set Day-Night**

Table 5-12

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	SET
Description	Set day-night
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInDayNight[0][0].Mode=BlackWhite
Success Return	OK
Comment	Parameters in URL: ParamName and paramValue are as table below. In table below, ***head*** = VideoInDayNight[***ChannelNo***][***ConfigNo***]

Appendix:

ParamName	ParamValue type	Description
head. Type	string	The range is {"Electron", "Mechanism"}, the way of ICR switching.
head. Mode	string	The range is {"**Color**", "**Brightness**", "**BlackWhite**}. "Color": Always "color" "Brightness": Day/Night Auto "BlackWhite": Always black-and-white
head. Sensitivity	integer	Range is [1-3]. Sensitivity of switching mode
head. Delay	integer	Range is [2-10]. Delay seconds when switching mode.

5.6 Zoom and Focus

To get the capability set of video input, refer to 4.5.12. For instance, you can use the following URL:
http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getCaps&channel=1
If ElectricFocus or SyncFocus of the return value is true, use 错误!未找到引用源。 错误!未找到引用源。; Otherwise, use 错误!未找到引用源。 错误!未找到引用源。.

5.6.1 Adjust Focus

Table 5-13

Syntax	http://<*server*>/cgi-bin/devVideoInput.cgi?action=adjustFocus&focus=<***zoomNo***>[&channel=<***ChannelNo***>]
Method	GET
Description	Ajust magnification and the focus.
Example	http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=adjustFocus&focus=0.5&zoom=-0.5
Success Return	OK
Comment	Parameters in URL: ***focusNo*** : float, the range is between 0 and 1; -1 means reset to position 0. ***zoomNo*** : float, the range is between 0 and 1; -1 means reset to position 0. ***ChannelNo***: integer, video channel index which starts from 1.

5.6.2 Adjust Focus Continuously

Table 5-14

Syntax	http://<*server*>/cgi-bin/devVideoInput.cgi?action=adjustFocusContinuously&focus=<***focusNo***>&zoom=<***zoomNo***>[&channel=<***ChannelNo***>]
Method	GET
Description	Adjust magnification and the focus continuously.
Example	If we want to adjust focus, the API like this: http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=adjustFocusContinuously&focus=0.02&zoom=-1 and when the motor is moving, we send below command to let it stop: http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=adjustFocusContinuously&focus=0&zoom=-1

Success Return	OK
Comment	Parameters in URL: ***focusNo***: float, the range is -1 < focus < 1; 0 means stop. ***zoomNo***: float, the range is -1 < zoom < 1; 0 means stop. ***ChannelNo***: integer, video channel index which starts from 1. The value means the moving speed of motor lens, positive value means move forwards, negative value means move backwards. This command is used to drive the lens move continuously, until it reaches end. When the motor is moving, you can send this command again with "focus" or "zoom" parameter as 0 to stop it immediately. In this command, when you adjust the focus parameter, the zoom parameter should be -1, and the focus parameter should be -1 when adjust the zoom parameter.

5.6.3 Auto Focus

Table 5-15

Syntax	http://<*server*>/cgi-bin/devVideoInput.cgi?action=autoFocus[&channel=<***ChannelNo***>]
Method	GET
Description	Auto focus.
Example	http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=autoFocus
Success Return	OK
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1.

5.6.4 Get Focus Status

Table 5-16

Syntax	http://<*server*>/cgi-bin/devVideoInput.cgi?action=getFocusStatus[&channel=<***ChannelNo***>]
Method	GET
Description	Get device focus status.
Example	http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getFocusStatus
Success Return	status.Focus=0.5 status.Zoom=0.5 status.Status=Normal
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. Parameters in Response: The range of status**.Status** is "Normal" and "Autofocus". This command must be continual executed until status.Status is "Normal".

5.6.5 Zoom Config

- Get zoom

Table 5-17

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInZoom
Method	GET
Description	Get zoom
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoInZoom
Success Return	head.DigitalZoom=true head.Speed=7
Comment	Parameters in URL: ***head*** = table.VideoInZoom [***ChannelNo***] [***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means

- Set zoom

Table 5-18

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	SET
Description	Set zoom
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInZoom[0][0].DigitalZoom=false&VideoInZoom[0][0].Speed=8
Success Return	OK
Comment	Parameters in URL: ***head*** = VideoInZoom [***ChannelNo***] [***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

Appendix:

ParamName	ParamValue type	Description
head. DigitalZoom	integer	true: Enable Digital Zoom false: Disable Digital Zoom
head. Speed	integer	Range is 0—100

5.6.6 Focus Config

- Get focus

Table 5-19

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInFocus
Method	GET
Description	Get focus
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoInFocus

Success Return	head.FocusLimit=100 head.FocusLimitSelectMode=Manual head.Mode=3 head.Sensitivity=1
Comment	—

- Set focus

Table 5-20

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	SET
Description	Set focus
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInFocus[0][0].FocusLimit=1000
Success Return	OK
Comment	Parameters in URL: paramName and paramValue are as table below. In table below, ***head*** = VideoInFocus [***ChannelNo***] [***ConfigNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

Appendix:

ParamName	ParamValue type	Description
head. Mode	integer	2—Auto focus, 3—Half auto focus, 4—Manual focus
head. FocusLimit	integer	100, 1000, 2000, 3000, 5000. If xxxx1 is included in the queue, replace 1 with 0, and add + sign at the end; for instance, 200001 will be displayed as 200000mm+. Get the "FocusLimit" capacity to determine the value range.
head. Sensitivity	integer	Range is 0,1,2 0—high, 1—default, 2—low
head. FocusLimitSelectMode	string	Manual or Auto.

5.7 Lighting

5.7.1 Lighting Config

- Get lighting

Table 5-21

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=Lighting

Method	GET
Description	Get lighting
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=Lighting
Success Return	head.Correction=50 head.FarLight[0].Angle=0 head.FarLight[0].Light=0 head.Mode=ZoomPrio head.NearLight[0].Angle=0 head.NearLight[0].Light=0 head.MiddleLight[0].Angle=50 head.MiddleLight[0].Light=50
Comment	***head*** = table.Lighting[***ChannelNo***][***ConfigNo***]: ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***ConfigNo***: array index, can be 0, 1 or 2. 0 means config for day, 1 means config for night, and 2 means config for normal scene.

- Set lighting

Table 5-22

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set lighting
Example	Turn on light: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Lighting[0][0].FarLight[0].Light=10&Lighting[0][0].NearLight[0].Light=90&Lighting[0][0].Mode=Manual Shift the light to ZoomPrio mode: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&Lighting[0][0].Correction=50&Lighting[0][0].Mode=ZoomPrio
Success Return	OK
Comment	Parameters in URL: paramName and paramValue are as table below. In table below, ***head*** = Lighting [***ChannelNo***] [***ConfigNo***] Some lighting equipment has the following restriction: Lighting[0][0].FarLight[0].Light + Lighting[0][0].NearLight[0].Light+ Lighting[0][0].MiddleLight [0]. Light <=100.

Appendix:

ParamName	ParamValue type	Description
head. Mode	string	Light mode. The range is {"Manual", "Auto", "Off", "ZoomPrio"}.

ParamName	ParamValue type	Description
head. Correction	integer	Light compensation. The range is [0—100], effective in ZoomPrio mode.
head. FarLight[Index]. Light	integer	Range is [0—100]. The luminance of far light.
head. MiddleLight [Index]. Light	integer	Range is [0—100]. The luminance of middle light.
head. NearLight [Index]. Light	integer	Range is [0—100]. The luminance of near light.

5.8 Video in Options

5.8.1 Video in Options Config

It's not recommended to use the CGI command from " video in options " ; It's now recommended to use the commands in 5.1 – 5.7.

- Get video in options

Table 5-23

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoInOptions
Method	GET
Description	Get video in options config, such as Backlight, ExposureSpeed, DayNightColor, DayOptions, NightOptions, and NormalOptions.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoInOptions
Success Return	***head***. Backlight=0 ***head***. DayNightColor=false ***head***. ExposureSpeed=0 ***head***. ExposureValue1=0.100000 ***head***. ExposureValue2=80.000000 ***head***. ExternalSync=0 ***head***. ExternalSyncPhase=0 ***head***. FlashControl.Mode=0 ***head***. FlashControl.Pole=0 ***head***. FlashControl.Value=0 ***head***. FlashControl.PreValue=0 ***head***. Flip=false ***head***. Gain=50 ***head***. GainAuto=true ***head***. IrisAuto=false ***head***. Mirror=false ***head***. NightOptions.AntiFlicker=0 ***head***. NightOptions.Backlight=0 ***head***. NightOptions.BacklightRegion[0]=3096 ***head***. NightOptions.BacklightRegion[1]=3096 ***head***. NightOptions.BacklightRegion[2]=5096

	head. NightOptions.BacklightRegion[3]=5096 ***head***. NightOptions.BrightnessThreshold=50 ***head***. NightOptions.DayNightColor=2 ***head***. NightOptions.ExposureMode=0 ***head***. NightOptions.ExposureSpeed=0 ***head***. NightOptions.ExposureValue1=0 ***head***. NightOptions.ExposureValue2=40 ***head***. NightOptions.ExternalSyncPhase=125 ***head***. NightOptions.Flip=false ***head***. NightOptions.Gain=50 ***head***. NightOptions.GainAuto=true ***head***. NightOptions.GainBlue=50 ***head***. NightOptions.GainGreen=50 ***head***. NightOptions.GainMax=50 ***head***. NightOptions.GainMin=0 ***head***. NightOptions.GainRed=50 ***head***. NightOptions.GlareInhibition=0 ***head***. NightOptions.IrisAuto=true ***head***. NightOptions.Mirror=false ***head***. NightOptions.Profile=3 ***head***. NightOptions.ReferenceLevel=50 ***head***. NightOptions.Rotate90=0 ***head***. NightOptions.SunriseHour=0 ***head***. NightOptions.SunriseMinute=0 ***head***. NightOptions.SunriseSecond=0 ***head***. NightOptions.SunsetHour=23 ***head***. NightOptions.SunsetMinute=59 ***head***. NightOptions.SunsetSecond=59 ***head***. NightOptions.SwitchMode=4 ***head***. NightOptions.WhiteBalance=Auto ***head***. NightOptions.WideDynamicRange=0 ***head***. NightOptions.WideDynamicRangeMode=0 ***head***. NormalOptions.AntiFlicker=0 ***head***. NormalOptions.Backlight=0 ***head***. NormalOptions.BacklightRegion[0]=3096 ***head***. NormalOptions.BacklightRegion[1]=3096 ***head***. NormalOptions.BacklightRegion[2]=5096 ***head***. NormalOptions.BacklightRegion[3]=5096 ***head***. NormalOptions.BrightnessThreshold=50 ***head***. NormalOptions.DayNightColor=1 ***head***. NormalOptions.ExposureMode=0 ***head***. NormalOptions.ExposureSpeed=0 ***head***. NormalOptions.ExposureValue1=0 ***head***. NormalOptions.ExposureValue2=40 ***head***. NormalOptions.ExternalSyncPhase=125 ***head***. NormalOptions.Flip=false ***head***. NormalOptions.Gain=50

	head. NormalOptions.GainAuto=true ***head***. NormalOptions.GainBlue=50 ***head***. NormalOptions.GainGreen=50 ***head***. NormalOptions.GainMax=50 ***head***. NormalOptions.GainMin=0 ***head***. NormalOptions.GainRed=50 ***head***. NormalOptions.GlareInhibition=0 ***head***. NormalOptions.IrisAuto=true ***head***. NormalOptions.Mirror=false ***head***. NormalOptions.Profile=0 ***head***. NormalOptions.ReferenceLevel=50 ***head***. NormalOptions.Rotate90=0 ***head***. NormalOptions.SunriseHour=0 ***head***. NormalOptions.SunriseMinute=0 ***head***. NormalOptions.SunriseSecond=0 ***head***. NormalOptions.SunsetHour=23 ***head***. NormalOptions.SunsetMinute=59 ***head***. NormalOptions.SunsetSecond=59 ***head***. NormalOptions.SwitchMode=0 ***head***. ReferenceLevel=50 ***head***. ReferenceLevelEnable=false ***head***. Rotate90=0 ***head***. SignalFormat=BT656 ***head***. WhiteBalance=Disable
Comment	Parameters in Response: ***head*** = table.VideoInOptions[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

- Set video in options

Table 5-24

Syntax	http://<server>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set video in options config, such as Backlight, ExposureSpeed, DayNightColor, DayOptions, NightOptions, and NormalOptions.
Example	Set Auto Exposure: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].ExposureMode=0&VideoInOptions[0].ExposureSpeed=0 Set Low Noise: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].ExposureMode=1&VideoInOptions[0].ExposureSpeed=0&VideoInOptions[0].GainMin=0&VideoInOptions[0].GainMax=60 Set Low Motion Blur: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].E

	xposureMode=2&VideoInOptions[0].ExposureSpeed=0&VideoInOptions[0].GainMin=0&VideoInOptions[0].GainMax=50&VideoInOptions[0].ExposureValue1=0&VideoInOptions[0].ExposureValue2=20 Set Manual: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].ExposureMode=4&VideoInOptions[0].ExposureSpeed=32&VideoInOptions[0].GainMin=0&VideoInOptions[0].GainMax=50&VideoInOptions[0].ExposureValue1=40&VideoInOptions[0].ExposureValue2=40 Set SmartIRExposure: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].SmartIRExposure=true Set Video Rotate: Filp: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].Flip=true Mirror: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].Mirror=true Or turn 90°: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].Rotate90=1 Set White Balance: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].WhiteBalance=Night Or http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].WhiteBalance=Custom&VideoInOptions[0].GainRed=50&VideoInOptions[0].GainBlue=50&VideoInOptions[0].GainGreen=50 (Sometimes you should set mode first before set GainRed or GainBlue: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoInOptions[0].WhiteBalance=Custom)
Success Return	OK
Comment	Parameters in URL: In table below, ***head*** =VideoInOptions[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

Appendix:

ParamName	ParamValue type	Description

ParamName	ParamValue type	Description
head. Backlight	integer	Range is [0—n] n depends on capability in **GetVideoInputCaps** 0: backlight closed. 1: backlight grade 1 ... n - backlight grade n
head. DayNightColor	integer	Range is {0,1,2} 0: always multicolor 1: autoswitch along with brightness, 2: always monochrome
head. ExposureMode	integer	Range is {0,1,2, 4} 0: AutoExposure 1: Gain first 2: Exposure first 4: Manual.
head. ExposureSpeed	integer	Range is [0 — n+1] n depends on capability in **GetVideoInputCaps** 0: AutoExposure 1-n-1: manual Exposure grade n: AutoExposure with time limit. n+1: manualExposure with user-defined time (n is supported maximum exposure grade)
head. ExposureValue1	float	Range is [0.1—80], unit is millisecond If ExposureSpeed is 0(AutoExposure enable), it's lower limit of AutoExposure time, otherwise it's time of manualExposure
head. ExposureValue2	float	Range is [0.1-80], unit is millisecond Upper limit of AutoExposure time, should be bigger than ExposureValue1
head. ExternalSync	integer	Range is {0,1} External Synchronous 0: Internal Synchronization 1: External Synchronous
head. ExternalSyncPhase	integer	Range is [0°—360°] External Synchronous Signal Phase
head. SmartIRExposure	bool	true: enable, false: disable
head. FlashControl.Mode	integer	Range is {0,1,2} 0: forbid flash 1: always flash 2: auto flash

ParamName	ParamValue type	Description
head. FlashControl.Pole	integer	Range is {0,1, 2, 3} Trigger mode: 0: low level 1: high level 2: rising-edge 3: falling-edge
head. FlashControl.Value	integer	Range is [0—15] Flashlight time-unit: 0: 0us, 1: 64us, 2: 128us, 3: 192us … 15 - 960us
head. FlashControl.PreValue	integer	Range is [0—100] It is threshold of brightness value: if brightness is less than this value, flash light will begin to work.
head. Flip	bool	true: enable video flip function false: disable video flip function
head. Gain	integer	Range is [0—100] If GainAuto is true, it's upper limit of auto gain, else it's the fixed gain adjust value.
head. GainBlue	integer	Range is [0—100] Gain for blue value, Value is effective when WhiteBalance is "Custom."
head. GainRed	integer	Range is [0—100] Gain for red value, Value is effective when WhiteBalance is "Custom."
head. GainGreen	integer	Range is [0—100] Gain for green value, Value is effective when WhiteBalance is "Custom."
head. GainAuto	bool	true: GainAuto false: No GainAuto
head. IrisAuto	bool	true: IrisAuto false: No IrisAuto
head. Mirror	bool	true: enable video mirror function false: disable video mirror function

ParamName	ParamValue type	Description
head. WhiteBalance	String	White balance Mode. Range is {Disable, Auto, Custom, Sunny, Cloudy, Home, Office, Night} Some IPC supports common modes: "Disable", "Auto", "Sunny", "Night", "Outdoor", "Custom" Sometimes the device support other advanced modes: "CustomColorTemperature", "Indoor", "ATW", "Manual", "AutoOutdoor", "ManualDatum" and so on.
head. ReferenceLevel	integer	Range is [0—100] The expected average brightness level of video frames.
head. Rotate90	integer	Range is {0,1,2} Video rotation: 0: No rotate 1: clockwise rotate 90° 2: anticlockwise rotate 90°
head. SignalFormat	String	Range is {Inside, BT656, 720p, 1080p, 1080i, 1080sF} Input Signal Mode
head. AntiFlicker	integer	Range is {0,1,2} AntiFlicker mode: 0: Outdoor 1: 50 Hz AntiFlicker 2: 60 Hz AntiFlicker
head. GlareInhibition	integer	Range is [0—100] GlareInhibition: 0: Close GlareInhibition.
head. NightOptions.BrightnessThreshold	integer	NightOptions contain a set of parameters used when brightness is not enough. Range is [0—100] when brightness is less than the BrightnessThreshold, parameters change to Nightoptions.
head. NightOptions.IrisAuto	bool	true: IrisAuto false: No IrisAuto
head. NightOptions.SunriseHour	integer	Range is [00—23] Sunrise hour.
head. NightOptions.SunriseMinute	integer	Range is [00—59] Sunrise minute
head. NightOptions.SunriseSecond	integer	Range is [00—59] Sunrise second
head. NightOptions.SunsetHour	integer	Sunset time. Its range is same with sunrise time, and it should be after sunrise time.

ParamName	ParamValue type	Description
head. NightOptions.SunsetMinute	integer	NightOptions are used if time is after sunset time and before sunrise time.
head. NightOptions.SunsetSecond	integer	
head. NightOptions.SwitchMode	integer	Range is {0,1,2} 0: NoSwitch, always use day options. 1: Switch depends on brightness. 2: Switch depends on time, switch to NightOptions when time is after sunset time and before sunrise. 3: NoSwitch, always use NightOptions. 4: No switch, always use NormalOptions.
head. NightOptions.Profile	integer	Range is {0,1,2,3} 0: use temporary day options. 1: use temporary NightOptions. 2: use temporary NormalOptions. 3: depends on *head*.NightOptions.SwitchMode
head. NightOptions.ExposureSpeed	integer	Range is the same as relevant items of day options in this table. Example: Value range of *head*.NightOptions.ExposureSpeed is the same with *head*. ExposureSpeed.
head. NightOptions.ExposureValue1	float	
head. NightOptions.ExposureValue2	float	
head. NightOptions.Gain	integer	
head. NightOptions.GainAuto	bool	
head. NightOptions.GainBlue	integer	
head. NightOptions.GainGreen	integer	
head. NightOptions.GainRed	integer	
head. NightOptions.WhiteBalance	String	
head. NightOptions.ReferenceLevel	integer	
head. NightOptions.ExternalSyncPhase	integer	
head. NightOptions.AntiFlicker	integer	
head. NightOptions.Backlight	integer	
head. NightOptions.DayNightColor	integer	

ParamName	ParamValue type	Description
head. NightOptions.ExposureMode	integer	
head. NightOptions.GlareInhibition	integer	
head. NightOptions.Mirror	integer	
head. NightOptions.Flip	integer	
head. NightOptions.Rotate90	integer	
head. NomalOptions.BrightnessThreshold	integer	NomalOptions contain a set of parameters similar with NightOptions. Range is the same as relevant items of NightOptions in this table.
head. NormalOptions.IrisAuto	bool	
head. NormalOptions.SunriseHour	integer	
head. NormalOptions.SunriseMinute	integer	
head. NormalOptions.SunriseSecond	integer	
head. NormalOptions.SunsetHour	integer	
head. NormalOptions.SunsetMinute	integer	
head. NormalOptions.SunsetSecond	integer	
head. NormalOptions.ExposureSpeed	integer	
head. NormalOptions.ExposureValue1	float	
head. NormalOptions.ExposureValue2	float	
head. NormalOptions.Gain	integer	
head. NormalOptions.GainAuto	bool	
head. NormalOptions.GainBlue	integer	
head. NormalOptions.GainGreen	integer	
head. NormalOptions.GainRed	integer	

ParamName	ParamValue type	Description
head. NormalOptions.WhiteBalance	String	
head. NormalOptions.ReferenceLevel	integer	
head. NormalOptions.ExternalSyncPhase	integer	
head. NormalOptions.AntiFlicker	integer	
head. NormalOptions.Backlight	integer	
head. NormalOptions.DayNightColor	integer	
head. NormalOptions.ExposureMode	integer	
head. NormalOptions.GlareInhibition	integer	
head. NormalOptions.Mirror	integer	
head. NormalOptions.Flip	integer	
head. NormalOptions.Rotate90	integer	

6 Storage APIs

6.1 Storage Devices

6.1.1 Get Hard Disk Information

Table 6-1

Syntax	http://<*server*>/cgi-bin/storageDevice.cgi?action=factory.getPortInfo
Method	GET
Description	Get the storage device port info.
Example	http://192.168.1.108/cgi-bin/storageDevice.cgi?action=factory.getPortInfo
Success Return	info.Total=2 info.Plug=1 info.Mask=1 info.Bad=0 info.IDE=1 info.Esata=4
Comment	—

6.1.2 Get All the Names of Storage Devices

Table 6-2

Syntax	http://<*server*>/cgi-bin/storageDevice.cgi?action=factory.getCollect
Method	GET
Description	Get all the names of storage devices.
Example	http://192.168.1.108/cgi-bin/storageDevice.cgi?action=factory.getCollect
Success Return	list[0]="/dev/sda0" list[1]="/dev/sda1" list[2]="/dev/sg1"
Comment	—

6.1.3 Get Storage Device Information

Table 6-3

Syntax	http://<*server*>/cgi-bin/storageDevice.cgi?action=getDeviceAllInfo
Method	GET
Description	Get all the storage device information.
Example	http://192.168.1.108/cgi-bin/storageDevice.cgi?action=getDeviceAllInfo

Success Return	list[0].Detail[0].IsError=false list[0].Detail[0].Pointer=27023434 list[0].Detail[0].TotalBytes=0 list[0].Detail[0].Type=ReadWrite list[0].Detail[0].UsedBytes=0 list[0].Name="/dev/sda" list[0].State=Success
Comment	—

6.1.4 Get Storage Capability

Table 6-4

Syntax	http://<*server*>/cgi-bin/storage.cgi?action=getCaps
Method	GET
Description	Get storage capabilities.
Example	http://192.168.1.108/cgi-bin/storage.cgi?action=getCaps
Success Return	caps.RedundantDisk.Support=false caps.SupportRemoteLimit=true
Comment	—

6.1.5 Format Camera SD-Card

Table 6-5

Syntax	http://<*server*>/cgi-bin/storageDevice.cgi?action=setDevice&type=FormatPatition&path=<***path***>
Method	GET
Description	Format camera SD card.
Example	http://192.168.1.108/cgi-bin/storageDevice.cgi?action=setDevice&type=FormatPatition&path= /dev/sda
Success Return	OK
Comment	If set successfully, return true, else return false. **path** : The **stringValue** is got from cgi API "*Get storage device information*" (/cgi-bin/storageDevice.cgi?action=getDeviceAllInfo). **stringValue is list[0].Name.**

6.2 NAS

6.2.1 NAS Information

- Get NAS config

Table 6-6

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=NAS
Method	GET
Description	Get all the directories on the NAS server.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=NAS

Success Return	table.NAS[0].Name=" FTP1" table.NAS[0].Enable = true table.NAS[0].Protocol ="FTP" table.NAS[0].Address ="www.ttt.com" table.NAS[0].Port =21 table.NAS[0].UserName ="anonymity" table.NAS[0].Password ="none" table.NAS[0].Directory ="share"
Comment	—

- Set NAS config

Table 6-7

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set NAS config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&NAS[0].Name=nas01&NAS[0].Enable=true
Success Return	OK
Comment	Parameters in URL: In table below, ***Head*** =NAS[***index***] ***index****:* The index of the NAS Server

Appendix:

ParamName	ParamValue type	Description
Head. Name	string	NAS name.
Head. Enable	bool	Enable/Disable the NAS.
Head. Protocol	string	The range is {"FTP", "SMB"}
Head. Address	string	The IP address or host name.
Head. Port	integer	NAS port.
Head. UserName	string	NAS username.
Head. Password	string	NAS password.
Head. Directory	string	Directory name.

6.3 Storage Point

6.3.1 Record Storage Point

- Get record storage point config

Table 6-8

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=RecordStoragePoint
Method	GET
Description	Get record storage point config.

Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=RecordStoragePoint
Success Return	table.RecordStoragePoint [0].TimingRecord.Local ="local" table.RecordStoragePoint [0].TimingRecord. Redundant =" Redundant" table.RecordStoragePoint [0].TimingRecord. Remote =" FTP" table.RecordStoragePoint [0].TimingRecord. AutoSync = false table.RecordStoragePoint [0].TimingRecord. AutoSyncRange =0 table.RecordStoragePoint [0].TimingRecord. LocalForEmergency =false table.RecordStoragePoint [0].TimingRecord. CompressBefore =15
Comment	—

- Set record storage point config

Table 6-9

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set record storage point config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&RecordStoragePoint[0].TimingRecord.Local=local
Success Return	OK
Comment	Parameters in URL: In table below, ***ch***: integer, array index starts from 0, which means video channel(equals to video channel index -1, and so 0 means channel 1). ***recType***: The range is {"TimingRecord", "VideoDetectRecord", "AlarmRecord", "EventRecord", "TimingSnapShot", "VideoDetectSnapShot", "AlarmSnapShot", "EventSnapShot"}

Appendix:

ParamName	ParamValue type	Description
RecordStoragePoint [***ch***].[***recType***].Local	string	Local directory name.
RecordStoragePoint [***ch***].[***recType***]. Redundant	string	Redundant directory name.
RecordStoragePoint [***ch***].[***recType***]. Remote	string	Remote directory name.
RecordStoragePoint [***ch***].[***recType***]. AutoSync	bool	When remote directory recovers, auto synchronize local directory to remote directory or not.
RecordStoragePoint [***ch***].[***recType***]. AutoSyncRange	integer	From the remote directory recovering time, how long the data needs to be synchronized. The unit is hour. If it is 0, all the data needs to be synchronized.

ParamName	ParamValue type	Description
RecordStoragePoint[*ch*].[*recType*].LocalForEmergency	bool	When the remote directory is unusable, save the data the local directory or not.
RecordStoragePoint[*ch*].[*recType*].CompressBefore	integer	The days' data which will be compressed.

6.3.2 Storage Group

- Get storage group config

Table 6-10

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=StorageGroup
Method	GET
Description	Get storage group config.
Example	http://192.168.1.168/cgi-bin/configManager.cgi?action=getConfig&name=StorageGroup
Success Return	table.StorageGroup[0].Channels[0].MaxPictures=0 table.StorageGroup[0].FileHoldTime=0 table.StorageGroup[0].Memo=For Reading & Writing Files table.StorageGroup[0].Name=ReadWrite table.StorageGroup[0].OverWrite=true table.StorageGroup[0].PicturePathRule=%y-%M-%d/%c/jpg/%h/%m/%s[%E][%O@%S][%R].jpg table.StorageGroup[0].RecordPathRule=%y-%M-%d/%c/dav/%h/%h.%m.%s-%h.%m.%s[%E][%O@%S][%R].dav table.StorageGroup[1].Channels[0].MaxPictures=0 table.StorageGroup[1].FileHoldTime=0 table.StorageGroup[1].Memo=For FTP Files table.StorageGroup[1].Name=Remote table.StorageGroup[1].OverWrite=true table.StorageGroup[1].PicturePathRule=%y-%M-%d/%c/jpg/%h/%m/%s[%E][%O@%S][%R].jpg table.StorageGroup[1].RecordPathRule=%y-%M-%d/%c/dav/%h/%h.%m.%s-%h.%m.%s[%E][%O@%S][%R].da
Comment	—

- Set storage group config

Table 6-11

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set storage group config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&StorageGroup[*0*].Name=main
Success Return	OK

Comment	Parameters in URL: In table below, ***Index*** = Storage Group index ***ch***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

Appendix:

ParamName	ParamValue type	Description
StorageGroup[***Index***]. Name	string	Storage group name.
StorageGroup[***Index***]. Memo	string	Storage group memo.
StorageGroup[***Index***]. FileHoldTime	integer	How many days the file will hold.
StorageGroup[***Index***]. OverWrite	bool	Over write or not when there is not enough storage.
StorageGroup[***Index***]. Channels[***ch***]. MaxPictures	Integer	The max pictures beyond which the old pictures will be over written. If it is 0, the old pictures will be not over written.
StorageGroup[***Index***]. Channels[***ch***]. Path	string	The channel path.

6.4 SDEncrypt

6.4.1 Encrypt SD Card

Table 6-12

Syntax	http://<*server*>/cgi-bin/SDEncrypt.cgi?action=encrypt&deviceName=<***deviceName***>&password=<***password***>
Method	GET
Description	SD encryption operation.
Example	http://192.168.1.108/cgi-bin/SDEncrypt.cgi?action=encrypt&deviceName=/dev/mmc0&password=123456
Success Return	OK
Comment	If set successfully, return true, else return false. deviceName: The stringValue is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo). password: The stringValue Card is Encrypted

6.4.2 Decrypt SD Card

Table 6-13

Syntax	http://<*server*>/cgi-bin/SDEncrypt.cgi?action=decrypt&deviceName=<***deviceName***>&password=<***password***>
Method	GET
Description	SD decrypt operation.

Example	http://192.168.1.108/cgi-bin/SDEncrypt.cgi?action=decrypt&deviceName=/dev/mmc0&password=123456
Success Return	OK
Comment	If set successfully, return true, else return false. ***deviceName***: The stringValue is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo). ***password***: The stringValue ard is Encrypted

6.4.3 Clear SD Card Password

Table 6-14

Syntax	http://<*server*>/cgi-bin/SDEncrypt.cgi?action=clearPassword&deviceName=<**deviceName**>&password=<**password**>
Method	GET
Description	SD clear password operation.
Example	http://192.168.1.108/cgi-bin/SDEncrypt.cgi?action=clearPassword&deviceName=/dev/mmc0&password=123456
Success Return	OK
Comment	If set successfully, return true, else return false. ***deviceName***: The stringValue is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo). password: The stringValue ard is Encrypted

6.4.4 Modify SD Card Password

Table 6-15

Syntax	http://<*server*>/cgi-bin/SDEncrypt.cgi?action=modifyPassword&deviceName=<**deviceName**>&password=<**password**>&oldPassword=<**oldPassword**>
Method	GET
Description	SD change password operation.
Example	http://192.168.1.108/cgi-bin/SDEncrypt.cgi?action=clearPassword&deviceName=/dev/mmc0&password=123456&oldPassword=admin123
Success Return	OK
Comment	If set successfully, return true, else return false. deviceName: The stringValue is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo). password: The new stringValue Card is Encrypted oldPassword: The old stringValue Card is Encrypted

6.4.5 Get SD Card Operate Error Policy

Table 6-16

Syntax	http://<*server*>/cgi-bin/SDEncrypt.cgi?action=getOperateErrorPolicy&deviceName=<**de**

	viceName>&operate=<**operate**>
Method	GET
Description	When decrypt, clearPassword, modifyPassword failed, get the error info, leftTimes and left time.
Example	http://192.168.1.108/cgi-bin/SDEncrypt.cgi?action=getOperateErrorPolicy&deviceName=/dev/mmc0&operate=decrypt
Success Return	policy.leftTimes=5 policy.lockSeconds=30
Comment	If set successfully, return true, else return false. deviceName: the stringValue is got from cgi API Get storage device information (cgi-bin/storageDevice.cgi?action=getDeviceAllInfo). Operate: operate type, can be: decrypt, modifyPassword and clearPassword leftTimes : remain operate times, max is 5 lockSeconds : lock operate time, unit is seconds, max is 30

6.4.6 Storage Health Alarm Settings

- Get StorageHealthAlarm config

Table 6-17

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=StorageHealthAlarm
Method	GET
Description	Get SD health info alarm config.
Example	http://<server>/cgi-bin/configManager.cgi?action=getConfig&name=StorageHealthAlarm
Success Return	table.StorageHealthAlarm.Enable=true table.StorageHealthAlarm.LowerLimit=10 table.StorageHealthAlarm.EventHandler= (output of EventHandler is described in GetEventHandler)
Comment	—

- Set StorageHealthAlarm config

Table 6-18

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set SD health info alarm config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&StorageHealthAlarm.Enable=true
Success Return	OK
Comment	—

7.1 GUI

7.1.1 GUISet

- Get GUISet config

Table 7-1

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=GUISet
Method	GET
Description	Get the GUI settings. Every video out screen has a group setting.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=GUISet
Success Return	table.GUISet[***index***]. WindowAlpha =128 table.GUISet[***index***]. TimeTitleEnable =true table.GUISet[***index***]. TimeTitlePos[0]=0 table.GUISet[***index***]. TimeTitlePos[1]=0 table.GUISet[***index***]. TimeTitlePos[2]=8191 table.GUISet[***index***]. TimeTitlePos[3]=8191 table.GUISet[***index***]. MenuShowOption =0 table.GUISet[***index***]. MenuAutoHideTime =10 table.GUISet[***index***]. AutoLogout =10 table.GUISet[***index***]. ChannelTitleShowEnable =true table.GUISet[***index***]. ChannelTitlePos[0]=0 table.GUISet[***index***]. ChannelTitlePos[1]=0 table.GUISet[***index***]. ChannelTitlePos[2]=8191 table.GUISet[***index***]. ChannelTitlePos[3]=8191 table.GUISet[***index***]. AutoGuideEnable =true …
Comment	Parameters in Response : ***index***: the array index which starts from 0.

- Set GUISet config

Table 7-2

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set the GUI settings. Every video out screen has a group setting.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&GUISet[0].WindowAlpha=192&GUISet[0].TimeTitleEnable=false&GUISet[0].MenuShowOption=1
Success Return	OK

Comment	Parameters in URL: The paramName and paramValue are in the table below. in table below, ***index*** : the array index which starts from 0.

Appendix:

ParamName	ParamValue type	Description
GUISet[***index***].WindowAlpha	integer	Diaphaneity of the window background.
GUISet[***index***].TimeTitleEnable	bool	Show the time title or not.
GUISet[***index***].TimeTitlePos[0]	integer	The position of the time title.
GUISet[***index***].TimeTitlePos[1]	integer	
GUISet[***index***].TimeTitlePos[2]	integer	
GUISet[***index***].TimeTitlePos[3]	integer	
GUISet[***index***].MenuShowOption	integer	0: Show the directory. 1: Hide the directory. 2: Timing-hide the directory.
GUISet[***index***].MenuAutoHideTime	integer	How many seconds to hide the directory.
GUISet[***index***].AutoLogout	integer	How many minutes to auto logout. The range is [0-120]. 0 expresses not logout.
GUISet[***index***].ChannelTitleShowEnable	bool	Show the channel title or not.
GUISet[***index***].ChannelTitlePos[0]	integer	The position of the channel title.
GUISet[***index***].ChannelTitlePos[1]	integer	
GUISet[***index***].ChannelTitlePos[2]	integer	
GUISet[***index***].ChannelTitlePos[3]	integer	
GUISet[***index***].AutoGuideEnable	bool	Auto guide or not when startup.

7.2 Split Screen

7.2.1 Split Screen Mode

- Get split screen mode

Table 7-3

Syntax	http://<*server*>/cgi-bin/split.cgi?action=getMode&channel=<***ChannelNo***>
Method	GET
Description	Get the split screen mode.
Example	http://192.168.1.108/cgi-bin/split.cgi?action=getMode&channel=1
Success Return	mode=split1 group=4
Comment	Parameters in URL: ***ChannelNo***: the display screen No. Start from 1 and <= 2.

- Set split screen mode

Table 7-4

Syntax	http://<*server*>/cgi-bin/split.cgi?action=setMode&channel=<***ChannelNo***>&mode=<***mode***>&group=<***group***>
Method	GET
Description	Set the split screen mode.
Example	http://192.168.1.108/cgi-bin/split.cgi?action=setMode&channel=1&mode=split4&group=*1*
Success Return	OK
Comment	Parameters in URL: ***ChannelNo***: the display screen No. Start from 1. ***mode***:enum{split1,split2,split4,split6,split8,split9,split12,split16,split20,split25,split36,split64,split144,pip1,pip3, "Free", "CompositeSplit1" / "FitDisplayUnit1", "CompositeSplit1" / "FitDisplayUnit4"}; ***group***: the No. of a group which contains certain number channels. For example, if 16 video channels display in split4 Mode which contains 4 video channels on Screen, then there are 4 groups and each group contains 4 video channels.

7.3 Moniter Tour

7.3.1 Moniter Tour

- Get moniter tour config

Table 7-5

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=MonitorTour
Method	GET
Description	Get monitor tour config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=MonitorTour
Success Return	table.MonitorTour[ch].Enable=128 table.MonitorTour[ch].Interval=true table.MonitorTour[ch].Mask.Split1=0,1,5 table.MonitorTour[ch].Mask.Split8=0,1,5 table.MonitorTour[ch].Collections=Favortite1, Favortite2…
Comment	—

- Set moniter tour config

Table 7-6

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set monitor tour config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&MonitorTour[0].Enable=true
Success Return	OK.
Comment	Parameters in URL:

	The paramName and paramValue are in the table below.

Appendix:

ParamName	ParamValue type	Description
MonitorTour[ch].Enable	bool	MonitorTour or not.
MonitorTour[ch].Interval	integer	MonitorTour interval.
MonitorTour[ch].Mask.Split1	—	Channel array for split1
MonitorTour[ch].Mask.Split8	—	Channel array for split8
MonitorTour[ch].Collections	—	Split collections

7.3.2 Enable Tour

Table 7-7

Syntax	http://<*server*>/cgi-bin/split.cgi?action=enableTour&channel=<***ChannelNo***>&enable=<***flag***>
Method	GET
Description	Enable tour in every video channel on a screen or not.
Example	http://192.168.1.108/cgi-bin/split.cgi?action=enableTour&channel=1&enable=*true*
Success Return	OK
Comment	***ChannelNo***: the display screen No. Start from 1 and <= 2. ***flag*** : true or false

7.3.3 Monitor Collection

- Get monitor collection config

Table 7-8

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=MonitorCollection
Method	GET
Description	Get monitor collection config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=MonitorCollection
Success Return	table.MonitorCollection.collectionname. Mode=Split1 table.MonitorCollection.collectionname.Windows[***winno***].Enable= true table.MonitorCollection.collectionname.Windows[***winno***].Device=device1 table.MonitorCollection.collectionname.Windows[***winno***].VideoChannel=5 table.MonitorCollection.collectionname.Windows[***winno***].VideoStream=Main table.MonitorCollection.collectionname.Windows[***winno***].AudioChannel=5 table.MonitorCollection.collectionname.Windows[***winno***].AudioStream=Main …
Comment	Parameters in Response : ***winno*** : integer, the array index which equals to the window index in a screen and starts from 0.

- Set monitor collection config

Table 7-9

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set monitor collection config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&MonitorCollection.Favorite1.Mode=split4&MonitorCollection.Favorite1.Windows[1].Enable=true&MonitorCollection.Favorite1.Windows[1].VideoChannel=2
Success Return	OK
Comment	Parameters in URL: The paramName and paramValue are in the table below. In table below: ***Collect***= MonitorCollection.***collectionname***. ***collectionname***: can be any name. ***winno***: integer, the array index which equals to the window index in a screen and starts from 0.

Appendix:

ParamName	ParamValue type	Description
Collect. Mode	string	The range is the same as **SetSplitMode**.
Collect. Windows[***winno***]. Enable	bool	Enable the window or not.
Collect. Windows[***winno***]. Device	string	The device Id.
Collect. Windows[***winno***]. VideoChannel	integer	The video channel.
Collect. Windows[***winno***]. VideoStream	string	The range is {"Main", "Extra1", "Extra2", "Extra3", "Auto"}.
Collect. Windows[***winno***]. AudioChannel	integer	The audio channel.
Collect. Windows[***winno***]. AudioStream	string	The range is {"Main", "Extra1", "Extra2", "Extra3", "Auto"}.

8 Video Analyse APIs

8.1 Video Analyse

8.1.1 Get Video Analyse Capability

Table 8-1

Syntax	http://<*server*>/cgi-bin/devVideoAnalyse.cgi?action=getcaps&channel=<***ChannelNo***>
Method	GET
Description	Get video analyse capabilities.
Example	http://192.168.1.108/cgi-bin/devVideoAnalyse.cgi?action=getcaps&channel=1
Success Return	caps.CalibrateBoxs[0]=2 caps.CalibrateBoxs[1]=3 caps.ComplexSizeFilter=false caps.MaxCelibateAreas=10 caps.MaxExcludeRegions=0 caps.MaxInternalOptions=512 caps.MaxModules=1 caps.MaxPointOfLine=20 caps.MaxPointOfRegion=20 caps.MaxRules=10 caps.MaxStaffs=4 caps.SpecifiedObjectFilter=true caps.SupportedRules[0]=CrossLineDetection caps.SupportedRules[1]=CrossRegionDetection caps.SupportedRules[2]=LeftDetection caps.SupportedRules[3]=TakenAwayDetection caps.SupportedScene[0]=Normal caps.SupportedScene[1]=FaceDetection caps.SupportedScene[2]=VideoDiagnosis caps.SupportedScenes.FaceDetection.SupportedCalibrateParams.Groud.HorizontalStaffs[0]=0 caps.SupportedScenes.FaceDetection.SupportedCalibrateParams.Groud.HorizontalStaffs[1]=0 caps.SupportedScenes.FaceDetection.SupportedCalibrateParams.Groud.VerticalStaffs[0]=0 caps.SupportedScenes.FaceDetection.SupportedCalibrateParams.Groud.VerticalStaffs[1]=0 caps.SupportedScenes.StereoNumber.SupportedRules.ManNumDetection.SupportLocalDataStore=false caps.SupportedScenes.NumberStat.CameraType=1 caps.SupportedScenes.NumberStat.SupportedRules.NumberStat.MaxRules=8
Comment	Parameters in URL:

	ChannelNo: integer, video channel index which starts from 1.

8.1.2 Video Analyse Global

- Get video analyse global config

Table 8-2

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoAnalyseGlobal
Method	GET
Description	Get video analyse global config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoAnalyseGlobal
Success Return	***head***.Scene.Type=Normal ***head***.Scene.PtzPresetId=1 ***head***.Scene.Depth=Far ***head***.Scene.Detail.CameraAngle=30 ***head***.Scene.Detail.CameraDistance=10.000000 ***head***.Scene.Detail.CameraHeight=6.200000 ***head***.TimePeriod.Day[0]=8:00:00 ***head***.TimePeriod.Day[1]=20:00:00 ***head***.TimePeriod.Night[0]=20:00:00 ***head***.TimePeriod.Night[1]=8:00:00 …
Comment	Parameters in Response : ***head*** =table.VideoAnalyseGlobal[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel(equals to video channel index -1, and so 0 means channel 1).

- Set video analyse global config

Table 8-3

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>...]
Method	GET
Description	Set video analyse global config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoAnalyseGlobal[0].Scene.Type=Normal&VideoAnalyseGlobal[0].Scene.PtzPresetId=1
Success Return	OK
Comment	Parameters in URL: paramName and paramValue are as table below. In table below, head =***VideoAnalyseGlobal***[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ParamName start with head.Scene.Detail depends on head.Scene.Type.

Appendix

ParamName	ParamValue type	Description

ParamName	ParamValue type	Description
head.Scene.Type	string	Scene class, the range is { "Normal", "Indoor", "ATM", "Traffic", "FaceRecognition", "FaceDetection", "Prison", "NumberStat", "HeatMap", "VideoDiagnosis", "VehicleAnalyse", "TrafficPatrol", "CourseRecord", "Vehicle" }
head.Scene.PtzPresetId	integer	Range is 0—255, 0 means that the scene is unassociated with PTZ.
head.Scene.Depth	string	Picture distance feature, the range is { "Normal", "Far", "Middle", "Near" }
head.Scene.Detail.*Value*	string	Detail config of a scene. For example, when Scene.Type is "Normal", it's detail includes CameraAngle, CameraDistance, CameraHeight, etc.
head.TimePeriod.Day[0]	string	The start time of Day, it's format is hh:mm:ss
head.TimePeriod.Day[1]	string	The end time of Day
head.TimePeriod.Night[0]	string	The start time of Night, it's format is hh:mm:ss
head.TimePeriod.Night[1]	string	The end time of Night

8.1.3 Video Analyse Rule

- Get video analyse rule

Table 8-4

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoAnalyseRule
Method	GET
Description	Get video analyse rules config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoAnalyseRule
Success Return	***head***.Name= line1 ***head***.Type=CrossLineDetection ***head***.VideoAnalyseRule[0][0].Enable =true ***head***.VideoAnalyseRule[0][0].EventHandler= (output of EventHandler is described in GetEventHandler) …
Comment	Parameters in Response : ***head*** =table.VideoAnalyseRule[***ChannelNo***] [***RuleNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***RuleNo*** =rule index.

- Set video analyse rule

Table 8-5

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set video analyse rules config.

Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoAnalyseRule[0][0].Name=myAnalyseRule1&VideoAnalyseRule[0][0].Type=CrossLineDetection
Success Return	OK
Comment	Parameters in URL: paramName and paramValue are as table below. In table below, ***head*** =VideoAnalyseRule[***ChannelNo***] [***RuleNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***RuleNo*** =rule index. ParamName start with head.Config is only effective with {"CrossLineDetection", "CrossRegionDetection", "LeftDetection", "TakenAwayDetection"}.

Appendix

ParamName	ParamValue type	Description
head. Name	string	Rule name, it must be unique.
head. Type	string	The range is {"CrossLineDetection", "CrossRegionDetection", "LeftDetection", "TakenAwayDetection", "VideoAbnormalDetection", "FaceDetection", "AudioMutation", "AudioAnomaly", "VideoUnFocus", "WanderDetection", "RioterDetection", "ParkingDetection", "MoveDetection", "NumberStat"}.
head. Enable	bool	Enable/Disable this rule.
head. EventHandler	—	Setting of EventHandler is described in **SetEventHandler**.
head. Config.DetectLine[0][0]	integer	The start point of DetectLine 0;
head. Config.DetectLine[0][1]	integer	The end point of DetectLine 0;
head. Config.DetectLine[1][0]	integer	The start point of DetectLine 1;
head. Config.DetectLine[1][1]	integer	The end point of DetectLine 1;
head. Config.Direction	string	The range is {"LeftToRight", "RightToLeft", "Both"}.
head. Config .SizeFilter.MaxSize[0]	integer	Maximum width. The width of the object must not be beyond maximum width. Adapt to {"CrossLineDetection", "CrossRegionDetection", "LeftDetection", "TakenAwayDetection", "FaceDetection", "WanderDetection", "RioterDetection", "ParkingDetection", "MoveDetection"}.
head. Config .SizeFilter.MaxSize[1]	integer	Maximum height. The height of the object must not be beyond maximum height.
head. Config .SizeFilter.MinSize[0]	integer	Minimum width. The width of the object must not be less than minimum width.
head. Config .SizeFilter.MinSize[1]	integer	Minimum height. The height of the object must not be beyond minimum height.

ParamName	ParamValue type	Description
head. Config.DetectRegion[0][0]	integer	The start point of DetectRegion 0; Adapt to {"CrossRegionDetection", "LeftDetection", "TakenAwayDetection", "WanderDetection", "RioterDetection", "ParkingDetection", "MoveDetection"}.
head. Config.DetectRegion[0][1]	integer	The end point of DetectRegion 0;
head. Config.DetectRegion[1][0]	integer	The start point of DetectRegion 1;
head. Config.DetectRegion[1][1]	integer	The end point of DetectRegion 1;
head. Config.DetectRegion[2][0]	integer	The start point of DetectRegion 2;
head. Config.DetectRegion[2][1]	integer	The start point of DetectRegion 2;
head. Config. MinDuration	integer	Range is 1—600, adapt to {"LeftDetection", "TakenAwayDetection", "WanderDetection"}. Range is 10-300, adapt to {"RioterDetection"}. Range is 6-300, adapt to {"ParkingDetection"}.
head. Config. Sensitivity	integer	Range is 1—10, adapt to {"RioterDetection", "MoveDetection"}.
head. Config. EnterThreshold	integer	Range is 0 — 100000000, adapt to {"NumberStat"}.
head. Config. ExitThreshold	integer	Range is 0 — 100000000, adapt to {"NumberStat"}.
head. Config. InsideThreshold	integer	Range is 0 — 100000000, adapt to {"NumberStat"}.

8.2 Number of People

8.2.1 Video Widget Number Status

- Get video widget number status

Table 8-6

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=VideoWidgetNumberStat
Method	GET
Description	Get OSD config when display number status information of people.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=VideoWidgetNumberStat
Success Return	***head***.EncodeBlend=true ***head***.ShowEnterNum=true ***head***.ShowExitNum=true ***head***.TextAlign=0

	…
Comment	Parameters in Response : ***head*** =table.VideoWidgetNumberStat[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

- Set video widget number status

Table 8-7

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set OSD config when display number status information of people.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&VideoWidgetNumberStat[0].EncodeBlend=true&VideoWidgetNumberStat[0].ShowEnterNum=true
Success Return	OK
Comment	Parameters in URL: paramName and paramValue are as table below. In table below, ***head*** =VideoWidgetNumberStat[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel(equals to video channel index -1, and so 0 means channel 1).

Appendix

ParamName	ParamValue type	Description
head. EncodeBlend	bool	Enable/Disable
head. ShowEnterNum	bool	Enable/Disable
head. ShowExitNum	bool	Enable/Disable
head. TextAlign	integer	0 for left, 2 for right

8.2.2 Get Heat Map Information

Table 8-8

Syntax	http://<*server*>/cgi-bin/heatMap.cgi?action=getPicByTime&channel=<***ChannelNo***>&StartTime=<***start***>&EndTime=<***end***>
Method	GET
Description	Get binary data of heat map.
Example	http://192.168.1.108/cgi-bin/heatMap.cgi?action=getPicByTime&channel=1&StartTime=2015-08-20%2000:00:00&EndTime=2015-08-21%2023:59:59
Success Return	Content-Type: application/binarytet-stream Content-Length:<***heatMap size***> <***HeatMap data***>
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. ***start***/***end***: the start/end time of Heat Map info. 24 hour Format, as: yyyy-mm-dd hh:mm:ss.

	Parameters in Response: ***heatMap size***: width*height + 16. ***HeatMap data****:* format as table below.

Appendix: HeatMap Data Format

0	1	2	3	4	…	15	16	17	18	…
Width		Height		Reserved			Data: every byte symbolize a pixel			

8.2.3 Get People Heat Map Information

Table 8-9

URL	http://<*server*>/cgi-bin/videoStatServer.cgi?action=getHeatMap		
Method	GET		
Description	Get People Heat Map Information		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1.
PlanID	int	O	The plan id, only valid for dome camera.
StartTime	string	R	The start time, ex: "2010-05-12 20:00:00"
EndTime	string	R	The end time, ex: "2010-05-12 22:00:00"
HeatMapType	string	R	The heat map type, it can be: “AverageStayTime” : average stay time heat map “HumanStatistics” : human statistics heat map “HumanTrack” : human track heat map
[Response Params] (multipart binary data)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=getHeatMap&channel=1&PlanID=2&StartTime=2010-05-12%2020:00:00&EndTime=2010-05-12%2022:00:00&HeatMapType=AverageStayTime		
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<*boundary*> Connection: closed --<*boundary*> Content-Type: application/octet-stream Content-Length: <*length*> <*heat map binary data*> --<*boundary*> Content-Type: application/octet-stream Content-Length: <*length*> <*heat map binary data*> --<*boundary*>		

	…

Appendix A: The heat map binary data format when HeatMapType is AverageStayTime.

octet	0	1	2	3	4~7	8~11	12~15
value	version	channel	preset	reserved	total data length	totol lines	lines in this packet
octet	16~31						
value	reserved						
octet	32~35				36~39	40~43	44~44+4*（N-1）
value	line number				data num in this line	data 1	data N
octet	44+4*（N-1）+1~ 44+4*（N-1）+4				……	……	……
value	next line number				data num in next line	data 1	data N

Note: If “lines in this packet” is 0, then all data has been sent, and connection will be closed.

Appendix B: The heat map binary data format when HeatMapType is HumanStatistics.

octet	0	1	2	3	4~7	8~11	12~15
value	version	channel	preset	reserved	total data length	total lines	lines in this packet
octet	16~31						
value	reserved						
octet	32~35				36~39	40~43	44~44+4*（N-1）
value	line number				data num in this line	data 1	data N
octet	44+4*（N-1）+1~ 44+4*（N-1）+4				……	……	……
value	next line number				data num in next line	data 1	data N

Note: If “lines in this packet” is 0, then all data has been sent, and connection will be closed.

Appendix C: The heat map binary data format when HeatMapType is HumanTrack.

octet	0	1	2	3	4~7	8~11	12~15
value	version	channel	preset	reserved	total data length	total tracks	tracks in this packet
octet	16~31						
value	reserved						
octet	32~35				36~39	40~43	44~44+4*（N-1）
value	Object ID				track point num in this object	track point 1	track point N
octet	44+4*（N-1）+1~ 44+4*（N-1）+4				……	……	……

value	next object ID	track point num in next object	track point 1	track point N

Note: If “tracks in this packet” is 0, then all data has been sent, and connection will be closed.

8.2.4 Subscribe People Realtime Trace Information

Table 8-10

URL	http://<*server*>/cgi-bin/videoStatServer.cgi?action=attachRealTraceProc		
Method	GET		
Description	Subscribe the people realtime trace information.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1.
heartbeat	int	O	Send heartbeat interval, range is [1, 60],unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat meaage are "Heartbeat". If this parameter is not present, its default value is 60.
[Response Params] (multipart binary data)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=attachRealTraceProc&channel=1&heartbeat=5		
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<*boundary*> Connection: closed --<*boundary*> Content-Type: application/octet-stream Content-Length: <*length*> <*realtime trace binary data*> --<*boundary*> Content-Type: text/plain Content-Length: 11 Heartbeat -<*boundary*> Content-Type: application/octet-stream Content-Length: <*length*> <*realtime trace binary data*> --<*boundary*> …		

Appendix A: The realtime trace binary data format.

octet	0	1	2	3	4~7	8~11	12~15
value	version	chan	pres	reser	total data length	reserved	object num in this

		nel	et	ved			packet
octet	16~31						
value	reserved						
octet	32~35			36~37	38~39	40~43	44~47
value	object ID 1			object coord x，	object coord y	current time in utc	reserved
octet	48~51			52~53	54~55	56~57	58~61
value	object ID 2			object coord x	object coord y	current time in utc	reserved

8.2.5 Get People Histroy Trace Information

Table 8-11

URL	http://<*server*>/cgi-bin/videoStatServer.cgi?action=getHistoryTrace		
Method	GET		
Description	Get people history trace information.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1.
PlanID	int	O	The plan id, only valid for dome camera.
StartTime	string	R	The start time, ex: "2010-05-12 20:00:00"
EndTime	string	R	The end time, ex: "2010-05-12 22:00:00"
[Response Params] (multipart binary data)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=getHistoryTrace&channel=1&PlanID=2&StartTime=2010-05-12%2020:00:00&EndTime=2010-05-12%2022:00:00		
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<*boundary*> Connection: closed --<*boundary*> Content-Type: application/octet-stream Content-Length: <*length*> <*history trace binary data*> --<*boundary*> Content-Type: application/octet-stream Content-Length: <*length*> <*history trace binary data*> --<*boundary*> …		

Appendix A: The history trace binary data format.

octet	0	1	2	3	4~7	8~11	12~15
value	version	channel	preset	reserved	total data length	total object num	object num in this packet
octet	16~31						
value	reserved						
octet	32~35				36~37	38~39	40~43	44~47
value	object ID 1				object coord x	object coord y	current time in utc	reserved
octet	48~51				52~53	54~55	56~57	58~61
value	object ID 2				object coord x	object coord y	current time in utc	reserved

Note: If “object num in this packet” is 0, then all data has been sent, and connection will be closed.

8.2.6 Subscribe Heat Map Raw Data

Table 8-12

URL	http://<*server*>/cgi-bin/HeatMapManager.cgi?action=attachRaw		
Method	GET		
Description	Subscribe the heap map raw data.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1.
heartbeat	int	O	Send heartbeat interval, range is [1, 60],unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat meaage are "Heartbeat". If this parameter is not present, its default value is 60.
[Response Params] (multipart binary data)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/HeatMapManager.cgi?action=attachRaw&channel=1&heartbeat=5		
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<*boundary*> Connection: closed --<*boundary*> Content-Type: application/octet-stream Content-Length: <*length*> <*heap map raw data*> --<*boundary*> Content-Type: text/plain Content-Length: 11		

	Heartbeat -<*boundary*> Content-Type: application/octet-stream Content-Length: <*length*> <*heap map raw data*> --<*boundary*> …

Appendix A: The heap map raw data format.

octet	0	1	2	3	4 ~ 23
value	width		height		StartTime, ex: “2012-01-04 00:00:00”
octet	24~43				44 ~ 47
value	EndTime, ex: “2012-01-04 01:00:00”				reserved
octet	48	49	…		
value	Data: every byte symbolize a pixel				

8.2.7 Clear statistics in time section

Table 8-13

URL	http://<*server*>/cgi-bin/videoStatServer.cgi?action=clearSectionStat[&AreaID=<***AreaID***>][&PtzPresetId=<***PtzPresetId***>]		
Method	GET		
Description	Clear statistics in time section		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1.
AreaID	int	O	The area index which starts from 1, valid for multiple area device, if omit means clear all area statistics.
PtzPresetId	int	O	The preset index of ptz. value starts from 1.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=clearSectionStat&AreaID=2		
Response	OK		

8.3 FishEye

8.3.1 Get FishEye Capability

The method described in the Section 4.5.13(Get video input capability) should be used fisrst.In the response, it will contain these message "caps.FishEye=false", if the value of the FishEye is true, Then you can use the method described below to get the detail capability.

Table 8-14

Syntax	http://<*server*>/cgi-bin/devVideoInput.cgi?action=getCapsEx&channel=<***ChannelNo***>&name=VideoInFishEye
Method	GET
Description	Clear the people count information.
Example	http://192.168.1.108/cgi-bin/devVideoInput.cgi?action=getCapsEx&channel=1&name=VideoInFishEye
Success Return	caps.***Type***=Chip caps.***MountMode***[0]=WallMode caps.***MountMode***[1]=CeilMode …… caps.***CalibrateMode***[0]=Original caps.***CalibrateMode***[1]=Config caps.***CalibrateMode***[2]=Panorama …… caps.***EPtzCmd***[0]=Up caps.***EPtzCmd***[1]=Down …….
Comment	Parameters in Response: ***Type****:* string, it can be Chip, Plugin, and ChipAndPlugin. Chip means only support calibrate by device. Plugin means only support calibrate by plugin. ChipAndPlugin means support both. ***MountMode****:* string and array.and ***MountMode*** means the install mode, it can be WallMode,CeilMode, FloorMode,180CeilMode. ***CalibrateMode****:* string and array. It can be Original, Config, Panorama, DoublePanorama, OriginalPlusThreeEPtzRegion, Single, FourEPtzRegion, TwoEPtzRegion, and Normal. ***EPtzCmd****:* string and array. It can be ZoomIn, ZoomOut, Up, Down, Left, Right, RotateClock, RotateAntiClock, Stop, TapView, and ShowRegion.

8.3.2 FishEye Setting

- Get FishEye config

Table 8-15

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=FishEye
Method	GET
Description	Get FishEye config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=FishEye
Success Return	***head***.***PlaceHolder***=1 ***head***.***CalibrateMode***=Original ……..
Comment	Parameters in Response: ***head*** = table.FishEye[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***PlaceHolder***：integer， it can be 1(CeilMode), 2(WallMode), 3(FloorMode).

	***CalibrateMode**:* string. It can be Original, Config, Panorama, DoublePanorama, OriginalPlusThreeEPtzRegion, Single, FourEPtzRegion, TwoEPtzRegion, and Normal.

- Set FishEye config

Table 8-16

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set FishEye config.
Example	Change the placeholder setting of channel 1: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&FishEye[0].PlaceHolder=1 Change the calibratemode setting of channel 1: http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&FishEye[0].CalibrateMode=Panorama Also can use http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&FishEye[0].CalibrateMode=Panorama&FishEye[0].PlaceHolder=1 to change the placeholder and calibratemode in one message.
Success Return	OK
Comment	—

8.4 CrowdDistriMap

8.4.1 Get Channel Caps

Table 8-17

Syntax	http://<*server*>/cgi-bin/crowdDistriMap.cgi?action=getCaps
Method	GET
Description	Get Channel CrowdistriMap Caps.
Example	http://192.168.1.108/cgi-bin/crowdDistriMap.cgi?action=getCaps
Success Return	CrowdCapsList[0].channel=<***ChannelNo***> CrowdCapsList[0].Support=<***Support***> CrowdCapsList[1].channel=<***ChannelNo***> CrowdCapsList[1].Support=<***Support***> …
Comment	Parameters in URL and Response: ***ChannelNo*** : video channel index ***Support*** : true or false, support or not

8.4.2 Subscribe to Realtime Crowd Stat

Table 8-18

Syntax	http://<*server*>/cgi-bin/crowdDistriMap.cgi?action=attach&channel=<***ChannelNo***>[&*he*

	artbeat=<***Heartbeat***>]
Method	GET
Description	Subscribe the crowd distribuite map information, return info at regular time.
Example	http://<*server*>/cgi-bin/crowdDistriMap.cgi?action=attach&channel=1&heartbeat=5
Success Return	HTTP/1.1 200 OK Content-Type: multipart/x-mixed-replace; boundary=<*boundary*> Connection: close --<*boundary*> Content-Type: text/plain Content-Length: <data length> CrowdStatData[0].Channel=1 CrowdStatData[0].GloabalPeopleNum =10 CrowdStatData[0].RegionNum =1 CrowdStatData[0].RegionPeopleList[0].RegionID=0 CrowdStatData[0].RegionPeopleList[0].Region[0][0]=10 CrowdStatData[0].RegionPeopleList[0].Region[0][1]=10 CrowdStatData[0].RegionPeopleList[0].Region[1][0]=10 CrowdStatData[0].RegionPeopleList[0].Region[1][1]=100 … CrowdStatData[0].RegionPeopleList[0].RegionPeopleNum=100 CrowdStatData[0].CrowdEventNum =2 CrowdStatData[0].CrowdList[0].Center=2 CrowdStatData[0].CrowdList[0].Radius=2 CrowdStatData[0].RegionEventNum =2 CrowdStatData[0].RegionList[0].Region[0][0]=10 CrowdStatData[0].RegionList[0].Region[0][1]=10 CrowdStatData[0].RegionList[0].Region[1][0]=10 CrowdStatData[0].RegionList[0].Region[1][0]=100 … CrowdStatData[0].RegionList[0].RegionID=0 CrowdStatData[0].RegionList[0].PeopleNum=100 --<*boundary*> Content-Type: text/plain Content-Length: 11 Heartbeat --<*boundary*> Content-Type: text/plain Content-Length: <data length> CrowdStatData[0].Channel=1 …
Comment	Parameters in URL and Response ***ChannelNo***: integer, video channel index to subscribe

	Region is ***Polygon***, has a list of ***points***, every point has ***x*** and ***y*** , so Region[0][0] is first point's x, Region[0][1] is first point's y, Region[1][0] is second point's x, Region[1][1] is second point's y, and so on ***Heartbeat****:* integer, range is [1,60],unit is second.If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client,the heartbeat meaage are "Heartbeat".

8.4.3 Get Current Crowd Stat

Table 8-19

Syntax	http://<*server*>/cgi-bin/crowdDistriMap.cgi?action=getSummary&channel=<***ChannelNo***>
Method	GET
Description	Get crowd distribuite map information, return info only once.
Example	http://<*server*>/cgi-bin/crowdDistriMap.cgi?action=getSummary&channel=1
Success Return	CrowdStatData[0].Channel=1 CrowdStatData[0].GloabalPeopleNum =10 CrowdStatData[0].RegionNum =1 CrowdStatData[0].RegionPeopleList[0].RegionID=0 CrowdStatData[0].RegionPeopleList[0].Region[0][0]=10 CrowdStatData[0].RegionPeopleList[0].Region[0][1]=10 CrowdStatData[0].RegionPeopleList[0].Region[1][0]=10 CrowdStatData[0].RegionPeopleList[0].Region[1][1]=100 … CrowdStatData[0].RegionPeopleList[0].RegionPeopleNum=100 CrowdStatData[0].CrowdEventNum =2 CrowdStatData[0].CrowdList[0].Center=2 CrowdStatData[0].CrowdList[0].Radius=2 CrowdStatData[0].RegionEventNum =2 CrowdStatData[0].RegionList[0].Region[0][0]=10 CrowdStatData[0].RegionList[0].Region[0][1]=10 CrowdStatData[0].RegionList[0].Region[1][0]=10 CrowdStatData[0].RegionList[0].Region[1][0]=100 … CrowdStatData[0].RegionList[0].RegionID=0 CrowdStatData[0].RegionList[0].PeopleNum=100
Comment	Similar to above "Subscribe to Realtime Crowd Stat", but reply stat only once, not at regular time.

8.5 Intelligent

8.5.1 Get Intelligent Caps

URL	http://<*server*>/cgi-bin/intelli.cgi?action=getCaps
Method	GET

Description	Get video analyse intelligent capabilities.		
[Request Params] (key=value)			
capsName	string	O	Name of capability, it can be: AnalyseMode: get analyse mode capacity MultiChannelMode: get multi channel seperate intelligent capacity and combined intelligent capacity MultiVideoAnalyse: get all intelligent capacity from the multichannel device in each channel if omit, means get all capacity
[Response Params] (key=value)			
caps	object	R	The video service capabilities.
+AnalyseMode	string	O	Analyse mode, can be : RealStream, RecordFileStream, PicFileStream
+MultiChannelMode	object	O	Multi channel description
++IndepMode	Array<Array<object>>	O	Multi channel open at one time separately
+++Channel	int	R	Video channel number, start from 0
+++Type	string	R	Business class
++CompMode	Array<Array<object>>	O	Multi channel combined mode
+++Channel	int	R	Video channel number, start from 0
+++Type	string	R	Business class
+MultiVideoAnalyse	Array<object>	O	List of channel capabilities
++caps	object	R	Video analyse capabilities
+Algorithm	Array<object>	O	Algorithm version information
++Class	string	R	Intelligent analyse class, ex : ObjectDetect
++AlgorithmVersion	string	R	Algorithm version
++AlgorithmVendor	sting	R	Algorithm Vendor name
+TotalCapacity	Array<object>	O	Intelligent analyse total capacity
++Class	string	R	Intelligent analyse class, ex : ObjectDetect
++Type	Array<string>	R	Intelligent analyse rules, ex : FaceDetection
++Number	int	R	The maximum number of video channels that can be analysed at same time
[Example]			
Request	GET http://192.168.1.108/cgi-bin/intelli.cgi?action=getCaps		
Response	caps.AnalyseMode=RealStream caps.MultiChannelMode.IndepMode[0][0].Channel=0 caps.MultiChannelMode.IndepMode[0][0].Type=Normal caps.MultiChannelMode.IndepMode[0][1].Channel=1		

	caps.MultiChannelMode.IndepMode[0][1].Type=ObjectDetect … caps.MultiChannelMode.CompMode[0][0].Channel=0 caps.MultiChannelMode.CompMode[0][0].Type=Normal caps.MultiChannelMode.CompMode[0][1].Channel=1 caps.MultiChannelMode.CompMode[0][1].Type=ObjectDetect … caps.Algorithm[0].Class=Normal caps.Algorithm[0].AlgorithmVersion=V2.8 caps.Algorithm[0].AlgorithmVendor=Dahua caps.Algorithm[1].Class=ObjectDetect caps.Algorithm[1].AlgorithmVersion=V2.8 caps.Algorithm[1].AlgorithmVendor=Dahua … caps.TotalCapacity[0].Class=Normal caps.TotalCapacity[0].Type[0]=FaceDetection caps.TotalCapacity[0].Number=3 caps.TotalCapacity[1].Class=ObjectDetect caps.TotalCapacity[1].Type[0]=FaceDetection caps.TotalCapacity[1].Number=3 …

8.5.2 Subscribe Resource Usage Info

URL	http://<*server*>/cgi-bin/intelli.cgi?action=attachResource		
Method	GET		
Description	Subscribe the intelligent analyse resource		
[Request Params] (key=value format in URL)			
heartbeat	int	O	Send heartbeat interval, range is [1, 60],unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat meaage are "Heartbeat". If this parameter is not present, its default value is 60.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
RemainCapacity	Array<object>	R	Remain capacity of intelligent analyse
+Class	string	R	Intelligent analyse class, ex : ObjectDetect
+Number	int	O	Remain number of video channels that can be analysed
[Example]			
Request	GET http://192.168.1.108/cgi-bin/intelli.cgi?action=attachResource&heartbeat=5		
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<*boundary*> Connection: closed --<*boundary*>		

	Content-Type: text/plain Content-Length: <*length*> RemainCapacity[0].Class=Normal RemainCapacity[0].Number=1 RemainCapacity[1].Class=ObjectDetect RemainCapacity[1].Number=2 --<*boundary*> Content-Type: text/plain Content-Length: 11 Heartbeat --<*boundary*> Content-Type: text/plain Content-Length: <*length*> RemainCapacity[0].Class=Normal RemainCapacity[0].Number=1 RemainCapacity[1].Class=ObjectDetect RemainCapacity[1].Number=1 …

9 Intelligent Traffic APIs

9.1 Traffic Snap

9.1.1 Get the Specific Parking Space Status

Table 9-1

Syntax	http://<*server*>/cgi-bin/trafficSnap.cgi?action=getParkingSpaceStatus&channel=<***ChannelNo***>&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Get specific parking space(s) status.
Example	http://192.168.1.108/cgi-bin/trafficSnap.cgi?action=getParkingSpaceStatus&condition.Lane[0]=0&condition.Lane[1]=255
Success Return	A list of parking space status status[0].Lane=0 status[0].PictureId=5 status[0].**TrafficCar**.CountInGroup=1 … status[1].Lane=1 status[1].PictureId=4 status[1].**TrafficCar**.CountInGroup=1 …
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. ***paramName*** and ***paramValue***: detail in table below. In table below, ***index***: The index of type array, start from 0 Parameters in Response : ***TrafficCar***: the members refer to TrafficCar

Appendix

ParamName	ParamValue type	Description
condition. Lane[***index***]	int	The Lane value
condition. ResponseLevel	int	The Level value , refer to condition

9.1.2 Open Strobe

Table 9-2

Syntax	http://<*server*>/cgi-bin/trafficSnap.cgi?action=openStrobe&channel=<***ChannelNo***>&info.openType=<***OpenType***>&info.plateNumber=<***PlateNumber***>

Method	GET
Description	Open the strobe manually.
Example	http://192.168.1.108/cgi-bin/trafficSnap.cgi?action=openStrobe&channel=1&info.openType=Normal&info.plateNumber=046XRW
Success Return	OK
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. ***OpenType***: string, For now, the value is fixed to "Normal" ***PlateNumber***: string, the plateNumber of the TrafficCar.

9.1.3 Open/Close Unlicensed Vehicle Detection

Table 9-3

Syntax	http://<*server*>/cgi-bin/trafficSnap.cgi?action=<***Action***>&channel=<***ChannelNo***>&name=UnlicensedVehicle
Method	GET
Description	Open or Close the unlicensed vehicle detection.
Example	http://192.168.1.108/cgi-bin/trafficSnap.cgi?action=open&channel=1&name=UnlicensedVehicle
Success Return	OK
Comment	Parameters in URL: ***Action***: string, this value can be open and close. ***ChannelNo***: integer, video channel index which starts from 1.

9.1.4 Snap

Table 9-4

Syntax	http://<*server*>/cgi-bin/trafficSnap.cgi?action=manSnap&channel=<***ChannelNo***>
Method	GET
Description	Take a snapshot manually. For intelligent traffic device, it should use this method to take a snapshot. But, the response is not image data.If you want to get the image data, please follow these steps: 1. Use the method mentioned chapter (4.4.3 Subscribe to snapshot) to subscribe the image data, and the eventcode is "TrafficManualSnap". 2. Use the "manSnap"to take a snapshot manually. 3. In the connection which built in the Step 1, the device will send the image data.
Example	http://192.168.1.108/cgi-bin/trafficSnap.cgi?action=manSnap&channel=1
Success Return	OK
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1.

9.2 Traffic Parking

9.2.1 Get All Status of Parking Spaces

Table 9-5

Syntax	http://<*server*>/cgi-bin/trafficParking.cgi?action=getAllParkingSpaceStatus
Method	GET
Description	Get all valid parking spaces status of one device.
Example	http://192.168.1.108/cgi-bin/trafficParking.cgi?action=getAllParkingSpaceStatus
Success Return	A list of parking space status status[0].Lane=0 status[0]. CustomParkNo = A2701 status[0].**Status** = Park … status[1].Lane=1 status[1]. **Status** = NoPark …
Comment	Parameters in Response : ***Status*** : Park or NoPark

9.2.2 Parking Space Light State

- Get parking space light state

Table 9-6

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=ParkingSpaceLightState
Method	GET
Description	Use this method. It can get the light state config. For example, it can know that when the space is free, then the light should be green, and the space is full, the light should be red.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=ParkingSpaceLightState
Success Return	table.ParkingSpaceLightState.SpaceFree.Blue=0 table.ParkingSpaceLightState.SpaceFree.Green=1 table.ParkingSpaceLightState.SpaceFree.Pink=0 table.ParkingSpaceLightState.SpaceFree.Purple=0 table.ParkingSpaceLightState.SpaceFree.Red=0 table.ParkingSpaceLightState.SpaceFree.White=0 table.ParkingSpaceLightState.SpaceFree.Yellow=0 table.ParkingSpaceLightState.SpaceFull.Blue=0 table.ParkingSpaceLightState.SpaceFull.Green=0 table.ParkingSpaceLightState.SpaceFull.Pink=0 table.ParkingSpaceLightState.SpaceFull.Purple=0 table.ParkingSpaceLightState.SpaceFull.Red=1 table.ParkingSpaceLightState.SpaceFull.White=0

	table.ParkingSpaceLightState.SpaceFull.Yellow=0 table.ParkingSpaceLightState.SpaceOrder.Blue=0 table.ParkingSpaceLightState.SpaceOrder.Green=0 table.ParkingSpaceLightState.SpaceOrder.Pink=0 table.ParkingSpaceLightState.SpaceOrder.Purple=0 table.ParkingSpaceLightState.SpaceOrder.Red=0 table.ParkingSpaceLightState.SpaceOrder.White=0 table.ParkingSpaceLightState.SpaceOrder.Yellow=1 table.ParkingSpaceLightState.SpaceOverLine.Blue=0 table.ParkingSpaceLightState.SpaceOverLine.Green=0 table.ParkingSpaceLightState.SpaceOverLine.Pink=0 table.ParkingSpaceLightState.SpaceOverLine.Purple=0 table.ParkingSpaceLightState.SpaceOverLine.Red=0 table.ParkingSpaceLightState.SpaceOverLine.White=0 table.ParkingSpaceLightState.SpaceOverLine.Yellow=1 table.ParkingSpaceLightState.SpaceSpecial.Blue=0 table.ParkingSpaceLightState.SpaceSpecial.Green=0 table.ParkingSpaceLightState.SpaceSpecial.Pink=0 table.ParkingSpaceLightState.SpaceSpecial.Purple=0 table.ParkingSpaceLightState.SpaceSpecial.Red=0 table.ParkingSpaceLightState.SpaceSpecial.White=0 table.ParkingSpaceLightState.SpaceSpecial.Yellow=1
Comment	Parameters in Response : There are five conditions about the space light state: SpaceFree, SpaceFull, SpaceOrder, SpaceOverLine, and SpaceSpecial. For each condition, you can choose the light state. The state value is integer. It can be 0, 1, 2. 0 means close; 1 means open; 2 means twinkle.

- Set parking space light state

Table 9-7

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<*para**m**Value*>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set the light state config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&ParkingSpaceLightState.SpaceFree.Blue=1&ParkingSpaceLightState.SpaceFree.Green=0&ParkingSpaceLightState.SpaceFree.Pink=0&ParkingSpaceLightState.SpaceFree.Purple=0&ParkingSpaceLightState.SpaceFree.Red=0&ParkingSpaceLightState.SpaceFree.White=0&ParkingSpaceLightState.SpaceFree.Yellow=0
Success Return	OK
Comment	—

9.2.3 Set Order State

Table 9-8

Syntax	http://<*server*>/cgi-bin/trafficParking.cgi?action=setOrderState&state[0].Lane=<***LaneNumber***>&state[0].State=<***State***>
Method	GET
Description	Set the light order state, order or not
Example	http://192.168.1.108/cgi-bin/trafficParking.cgi?action=setOrderState&state[0].Lane=0&state[0].State=Ordered
Success Return	OK
Comment	Parameters in URL: ***LaneNumber***: integer, for now, fixed to 0. ***State***: string, the value can be "Ordered"or "Free".

9.2.4 Set Light State

Table 9-9

Syntax	http://<*server*>/cgi-bin/trafficParking.cgi?action=setLightState&state[***Index***].LightNo=<***LaneNumber***>&state[***Index***].Color=<***Color***>&state[***Index***].State=<***State***>&state[***Index***].Enable=<***Enable***>
Method	GET
Description	Set the light state.
Example	http://192.168.1.108/cgi-bin/trafficParking.cgi?action=setLightState&state[0].LightNo=0&state[0].Color=Red&state[0].State=0&state[0].Enable=true
Success Return	OK
Comment	Parameters in URL: ***Index***: integer, start with 0. ***LaneNumber***: integer, the No. of the Lane. ***Color***: string, it can be Red, Yellow, Blue, Green, Purple, White, Pink. ***State***: integer, it can be 0, 1, 2.0 means close;1 means open;2 means twinkle; ***Enable****:* true or false, enable or not.

9.2.5 [Config] Parking Space Access Filter Setting

- Get parking space access filter setting

Table 9-10

URL	http://<*server*>/cgi-bin/ConfigManager.cgi?action=getConfig&name=ParkingSpaceAccessFilter
Method	GET
Description	Get Parking Space Access Filter config. Using this method, we can get the accessible address of the device.
[Request Params] (None)	
[Response Params] (key=value format)	

Name	Type	R/O	Param Description
table	object	R	config table object
+ParkingSpaceAccessFilter	object	R	ParkingSpaceAccessFilter config object
++Enable	bool	R	Enable filter or not
++Type	string	R	Filter type, can be: "**BannedList**", "**TrustList**".
++TrustList	Array<string>	O	IP address list that trust
++BannedList	Array<string>	O	IP address list that banned.
[Example]			
Request	GET http://10.0.0.8/cgi-bin/ConfigManager.cgi?action=getConfig&name=ParkingSpaceAccessFilter		
Response	table.ParkingSpaceAccessFilter.Enable=false table.ParkingSpaceAccessFilter.Type=TrustList table.ParkingSpaceAccessFilter.TrustList[0]=172.24.2.14 table.ParkingSpaceAccessFilter.BannedList[0]=172.24.2.15		

- Set parking space access filter setting

Table 9-11

URL	http://<*server*>/cgi-bin/ConfigManager.cgi?action=setConfig		
Method	GET		
Description	Set Parking Space Access Filter config.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
ParkingSpaceAccessFilter	object	R	ParkingSpaceAccessFilter config object
+Enable	bool	R	Enable filter or not
+Type	string	R	Filter type, can be: "**BannedList**", "**TrustList**".
+TrustList	Array<string>	O	IP address list that trust
+BannedList	Array<string>	O	IP address list that banned.
[Response Params] (OK)			
[Example]			
Request	GET http://10.0.0.8/cgi-bin/ConfigManager.cgi?action=setConfig&ParkingSpaceAccessFilter.Enable=true&ParkingSpaceAccessFilter.Type=TrustList&ParkingSpaceAccessFilter.TrustList[0]=172.24.2.14&ParkingSpaceAccessFilter.BannedList[0]=172.24.2.15		
Response	OK		

9.2.6 Set OverLine State

Table 9-12

Syntax	http://<server>/cgi-bin/trafficParking.cgi?action=setOverLineState&state[0].Lane=<***LaneNumber***>&state[0].State=<***State***>
Method	GET
Description	Set the spaceState overLine state or StopOverLine state

Example	http://192.168.1.108/cgi-bin/trafficParking.cgi?action=setOverLineState&state[0].Lane=0&state[0].State=OverLine
Success Return	OK
Comment	Parameters in URL: ***LaneNumber***: integer, for now, fixed to 0. ***State***: string, the value can be “OverLine” or “StopOverLine”.

9.3 Traffic Record Import Export

9.3.1 Traffic BlackList / RedList

- Import blacklist or redlist record

Table 9-13

Syntax	http://<*server*>/cgi-bin/trafficRecord.cgi?action=uploadFile&Type=<***Type***>&format=<***Format***>&code=<***Code***>
Method	POST
Description	Upload a blacklist or redlist(whitelist) file into the deivce, and the device will import the data of the file.This operation may take a long time.
Example	POST /cgi-bin/trafficRecord.cgi?action=uploadFile&Type=TrafficBlackList&format=CSV&code=UTF-8 HTTP/1.1 Host: 192.168.1.108 Connection: keep-alive Content-Length:XXXX Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryooT6JTCbuezAQeDy ------WebKitFormBoundaryooT6JTCbuezAQeDy Content-Disposition: form-data; name="blackfile"; filename="TrafficBlackList.CSV" Content-Type: application/vnd.ms-excel File data…. ------WebKitFormBoundaryooT6JTCbuezAQeDy--
Success Return	OK
Comment	Parameters in URL: ***Type***: string, the value can be “TrafficBlackList” or “TrafficRedList”. ***Format***: string, for the blacklist or redlist(whitelist), the value is fixed to "CSV". ***Code***: string, for the blacklist, the value can be "utf-8"or "GB2312".

- Export blacklist or redlist record async

Table 9-14

Syntax	http://<server>/cgi-bin/recordUpdater.cgi?action=exportAsyncFile&name=<***Name***>&format=<***Format***>&code=<***Code***>

Method	GET
Description	Let the device export the record into a file in the device. This method is non-blocking. To get the result, it should use the “getFileExportState” method.
Example	http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=exportAsyncFile&name=TrafficBlackList&format=CSV&code=utf-8
Success Return	OK
Comment	Parameters in URL: ***Name***: string, the value can be “TrafficBlackList” or “TrafficRedList”. ***Format***: string, for the blacklist or redlist(whitelist), the value is fixed to “CSV”. ***Code***: string, for the blacklist or whitelist, the value can be “utf-8” or “GB2312”.

- Get export blacklist or redlist record state

Table 9-15

Syntax	http://<server>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=<***Name***>
Method	GET
Description	Get the blacklist or redlist(whitelist) export state
Example	http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficBlackList
Success Return	state=0
Comment	Parameters in URL: ***Name***: string, the value can be “TrafficBlackList” or “TrafficRedList”. ***state***: integer, the value can be 0(success),1(fail), 2(running), 3(the file is invalid), 4(the file is too large) or 5(have duplicate info in the file).

- Get export blacklist or redlist record result

Table 9-16

Syntax	http://<server>/cgi-bin/trafficRecord.cgi?action=downloadFile&Type=<***Type***>
Method	GET
Description	Download the blacklist or redlist(whitelist) record
Example	http://192.168.1.108/cgi-bin/trafficRecord.cgi?action=downloadFile&Type=TrafficBlackList
Success Return	File data
Comment	Parameters in URL: ***Type***: string, the value can be “TrafficBlackList” or “TrafficRedList”.

9.3.2 Traffic Flow

- Export traffic flow record

Table 9-17

Syntax	http://<*server*>/cgi-bin/recordUpdater.cgi?action=exportAsyncFile&name=TrafficFlow&format=<***Format***>&code=<***Code***>
Method	GET
Description	Let the deivce export the record into a file in the device.This method is non-blocking.To get the result, it should use the "getFileExportState"method.

Example	http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=exportAsyncFile&name=TrafficFlow&format=CSV&code=utf-8
Success Return	OK
Comment	Parameters in URL: ***Format***: string, for the TrafficFlow, the value is fixed to "CSV". ***Code***: string, for the TrafficFlow, the value can be "utf-8"or "GB2312".

Table 9-18

Syntax	http://<*server*>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficFlow
Method	GET
Description	Get the export state.
Example	http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficFlow
Success Return	state=0
Comment	Parameters in Response: ***state***: integer,the value can be 0(success),1(fail), 2(running), 3(the file is invalid), 4(the file is too large) or 5(have duplicate info in the file).

Table 9-19

Syntax	http://<*server*>/cgi-bin/trafficRecord.cgi?action=downloadFile&Type=TrafficFlow
Method	GET
Description	Download the traffic flow record.
Example	http://192.168.1.108/cgi-bin/trafficRecord.cgi?action=download&filename=TrafficFlow
Success Return	File data
Comment	—

9.3.3 Traffic Snap Event Info

- Export traffic snap event info record

Table 9-20

Syntax	http://<*server*>/cgi-bin/recordUpdater.cgi?action=exportAsyncFileByConditon&name=TrafficSnapEventInfo&format=<***Format***>&code=<***Code***>&condition.startTime=<***StartTime***>&condition.endTime=<***EndTime***>
Method	GET
Description	Let the deivce export the record into a file in the device.This method is non-blocking.To get the result, it should use the "getFileExportState"method.
Example	http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=exportAsyncFileByConditon&name=TrafficSnapEventInfo&format=CSV&code=utf-8&condition.startTime=123456700&condition.endTime=123456800
Success Return	OK
Comment	Parameters in URL: ***Format***: string, for the TrafficSnapEventInfo, the value is fixed to "CSV".

	Code: string, for the TrafficSnapEventInfo, the value can be "utf-8"or "GB2312". ***StartTime***: string, the start of the record's Time. ***EndTime***: string, The end of the record's Time.

Table 9-21

Syntax	http://<*server*>/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficSnapEventInfo
Method	GET
Description	Get the export state.
Example	http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=getFileExportState&name=TrafficSnapEventInfo
Success Return	state=0
Comment	Parameters in Response: ***state***: integer,the value can be 0(success),1(fail), 2(running), 3(the file is invalid), 4(the file is too large) or 5(have duplicate info in the file).

Table 9-22

Syntax	http://<*server*>/cgi-bin/trafficRecord.cgi?action=downloadFile&Type=TrafficSnapEventInfo
Method	GET
Description	Download the traffic snap event info record.
Example	http://192.168.1.108/cgi-bin/trafficRecord.cgi?action=download&filename=TrafficSnapEventInfo
Success Return	File data
Comment	—

10 Thermography and Radiometry APIs

10.1 Thermography Manager

10.1.1 Get Capability of Thermography

Table 10-1

Syntax	http://<*server*>/cgi-bin/ThermographyManager.cgi?action=getCaps&channel=<***ChannelNo***>
Method	GET
Description	Get thermography capability.
Example	http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=getCaps&channel=1
Success Return	caps.**PresetModes** = Indoor caps.Brightness.Max = 100 caps.Brightness.Min = 0 caps.Brightness.Step = 1 caps.Sharpness.Max= 100 caps.Sharpness.Min = 0 caps.Sharpness.Step = 5 caps.EZoom.Max= 24 caps.EZoom.Min = 0 caps.EZoom.Step = 1 caps. ThermographyGamma.Max= 8 caps. ThermographyGamma.Min = -8 caps. ThermographyGamma.Step = 1 caps. SmartOptimizer.Max= 100 caps. SmartOptimizer.Min = 0 caps. SmartOptimizer.Step = 5 caps. Agc.Max= 255 caps. Agc.Min = 0 caps. Agc.Step = 5 caps. AgcMaxGain.Max= 255 caps. AgcMaxGain.Min = 0 caps. AgcMaxGain.Step = 5 caps. AgcPlateau.Max= 100 caps. AgcPlateau.Min = 0 caps. AgcPlateau.Step = 5 caps.**PresetColorization**[i]= Ironbow2 caps.**PresetROIModes**[j]= Full Screen
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. Parameters in Response:

	PresetModes: the preset mode. Range is { "Indoor", "Outdoor", "Default" } PresetColorization: Preset colorization mode. Range is { "WhiteHot", "BlackHot", "Fusion", "Rainbow", "Globow", "Ironbow1", "Ironbow2", "Sepia", "Color1", "Color2", "Icefire", "Rain", "RedHot", "GreenHot"}. **PresetROIModes**: Preset ROI mode. Range is {"Full Screen", "Sky", "Ground", "Horizon", "Center 75%", "Center 50%", "Center 25%", "Custom"}

10.1.2 Thermography Options

- Get thermography options config

Table 10-2

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=ThermographyOptions
Method	GET
Description	Thermography options contain EZoom, Colorization, and SmartOptimizer and so on.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=ThermographyOptions
Success Return	***head***.EZoom=0 ***head***.Colorization=White Hot ***head***.SmartOptimizer=10 ***head***.OptimizedRegion.Type=Custom ***head***.OptimizedRegion.Enable= true ***head***.OptimizedRegion.**Regions**[*i*][0u]=0 ***head***.OptimizedRegion.**Regions**[*i*][1u]=0 ***head***.OptimizedRegion.**Regions**[*i*][2u]=0 ***head***.OptimizedRegion.**Regions**[*i*][3u]=0 ***head***.Agc=10 ***head***.AgcMaxGain=10 ***head***.AgcPlateau=10 ***head***.Mode="HighTemperature" ***head***.Auto.LowToHigh=13 ***head***.Auto.LHROI=15 ***head***.Auto.HighToLow=12 ***head***.Auto.HLROI=95
Comment	Parameters in Response: ***head*** = table.ThermographyOptions [***ChannelNo***][0] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). **Regions**: the region is a rectangle ***i:*** the array index starts from 0.

- Set thermography options config

Table 10-3

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set thermography options.

Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&ThermographyOptions[0][0].OptimizedRegion.Type=Gound
Success Return	OK
Comment	Parameters in URL: The paramName and paramValue are in the table below. In table below, ***head*** = ThermographyOptions[***ChannelNo***][0] ***ChannelNo:*** integer, array index starts from 0, which means video channel(equals to video channel index -1, and so 0 means channel 1). ***i:*** the array index starts from 0.

Appendix

ParamName	ParamValue type	Description
head. EZoom	integer	Range is [0—24]. Range and step are got from interface in "10.1.1 Get Capability of Thermography".
head. Colorization	String	Range is {"White Hot", "Black Hot", "Ironbow2", "IceFire"…}. Range and step are got from interface in "10.1.1 Get Capability of Thermography".
head. SmartOptimizer	integer	Range is [0—100]. Range and step are got from interface in "10.1.1 Get Capability of Thermography".
head. OptimizedRegion.Type	String	Range is {"Full Screen", "Sky", "Ground", "Horizontal", "Center 75%", "Center 50%", "Center 25%", "Custom"}.
head. OptimizedRegion.Enable	bool	true: enable false: not enable
head. OptimizedRegion.Regions[*i*][0u]	integer	Range is [0—8191]. ***i:*** the region index, starts from 0.
head. OptimizedRegion.Regions[*i*][1u]	integer	Range is [0—8191]. ***i:*** the region index, starts from 0.
head. OptimizedRegion.Regions[*i*][2u]	integer	Range is [0—8191]. ***i:*** the region index, starts from 0.
head. OptimizedRegion.Regions[*i*][3u]	integer	Range is [0—8191]. ***i:*** the region index, starts from 0.
head. Agc	integer	Range is [0—255]. Range and step are got from interface in "10.1.1 Get Capability of Thermography".
head. AgcMaxGain	integer	Range is [0—255]. Range and step are got from interface in "10.1.1

ParamName	ParamValue type	Description
		Get Capability of Thermography".
head. AgcPlateau	integer	Range and step are got from interface in "10.1.1 Get Capability of Thermography".
head. Mode	string	Range is {"HighTemperature", "LowTemperature", "Auto"}.
head. Auto.LowToHigh	integer	UInt32
head. Auto.LHROI	integer	UInt32, percentage range is[0—100]
head. Auto.HighToLow	integer	UInt32
head. Auto.HLROI	integer	UInt32, percentage range is[0—100]

10.1.3 Get ExternSystem Information

Table 10-4

Syntax	http://<*server*>/cgi-bin/ThermographyManager.cgi?action=getExternSystemInfo&channel=<***ChannelNo***>
Method	GET
Description	Get extern system Info.
Example	http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=getExternSystemInfo&channel=1
Success Return	sysInfo.SerialNumber = 11111111123 sysInfo.SoftwareVersion = 2222222222222 sysInfo.FirmwareVersion= 3333333333333 sysInfo.LibVersion = 4444444444
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1.

10.1.4 Get Information of Preset Mode

Table 10-5

Syntax	http://<*server*>/cgi-bin/ThermographyManager.cgi?action=getPresetParam&channel=<***ChannelNo***>&mode=<***modeType***>
Method	GET
Description	Get preset mode info.
Example	http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=getPresetParam&channel=1&mode=Default
Success Return	presetInfo.Brightness = 50 presetInfo.Sharpness= 50 presetInfo.EZoom= 12 presetInfo.ThermographyGamma= 0 presetInfo.Colorization= "White Hot" presetInfo.SmartOptimizer= 10 presetInfo.OptimizedRegion.Type= Full Screen presetInfo.OptimizedRegion.Enable= Full Screen presetInfo.OptimizedRegion.**Regions[i]**[0u]=0 presetInfo.OptimizedRegion.**Regions[i]**[1u]=0

	presetInfo.OptimizedRegion.**Regions[i]**[2u]=0 presetInfo.OptimizedRegion.**Regions[i]**[3u]=0 presetInfo.Agc= 10 presetInfo.AgcMaxGain=10 presetInfo.AgcPlateau = 10
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. ***modeType***: depends on capability , get from interface in getCaps Parameters in Response : **Regions** : the region is a rectangle **i :** the array index.

10.1.5 Get Optimized Region Information

Table 10-6

Syntax	http://<*server*>/cgi-bin/ThermographyManager.cgi?action=getOptimizedRegion&channel=<***ChannelNo***>
Method	GET
Description	Get optimized region info.
Example	http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=getOptimizedRegion&channel=1
Success Return	optimizedRegion.Type= Full Screen optimizedRegion.Enable= true optimizedRegion.**Regions[*i*]**[0u]=0 optimizedRegion.**Regions[*i*]**[1u]=0 optimizedRegion.**Regions[*i*]**[2u]=0 optimizedRegion.**Regions[*i*]**[3u]=0
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. Parameters in Response: **Regions** : the region is a rectangle ***i:*** the region index.

10.1.6 Enable Shutter

Table 10-7

Syntax	http://<*server*>/cgi-bin/ThermographyManager.cgi?action=enableShutter&channel=<***ChannelNo***>&enable=<***Enable***>
Method	GET
Description	Shutter control, whether enable shutter.
Example	http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=enableShutter&channel=1&enable=true
Success Return	OK
Comment	Parameters in URL:

	ChannelNo: integer, video channel index which starts from 1. ***Enable***: true or false, enable or not.

10.1.7 Fix Focus

Table 10-8

Syntax	http://<*server*>/cgi-bin/ThermographyManager.cgi?action=fixFocus&linkVideoChannel[0]=<***ChannelNo***>&linkVideoChannel[1]=<***ChannelNo***>[&speed=<***SpeedValue***>]
Method	GET
Description	The visual channel change focus to the same as the thermography channel.
Example	http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=fixFocus&linkVideoChannel[0]=1&linkVideoChannel[1]=2
Success Return	OK
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. ***SpeedValue***: float, range is 0.0-1.0.

10.1.8 Do Flat Field Correction

Table 10-9

Syntax	http://<*server*>/cgi-bin/ThermographyManager.cgi?action=doFFC&channel=<***ChannelNo***>
Method	GET
Description	Do flat field correction.
Example	http://192.168.1.108/cgi-bin/ThermographyManager.cgi?action=doFFC&channel=1
Success Return	OK
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1.

10.2 Radiometry

10.2.1 Get Capability of Radiometry

Table 10-10

Syntax	http://<*server*>/cgi-bin/RadiometryManager.cgi?action=getCaps[&channel=<***ChannelNo***>]
Method	GET
Description	Get the capabilities of radiometry manager.
Example	http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=getCaps&channel=1
Success Return	caps.TotalNum.MaxNum=8 caps.TotalNum.Spot.MaxSpots=8 caps.TotalNum.Line.MaxLines=1 caps.TotalNum.Area.MaxAreas=8 caps.TemperPresets.MaxPresets=256

	caps.MeterInfo.Type[0u]=Spot caps.MeterInfo.Type[1u]=Area caps.MeterInfo.ObjectEmissivity.Max=100 caps.MeterInfo.ObjectEmissivity.Min=0 caps.MeterInfo.ObjectEmissivity.Default=0 caps.MeterInfo.ObjectEmissivity.Step=1 caps.MeterInfo.ObjectDistanceMeter.Max=100 caps.MeterInfo.ObjectDistanceMeter.Min=0 caps.MeterInfo.ObjectDistanceMeter.Default=0 caps.MeterInfo.ObjectDistanceMeter.Step=1 caps.MeterInfo.ReflectedTemperature.Max=100 caps.MeterInfo.ReflectedTemperature.Min=0 caps.MeterInfo.ReflectedTemperature.Default=0 caps.MeterInfo.ReflectedTemperature.Step=1 caps.MeterInfo.RelativeHumidity.Max=100 caps.MeterInfo.RelativeHumidity.Min=0 caps.MeterInfo.RelativeHumidity.Default=0 caps.MeterInfo.RelativeHumidity.Step=1 caps.MeterInfo.AtmosphericTemperature.Max=100 caps.MeterInfo.AtmosphericTemperature.Min=0 caps.MeterInfo.AtmosphericTemperature.Default=0 caps.MeterInfo.AtmosphericTemperature.Step=1 caps.Statistics.MinPeriod=60 caps.Isotherm.MaxTemp=327.0 caps.Isotherm.MinTemp=-20.0
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1.

10.2.2 Heat Image Thermometry

- Get heat image thermometry config

Table 10-11

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=HeatImagingThermometry
Method	GET
Description	Get HeatImagingThermometry Config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=HeatImagingThermometry

Success Return	table.RelativeHumidity = 50 table.AtmosphericTemperature =20 table.ObjectEmissivity =1 table.ObjectDistance =100 table.ReflectedTemperature=20 table.TemperatureUnit= Centigrade table.Isotherm.Enable=true table.Isotherm.MaxValue=50 table.Isotherm.MinValue=0 table.Isotherm.ColorBarDisplay=true table.HotSpotFollow=true table.TemperEnable=true
Comment	—

- Set heat image thermometry config

Table 10-12

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set HeatImagingThermometry Config
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&HeatImagingThermometry.RelativeHumidity=50&HeatImagingThermometry.ObjectDistance=20.3
Success Return	OK
Comment	Parameters in URL: The paramName and paramValue are in the table below.

Appendix

ParamName	ParamValue type	Description
HeatImagingThermometry.RelativeHumidity	integer	The Relative Humidity range and step are got from interface in **getCaps**.
HeatImagingThermometry.AtmosphericTemperature	float	The Atmospheric Temperature range and step are got from interface in **getCaps**.
HeatImagingThermometry.ObjectEmissivity	float	The Object Emissivity range and step are got from interface in **getCaps**.
HeatImagingThermometry.ObjectDistance	float	The Object Distance range and step are got from interface in **getCaps**. Unit is meter.
HeatImagingThermometry.ReflectedTemperature	float	The Reflected Temperature range and step are got from interface in **getCaps**

ParamName	ParamValue type	Description
HeatImagingThermometry.TemperatureUnit	string	Range is {Centigrade, Fahrenheit}.
HeatImagingThermometry.Isotherm. Enable	bool	true or false
HeatImagingThermometry.Isotherm. MaxValue	float	MaxValue range is got form interface in getCaps. MaxValue must be bigger than MinVaue
HeatImagingThermometry.Isotherm. MinValue	float	MinValue range is got form interface in getCaps. MinValue must be smaller than MaxVaue.
HeatImagingThermometry.Isotherm. ColorBarDisplay	bool	true or false
HeatImagingThermometry.HotSpotFollow	bool	true or false
HeatImagingThermometry.TemperEnable	bool	true or false

10.2.3 Thermometry Rule

- Get thermometry rule config

Table 10-13

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=ThermometryRule
Method	GET
Description	Get thermometry rule.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=ThermometryRule
Success Return	***head***.Enable = true ***head***.PresetId =0 ***head***.RuleId=0 ***head***.Name=SpotName ***head***.Type=Spot ***head***.MeterRegion.Coordinates[***PointNo***][0]= 0 ***head***.MeterRegion.Coordinates[***PointNo***][1]= 0 … ***head***.T=3 ***head***.***Alarm***.Id=0 ***head***.***Alarm***.Enable=true ***head***.***Alarm***.Result =Max ***head***.***Alarm***.AlarmCondition=Below ***head***.*Alarm*.Threshold=20.0 ***head***.***Alarm***.Hysteresis=0.1 ***head***.***Alarm***.Duration=30 ***head***.LocalParameters.Enable=true ***head***.LocalParameters.ObjectEmissivity=0.95 ***head***.LocalParameters.ObjectDistance=0.95 ***head***.LocalParameters.RefalectedTemp=0

Comment	Parameters in Response : ***head*** =table.ThermometryRule[***ChannelNo***][***RuleNo***] *ChannelNo*: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***RuleNo*** =rule index. ***PointNo*** = point index ***Alarm***= AlarmSetting[***AlarmNo***] ***AlarmNo*** = alarm index

- Set thermometry rule config

Table 10-14

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set thermometry rule.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&ThermometryRule[0][0].Name=name1
Success Return	OK
Comment	Parameters in URL: The paramName and paramValue are in the table below. In table below, ***head*** = ThermometryRule[***ChannelNo***][***RuleNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). ***PointNo*** = point index ***RuleNo*** =rule index. ***Alarm***= AlarmSetting[***AlarmNo***] ***AlarmNo*** = alarm index

Appendix

ParamName	ParamValue type	Description
head. Enable	bool	Enable/Disable
head. PresetId	integer	Range [0—PresetMax] PresetMax is got from interface in **GetCurrentProtocolCaps**.
head. RuleId	integer	Range [0—MaxNum] MaxNum is got from interface in **getCaps**.
head. Name	string	Radiometry rule name. char[64]
head. Type	string	Range is {Spot, Line, Area }.
head. MeterRegion.Coordinates[***PointNo***] [0]	integer	Range [0—8091] The Xscale of Region/Line point

ParamName	ParamValue type	Description
head. MeterRegion.Coordinates[***PointNo***] [1]	integer	Range [0—8091] The Yscale of Region/Line point
head. T	integer	Temperature Sample period. Unit is Second.
head. *Alarm*.Id	integer	Range [0 — 65535],unique alarm id
head. *Alarm*.Enable	bool	Enable/Disable
head. *Alarm*.Result	string	Depend on the vaule of Type Spot : {Vaule} Line: { Max, Min, Aver} Area: {Max, Min, Aver, Std, Mid, ISO}
head. *Alarm*. AlarmCondition	string	Range is {Below, Match , Above }
head. *Alarm*. Threshold	float	Alarm threshold
head. *Alarm*. Hysteresis	float	Alarm hysteresis
head. *Alarm*. Duration	integer	The duration time of alarm. Unit is second
head. LocalParameters.Enable	bool	Enable/Disable
head. LocalParameters. ObjectEmissivity	float	Range [0 — 1] Accuracy is 0.01
head. LocalParameters. ObjectDistance	float	Object distance The range is got from interface in **getCaps**.
head. LocalParameters. ReflectedTemp	float	Object Reflected Temperature The range is got from interface in **getCaps**.

10.2.4 Heat Image Temper Event

- Get heat image temper event config

Table 10-15

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=HeatImagingTemper
Method	GET
Description	Get Heat Imaging Temper config
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=HeatImagingTemper
Success Return	***head***.Enable=false ***head***.EventHandler. paramName = paramValue
Comment	Parameters in Response: ***head***= table.HeatImagingTemper[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

- Set heat image temper event config

Table 10-16

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set Heat Imaging Temper config
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&HeatImagingTemper[0].Enable=false&HeatImagingTemper[0].EventHandler.BeepEnable=false
Success Return	OK
Comment	Parameters in URL: The paramName and paramValue are in the table below. In table below, ***head***= HeatImagingTemper[***ChannelNo***] ***ChannelNo***: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

Appendix

ParamName	ParamValue type	Description
head.Enable	bool	Enable/Disable Heat Imaging Temper feature.
head.EventHandler	—	Setting of EventHandler is described in **SetEventHandler**.

10.2.5 Get Temperature of Particular Point

Table 10-17

Syntax	http://<*server*>/cgi-bin/RadiometryManager.cgi?action=getRandomPointTemper&channel=<***ChannelNo***>&coordinate[0]=*x* &coordinate[1]=*y*
Method	GET
Description	Get temperature values of random point.
Example	http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=getRandomPointTemper&channel=1&coordinate[0]=1024&coordinate[1]=1024
Success Return	TempInfo.Type=Spot TempInfo.TemperAver=27.5
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. ***x*** : The Xscale of the point ***y*** : The Yscale of the point

10.2.6 Get Temperature of Particular Condition

Table 10-18

Syntax	http://<*server*>/cgi-bin/RadiometryManager.cgi?action=getTemper&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET

Description	Get temperature values from rules which have been set.
Example	http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=getTemper&condition.PresetId=0&condition.RuleId=0&condition.Type=Spot&condition.Name=Spot1&condition.channel=1
Success Return	TempInfo.Type=Spot TempInfo.TemperAver=27.5
Comment	Parameters in URL: The paramName and paramValue are in the table below.

Appendix

ParamName	ParamValue type	Description
condition. Channel	integer	Video channel index. Start from 1
condition. PresetId	integer	Range [0- PresetMax] PresetMax is got from interface in **GetCurrentProtocolCaps**.
condition. RuleId	integer	Range [0- MaxNum] MaxNum is got from interface in **getCaps**.
condition. Type	string	Range is {Spot, Line, Area}.
condition. Name	string	Name is got from interface in **GetThermometryRuleConfig**.

10.2.7 Query Temperature Information

1. Start to query temperature information

Table 10-19

Syntax	http://<*server*>/cgi-bin/RadiometryManager.cgi?action=startFind&*condition.StartTime*=<***StartTimeValue***>&*condition.EndTime*=<***EndTimeValue***>&*condition.Type*=<***TypeValue***>&*condition.channel*=<***ChannelValue***>&*condition.Period*=<***PeriodValue***>
Method	GET
Description	Start to query the history data of temperature values.
Example	http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=startFind&condition.StartTime=2010-04-01%200:00:00&condition.EndTime=2010-04-08%200:00:00&condition.Type=Spot&condition.channel=1&condition.Period=5
Success Return	token=46878 totalCount=333
Comment	The parameters in bold face are as table below.

Appendix

ParamName	ParamValue type	Description
condition.StartTime	string	The start time to find.
condition.EndTime	string	The end time to find.
condition.Type	string	The type of data. Range is {Spot, Line, Area}
condition.channel	integer	Video channel index. Start from 1
condition.Period	integer	Range is {5, 10, 15, 30}, minute

2. Get the data of temperature

Table 10-20

Syntax	http://<*server*>/cgi-bin/RadiometryManager.cgi?action=doFind&**token**=<*tokenvalue*>&**beginNumber**=<*BeginNumber*>&**count**=<*findNum*>
Method	GET
Description	Get the history data of temperature.
Example	http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=doFind&token=46878&beginNumber=16&count=16
Success Return	found=12 info[i].Time=2010-04-08 16:12:46 info[i].PresetId=0 info[i].RuleId=0 info[i].Type=Spot info[i].Name=xxxx info[i].Coordinate[0]=1024 info[i].Coordinate[1]=2048 info[i].Channel=0 info[i].TemperatureUnit=Centigrade info[i].QueryTemperInfo.TemperAve=50.1 info[i].QueryTemperInfo.TemperMax=50.2 info[i].QueryTemperInfo.TemperMin=50.0
Comment	Parameters in URL: ***token***: query token, get from interface of the first step above. ***beginNumber***: the begin index in this query. ***count***: the number you want to query. Params in Resp: ***i:*** the array index.

3. Stop finding temperature information

Table 10-21

Syntax	http://<*server*>/cgi-bin/RadiometryManager.cgi?action=stopFind&**token**=<*tokenvalue*>
Method	GET
Description	Stop to find the history data of temperature values.
Example	http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=stopFind&token=46878
Success Return	OK
Comment	**token**: query token, get from interface of the first step.

10.2.8 Subscribe to Temperature Information

Table 10-22

Syntax	http://<*server*>/cgi-bin/RadiometryManager.cgi?action=attachTemper&channel=<***ChannelNo***>
Method	GET
Description	Subscribe to temperature information of a channel.
Example	http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=attachTemper&channel=2
Success Return	--<*boundary*>\r\n Content-Type: text/plain\r\n

	Content-Length: <data length>\r\n\r\n info[i].Time=2010-04-08 16:12:46 info[i].PresetId=0 info[i].RuleId=0 info[i].Type=Spot info[i].Name=xxxx info[i].Coordinate[0]=1024 info[i].Coordinate[1]=2048 info[i].Channel=0 info[i].TemperatureUnit=Centigrade info[i].QueryTemperInfo.TemperAve=50.1 info[i].QueryTemperInfo.TemperMax=50.2 info[i].QueryTemperInfo.TemperMin=50.0
Comment	Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. Params in Resp : ***i:*** the array index.

10.2.9 Subscribe to Radiometry Data

Table 10-23

Syntax	http://<*server*>/cgi-bin/RadiometryManager.cgi?action=attachProc&channel=<***ChannelNo***>
Method	GET
Description	Subscribe to radiometry data of a channel. It needs to cooperate with interface below.
Example	http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=attachProc&channel=2
Success Return	--<*boundary*>\r\n Content-Type: text/plain\r\n Content-Length: <data length>\r\n\r\n dataInfo.Height=0 dataInfo.Width=0 dataInfo.Channel=0 dataInfo.Time=2010-05-25 00:00:00 dataInfo.Length=0 dataInfo.sensorType="Tau" dataInfo.Unzip.ParamR=1 dataInfo.Unzip.ParamB=1 dataInfo.Unzip.ParamF=1 dataInfo.Unzip.ParamO=1 --<*boundary*>\r\n Content-Type: application/http\r\n Content-Length: <data length>\r\n\r\n <Binary data>
Comment	***ChannelNo***: integer, video channel index which starts from 1.

10.2.10 To Fetch Radiometry Data

Table 10-24

Syntax	http://<*server*>/cgi-bin/RadiometryManager.cgi?action=toFetch&channel=<***ChannelNo***>
Method	GET
Description	Start to fetch radiometry data.
Example	http://192.168.1.108/cgi-bin/RadiometryManager.cgi?action=toFetch&channel=2
Success Return	status=Ready
Comment	***ChannelNo***: integer, video channel index which starts from 1. ***status***: Range is {Ready, Busy}. "Ready"means service available and "Busy"means service busy.

10.2.11 Get FireWarning Config

Table 10-25

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=FireWarning
Method	GET
Description	Get FireWarning config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=FireWarning
Success Return	head.Enable=true head.PresetId=0 head.Row = 31 head.Col = 40 head.Mode="Auto" head.TimeDurationEnable=false head.FireDuration= 15 head.DetectWindow[windowsNum].Regions[0]=123468789 head.DetectWindow[windowsNum].Regions[1]=123468789 head.DetectWindow[windowsNum].Regions[2]=123468789 head.DetectWindow[windowsNum].Regions[3]=123468789 head.DetectWindow[windowsNum].Postion[0]=0 head.DetectWindow[windowsNum].Postion[1]=0 head.DetectWindow[windowsNum].Postion[2]=0 head.DetectWindow[windowsNum].Postion[3]=0 head.DetectWindow[windowsNum].Sensitivity = 95 head. DetectWindow[windowsNum].Id=1 head.DetectWindow[windowsNum].Name="windName" head.EventHandler=(output of EventHandler is described in GetEventHandler)
Comment	Parameters in Response: head= table.FireWarning[ChannleNo][RuleNum] ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1). RuleNum: integer,array index starts from 0,which means rules in each Preset, If FireWarningMode is "SpaceExClude", then only the first rule used.

	windowsNum: support 4 Nums if FireWarningMode is "PtzPreset", 8 if "SpaceExClude" Postion: only valid in "SpaceExClude"Mode

10.2.12 Set FireWarning Config

Table 10-26

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<paramName>=<paramValue>[&<paramName>=<paramValue>…]
Method	GET
Description	Set FireWarning config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&FireWarning[0][0].Eanble=false
Success Return	OK
Comment	head = FireWarning[ChNum][RuleNum]

Appendix:

ParamName	ParamValue type	Description
head.Enable	bool	whether fire detect take effect
head.PresetId	integer	The PresetId
head.Row	integer	rows of fire detect area
head.Col	integer	cols of fire detect area
head.Mode	string	Range is {"Auto","Normal"}.
head.TimeDurationEnable	bool	whether include fire detect duration,only take effect in SpaceExClude mode
head.FireDuration	integer	fire last times
head.DetectWindow[windowsNum].Regions[0]	integer	detect area mask
head.DetectWindow[windowsNum].Sensitivity	integer	Rage {0,100}
head. DetectWindow[windowsNum].Id	integer	Rage {0,…}
head.DetectWindow[windowsNum].Name	string	detect window name
head.EventHandler	EventHandler	Setting of EventHandler is described in SetEventHandler.

10.2.13 Get FireWarningMode Config

Table 10-27

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=FireWarningMode
Method	GET
Description	Get FireWarningMode config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=FireWarningMode
Success Return	head.Mode="PtzPreset"
Comment	Parameters in Response: head= table.FireWarningMode[ChannelNo]

	ChannelNo: integer, array index starts from 0, which means video channel (equals to video channel index -1, and so 0 means channel 1).

10.2.14 Set FireWarningMode Config

Table 10-28

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set FireWarningMode config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&FireWarningMode[0]. Mode ="SpaceExClude"
Success Return	OK
Comment	***head*** = FireWarningMode[ChannelNo]

Appendix:

ParamName	ParamValue type	Description
head.Mode	string	Rage is {"PtzPreset","SpaceExClude"}

10.2.15 Get Current Hot Cold Spot

URL	http://<*server*>/cgi-bin/TemperCorrection.cgi?action=getCurrentHotColdSpot		
Method	GET		
Description	Get the max and min temperature values.		
[Request Params] (key=value format in URL)			
channel	int	R	video channel index which starts from 1.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
info	object	R	The current hot code spot info.
+HotPoint	array<int>	R	The hot spot position, must be two int, means x and y value, coordinate remap to 0 — 8192.
+HotSpotValue	double	R	The hot spot temperature value.
+ColdPoint	array<int>	O	The cold spot position, must be two int, means x and y value, coordinate remap to 0 — 8192.
+ColdSpotValue	double	O	The cold spot temperature value.
+TemperatureUnit	int	O	The temperature unit : 0 Centigrade, 1 Fahrenheit
[Example]			
Request	GET http://192.168.1.108/cgi-bin/TemperCorrection.cgi?action=getCurrentHotColdSpot&channel=1		
Response	info.HotPoint[0]=1150 info.HotPoint[1]=2320 info.HotSpotValue=35.5 info.ColdPoint[0]=5452 info.ColdPoint[1]=6192 info.ColdSpotValue=24.3		

	info.TemperatureUnit=0

11 Access Control APIs

11.1 Door

11.1.1 Open Door

Table 11-1

Syntax	http://<*server*>/cgi-bin/accessControl.cgi?action=openDoor&channel=<***ChannelNo***>[&UserID=<***UserID***>&Type=<***Type***>]
Method	GET
Description	Open the door.
Example	http://192.168.1.108/cgi-bin/accessControl.cgi?action=openDoor&channel=1&UserID=101&Type=Remote
Success Return	OK
Comment	Parameters in URL: ***ChannelNo***: integer, the index of door, starts from 1. ***UserID***: remote user ID. ***Type***: the open type, default value is "*Remote*".

11.1.2 Get Door Status

Table 11-2

Syntax	http://<*server*>/cgi-bin/accessControl.cgi?action=getDoorStatus&channel=<***ChannelNo***>
Method	GET
Description	Get status of the door.
Example	http://192.168.1.108/cgi-bin/accessControl.cgi?action=getDoorStatus&channel=1
Success Return	Info.status=Open
Comment	Parameters in URL : ***ChannelNo***: integer, the index of door, starts from 1. Parameters in Response : ***status*** : the range is {Open, Break, Close}

11.1.3 Close Door

Table 11-3

Syntax	http://<*server*>/cgi-bin/accessControl.cgi?action=closeDoor&channel=<***ChannelNo***>[&UserID=<***UserID***>&Type=<***Type***>]
Method	GET
Description	Close the door.
Example	http://192.168.1.108/cgi-bin/accessControl.cgi?action=closeDoor&channel=1&UserID=101&Type=Remote

Success Return	OK
Comment	Parameters in URL: ***ChannelNo***: integer, the index of door, starts from 1. ***UserID***: remote user ID. ***Type***: the open type, default value is "*Remote*".

11.2 Access Control

11.2.1 Add Access Control Custom Password

Table 11-4

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCustomPassword		
Method	GET		
Description	Insert access control custom password.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access control custom password record name, should be "**AccessControlCustomPassword**".
UserID	string	R	The user's id.
OpenDoorPassword	string	R	Ths user’s open door password.
AlarmPassword	string	O	Ths user’s alarm password. (This param is not supported by video talk device)
Doors	array<int>	O	The index of the doors that custom password can open. (This param is not supported by video talk device)
TimeSections	array<int>	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)
VTOPosition	string	O	VTO position number. (This param is not supported by access control device)
ValidDateStart	string	O	The start of valid date, format is "**yyyyMMdd hhmmss**".
ValidDateEnd	string	O	The end of valid date, format is "**yyyyMMdd hhmmss**".
ValidCounts	int	O	The password’s valid counts.
OriginSmartGateWay	string	O	The origin smart gateway address. (Video talk device customize.)
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
RecNo	int	R	The record id.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCustomPassword&UserID=102&OpenDoorPassword=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&VTOPosition=01018001&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811&ValidCounts=30		

Response	RecNo=12345

11.2.2 Modify Access Control Custom Password

Table 11-5

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=update&name=AccessControlCustomPassword		
Method	GET		
Description	Update access control custom password. **Note:** You should provide at lease one optional param to update.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access control custom password record name, should be "**AccessControlCustomPassword**".
recno	int	R	The record id.
UserID	string	R	The user's id.
OpenDoorPassword	string	R	Ths user’s open door password.
Doors	array<int>	R	The index of the doors that custom password can open.
…<other param>	—	—	…<See above insert command for other params of the record, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=update&name=AccessControlCustomPassword&recno=12345&UserID=102&OpenDoorPassword=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811		
Response	OK		

11.2.3 Delete Access Control Custom Password

- Delete access control custom password record by recno

Table 11-6

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=remove&name=AccessControlCustomPassword		
Method	GET		
Description	Remove the access control custom password record by recno.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be "**AccessControlCustomPassword**".
recno	int	R	The record id.
[Response Params] (OK)			
[Example]			

Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=remove&name=AccessControlCustomPassword&recno=12345
Response	OK

- Delete all the access control custom password records

Table 11-7

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=clear&name=AccessControlCustomPassword		
Method	GET		
Description	Remove all the access control custom password records		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be "**AccessControlCustomPassword**".
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=clear&name=AccessControlCustomPassword		
Response	OK		

11.2.4 Find Access Control Custom Password

- Find Access control custom password by condition

Table 11-8

URL	http://<*server*>/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCustomPassword		
Method	GET		
Description	Find Access control custom password by condition.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Record Name, should be "**AccessControlCustomPassword**".
count	int	O	Max result to return, default is 1024.
StartTime	string	O	The start of the record's CreateTime.
EndTime	string	O	The End of the record's CreateTime.
condition	object	O	Search condition.
+UserID	string	O	The user's id.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
totalCount	int	O	The total record num that find.
found	int	O	The record num that returned.
records	array<object>	R	The records that returned.
+RecNo	int	R	The record id.
+CreateTime	int	R	The create time of record.
+UserID	string	R	The user's id.

+OpenDoorPassword	string	R	Ths user's open door password.
+AlarmPassword	string	O	Ths user's alarm password. (This param is not supported by video talk device)
+Doors	array<int>	O	The index of the doors that custom password can open. (This param is not supported by video talk device)
+TimeSections	array<int>	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)
VTOPosition	string	O	VTO position number. (This param is not supported by video talk device)
+ValidDateStart	string	O	The start of valid date, format is "**yyyyMMdd hhmmss**".
+ValidDateEnd	string	O	The end of valid date, format is "**yyyyMMdd hhmmss**".
+ValidCounts	int	O	The password's valid counts.
+OriginSmartGateWay	string	O	The origin smart gateway address. (Video talk device customize.)

[Example]

Request	GET http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCustomPassword&condition.UserID=103&StartTime=123456700&EndTime=123456800&count=100
Response	totalCount=1000 found=100 records[0].RecNo=12345 records[0].CreateTime=123456789 records[0].UserID=103 records[0].OpenDoorPassword=123456 records[0].Doors[0]=1 records[0].Doors[1]=3 records[0].Doors[2]=5 records[0].VTOPosition=01018001 records[0].ValidStart=20151022 093811 records[0].ValidEnd=20151222 093811 … records[1].RecNo=13579 records[1].CreateTime=123456799 records[1].UserID=103 records[0].OpenDoorPassword=123456 records[1].Doors[0]=2 records[1].Doors[1]=4 records[1].Doors[2]=6 records[1].VTOPosition=01018002 records[1].ValidStart=20151022 093811 records[1].ValidEnd=20151222 093811 …

- Find Access control custom password by recno

Table 11-9

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=get&name=AccessControlCustomPassword		
Method	GET		
Description	Find Access control custom password by recno.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Record Name, should be "**AccessControlCustomPassword**".
recno	int	R	The record id.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
record	object	R	The record that returned.
+RecNo	int	R	The record id.
+CreateTime	int	R	The create time of record.
+UserID	string	R	The user's id.
…<other param>	—	—	…<See above find command for other params of the record.>
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=get&name=AccessControlCustomPassword&recno=3		
Response	record.RecNo=3 record.CreateTime=123456789 record.UserID=103 record.OpenDoorPassword=123456 record.Doors[0]=1 record.Doors[1]=3 record.Doors[2]=5 record.VTOPosition=01018001 record.ValidStart=20151022 093811 record.ValidEnd=20151222 093811 …		

11.2.5 Get the Total Number of Records of Access Control Custom Password

Table 11-10

URL	http://<*server*>/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCard		
Method	GET		
Description	Get the total number of records of the access control custom password.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be "**AccessControlCustomPassword**".
[Response Params] (key=value)			
count	int	R	The total number of records.
[Example]			

Request	GET http://192.168.1.108/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCustomPassword
Response	count=150

11.2.6 Get Access Control Caps

Table 11-11

URL	http://<*server*>/cgi-bin/accessControlManager.cgi?action=getCaps		
Method	GET		
Description	Get access control capabilities.		
[Request Params] (none)			
[Response Params] (key=value)			
caps	object	R	The access control capabilities.
+AccessControlChannels	int	R	The access control channel number.
+AccessControlAlarmRecord	bool	O	Support log access control alarm record or not. (This param is not supported by video talk device)
+CustomPasswordEncryption	int	O	The custom password crypt type, 0 : plain text, 1 : MD5
+SupportFingerPrint	int	O	Support fingerprint type, 0 : unknown, 1 : not support, 2 : support
+OnlySingleDoorAuth	int	O	Support single door auth type, 0 : not support, 1 : support, (This param is not used by video talk device)
+AsynAuth	int	O	Support async auth type, 0 : not support, 1 : support, (This param is not supported by video talk device)
+SpecialDaysSchedule	object	O	Special days schedule capabilities, (This param is not supported by video talk device)
++Support	bool	O	Support special days schedule or not. (This param is not supported by video talk device)
++MaxSpecialDaysSchedules	uint	O	Max special days schedule number. (This param is not supported by video talk device)
++MaxTimePeriodsPerDay	uint	O	Max time periods per day. (This param is not supported by video talk device)
++MaxSpecialDayGroups	uint	O	Max special day groups. (This param is not supported by video talk device)
++MaxDaysInSpecialDayGroup	uint	O	Max special days in special day group. (This param is not supported by video talk device)
[Example]			
Request	GET http://192.168.1.108/cgi-bin/accessControlManager.cgi?action=getCaps		

Response	caps.AccessControlChannels=3 caps.AccessControlAlarmRecord=true caps.CustomPasswordEncryption=1 caps.SupportFingerPrint=1 caps.OnlySingleDoorAuth=1 caps.AsynAuth=1 caps.SpecialDaysSchedule.Support=true caps.SpecialDaysSchedule.MaxSpecialDaysSchedules=5 caps.SpecialDaysSchedule.MaxTimePeriodsPerDay=5 caps.SpecialDaysSchedule.MaxSpecialDayGroups=5 caps.SpecialDaysSchedule.MaxDaysInSpecialDayGroup=5

11.2.7 [Config] Access Control General Setting

- Get access control general setting

Table 11-12

URL	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=AccessControlGeneral		
Method	GET		
Description	Get access control general setting.		
[Request Params] (None)			
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
table	object	R	config table object
+AccessProperty	string	O	Access property, can be : “unidirect”, “bidirect” (This param is supported by access control device.)
+ABLock	object	O	AB Lock setting (This param is supported by access control device.)
++Enable	bool	O	Enable AB Lock or not.
++Doors	Array< Array<int> >	O	AB Lock groups, each group has several doors, one door can be opened only when all other doors in the AB lock group are in closed state.
+CustomPasswordEnable	bool	O	Whether to enable custom password. (This param is supported by video talk device.)
+CommonPassword	string	O	The common password. (This param is supported by video talk device.)
+ButtonExitEnable	bool	O	Whether to enable the open door button. (This param is supported by video talk device.)
+CheckSensorBeforeLock	bool	O	Whether to check the sensor before lock the door. (This param is supported by video talk device.)
+CheckSensorTime	int	O	The check sensor time, in seconds. (This param is supported by video talk device.)
+DuressPassword"	string	O	The duress password. (This param is supported by video talk device.)
+DuressEnable	bool	O	Whether to enable duress password. (This param is supported by video talk device.)

[Example]	
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=AccessControlGeneral
Response	table.AccessControlGeneral.AccessProperty=bidirect table.AccessControlGeneral.ABLock.Enable=true table.AccessControlGeneral.ABLock.Doors[0][0]=1 table.AccessControlGeneral.ABLock.Doors[0][1]=2 table.AccessControlGeneral.ABLock.Doors[0][2]=3 table.AccessControlGeneral.ABLock.Doors[1][0]=4 table.AccessControlGeneral.ABLock.Doors[1][1]=5 table.AccessControlGeneral.ABLock.Doors[1][2]=6 table.AccessControlGeneral.CustomPasswordEnable=true table.AccessControlGeneral.CommonPassword=123456 table.AccessControlGeneral.ButtonExitEnable=true table.AccessControlGeneral.CheckSensorBeforeLock=true table.AccessControlGeneral.CheckSensorTime=30 table.AccessControlGeneral.DuressPassword=654321 table.AccessControlGeneral.DuressEnable=true

- Set access control general setting

Table 11-13

URL	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set access control general setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
AccessControlGeneral	object	R	AccessControlGeneral config object
+AccessProperty	string	O	Access property, can be : “unidirect”, “bidirect” (This param is supported by access control device.)
…<other param>	—	—	…<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&AccessControlGeneral.AccessProperty=bidirect&AccessControlGeneral.ABLock.Enable=false&AccessControlGeneral.CustomPasswordEnable=true&AccessControlGeneral.CommonPassword=123456&AccessControlGeneral.ButtonExitEnable=true&AccessControlGeneral.CheckSensorBeforeLock=true		
Response	OK		

11.2.8 [Config] Access Control Setting

- Get access control setting

Table 11-14

URL	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=AccessControl		
Method	GET		
Description	Get access control setting.		
[Request Params] (None)			
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
table	array<object>	R	The config table array for every access control channel
+Enable	bool	O	Whether to enable config for this channel.
+State	string	O	Door state, can be : “Normal”, “CloseAlways”, “OpenAlways”, “NoPersonNC”, “NoPersonNO”
+Method	int	O	Open door method, can be following value, default is 2 : 0 : only by password 1 : only by access card 2 : by password or access card 3 : by access card first then password 4 : by password first then access card 5 : different method in differenct time range 6 : only by fingerprint 7 : by password or access card or fingerprint 8 : by password and access card and fingerprint together 9 : by password and fingerprint together 10 : by access card and fingerprint together 11 : by multiple access user 12 : by face (match with picture in ID card) 13 : by face and ID card 14 : by face (match with picture in ID card) or access card or fingerprint 15 : by face and ID card, or access card or fingerprint 16 : by UserID and password 17 : only by face 18 : by face and password together 19 : by fingerprint and password together 20 : by fingerprint and face together 21 : by access card and face together 22 : by face or password 23 : by fingerprint or password 24 : by fingerprint or face 25 : by access card or face 26 : by access card or fingerprint 27 : by fingerprint and face and password together 28 : by access card and face and password together 29 : by access card and fingerprint and password together 30 : by access card and fingerprint and face together 31 : by fingerpint or face or password 32 : by access card or face or password 33 : by access card or fingerprint or face 34 by access card and fingerprint and face and password

			together 35 : by access card or fingerprint or face or password 36 : by face and ID card, or access card or face 37 : by face (match with picture in ID card) or access card or face 38 : by access card and password, or fingerprint and password
+OpenAlwaysTime	int	O	The time range that working, value is index in AccessTimeSechdule config.
+CloseAlwaysTime	int	O	The time range that always closed, value is index in AccessTimeSechdule config.
+HolidayTime	int	O	The time range that working in holidays, value is index in AccessTimeSechdule config.
+UnlockHoldInterval	int	O	The Unlock holding interval, unit is milliseconds, value should between 250 to 20000. (This param is also supported by video talk device.)
+UnlockReloadInterval	int	O	The Unlock reload interval, unit is milliseconds. (This param is also supported by video talk device.)
+AccessProtocol	string	O	The access control process, can be : "Local", "Dahua", "Remote", "Private" (This param is also supported by video talk device.)
+BreakInAlarmEnable	bool	O	Whether to enable the breakin alarm.
+RepeatEnterAlarm	bool	O	Whether to enable the repeat enter alarm.
+DoorNotClosedAlarmEnable	bool	O	Whether to enable the alarm when door not closed.
+DuressAlarmEnable	bool	O	Whether to enable the duress alarm.
+FirstEnter	object	O	The first enter setting.
++Enable	bool	O	Whenther to enable the first enter function,
++Status	string	O	The status, can be : “KeepOpen” “Normal”
++Time	int	O	The time range that enable first enter function, value is index in AccessTimeSechdule config.
+CardNoConvert	int	O	The card number convert setting : 0 – not convert, 1 – convert using NOT operation, 2 : use HIDpro convert
+MaliciousAccessControlEnable	bool	O	Whether to enable malicious access alarm.
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=AccessControl		
Response	table.AccessControl[0].Enable=true table.AccessControl[0].State=Normal table.AccessControl[0].Method=2 table.AccessControl[0].OpenAlwaysTime=1 table.AccessControl[0].CloseAlwaysTime=2 table.AccessControl[0].UnlockHoldInterval=500		

	table.AccessControl[0].AccessProtocol=Local table.AccessControl[0].BreakInAlarmEnable=true table.AccessControl[0].DuressAlarmEnable=true table.AccessControl[0].CardNoConvert=0 table.AccessControl[0].MaliciousAccessControlEnable=true …

- Set access control general setting

Table 11-15

URL	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set access control setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
AccessControl	array<object>	R	AccessControl config object array
+Enable	bool	O	Whether to enable config for this channel.
…<other param>	—	—	…<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&AccessControl[0].Enable=true&AccessControl[0].State=Normal&AccessControl[0].Method=2&AccessControl[0].OpenAlwaysTime=1&AccessControl[0].CloseAlwaysTime=2&AccessControl[0].UnlockHoldInterval=500&AccessControl[0].AccessProtocol=Local&AccessControl[0].BreakInAlarmEnable=true&AccessControl[0].DuressAlarmEnable=true&AccessControl[0].CardNoConvert=0&AccessControl[0].MaliciousAccessControlEnable=true		
Response	OK		

11.2.9 [Config] Wiegand Setting

- Get Wiegand setting

Table 11-16

URL	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=Wiegand		
Method	GET		
Description	Get Wiegand setting.		
[Request Params] (None)			
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
table	object	R	config table object
+Mode	int	R	Wiegand mode, 1 : wiegand input, 2 : wiegand output
+PulseWidth	int	R	The pulse width, unit is microseconds
+PulseStep	int	R	The pulse step, unit is microseconds
+TransferMode	int	R	The transfer mode, 0 : wiegand 34bit transfer, 4 byte card

			number, 2 bit checksum, 1 : wiegand 66bit transfer, 8 byte card number, 2 bit checksum, 2 : wiegand 26bit transfer, 3 byte card number, 2 bit checksum,
+OutType	bool	R	The output type, 0 : output ID, 1 : output card number
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=Wiegand		
Response	table.Wiegand.Mode=1 table.Wiegand.PulseWidth=200 table.Wiegand.PulseStep=1000 table.Wiegand.TransferMode=1 table.Wiegand.OutType=1		

- Set Wiegand setting

Table 11-17

URL	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set Wiegand setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
Wiegand	object	R	Wiegand config object
+Mode	int	O	Wiegand mode, 1 : wiegand input, 2 : wiegand output
…<other param>	—	—	…<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&Wiegand.Mode=1&Wiegand.PulseWidth=200&Wiegand.PulseStep=1000&Wiegand.TransferMode=1&Wiegand.OutType=1		
Response	OK		

11.2.10 [Config] Access Time Schedule Setting

The following Apis are not supported by video talk device.

- Get access time schedule setting

Table 11-18

URL	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=AccessTimeSchedule		
Method	GET		
Description	Get access time schedule setting.		
[Request Params] (None)			
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
table	array<object>	R	config table object array
+Name	string	O	The schedule name, max string length is 63.

+Enable	bool	R	Whether to enable this time schedule
+TimeSechdule	array< array< string > >	O	The time schedule array, first array has max 8 iterm, refer to 7 day in a week and last one is holiday, each item is an array has max 6 time section, each time section is a string, format is “enable, hour:minite:second –hour:minite:second”, for example : "1 00:00:00-12:00:00"
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=AccessTimeSchedule		
Response	table.AccessTimeSchedule[0].Name=TS1 table.AccessTimeSchedule[0].Enable=true table.AccessTimeSchedule[0].TimeSechdule[0][0]=1 00:00:00-12:00:00 table.AccessTimeSchedule[0].TimeSechdule[0][1]=1 15:00:00-20:00:00 table.AccessTimeSchedule[0].TimeSechdule[1][0]=1 00:00:00-12:00:00 table.AccessTimeSchedule[0].TimeSechdule[1][1]=1 15:00:00-20:00:00 …		

- Set access time schedule setting

Table 11-19

URL	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set access time schedule setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
AccessTimeSchedule	array<object>	R	The access time schedule config object array
+Name	string	O	The schedule name, max string length is 63.
…<other param>	—	—	…<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&AccessTimeSchedule[0].Name=TS1&AccessTimeSchedule[0].Enable=true&AccessTimeSchedule[0].TimeSechdule[0][0]=1%2000:00:00-12:00:00&AccessTimeSchedule[0].TimeSechdule[0][1]=1%2015:00:00-20:00:00&AccessTimeSchedule[0].TimeSechdule[1][0]=1%2000:00:00-12:00:00&AccessTimeSchedule[0].TimeSechdule[1][1]=1%2015:00:00-20:00:00		
Response	OK		

11.2.11 [Config] Special Day Group Setting

The following Apis are not supported by video talk device.

- Get special day group setting

Table 11-20

URL	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=SpecialDayGroup

Method	GET		
Description	Get special day group setting.		
[Request Params] (None)			
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
table	array<object>	R	config table object array
+Name	string	O	The special day group name
+Enable	bool	R	Whether to enable this special day group
+Days	array<object>	O	The special days in group
++SpecialDayName	string	O	The special day name.
++StartTime	string	O	The special day start datetime
++EndTime	string	O	The special day end datetime
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=SpecialDayGroup		
Response	table.SpecialDayGroup[0].Name=SpecialDayGroup1 table.SpecialDayGroup[0].Enable=true table.SpecialDayGroup[0].Days[0].SpecialDayName=NationalDay table.SpecialDayGroup[0].Days[0].StartTime=2017-10-01 00:00:00 table.SpecialDayGroup[0].Days[0].StartTime=2017-10-07 23:59:59 …		

- Set special day group setting

Table 11-21

URL	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set special day group setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
SpecialDayGroup	array<object>	R	The special day group config object array
+Name	string	O	The special day group name
…<other param>	—	—	…<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&SpecialDayGroup[0].Name=SpecialDayGroup1&SpecialDayGroup[0].Enable=true&SpecialDayGroup[0].Days[0].SpecialDayName=NationalDay&SpecialDayGroup[0].Days[0].StartTime=2017-10-01%2000:00:00&SpecialDayGroup[0].Days[0].StartTime=2017-10-07%2023:59:59		
Response	OK		

11.2.12 [Config] Special Days Schedule Setting

The following Apis are not supported by video talk device.

- Get special days schedule setting

Table 11-22

URL	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=SpecialDaysSchedule		
Method	GET		
Description	Get special days schedule setting.		
[Request Params] (None)			
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
table	array<object>	R	config table object array
+Name	string	O	The special day schedule name
+Enable	bool	R	Whether to enable this special day schedule
+GroupNo	int	R	The special day group number, value is the index of SpecialDayGroup setting array.
+TimeSection	array<string>	R	The time section array, the format of the item is “enable, hour:minite:second – hour:minite:second”, for example : "1 00:00:00-12:00:00".
+Doors	array<int>	R	The doors array.
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=SpecialDaysSchedule		
Response	table.SpecialDaysSchedule[0].Name=SpecialDayGroup1 table.SpecialDaysSchedule[0].Enable=true table.SpecialDaysSchedule[0].GroupNo=1 table.SpecialDaysSchedule[0].TimeSection[0]=1 00:00:00-12:00:00 table.SpecialDaysSchedule[0].TimeSection[1]=1 15:00:00-20:00:00 table.SpecialDaysSchedule[0].Doors[0]=2 table.SpecialDaysSchedule[0].Doors[1]=3 …		

- Set special days schedule setting

Table 11-23

URL	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set special days schedule setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
SpecialDaysSchedule	array<object>	R	The special days schedule config object array
+Name	string	O	The special day schedule name
…<other param>	—	—	…<See above getConfig command for other params of the config object, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&SpecialDaysSchedule[0].		

	Name=SpecialDayGroup1&SpecialDaysSchedule[0].Enable=true&SpecialDaysSchedule[0].GroupNo=1&SpecialDaysSchedule[0].TimeSection[0]=1%2000:00:00-12:00:00&SpecialDaysSchedule[0].TimeSection[1]=1%2015:00:00-20:00:00&SpecialDaysSchedule[0].Doors[0]=2&SpecialDaysSchedule[0].Doors[1]=3
Response	OK

12 Intelligent Building APIs

12.1 Video Talk

The following Apis are supported by VTT / VTA products.

12.1.1 Subscribe Video Talk Status

Table 12-1

Syntax	http://<*server*>/cgi-bin/VideoTalkPeer.cgi?action=attachState
Method	GET
Description	Subscribe the video talk status. When client disconnect, it will unsubscribe.
Example	http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=attachState
Success Return	Notify the state: SID=315 state.**State**=Answer state.Talkback.Pack=RTP state.Talkback.Protocol=UDP state.Talkback.Type=Talk state.Talkback.Audio.AudioPort=6000 state.Talkback.Audio.Format[0].Compression=PCM state.Talkback.Audio.Format[0].Frequency=44000 state.Talkback.Audio.Format[0].Depth=16 state.Talkback.Audio.Format[1].Compression=G.711A state.Talkback.Audio.Format[1].Frequency=44000 state.Talkback.Audio.Format[1].Depth=16 state.Talkback.Video.VideoPort=7000 state.Talkback.Video.Format[0].Compression=H.264 state.Talkback.Video.Format[0].Frequency=90000 state.Talkback.Video.Format[1].Compression=MJPG … state.Talkback. MediaAddr=224.10.10.10
Comment	Parameters in Response: **State**: in range of {"Ringing", "Inviting", "Answer", "Refuse", "Cancel", "Hangup", "Busying" }

12.1.2 Unsubscribe Video Talk Status

Table 12-2

Syntax	http://<*server*>/cgi-bin/VideoTalkPeer.cgi?action=detachState&SID=<***sid***>
Method	GET
Description	Unsubscribe the video talk status.

Example	http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=detachState&SID=101
Success Return	OK
Comment	Parameters in URL: ***sid***: the subscribe id, which is the response of attachState

12.1.3 Invite Server on Video Talk

Table 12-3

Syntax	http://<*server*>/cgi-bin/VideoTalkPeer.cgi?action=invite[&Talkback.Protocol=<***protocol***>&Talkback.Type=<***type***>&Talkback.MediaAddr=<***addr***>…]
Method	GET
Description	Start the video talk conversation.
Example	http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=invite&Talkback.Protocol=UDP&Talkback.Type=Talk&Talkback.MediaAddr=224.10.10.10
Success Return	OK
Comment	Parameters in URL: ***protocol***: the transmit protocol ***type***: video talk type. ***addr***: addr to get stream

12.1.4 Cancel the Video Talk

Table 12-4

Syntax	http://<*server*>/cgi-bin/VideoTalkPeer.cgi?action=cancel
Method	GET
Description	Cancel video talk conversation.
Example	http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=cancel
Success Return	OK
Comment	—

12.1.5 Answer the Invitation

Table 12-5

Syntax	http://<*server*>/cgi-bin/VideoTalkPeer.cgi?action=answer&Talkback.Protocol=<***protocol***>&Talkback.Type=<***type***>&Talkback.MediaAddr=<***addr***>…
Method	GET
Description	Answer the call.
Example	http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=answer&Talkback.Protocol=UDP&Talkback.Type=Talk&Talkback.MediaAddr=224.10.10.10
Success Return	OK
Comment	Parameters in URL: ***protocol***: the transmit protocol

	type: video talk type. ***addr***: addr to get stream

12.1.6 Refuse to Answer the Video Talk Invitation

Table 12-6

Syntax	http://<*server*>/cgi-bin/VideoTalkPeer.cgi?action=refuse
Method	GET
Description	Refuse answer the call.
Example	http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=refuse
Success Return	OK
Comment	—

12.1.7 Hang Up

Table 12-7

Syntax	http://<*server*>/cgi-bin/VideoTalkPeer.cgi?action=hangup
Method	GET
Description	Close it when the conversation is over.
Example	http://192.168.1.108/cgi-bin/VideoTalkPeer.cgi?action=hangup
Success Return	OK
Comment	—

12.2 Video Talk Log

The following Apis are supported by VTO products.

12.2.1 Query Video Talk Log

Table 12-8

Syntax	http://<*server*>/cgi-bin/recordFinder.cgi?action=find&name=VideoTalkLog[&condition.CallType=<***Type***>&condition.EndState=<***State***>&count=<***countNo***>]
Method	GET
Description	Find the VideoTalkLog record.
Example	http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=VideoTalkLog&condition.CallType=Incoming&condion.EndState=Missed&count=500
Success Return	**totalCount**=1000 **found**=500 records[0].RecNo=789 records[0].CreateTime=123456789 records[0].**CallType**=Incoming records[0].**EndState**=Received records[0].PeerNumber=501 …

Comment	Parameters in URL: ***Type***: call type ***State***: end state of the call ***countNo***: the number of records to get Parameters in Response : **totalCount** : the record count which match condition **found** : the record count to return **CallType**: call type. The range is {"Incoming", "Outgoing"}. **EndState**: the range is {"EndState" , "Received"}

12.3 Announcement Record

The following Apis are supported by VTO products.

12.3.1 Insert Record

Table 12-9

Syntax	http://<*server*>/cgi-bin/recordUpdater.cgi?action=insert&name=Announcement&Content=<***Content***>&ExpirTime=<***ExpirTime***>&IssueTime=<***IssueTime***>&Title=<***Title***>&User=<***User***>&State=<***State***>&ReadFlag=<***ReadFlag***>
Method	GET
Description	Insert the Announcement record.
Example	http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insert&name=Announcement&Content=stringData&ExpirTime=2012-01-01%2012:00:00&IssueTime=2012-01-01%2012:00:00&Title=Anounce1&User=101&State=0&ReadFlag=0
Success Return	RecNo=<***RecNo***>
Comment	Parameters in URL: ***Content***: Announcement Content ***ExpirTime***: the time when the Announcement expire, format: 2012-01-01%2012:00:00 ***IssueTime***: Announcement issue time, format: 2012-01-01%2012:00:00 ***Title***: title of the announcement ***User***: the number the Announcement issued to ***State***: the state of the Announcement. 0 init, 1 send , 2 overdue ***ReadFlag***: the read flag. 0 not read, 1 read. Parameters in Response : ***RecNo***: the record index of the new record

12.4 Alarm Record

12.4.1 Query Alarm Record

Table 12-10

Syntax	http://<*server*>/cgi-bin/recordFinder.cgi?action=find&name=AlarmRecord[&StartTime=<***startTime***>&EndTime=<***endTime***>&count=<***countNo***>]

Method	GET
Description	Find the AlarmRecord record. (This api is supported by video talk device.)
Example	http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AlarmRecord&StartTime=2014-8-25%2000:02:32&EndTime=2014-8-25%2001:02:32&count=500
Success Return	**totalCount**=1000 **found**=500 records[0].RecNo=789 records[0].CreateTime=123456789 records[0].Channel=0 records[0].**SenseMethod**=DoorMagnetism records[0].RoomNumber=501 records[0].ReadFlag=0 records[0].**Comment**=Friend …
Comment	Parameters in URL: ***startTime***: The start time ,format : 2014-8-25%2000:01:32 ***endTime***: The end time, format: 2014-8-25%2000:02:32 ***countNo***: the number of records to get, The record count, default 1024 Parameters in Response : **totalCount** : the record count which match condition **found** : the record count to return **SenseMethod** :the range is { "DoorMagnetism", "PassiveInfrared", "GasSensor", "SmokingSensor", "WaterSensor", "ActiveInfrared", "CallButton", "UrgencyButton", "Steal", "Perimeter", "PreventRemove", "DoorBell" }

12.4.2 Query Access Control Alarm Record

Table 12-11

Syntax	http://<*server*>/cgi-bin/recordFinder.cgi?action=find&name=AccessControlAlarmRecord[&StartTime=<***startTime***>&EndTime=<***endTime***>&count=<***countNo***>]
Method	GET
Description	Find the AccessControlAlarmRecord record. (This api is supported by access control device.)
Example	http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlAlarmRecord&StartTime=2014-8-25%2000:02:32&EndTime=2014-8-25%2001:02:32&count=500
Success Return	**totalCount**=1000 **found**=500 records[0].RecNo=789 records[0].CreateTime=123456789 records[0].UserID=10113 records[0].**EventCode**=DoorMagnetism records[0].**DevAddrs**=1 records[0].IndexNum=0 records[0].Time=2017-05-10 16:00:01 …

Comment	Parameters in URL: ***startTime***: The start time ,format : 2014-8-25%2000:01:32 ***endTime***: The end time, format: 2014-8-25%2000:02:32 ***countNo***: the number of records to get, The record count, default 1024 Parameters in Response : **totalCount** : the record count which match condition **found** : the record count to return **EventCode** : alarm event code, the range is { "DoorNotClosed", "BreakIn", "RepeatEnter", "Duress", "AlarmLocal", "ChassisIntruded" }

13 DVR Custom APIs

13.1 File Finder

13.1.1 Create a File Finder

Table 13-1

Syntax	http://<server>/cgi-bin/FileFindHelper.cgi?action=startFind&condition.channel=<***ChannelNo***>&condition.startTime=<***start***>&condition.endTime=<***end***>&condition.streamType=<***stream***>[&condition.flags[0]=<***flag***>&condition.events[0]=<***event***>&combineMode.granularity=<***granularityValue***>&combineMode.types[0]=<***combineType***>]
Method	GET
Description	Start to find files.
Example	Find a file in channel 1. Event type is "AlarmLocal" or "VideoMotion". Time lines between 2014-1-1 12:00:00 and 2015-1-10 12:00:00. Combine "AlarmLocal" or "VideoMotion" files with granularity 16. URL is: http://172.23.1.66/cgi-bin/fileFindHelper.cgi?action=startFind&condition.channel=1&condition.startTime=2014-1-1%2012:00:00&condition.endTime=2015-1-10%2012:00:00&condition.streamType=Main&condition.flags[0]=Event&condition.events[0]=AlarmLocal&condition.events[1]=VideoMotion&combineMode.granularity=16&combineMode.types[0]=AlarmLocal&combineMode.types[0]=VideoMotion
Success Return	result=08137
Comment	Start to find a file with the above condition and combine files with certain type. If it succeeds, return to find id. Otherwise return to Error. Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. ***start*** / ***end***: the start/end time when recording. ***flag***: which flags of the file you want to find. It is an array. The index starts from 0. The range of flag is {"Timing", "Marked", "Event", "Restrict"}. If omitted, find files with all the flags. ***event***: by which event the record file is triggered. It is an array. The index starts from 0. The range of event is {"AlarmLocal", "VideoMotion", "VideoLoss"}. This condition can be omitted. If omitted, find files of all the events. ***stream***: which video stream type you want to find. The range of stream is {"Main", "Extra1", "Extra2", "Extra3"}. ***combineType***: which types of the file you want to combined. It is an array. The index starts from 0. The range of combine type is {"AlarmLocal", "VideoMotion", "Timing", "VideoLoss"}. This condition can be omitted. If omitted, file will not be combined. ***granularityValue***: by which granularity to combine files Example: File 1:

	items[0]. Channel =1 items[0]. StartTime =2011-1-1 12:00:00 items[0]. EndTime =2011-1-1 13:00:00 items[0]. Events[0]=AlarmLocal items[0]. VideoStream=Main items[0]. Length =790 items[0]. Duration = 3600 File 2: items[0]. Channel =1 items[0]. StartTime =2011-1-1 13:00:00 items[0]. EndTime =2011-1-1 14:00:00 items[0]. Events[0]=AlarmLocal items[0]. VideoStream=Main items[0]. Length =790 items[0]. Duration = 3600 file1 and file2 will be combined to file3 File 3: items[0]. Channel =1 items[0]. StartTime =2011-1-1 12:00:00 items[0]. EndTime =2011-1-1 14:00:00 items[0]. Events[0]=AlarmLocal items[0]. VideoStream=Main items[0]. Length =1580 items[0]. Duration = 7200

13.1.2 Create a Motion File Finder

Table 13-2

Syntax	http://<server>/cgi-bin/FileFindHelper.cgi?action=startMotionFind&condition.channel=<***ChannelNo***>&condition.startTime=<***start***>&condition.endTime=<***end***>&condition.streamType=<***stream***>&motionRegion.senseLevel=<***level***>[&motionRegion.rects[***rectNo***][0]=<***rect0***>&motionRegion.rects[***rectNo***][1]=<***rect1***>&motionRegion.rects[***rectNo***][2]=<***rect2***>&motionRegion.rects[***rectNo***][3]=<***rect3***>]
Method	GET
Description	Start to find motion files.
Example	Find a file in channel 1, event type is "AlarmLocal" or "VideoMotion", and time between 2014-1-1 12:00:00 and 2015-1-10 12:00:00, motion region is [0,0,21,17] URL is: http://172.23.1.66/cgi-bin/fileFindHelper.cgi?action=startMotionFind&condition.channel=1&condition.startTime=2014-1-1%2012:00:00&condition.endTime=2015-1-10%2012:00:00&condition.streamType=Main&condition.flags[0]=Event&condition.events[0]=AlarmLocal&condition.events[1]=VideoMotio&motionRegion.senseLevel=1&motionRegion.rects[1][0]=0&motionRegion.rects[1][1]=0&motionRegion.rects[1][2]=21&motionRegion.rects[1][3]=17
Success	result=08137

Return	
Comment	Start to find a file with the above condition and combine files with certain type. If success, return find id, else return Error. Parameters in URL: ***ChannelNo***: integer, video channel index which starts from 1. ***start*** / ***end***: the start/end time when recording. ***flag***: which flags of the file you want to find. It is an array. The index starts from 0. The range of flag is {"Timing", "Marked", "Event", "Restrict"}. If omitted, find files with all the flags. ***event***: by which event the record file is triggered. It is an array. The index starts from 0. The range of event is {"AlarmLocal", "VideoMotion"}. This condition can be omitted. If omitted, find files of all the events. ***stream***: which video stream type you want to find. The range of stream is {"Main", "Extra1", "Extra2", "Extra3"}. ***level****:* the motion sensitive level, range is 0–6, 0 represent all level ***rectNo****:* the rects array index, start from 1 ***rect0*** & ***rect1*** & ***rect2*** & ***rect3*** : relative coordinates, rect0 and rect2 range is 0—21, rect1 and rect3 range is 0—17. {0,0,0,0} top—left, {21,0,0,0} top—right, {0,17,0,0} bottom—left, {21,17,0,0} bottom—right

13.1.3 Get the File Information Found by the Finder

Table 13-3

Syntax	http://<server>/cgi-bin/FileFindHelper.cgi?action=findNext&findId=<***findId***>&count=<***fileCount***>
Method	GET
Description	Find the next files no more than *fileCount* number.
Example	http://192.168.1.108/cgi-bin/FileFindHelper.cgi?action=findNext&findId=08137&count=100
Success Return	found=1 items[0]. channel =1 items[0]. startTime =2011-1-1 12:00:00 items[0]. endTime =2011-1-1 13:00:00 items[0]. fileType =dav items[0]. events[0]=AlarmLocal items[0]. streamType=Main items[0]. length =790 items[0]. duration = 3600
Comment	***findId****:* The find Id is created by API Create a file finder or API Create a motion file finder. Must create a finder before finding files.

13.1.4 Stop the Finder

Table 13-4

Syntax	http://<server>/cgi-bin/FileFindHelper.cgi?action=stopFind&findId=<***findId***>
Method	GET
Description	Stop the searching operation.

Example	http://192.168.1.108/cgi-bin/FileFindHelper.cgi?action=stopFind&findId =08137
Success Return	OK
Comment	***findId**:* The find Id is created by API Create a file finder or API Create a motion file finder. Must create a finder before finding files.

13.1.5 Get Bound Files

Table 13-5

Syntax	http://<server>/cgi-bin/FileFindHelper.cgi?action=getBoundFile&condition.channel=<***ChannelNo***>&condition.startTime=<***start***>&condition.endTime=<***end***>&condition.streamType=<***stream***>[&condition.flags[0]=<***flag***>&condition.events[0]=<***event***>]
Method	GET
Description	Get bound files.
Example	http://<server>/cgi-bin/FileFindHelper.cgi?action=getBoundFile&condition.channel=1&condition.startTime=2014-1-1%2012:00:00&condition.endTime=2015-1-10%2012:00:00&condition.streamType=Main&condition.flags[0]=Timing
Success Return	found=2 items[0]. channel =1 items[0]. startTime =2011-1-1 12:00:00 items[0]. endTime =2011-1-1 13:00:00 items[0]. flags [0]= Timing items[0]. streamType=Main items[0]. length =790 items[0]. duration = 3600 items[1]. channel =1 items[1]. startTime =2011-1-1 13:00:00 items[1]. endTime =2011-1-1 14:00:00 items[1]. events[0]= Timing items[1]. streamType=Main items[1]. length =790 items[1]. duration = 3600
Comment	Params is same as FileFindHelper. startFind

13.2 BandLimit

13.2.1 Get Bandwidth Limit State

Table 13-6

Syntax	http://<*server*>/cgi-bi*n*/BandLimit.cgi?action=getLimitState
Method	GET
Description	Get bandwidth limit state.
Example	http://192.168.1.108/cgi-bin/bandLimit.cgi?action=getLimitState
Success Return	limit=true
Comment	—

13.3 Record Files Protection

13.3.1 Add Protection

Table 13-7

Syntax	http://<*server*>/cgi-bin/FileManager.cgi?action=addConditionList&condition.Types[0]=<***paramValue***>&condition.StartTime=<***paramValue***>&condition.EndTime=<***paramValue***>&condition.Channel[0]=<***paramValue***>
Method	GET
Description	Add protection or access control for record files.
Example	http://192.168.1.108/cgi-bin/FileManager.cgi?action=addConditionList&condition.Types[0]=RecordRestrict&condition.Types[1]=RecordProtect&condition.StartTime=2014-7-3%2021:02:32&condition.EndTime=2014-7-3%2023:02:32&condition.Channel[0]=1&condition.Channel[1]=3
Success Return	OK
Comment	In table below: ***TypeIndex***: The index of type array ***ChIndex***: The index of channel number array

Appendix:

ParamName	ParamValue type	Description
condition.Type[***TypeIndex***]	string	An array. The range is {"RecordProtect", "RecordRestrict"}.
condition.StartTime	string	The time format is "Y-M-D H-m-S", example 2011-7-3%2021:02:32
condition.EndTime	string	The time format is "Y-M-D H-m-S"
condition.Channel*[**ChIndex**]*	integer	Video channel index which starts from 1.

13.3.2 Cancel Protection

Table 13-8

Syntax	http://<*server*>cgi-bin/FileManager.cgi?action=cancelConditionList&condition.Types[0]=<***paramValue***>&condition.StartTime=<***paramValue***>&condition.EndTime=<***paramValue***>&condition.Channel[0]=<***paramValue***>
Method	GET
Description	Cancel protection of record files.
Example	http://192.168.1.108/cgi-bin/FileManager.cgi?action=cancelConditionList&condition.Types[0]=RecordRestrict&condition.Types[1]=RecordProtect&condition.StartTime=2014-7-3%2021:02:32&condition.EndTime=2014-7-3%2023:02:32&condition.Channel[0]=1&condition.Channel[1]=3
Success Return	OK
Comment	***paramValue*** as Appendix above.

13.3.3 Remove Protection

Table 13-9

Syntax	http://<*server*>/cgi-bin/FileManager.cgi?action=removeConditionList&condition.Types[0]=<***paramValue***>&condition.StartTime=<***paramValue***>&condition.EndTime=<***paramValue***>&condition.Channel[0]=<***paramValue***>
Method	GET
Description	Remove protection of record files.
Example	http://192.168.1.108/cgi-bin/FileManager.cgi?action=removeConditionList&condition.Types[0]=RecordRestrict&condition.Types[1]=RecordProtect&condition.StartTime=2014-7-3%2021:02:32&condition.EndTime=2014-7-3%2023:02:32&condition.Channel[0]=0&condition.Channel[1]=3
Success Return	OK
Comment	***paramValue*** as Appendix above.

13.4 Get Daylight

Table 13-10

Syntax	http://<*server*>/cgi-bin/global.cgi?action=getDST
Method	GET
Description	Get daylight saving time state.
Example	http://192.168.1.108/cgi-bin/global.cgi?action=getDST
Success Return	result = 1
Comment	***result****:* 1/0, yes or not in daylight saving time

14.1 Coaxial Control IO

14.1.1 Control White Light or Speaker

- control white light or speaker

Table 14-1

Syntax	http://<server>/cgi-bin/coaxialControlIO.cgi?action=control&[channel=<***channelno***>]&[info[0].Type=<***type***>]&[info[0].IO=<*io*>
Method	GET
Description	Control white light or speaker
Example	http://192.168.1.108/cgi-bin/coaxialControlIO.cgi?action=control&channel=1&info[0].Type=1&info[0].IO=0
Success Return	status.whitelight=on status.speaker=on
Comment	***channelno****:* video channel index for white light,for siren it's audio output channel, The channel number is default 0 if the request is not carried the param ***io****: enable or disable the function*

Appendix:

ParamName	ParamValue type	Description
type	integer	1:wight light, 2:speaker
io	bool	0:close, 1:open

- get white light and speaker status

Table 14-2

Syntax	http://<server>/cgi-bin/coaxialControlIO.cgi?action= getstatus&[channel=<***channelno***>]
Method	GET
Description	Get white light and speaker stauts
Example	http://192.168.1.108/cgi-bin/coaxialControlIO.cgi?action= getStatus&channel=1
Success Return	status.whitelight=on status.speaker=on
Comment	***channelno****:* video channel index

14.2 Pir Alarm

14.2.1 Configure Pir Parameter

- Get pir parameter

Table 14-3

Syntax	http://<*server*>/cgi-bin/pirAlarm.cgi?action=getPirParam&[channel=<***channelNo***>]

Method	GET
Description	Get pir parameter.
Example	http://192.168.1.108/cgi-bin/pirAlarm.cgi?action=getPirParam&channel=1
Success Return	***head***.Enable=true ***head***.PirLink.LightingLink.Enable=true ***head***.PirLink.LightingLink.LightLinkType=Filcker ***head***.PirLink.LightingLink.FilckerIntevalTime=5 ***head***.PirLink.LightingLink.LightDuration=10 ***head***.PirLink.LightingLink.WhiteLightTimeSection=TimeSection ***head***.PirLink.TimeSection[***weekday***][0]=1 00:00:00-24:00:00 ***head***.PirLink.TimeSection[***weekday***][1]=0 02:00:00-24:00:00 ***head***.PirLink.TimeSection[***weekday***][2]=0 03:00:00-24:00:00 ***head***.PirLink.TimeSection[***weekday***][3]=0 04:00:00-24:00:00 ***head***.PirLink.TimeSection[***weekday***][4]=0 05:00:00-24:00:00 ***head***.PirLink.TimeSection[***weekday***][5]=0 06:00:00-24:00:00 ***head***.RecordEnable=true ***head***.RecordChannels=[0, 1, 2] ***head***.RecordLatch=10 ***head***.AlarmOutEnable=true ***head***.AlarmOutChannels=[1, 4] ***head***.AlarmOutLatch=10 ***head***.SnapshotEnable=true ***head***.SnapshotChannels=[2, 4] ***head***.MailEnable=true ***head***.AlarmBellEnable=true ***head***.AlarmBellLatch=10 ***head***.Dejitter=0 ***head***.LogEnable=true ***head***.DetectWindow[0].Level=3 ***head***.DetectWindow[0].Id=0 ***head***.DetectWindow[0].Name=Region0 ***head***.DetectWindow[0].Sensitive=58 ***head***.DetectWindow[0].Threshold=4 ***head***.DetectWindow[0].Region[0]=3932160 ***head***.DetectWindow[0].Region[1]=3932160 … … ***head***. DetectWindow [1]…
Comment	if a device has one or more pir, PirParam is a array ***channel*** *:* video channel index ***weekday*** *:* range is [0-6] (Sunday - Saturday) ***head***=configEx[channelNo]

- set pir parameter

Table 14-4

Syntax	http://<server>/cgi-bin/pirAlarm.cgi?action=setPirParam&[channel=<***channelNo***>]&<***paramName***>=<***paramValue***>...&<***paramName***>=<***paramValue***>

Method	GET
Description	Set pir parameter.
Example	http://192.168.1.108/cgi-bin/pirAlarm.cgi?action=setPirParam&channel=1&configEx[1].Enable=true&configEx[1].PirLink.LightingLink.Enable=true&...
Success Return	OK
Comment	***channel*** *:* video channel index ***head***=configEx[channelNo]

Appendix:

ParamName	ParamValue type	Description
head.Enable	bool	Enable/Disable motion detect feature in a channel.
head.DetectWindow [***WinNum***].Level	integer	*WinNum* Index of detect window, there are 4 detect windows at present. Each window is divided into 18 lines and 22 blocks/line. Range is [1—6]. Sensitivity of pir 1: lowest sensitivity. 6: highest sensitivity.
head.DetectWindow [***WinNum***].Id	integer	It is the Id of a detect window.
head.DetectWindow [***WinNum***].Name	string	It is the name of a detect window.
head.DetectWindow [***WinNum***].Sensitive	integer	Range is [0—100]. It presents more sensitive if the value is larger.
head.DetectWindow [***WinNum***].Threshold	integer	Range is [0—100]. It presents the threshold value when trigger motion detect.

ParamName	ParamValue type	Description
head.DetectWindow[*WinN**um***].Region[***LineNum***]	integer	*LineNum* Index of region, region is divided into lines and each line has several blocks, a line is described by a 32 bit integer, a bit for a block. 0=Line 1 1=Line 2 … … Currently, region is divided into 18 lines and 22 blocks/line. A bit describes a block in the line. Bit = 1: motion in this block is monitored. Example: MotionDetect[0].Region[0] = 4194303 (0x3FFFFF):: motion in channel 0 line 0's 22 blocks is monitored. MotionDetect[0].Region[1] =0: motion in line 1's 22 blocks is not monitored. MotionDetect[0].Region[17] = 3: in the last line of channel 0, motion in the left two blocks is monitored.
head.TimeSection[***wd***][***ts***]	string	*wd* (week day) range is [0—6] (Sunday - Staurday) *ts* (time section) range is [0 — 23], timesection table index. Format: mask hh:mm:ss-hh:mm:ss Mask: [0—65535], hh: [0—24], mm: [0—59], ss: [0—59] Mask indicates record type by bits: Bit0: regular record Bit1: motion detection record Bit2: alarm record Bit3: card record
head. PirLink.RecordChannels[***ch***]	Integer	Range is {0, 1} 0 – do not record on video channel *ch* 1 – record on video channel *ch*
head. PirLink.RecordEnable	bool	Enable/Disable record function.
head. PirLink.RecordLatch	integer	Range is [10—300]. Unit is seconds, indicates the time to record after input alarm is cleared.

ParamName	ParamValue type	Description
head. PirLink.AlarmOutChannels[*ch*]	integer	Range is {0, 1}, *ch* is alarm out channel index. 0 — do not output alarm at alarm out channel *ch* 1 — output alarm at alarm out channel *ch*
head. PirLink.AlarmOutEnable	bool	Enable/Disable alarm out function.
head. PirLink.AlarmOutLatch	Integer	Range is [10—300]. Unit is seconds, indicates the time to output alarm after input alarm is cleared.
head. PirLink.SnapshotChannels[*ch*]	integer	Range is {0, 1} 0 — do not snapshot on video channel *ch* 1 — snapshot on video channel *ch*
head. PirLink.SnapshotEnable	bool	Enable/Disable snapshot function.
head. PirLink.Dejitter	integer	Range is [0—255]. Alarm signal dejitter seconds. Alarm signal change during this period is ignored.
head. PirLink.MailEnable	bool	Enable/Disable mail send for alarm.
head. PirLink.AlarmBellEnable	bool	Enable/Disable mail send for alarm.
head. PirLink.AlarmBellLatch	integer	Range is [10, 300] Unit is seconds
head. PirLink.LogEnable	bool	Enable/Disable log for alarm.

15 Other APIs

15.1 Discover Devices

15.1.1 Discover Devices on Internet

Table 15-1

Syntax	http://<*server*>/cgi-bin/deviceDiscovery.cgi?action=attach[&DeviceClass=<***deviceClass***>]
Method	GET
Description	Discover devices on internet.
Example	http://192.168.1.108/cgi-bin/deviceDiscovery.cgi?action=attach&DeviceClass=VTO
Success Return	deviceInfo[***index***].AlarmInputChannels=8 deviceInfo[***index***].AlarmOutputChannels=0 deviceInfo[***index***].DeviceClass=VTO deviceInfo[***index***].DeviceType=VTO2000A deviceInfo[***index***].HttpPort=80 deviceInfo[***index***].IPv4Address.DefaultGateway=172.12.0.1 deviceInfo[***index***].IPv4Address.DhcpEnable=false deviceInfo[***index***].IPv4Address.IPAddress=172.12.7.102 deviceInfo[***index***].IPv4Address.SubnetMask=255.255.0.0 deviceInfo[***index***].IPv6Address.DefaultGateway=2008::1 deviceInfo[***index***].IPv6Address.DhcpEnable=false deviceInfo[***index***].IPv6Address.IPAddress=2008::6/112 deviceInfo[***index***].Mac=00:01:5b:01:44:77 deviceInfo[***index***].MachineName=YZZ4DZ008D00031 deviceInfo[***index***].Port=37777 deviceInfo[***index***].RemoteVideoInputChannels=0 deviceInfo[***index***].SerialNo=YZZ4DZ008D00031 deviceInfo[***index***].Vendor=Multi deviceInfo[***index***].Version=1.200.0.0 deviceInfo[***index***].VideoInputChannels=1 deviceInfo[***index***].VideoOutputChannels=16
Comment	Parameters in URL: ***deviceClass***: in range of {VTO, VTH, VTT, VTS, VTNC, SHG} Parameters in Response: ***index***: the array index which starts from 0. ***Version***: Software Version

15.2 Flashlight

15.2.1 Flashlight Config

- Get Flashlight Config

Table 15-2

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=getConfig&name=FlashLight
Method	GET
Description	Get Flashlight config. It does not recommend using it.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=getConfig&name=FlashLight
Success Return	***head***.Brightness=50 ***head***.Enable=false ***head***.TimeSection[0][0]=1 00:00:00-23:59:59 ***head***.TimeSection[0][1]=0 00:00:00-23:59:59 … ***head***.TimeSection[6][5]=0 00:00:00-23:59:59
Comment	Parameters in Response: ***head*** = table.FlashLight

- Set flashlight config

Table 15-3

Syntax	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig&<***paramName***>=<***paramValue***>[&<***paramName***>=<***paramValue***>…]
Method	GET
Description	Set Flashlight config.
Example	http://192.168.1.108/cgi-bin/configManager.cgi?action=setConfig&FlashLight.Enable=true&FlashLight.TimeSection[1][0]=1%2012:00:00-18:00:00
Success Return	OK
Comment	—

Appendix:

ParamName	ParamValue type	Description
FlashLight. Enable	bool	Enable
FlashLight. Brightness	integer	Brightness
FlashLight. TimeSection[***wd***][***ts***]	string	It is an effective time period for flash light every day. ***wd*** (week day) range is [0—6] (Sunday—Saturday) ***ts*** (time section) range is [0—23], it's index of time section table. Format: mask hh:mm:ss-hh:mm:ss Mask: {0,1}, hh: [0—24], mm: [00—59], ss: [00—59] Mask 0: this time section is not used. Mask 1: this time section is used. Example: TimeSection[1][0]=1 12:00:00—18:00:00 Means flash light is effective between 12:00:00 and 18:00:00 at Monday.

15.3 Open Platform

15.3.1 Application Start and Stop

- Start Application

Table 15-4

Syntax	http://<*server*>/cgi-bin/installManager.cgi?action=start&appname=<***appname***>&appid=<***appid***>
Method	GET
Description	Start application.
Example	http://192.168.1.108/cgi-bin/installManager.cgi?action=start&appname=FaceDemo&appid=1
Success Return	OK
Comment	Parameters in URL: ***appname***: the application name, ***appid***: optional, the application id

- Stop Application

Table 15-5

Syntax	http://<*server*>/cgi-bin/installManager.cgi?action=stop&appname=<***appname***>&appid=<***appid***>
Method	GET
Description	Stop application.
Example	http://192.168.1.108/cgi-bin/installManager.cgi?action=stop&appname=FaceDemo&appid=1
Success Return	OK
Comment	Parameters in URL: ***appname***: the application name, ***appid***: optional, the application id

15.3.2 Install Application

- Install application with app data

Table 15-6

URL	http://<*server*>/cgi-bin/dhop.cgi?action=uploadApp
Method	POST
Description	Install application.
[Request Params] (None)	
[Response Params] (OK)	
[Example]	
Request	POST /cgi-bin/dhop.cgi?action=uploadApp HTTP/1.1 Host: 172.29.2.176 Content-Length: *413124*

	Content-Type:multipart/form-data;boundary=<*boundary*> --<*boundary*> Content-Disposition: form-data; name="dev_upgrade"; filename="demo.bin" Content-Type: application/octet-stream <app data> --<*boundary*>--
Response	OK

- Install Application with app download url

Table 15-7

URL	http://<*server*>/cgi-bin/dhop.cgi?action=installAppByUrl		
Method	GET		
Description	Install application.		
[Request Params] (key=value format in URL)			
Url	string	R	The app download url
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/dhop.cgi?action=installAppByUrl&Url=https://aaa/bbb/ccc/app.bin		
Response	OK		

15.3.3 Update Application and License

- Update Application with app download url

Table 15-8

URL	http://<*server*>/cgi-bin/dhop.cgi?action=updateAppByUrl		
Method	GET		
Description	Update Application		
[Request Params] (key=value format in URL)			
appName	string	R	The app name
Url	string	R	The app download url
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/dhop.cgi?action=updateAppByUrl&appName=xxx&Url=https://aaa/bbb/ccc/app.bin		
Response	OK		

- Update firmware with firmware download url

Table 15-9

URL	http://<*server*>/cgi-bin/dhop.cgi?action=updateFirmwareByUrl		
Method	GET		
Description	Update firmware		
[Request Params] (key=value format in URL)			
Url	string	R	The firmware download url
[Response Params] (OK)			

[Example]	
Request	GET http://192.168.1.108/cgi-bin/dhop.cgi?action=updateFirmwareByUrl&Url=https://aaa/bbb/ccc/firmware.bin
Response	OK

- Update application license with license download url

Table 15-10

URL	http://<*server*>/cgi-bin/dhop.cgi?action=updateLicenseByUrl		
Method	GET		
Description	Update application license.		
[Request Params] (key=value format in URL)			
appName	string	R	The app name
Url	string	R	The license download url
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/dhop.cgi?action=updateLicenseByUrl&appName=xxx&Url=https://aaa/bbb/ccc/license.bin		
Response	OK		

- Update application license with license data

Table 15-11

URL	http://<*server*>/cgi-bin/dhop.cgi?action=uploadLicense		
Method	POST		
Description	Update application license.		
[Request Params] (key=value format in URL)			
appName	string	R	The app name
[Response Params] (OK)			
[Example]			
Request	POST /cgi-bin/dhop.cgi?action=uploadLicense&appName=xxx HTTP/1.1 Host: 172.29.2.176 Content-Type: application/octet-stream Content-Length: <*length*> <*app license data*>		
Response	OK		

15.3.4 Uninstall Application

URL	http://<*server*>/cgi-bin/dhop.cgi?action=uninstall		
Method	GET		
Description	Uninstall application.		
[Request Params] (key=value format in URL)			
appName	string	R	The app name
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/dhop.cgi?action=uninstall&appName=xxx		

Response	OK

15.3.5 Download Application Log

URL	http://<*server*>/cgi-bin/dhop.cgi?action=downloadLog		
Method	GET		
Description	Download application log.		
[Request Params] (key=value format in URL)			
appName	string	R	The app name
[Response Params] (log file content)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/dhop.cgi?action=downloadLog&appName=xxx		
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: Application/octet-stream Content-Length: <*length*> <*log file data*>		

16.1 Video Analyse APIs

16.1.1 People Counting

16.1.1.1 Get Summary

Table 16-1

URL	http://<*server*>/cgi-bin/videoStatServer.cgi?action=getSummary		
Method	GET		
Description	Get summary information of video Stat.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
summary	object	R	Summary information of video Stat.
+Channel	int	R	Video channel index which starts from 0.
+RuleName	string	R	Rule type, it can be: ● "**ManNumDetection**": count the people num in region, and the detail stat is in "InsideSubtotal" param. ● "**NumberStat**": count the people entering and leaving the region, and the detail stat is in "EnteredSubtotal" and "ExitedSubtotal"param.
+EnteredSubtotal	object	O	People enter stat.
++Total	int	R	Total enter num.
++Today	int	R	Today enter num.
++Hour	int	R	This hour enter num.
++TotalInTimeSection	int	O	Today enter num after call clearSectionStat.
+ExitedSubtotal	object	O	People leave stat.
++Total	int	R	Total leave num.
++Today	int	R	Today leave num.
++Hour	int	R	This hour leave num.
++TotalInTimeSection	int	O	Today leave num after call clearSectionStat.
+InsideSubtotal	object	O	People inside region stat.
++Total	int	R	Today inside num.

++ManStayStat	array<object>	O	The entering and leaving stat of people that leaved.
+++EnterTime	string	O	People enter time, ex: 2012-01-04 00:00:00.
+++ExitTime	string	O	People leave time, ex: 2012-01-04 00:00:45.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=getSummary&channel=1		
Response	summary.Channel=0 summary.RuleName=NumberStat summary.EnteredSubtotal.Today=0 summary.EnteredSubtotal.Total=14 summary.EnteredSubtotal.TotalInTimeSection=0 summary.ExitedSubtotal.Today=0 summary.ExitedSubtotal.Total=32 summary.ExitedSubtotal.TotalInTimeSection=0 summary.InsideSubtotal.Total=65 summary.InsideSubtotal.ManStayStat[0].EnterTime=2012-01-04 00:00:00 summary.InsideSubtotal.ManStayStat[0].ExitTime=2012-01-04 00:00:45 summary.InsideSubtotal.ManStayStat[1].EnterTime=2012-01-04 00:00:00 summary.InsideSubtotal.ManStayStat[1].ExitTime=2012-01-04 00:00:45		

16.1.1.2 Query the Count of People

- Start to find

Table 16-2

URL	http://<*server*>/cgi-bin/videoStatServer.cgi?action=startFind		
Method	GET		
Description	Start to find video stat info, in response, there is a token for further info finding process, and there is a totalCount shows how many data count(s).		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1.
condition	object	R	Find condition.
+StartTime	string	R	Find time range start, ex: 2012-01-04 00:00:00.
+EndTime	string	R	Find time range end, ex: 2012-01-04 00:02:00.
+Granularity	string	R	The information granularity returned by the query requirements. The range is {Hour, Day, Week, Month, Season, Year}(**Note: Most devices only support Hour, Day, Week.**)
+RuleType	string	O	Rule type, it can be: **NumberStat**, **ManNumDetection.** If omit, default is **NumberStat**
+MinStayTime	int	O	Valid when ruleType is **ManNumDetection**, report people stay over this minimal time.
+PlanID	int	O	The plan id, only valid for dome camera.
+AreaID	Array<int>	O	The area id which starts from 1, max array size is 20.
+OtherRule	string	O	The other search rule, can be : AverageStayTime
[Response Params] (key=value format)			
Name	Type	R/O	Param Description

token	int	R	Token for this search, use this token to get result and stop search.
totalCount	int	R	Number of find result.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=startFind&channel=1&condition.StartTime=2011-01-01%2012:00:00&condition.EndTime=2011-01-10%2012:00:00&condition.Granularity=Hour&condition.RuleType=NumberStat&condition.MinStayTime=20&condition.AreaID[0]=2&condition.AreaID[1]=3		
Response	token=12345 totalCount=56		

- Get the find result

Table 16-3

URL	http://<*server*>/cgi-bin/videoStatServer.cgi?action=doFind		
Method	GET		
Description	Get the find result of Video Stat info with channel, token, begin Number and count.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1. **NOTE:** must be the same as startFind.
token	int	R	Token for this search, use this token to get result and stop searching.
beginNumber	int	R	The start count. It must be between 0 and totalCount -1.
count	int	R	The count of info for this query.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
found	int	R	Number of result that return.
info	array<object>	R	Result information of video Stat.
+Channel	int	R	Video channel index which starts from 0.
+PlanID	int	O	The plan id, only valid for dome camera.
+AreaID	int	O	The area id which starts from 1.
+AverageStayTime	int	O	The average stay time, only valid when startFind with OtherRule param's value is "AverageStayTime".
+RuleName	string	R	Rule name, it can be: ● "**ManNumDetection**": count people num in region, detail stat is in "InsideSubtotal"param ● "**NumberStat**": count people enter and leave region, detail stat is in "EnteredSubtotal"and "ExitedSubtotal"param.
+StartTime	string	O	Find time range start, ex: 2012-01-04 00:00:00.
+EndTime	string	O	Find time range end, ex: 2012-01-04 00:02:00.
+EnteredSubtotal	int	O	Total enter num.
+ExitedSubtotal	int	O	Total leave num.
+InsideSubtotal	int	O	Total inside num.

[Example]	
Request	GET http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=doFind&channel=1&token=12345&beginNumber=0&count=20
Response	found=20 info[0].Channel=0 info[0].AreaID=2 info[0].RuleName=NumberStat info[0].StartTime=2012-03-14 00:00:00 info[0].EndTime=2012-04-14 00:00:00 info[0].EnteredSubtotal=14 info[0].ExitedSubtotal=5 info[0].InsideSubtotal=65 info[1].Channel=0 info[1].AreaID=3 info[1].RuleName=NumberStat info[1].StartTime=2012-03-14 00:00:00 info[1].EndTime=2012-04-14 00:00:00 info[1].EnteredSubtotal=14 info[1].ExitedSubtotal=5 info[1].InsideSubtotal=65 …

- Stop the searching session

Table 16-4

URL	http://<*server*>/cgi-bin/videoStatServer.cgi?action=stopFind		
Method	GET		
Description	Stop query video stat by channel and token.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1. **NOTE:** must be the same as startFind.
token	int	R	Token for this search, use this token to get result and stop search.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=stopFind&channel=1&token=12345		
Response	OK		

16.1.1.3 Clear the People Count Information

Table 16-5

URL	http://<*server*>/cgi-bin/videoStatServer.cgi?action=clearSectionStat
Method	GET
Description	Clear the people count information.

[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=clearSectionStat&channel=1		
Response	OK		

16.1.1.4 Subscribe the People Count Information

Table 16-6

URL	http://<*server*>/cgi-bin/videoStatServer.cgi?action=attach		
Method	GET		
Description	Subscribe the people count information.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
channel	int	O	Video channel index which starts from 1, default is 1.
heartbeat	int	O	Send heartbeat interval, range is [1, 60],unit is second. If the URL contains this parameter, and the value is 5, it means every 5 seconds the device should send the heartbeat message to the client, the heartbeat meaage are "Heartbeat". If this parameter is not present, its default value is 60.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
summary	object	R	Summary information of video Stat.
+Channel	int	R	Video channel index which starts from 0.
+RuleName	string	R	Rule type, it can be: ● "**ManNumDetection**": Count people num in region, detail stat is in "InsideSubtotal" param. ● "**NumberStat**": Count people entering and leaving region, detail stat is in "EnteredSubtotal" and "ExitedSubtotal" param.
+EnteredSubtotal	object	O	People enter stat.
++Total	int	R	Total enter num.
++Today	int	R	Today enter num.
++Hour	int	R	This hour enter num.
++TotalInTimeSection	int	O	Today enter num after call clearSectionStat.
+ExitedSubtotal	object	O	People leave stat.
++Total	int	R	Total leave num.
++Today	int	R	Today leave num.
++Hour	int	R	This hour leave num.
++TotalInTimeSection	int	O	Today leave num after call clearSectionStat.
+InsideSubtotal	object	O	People inside region stat.

++Total	int	R	Today inside num.
++ManStayStat	array<object>	O	The entering and leaving stat of people that leaved.
+++EnterTime	string	O	People enter time, ex: 2012-01-04 00:00:00.
+++ExitTime	string	O	People leave time, ex: 2012-01-04 00:00:45.

[Example]

Request	GET http://192.168.1.108/cgi-bin/videoStatServer.cgi?action=attach&channel=1&heartbeat=5
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<*boundary*> Connection: closed --<*boundary*> Content-Type: text/plain Content-Length: <*length*> summary.Channel=0 summary.RuleName=NumberStat summary.EnteredSubtotal.Today=0 summary.EnteredSubtotal.Total=14 summary.EnteredSubtotal.TotalInTimeSection=0 summary.ExitedSubtotal.Today=0 summary.ExitedSubtotal.Total=32 summary.ExitedSubtotal.TotalInTimeSection=0 summary.InsideSubtotal.Total=65 summary.InsideSubtotal.ManStayStat[0].EnterTime=2012-01-04 00:00:00 summary.InsideSubtotal.ManStayStat[0].ExitTime=2012-01-04 00:00:45 summary.InsideSubtotal.ManStayStat[1].EnterTime=2012-01-04 00:00:00 summary.InsideSubtotal.ManStayStat[1].ExitTime=2012-01-04 00:00:45 --<*boundary*> Content-Type: text/plain Content-Length: 11 Heartbeat --<*boundary*> Content-Type: text/plain Content-Length: <*length*> summary.Channel=0 summary.RuleName=NumberStat summary.EnteredSubtotal.Today=0 summary.EnteredSubtotal.Total=14 …

16.1.1.5 [Event] NumberStat

Table 16-7

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When people number triggers the rule, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Number	int	R	Total number in detect region.
EnteredNumber	int	R	Total number that enter the detect region.
ExitedNumber	int	R	Total number that leave the detect region.
Type	string	R	Number overrun type, it can be: "EnterOver", "ExitOver", "InsideOver".
AreaID	int	O	The area id, begin from 1, if omit, means single area.
PresetID	int	O	The preset id, if omit, means preset is unknown.
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=NumberStat;action=Start;index=0;data={
	"Number": 120,
	"EnteredNumber": 180,
	"ExitedNumber": 60,
	"Type": "EnterOver",
	"Area": 2
	}
	--<*boundary*>		

16.1.1.6 [Event] ManNumDetection

Table 16-8

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When people number triggers the rule, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
ManList	Array<object>	R	The people info list.
+BoundingBox	Array<int>	R	The detected people bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
+Stature	int	R	The people's stature, unit is cm.
AreaID	int	O	The area id, begin from 1, if omit, means single area.
PresetID	int	O	The preset id, if omit, means preset is unknown.
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=ManNumDetection;action=pulse;index=0;data={
	"ManList": [{ "BoundingBox": [2992,1136,4960,5192], "Stature": 170 },
	{ "BoundingBox": [4392,4136,6960,6512], "Stature": 175 },
	{…}, …],
	"AreaID" : 2
	}
	--<*boundary*>

16.1.1.7 [Event] CrowdDetection

Table 16-9

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When crowd density overrun, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
CrowdList	Array<object>	O	The global crowd density overrun list.
+Center	Array<int>	R	The center point, must be two int, means x and y value, coordinate remap to 0 — 8192.
+Radius	int	R	The radius length.
RegionList	Array<object>	O	The people num overrun region list.
+RegionID	int	R	The region index.
+PeopleNum	int	R	The people count in region.
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=CrowdDetection;action=start;index=0;data={
	"CrowdList": [{ "Center" : [5734,2377],"Radius" : 10}, …{}],
	"RegionList" : [{ "RegionID" : 0, "PeopleNum" : 100 },…{}]
	}
	--<*boundary*>		

16.1.2 FaceRecognitionServer

16.1.2.1 Create Face Group

Table 16-10

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=createGroup

Method	GET		
Description	Create a face group.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupName	string	R	The face group name, max string length is 127.
groupDetail	string	O	The description detail of the face group, max string length is 255.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
groupID	string	R	The identity of the created face group, max string length is 63.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=createGroup&groupName=Test1&groupDetail=ForTest1		
Response	groupID=10000		

16.1.2.2 Modify Face Group

Table 16-11

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=modifyGroup		
Method	GET		
Description	Modify a face group.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupID	string	R	The identity of the face group, max string length is 63.
groupName	string	R	The name of the the face group, max string length is 127.
groupDetail	string	O	Description detail of the face group, max string length is 255.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=modifyGroup&groupID=10000&groupName=Test1&groupDetail=ForTest1		
Response	OK		

16.1.2.3 Delete Face Group

Table 16-12

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=deleteGroup		
Method	GET		
Description	Delete a face group.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupID	string	R	The identity of the face group, max string length is 63.
[Response Params] (OK)			
[Example]			

Request	GET http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=deleteGroup&groupID=10000
Response	OK

16.1.2.4 Deploy Face Group

There are two ways to deploy the group. One is based on the group (putDisposition), and the another one is based on the channel (setGroup).

- Put disposition to group

Table 16-13

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=putDisposition		
Method	GET		
Description	Deploy the face group to some video channels. If the video channel has been deployed already, it will change the similary.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupID	string	R	The identity of the face group, max string length is 63.
list	Array<object>	R	List of disposition info.
+channel	int	R	Video channel index which starts from 1.
+similary	int	R	The threshold of the face similary, 0 — 100.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
report	array<bool>	R	Result of putting disposition for each request channel.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=putDisposition&groupID=10000&list[0].channel=1&list[0].similary=80&list[1].channel=2&list[1].similary=70		
Response	report[0]=true report[1]=false		

- Delete some disposition from group

Table 16-14

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=deleteDisposition		
Method	GET		
Description	Remove the deployment of face group from some video channels.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupID	string	R	The identity of the face group, max string length is 63.
channel	Array<int>	R	Video channel index which starts from 1.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
report	array<bool>	R	Result of deleting disposition for each request channel.
[Example]			
Request	GET		

	http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=deleteDisposition&groupID=10000&channel[0]=1&channel[1]=2
Response	report[0]=true report[1]=false

- set disposition group to channel

Table 16-15

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=setGroup		
Method	GET		
Description	Deploy some face groups to one video channel. If the video channel has been deployed already, it will change the similary. **Note**: This method will do an overwrite operation.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	Video channel index which starts from 1.
list	Array<object>	O	List of disposition info, if not exist, remove all disposition from channel.
+groupID	int	R	The identity of the face group, max string length is 63.
+similary	int	R	The threshold of the face similary, 0 — 100.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=setGroup&channel=1&list[0].groupID=10000&list[0].similary=80&list[1].groupID=10002&list[1].similary=75		
Response	OK		

- get disposition group from channel

Table 16-16

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=getGroup		
Method	GET		
Description	Get the Deployment about the video channel. **Note**: If the video channel does not deploy any group, then the response will be success with empty http body.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
channel	int	R	Video channel index which starts from 1.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
groupID	Array<int>	R	The identity of the face group, max string length is 63.
similary	Array<int>	R	The threshold of the face similary, 0 — 100.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=getGroup&channel=1		
Response	groupID[0]=10001 groupID[1]=10003 groupID[2]=10006 …. similary[0]=80		

	similary[1]=75 similary[2]=85 ….

16.1.2.5 Find Face Group

Table 16-17

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=findGroup		
Method	GET		
Description	Find the face group. If the groupID is not present in the URL, it will return all the groups.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupID	string	O	The identity of the face group, max string length is 63.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
GroupList	Array<object>	R	The face group information list.
+groupID	string	R	The identity of the face group, max string length is 63.
+groupName	string	R	Name of the face group, max string length is 127.
+groupDetail	string	O	Description detail of the face group, max string length is 255.
+groupSize	int	R	The number of face in this face group.
+channels	Array<int>	R	Video channel index which starts from 0.
+similarity	Array<int>	R	The threshold of the face similary.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=findGroup		
Response	GroupList[0].groupID=00001 GroupList[0].groupName=Test1 GroupList[0].groupDetail=ForTest1 GroupList[0].groupSize=30 GroupList[0].channels[0]=1 GroupList[0].channels[1]=2 … GroupList[0].similarity[0]=80 GroupList[0].similarity[1]=75 … GroupList[1].groupID=00003 GroupList[1].groupName=Test3 GroupList[1].groupDetail=ForTest3 GroupList[1].groupSize=50 GroupList[1].channels[0]=1 GroupList[1].channels[1]=2 … GroupList[1].similarity[0]=70 GroupList[1].similarity[1]=85 …		

16.1.2.6 Re-Abstract Feature By Group

- Start ReAbstract

Table 16-18

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=groupReAbstract		
Method	GET		
Description	Abstract features for the groups. About the process of the re-extract, the device will use an event named "**FaceFeatureAbstract**" to report the process.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
groupID	Array<int>	R	The identity of the face group, max string length is 63.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
token	int	R	The identity of this operation.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=groupReAbstract&groupID[0]=10000&groupID[1]=10001		
Response	token=12345		

- Stop ReAbstract

Table 16-19

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=stopGroupReAbstract		
Method	GET		
Description	Stop the abstract features process.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
token	int	R	The identity of this operation.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=stopGroupReAbstract&token=12345		
Response	OK		

16.1.2.7 Add Person

Table 16-20

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=addPerson		
Method	POST		
Description	Add a person to a group.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
groupID	string	R	The identity of the face group that this person to add. Max string

			length is 63.
name	string	R	The person name, max string length is 63.
birthday	string	O	The person's birthday, ex: "1980-01-01".
sex	string	O	Sex, it can be "Male", "Female", "Unknown".
country	string	O	The country name, length must be 2, and value should be according to ISO3166.
province	string	O	The province name, max string length is 63.
city	string	O	The city name, max string length is 63.
certificateType	string	O	The certificate type. It can be: "IC', "Passport", "Military", "Unknown".
id	string	O	The ID of certificate type, max string length is 31.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
uid	string	R	The id for this Person, max string length is 31.
[Example]			
Request	POST http://<server>/cgi-bin/faceRecognitionServer.cgi?action=addPerson&groupID=10000&name=ZhangSan&birthday=1980-01-05&sex=Male&country=CN&province=ZheZhang&city=Hangzhou HTTP/1.1 Content-Type: image/jpeg Content-Length: <image size> <JPEG image data>		
Response	uid=0005		

16.1.2.8 Modify Person

Table 16-21

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=modifyPerson		
Method	POST		
Description	Modify a person's info. **Note**: If you do not want to change the image about the person, the request should not contain the image data. **Note:** You should provide at lease one optional param to update.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
uid	string	R	The identity of the Person, max string length is 31.
groupID	string	R	The identity of the Face Group that this Person in. max string length is 63.
name	string	O	The person's name, max string length is 63.
birthday	string	O	The person's birthday, ex: "1980-01-01".
sex	string	O	Sex, it can be "Male", "Female", "Unknown".
country	string	O	The country name, length must be 2, and value should be according to ISO3166.
province	string	O	The province name, max string length is 63.
city	string	O	The city name, max string length is 63.

certificateType	string	O	The certificate type. It can be: "IC', "Passport", "Military", "Unknown".
id	string	O	The ID of certificate type, max string length is 31.
[Response Params] (OK)			
[Example]			
Request	POST http://<server>/cgi-bin/faceRecognitionServer.cgi?action=modifyPerson&uid=0005&groupID=10000&name=ZhangSan&birthday=1980-01-05&sex=Male&country=CN&province=ZheZhang&city=Hangzhou HTTP/1.1 Content-Type: image/jpeg Content-Length: <*image size*> <JPEG image data>		
Response	OK		

16.1.2.9 Delete Person

Table 16-22

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=deletePerson		
Method	GET		
Description	Delete a person from a group.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
uid	string	R	The identity of the person, max string length is 31.
groupID	string	R	The identity of the face group that this Person in. max string length is 63.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=deletePerson&uid=001&groupID=10000		
Response	OK		

16.1.2.10 Find Person

- Start to find

Table 16-23

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=startFind		
Method	GET		
Description	Start to find person in face groups. **Note**: the returned token will be expired after 60 seconds without any doFind call.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
condition	Object	R	Search scope condition.
+GroupID	Array<string>	R	The list of identity of the face group, max string length is 63.

person	object	O	Person condition.
+Name	string	O	Person Name, max string length is 63.
+Sex	string	O	Sex, it can be "Male", "Female", "Unknown".
+Country	string	O	Country name, length must be 2, and value should be according to ISO3166.
+Province	string	O	Province name, max string length is 63.
+City	string	O	City name, max string length is 63.
+CertificateType	string	O	Certificate Type. It can be: "IC', "Passport", "Military", "Unknown".
+ID	string	O	Person ID of CertificateType, max string length is 31.
+FeatureState	int	O	Feature State, 0:Unknown, 1:Failed, 2:OK.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop search.
totalCount	int	R	Result num, return -1 means still searching.
[Example]			
Request	GET http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFind&condition.GroupID[0]=10000&condition.GroupID[1]=10003&person.Sex=Male&person.Country=CN&person.FeatureState=1		
Response	token=123456789 totalCount=24		

- Get find result

Table 16-24

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=doFind		
Method	GET		
Description	Get one result of person's information from the search result set. **Note**: the returned token will be expired after 60 seconds without any doFind call.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop search.
index	uint	R	The index in search result, should between 0 and totalCount –1.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
person	object	R	Person condition.
+UID	string	R	The identity of the person, max string length is 31.
+GroupID	string	R	The identity of the face group that this Person in. max string length is 63.
+Name	string	R	The person name, max string length is 63.
+Sex	string	O	Sex, it can be "Male", "Female", "Unknown".
+Birthday	string	O	The person's birthday, ex: "1980-01-01".
+Country	string	O	Country name, length must be 2, and value should be according to ISO3166.

+Province	string	O	Province name, max string length is 63.
+City	string	O	City name, max string length is 63.
+CertificateType	string	O	Certificate Type, can be: "IC', "Passport", "Military", "Unknown".
+ID	string	O	Person ID of CertificateType, max string length is 31.
+FeatureState	int	O	Feature State, 0:Unknown, 1:Failed, 2:OK.
[Example]			
Request	GET http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFind&token=123456789&index=0		
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<*boundary*> Content-Length: <length> --<*boundary*> Content-Type: text/plain Content-Length: <length> person.UID=0005 person.GroupID=10000 person.Name=ZhangSan person.Birthday=1980-01-01 person.Sex=Male person.Country=CN person.Province=ZheJiang person.City=HangZhou person.CertificateType=IC person.ID=1234567890 person.FeatureState=0 --<*boundary*> Content-Type: image/jpeg Content-Length: <image size> < jpeg image data ... > --<*boundary*>--		

- Stop finding

Table 16-25

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=stopFind		
Method	GET		
Description	Stop the search session.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
token	uint	R	The token for this search, use this token to get result and stop search.

[Response Params] (OK)	
[Example]	
Request	GET http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFind&token=123456789
Response	OK

16.1.2.11 Re-Abstract Features By Person

- Start ReAbstract

Table 16-26

URL	http://*<server>*/cgi-bin/faceRecognitionServer.cgi?action=reAbstract		
Method	GET		
Description	Abstract features for the persons. About the process of the re-extract, the device will use an event named "**FaceFeatureAbstract**"to report the process.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
UID	Array<int>	O	The list of identity of person, max string length is 31.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=reAbstract&uid[0]=001&uid[1]=002		
Response	OK		

- Stop ReAbstract

Table 16-27

URL	http://*<server>*/cgi-bin/faceRecognitionServer.cgi?action=stopReAbstract
Method	GET
Description	Stop the abstract features process.
[Request Params] (None)	
[Response Params] (OK)	
[Example]	
Request	GET http://192.168.1.108/cgi-bin/faceRecognitionServer.cgi?action=stopReAbstract
Response	OK

16.1.2.12 [Config] Face Recognition AlarmOut Setting

- Get FaceRecognitionAlarm config

Table 16-28

URL	http://*<server>*/cgi-bin/configManager.cgi?action=getConfig&name=FaceRecognitionAlarm
Method	GET
Description	Get the Face Recognition Alarm Out Setting.

[Request Params] (None)			
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
table	object	R	Config info table.
+FaceRecognitionAlarm	array<object>	R	Each face group has one config object in this array.
++GroupID	String	R	The face group ID, max string length is 63.
++GroupName	String	R	The face group name, max string length is 127.
++AlarmOutEnable	Bool	R	Enable AlarmOut or not.
++AlarmChannel	Array<object>	R	Each AlarmOut channel has one config object in this array.
+++AlarmRuleMask	Int	R	Alarm rule mask. ● Bit 0 : recognition success ● Bit 1 : recognition failed
+++AlarmOutLatch	Int	R	Alarm out delay, unit is second, value between 1 anf 300.
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=getConfig&name=FaceRecognitionAlarm		
Response	table.FaceRecognitionAlarm[0].GroupID=0017 table.FaceRecognitionAlarm[0].GroupName=wsd table.FaceRecognitionAlarm[0].AlarmOutEnable=true table.FaceRecognitionAlarm[0].AlarmChannel[0].AlarmRuleMask=0 table.FaceRecognitionAlarm[0].AlarmChannel[0].AlarmOutLatch=5 table.FaceRecognitionAlarm[0].AlarmChannel[1].AlarmRuleMask=0 table.FaceRecognitionAlarm[0].AlarmChannel[1].AlarmOutLatch=8 … table.FaceRecognitionAlarm[1].GroupID=0018 table.FaceRecognitionAlarm[1].GroupName=cst table.FaceRecognitionAlarm[1].AlarmOutEnable=true table.FaceRecognitionAlarm[1].AlarmChannel[0].AlarmRuleMask=0 table.FaceRecognitionAlarm[1].AlarmChannel[0].AlarmOutLatch=10 table.FaceRecognitionAlarm[1].AlarmChannel[1].AlarmRuleMask=0 table.FaceRecognitionAlarm[1].AlarmChannel[1].AlarmOutLatch=15 …		

- Set FaceRecognitionAlarm config

URL	http://<*server*>/cgi-bin/configManager.cgi?action=setConfig		
Method	GET		
Description	Set the Face Recognition Alarm Out Setting.		
[Request Params] (key=value format at URL)			
Name	Type	R/O	Param Description
FaceRecognitionAlarm	array<object>	R	Each face group has one config object in this array.
+GroupID	String	R	The face group ID, max string length is 63.

+GroupName	String	R	The face group name, max string length is 127.
+AlarmOutEnable	Bool	R	Enable AlarmOut or not.
+AlarmChannel	Array<object>	R	Each AlarmOut channel has one config object in this array.
++AlarmRuleMask	Int	R	Alarm rule mask. ● Bit 0 : recognition success ● Bit 1 : recognition failed
++AlarmOutLatch	Int	R	Alarm out delay, seconds, value between 1 and 300.
[Response Params] (OK)			
[Example]			
Request	GET http://10.0.0.8/cgi-bin/configManager.cgi?action=setConfig&FaceRecognitionAlarm[0].GroupID=0017&FaceRecognitionAlarm[0].GroupName=wsd&FaceRecognitionAlarm[0].AlarmOutEnable=true&FaceRecognitionAlarm[0].AlarmChannel[0].AlarmRuleMask=0&FaceRecognitionAlarm[0].AlarmChannel[0].AlarmOutLatch=5		
Response	OK		

16.1.2.13 Find Person by Picture

- Start to find

Table 16-29

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=startFindByPic		
Method	POST		
Description	Start to find person in face groups by picture. The search may last for some time, so the response may push at regular intervals until 100% Progress **Note**: If you want to find person in face groups by person info, see above "Find Person"API. **Note**: The returned token will be expired after 60 seconds without any doFind call.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
GroupID	array<string>	R	Face groups to find, max string length is 63.
Similarity	int	R	Similarity percent, 1 — 100.
MaxCandidate	int	O	Max Candidate result number.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
token	uint	R	The token of this search, use this token to get result and stop search.
progress	uint	R	Search Progress, 100 means finished.
totalCount	int	R	Result num, return -1 means still searching.
[Example]			
Request	POST http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFindByPic&GroupID[0]=00001&GroupID[1]=00003&Similarity=80&MaxCandidate=500 Content-Type: image/jpeg Content-Length: <*image size*>		

	<JPEG data>
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<*boundary*> Connection: closed --<*boundary*> Content-Type: application/json Content-Length: <*length*> { "token": 123456789, "progress": 20 "totalCount": -1 } --<*boundary*> Content-Type: application/json Content-Length: <*length*> { "token": 123456789, "progress": 60 "totalCount": -1 } --<*boundary*> Content-Type: application/json Content-Length: <*length*> { "token": 123456789, "progress": 100 "totalCount": 350 } --<*boundary*>--

- Get the find result

Table 16-30

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=doFindByPic		
Method	GET		
Description	Get the find result, reply using multipart format, first part is json string to describe all candidate person, then the following parts are the person's pictures, refer by UID and GroupID in part header Content-Info. **Note**: the returned token will be expired after 60 seconds without any doFind call.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop search.
index	uint	R	The index in search result, should between 0 and totalCount –1.

count	uint	R	Number of result person to get.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
Found	int	R	Number of result person that return.
Candidates	Array<object>	R	Candidates Person.
+Person	object	R	Person Info.
++UID	string	R	System id for this Person, max string length is 31.
++GroupID	string	R	The identity of the Face Group that this Person in. max string length is 63.
++Name	string	R	Person Name, max string length is 63.
++Birthday	string	O	Birthday ex: "1980-01-01".
++Sex	string	O	Sex, it can be "Male", "Female", "Unknown".
++Country	string	O	Country name, length must be 2, value should be according to ISO3166.
++Province	string	O	Province name, max string length is 63.
++City	string	O	City name, max string length is 63.
++CertificateType	string	O	Certificate Type. It can be "IC', "Passport", "Military" or "Unknown".
++ID	string	O	Person ID of CertificateType, max string length is 31.
++FeatureState	int	O	Feature State, 0:Unknown, 1:Failed, 2:OK.
+Similarity	int	R	Similarity.

[Example]

Request

```
GET
http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFindByPic&token=123456789&index=0&count=10
```

Response

```
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: multipart/x-mixed-replace; boundary=<boundary>
Connection: closed

--<boundary>
Content-Type: application/json
Content-Length: <length>

{ "Found" : 10,
   "Candidates" : [
      {
         "person" : {
            "UID" : "0001",
            "GroupID" : "001",
            "Name" : "ZhangSan",
            "Birthday" : "1980-01-05",
            "Sex" : "Male",
            ...
         },
         "Similarity" : 85
```

	},
	{
	"person" : {
	"UID" : "0002",
	"GroupID" : "002",
	"Name" : "LiSi",
	"Birthday" : "1980-01-06",
	"Sex" : "Male",
	...
	},
	"Similarity" : 80
	},
	{ ... }, ...
	]
	}
	--<*boundary*>
	Content-Info: UID=0001&GroupID=001
	Content-Type: image/jpeg
	Content-Length: <*length*>
	
	< jpeg image data ... >
	--<*boundary*>
	Content-Info: UID=0002&GroupID=002
	Content-Type: image/jpeg
	Content-Length: <*length*>
	
	< jpeg image data ... >
	--<*boundary*>
	...

- Stop finding

Table 16-31

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=stopFindByPic		
Method	GET		
Description	Stop the search session.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop search.
[Response Params] (OK)			
[Example]			
Request	GET http://<server>/cgi-bin/faceRecognitionServer.cgi?action=stopFindByPic&token=123456789		
Response	OK		

16.1.2.14 Find History Person by Picture

- Start to find

Table 16-32

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=startFindHistoryByPic		
Method	POST		
Description	Start to find person in capture history by picture. The search may last for some time, so the response may push at regular intervals until 100% Progress. **Note**: If you want to find person in capture history by person info, please refer to "mediaFileFind"API. **Note**: the returned token will be expired after 60 seconds without any doFind call.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Channel	int	R	Video channel index which starts from 0.
StartTime	string	R	Start time to search, ex: 2018-01-13T00:00:00Z.
EndTime	string	R	End time to search, ex: 2018-01-14T00:00:00Z.
Similarity	int	R	Similarity percent, 1 — 100.
MaxCandidate	int	O	Max Candidate result number.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop search.
progress	uint	R	Search Progress, 100 means finished.
totalCount	int	R	Result num, return -1 means still searching.
[Example]			
Request	POST http://<server>/cgi-bin/faceRecognitionServer.cgi?action=startFindHistoryByPic&Channel=0&StartTime=2018-01-13T00:00:00Z&EndTime=2018-01-14T00:00:00Z&Type=All&Similarity=80&MaxCandidate=500 Content-Type: image/jpeg Content-Length: <*image size*> <JPEG data>		

Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<*boundary*> Connection: closed --<*boundary*> Content-Type: application/json Content-Length: <*length*> { "token": 123456789, "progress": 20 "totalCount": -1 } --<*boundary*> Content-Type: application/json Content-Length: <*length*> { "token": 123456789, "progress": 60 "totalCount": -1 } --<*boundary*> Content-Type: application/json Content-Length: <*length*> { "token": 123456789, "progress": 100 "totalCount": 350 } --<*boundary*>--

- Get find result

Table 16-33

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=doFindHistoryByPic		
Method	GET		
Description	Get the find result, reply by multipart, first part is json string to describe all candidate person, then the following part is the person's picture, refer by UID in part header Content-Info. **Note**: the returned token will be expired after 60 seconds without any doFind call.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop search.
index	uint	R	The index in search result, should between 0 and totalCount –1.
count	uint	R	Number of result person to get, start from Index.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description

Found	int	R	Number of result person that return.
Candidates	Array<object>	R	Candidates Person.
+Person	object	R	Person Info.
++UID	string	R	System id for this Person, max string length is 63.
++Sex	string	O	Sex. It can be "Male", "Female" or "Unknown".
++Age	int	O	Age.
++Glasses	int	O	Glasses Status, 0: all, 1: not wear, 2: wear.
+Similarity	int	R	Similarity.

[Example]

Request

```
GET
http://<server>/cgi-bin/faceRecognitionServer.cgi?action=doFindHistoryByPic&token=123456789&index=0&count=12
```

Response

```
HTTP/1.1 200 OK
Server: Device/1.0
Content-Type: multipart/x-mixed-replace; boundary=<boundary>
Connection: closed

--<boundary>
Content-Type: application/json
Content-Length: <length>

{ "Found" : 12,
   "Candidates" : [
      {
         "person" : {
            "UID" : "0001",
            "Sex" : "Male",
            "Age" : 30,
            "Glasses": 1
         },
         "Similarity" : 85
      },
      {
         "person" : {
            "UID" : "0002",
            "Sex" : "Male",
            "Age" : 50,
            "Glasses": 2
         },
         "Similarity" : 80
      },
      { ... }, ...
   ]
}
--<boundary>
Content-Info: UID=0001
Content-Type: image/jpeg
```

	Content-Length: <*length*> < jpeg image data ... > --<*boundary*> Content-Info: UID=0002 Content-Type: image/jpeg Content-Length: <*length*> < jpeg image data ... > --<*boundary*> ...

- Stop finding

Table 16-34

URL	http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=stopFindHistoryByPic		
Method	GET		
Description	Stop finding.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
token	uint	R	Token for this search, use this token to get result and stop searching.
[Response Params] (OK)			
[Example]			
Request	GET http://<*server*>/cgi-bin/faceRecognitionServer.cgi?action=stopFindHistoryByPic&token=123456789		
Response	OK		

16.1.2.15 [Event] FaceDetection

Table 16-35

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When the video channel disposition with some face group, and the video channel detect a face, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Faces	Array<object>	R	The detected faces info.
+BoundingBox	Array<int>	R	The detected face bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
+Sex	string	O	Sex, it can be "Man", "Woman".
+Age	int	O	Age.

+Feature	array<string>	O	Face feature, can be some of the following: "WearGlasses", "SunGlasses", "NoGlasses", "Smile", "Anger", "Sadness", "Disgust", "Fear", "Surprise", "Neutral", "Laugh", "Happy", "Confused", "Scream".
+Eye	int	O	Eye status, 0: not detected, 1: close eye, 2: open eye.
+Mouth	int	O	Mouth status, 0: not detected, 1: close mouth, 2: open mouth.
+Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask.
+Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard.
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=FaceDetection;action=Start;index=0;data={
	"Faces": [{ "BoundingBox": [2992,136,6960,8192],
	"Sex": "Man",
	"Age": 40,
	"Feature": ["WearGlasses", "Smile"],
	"Eye": 2,
	"Mouth": 1,
	"Mask": 1,
	"Beard": 2
	}, {…}, …]
	}
	--<*boundary*>		

16.1.2.16 [Event] FaceRecognition

Table 16-36

Usage	Refer to "4.4.3 Subscribe to Snapshot"for how to subscribe event		
Description	When the video channel disposition with some face group, and the video channel detect a face, after recognize in the face groups, send this event.		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	R	Event info array.
+EventBaseInfo	object	R	Base info of event.
++Code	String	R	Event Code. It should be FaceRecognition.
++Action	String	R	Event Action. It can be "Start", "Stop" or "Pulse".
++Index	int	O	The channel index relate to this event, start from 0.
+UID	String	R	The identity of the Person, max string length is 31.
+Candidates	array<object>	O	The Candidates person list.
++Person	object	O	The information of candidate person.
+++UID	string	R	The identity of the Person, max string length is 31.
+++GroupID	string	R	The identity of the Face Group that this Person in. max string

			length is 63.
+++Name	string	O	Person Name, max string length is 63.
+++Birthday	string	O	Birthday ex: "1980-01-01".
+++Sex	string	O	Sex, it can be "Male", "Female", "Unknown".
+++Country	string	O	Country name. The length must be 2, and value should be according to ISO3166.
+++Province	string	O	Province name, max string length is 63.
+++City	string	O	City name, max string length is 63.
+++CertificateType	string	O	Certificate Type. It can be: "IC', "Passport", "Military", "Unknown".
+++ID	string	O	Person ID of CertificateType, max string length is 31.
++Similarity	int	O	Similarity of the Candidates person and the detected person, value between 1 — 100.
+Face	object	O	The attribute information of face.
++Sex	string	O	Sex, it can be "Man", "Woman".
++Age	int	O	Age.
++Feature	array<string>	O	Face feature, can be some of the following : "WearGlasses", "SunGlasses", "NoGlasses", "Smile", "Anger", "Sadness", "Disgust", "Fear", "Surprise", "Neutral", "Laugh", "Happy", "Confused", "Scream".
++Eye	int	O	Eye status, 0: not detected, 1: close eye, 2: open eye.
++Mouth	int	O	Mouth status, 0: not detected, 1: close mouth, 2: open mouth.
++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask.
++Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard.

[Example]

Event	--<*boundary*> Content-Type: text/plain Content-Length: <*length*> Events[0].EventBaseInfo .Code=FaceRecognition Events[0].EventBaseInfo .Action=Pulse Events[0].EventBaseInfo .Index=0 Events[0].UID=00105 Events[0].Candidates[0].Person.UID=0012 Events[0].Candidates[0].Person.GroupID=10000 Events[0].Candidates[0].Person.Name=ZhangSan Events[0].Candidates[0].Person.Birthday=1980-01-02 Events[0].Candidates[0].Person.Sex=Male … Events[0].Candidates[0].Similarity=80 Events[0].Candidates[1].Person.UID=0014 Events[0].Candidates[1].Person.GroupID=10000 Events[0].Candidates[1].Person.Name=Lisi Events[0].Candidates[1].Person.Birthday=1980-01-05 Events[0].Candidates[1].Person.Sex=Male

	… Events[0].Candidates[1].Similarity=75 … Events[0].Face.Sex=Man Events[0].Face.Age=20 Events[0].Face.Feature[0]=SunGlasses Events[0].Face.Feature[1]=Smile Events[0].Face.Eye=2 Events[0].Face.Mouth=1 Events[0].Face.Mask=1 Events[0].Face.Beard=2 --<*boundary*> Content-Type: image/jpeg Content-Length: <image size> <Jpeg image data> --<*boundary*>

16.1.2.17 [Event] FaceFeatureAbstract

Table 16-37

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When Re-Abstract Feature By Group or By Person, the abstract progress detail will send in this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Infos	Array<object>	R	Abstrace detail Info, max size is 100.
+State	String	R	Abstract state, it can be : ● "Success" : Abstract success, ● "False" : Failed to abstract; ● "Process" : In Process;
+Process	int	O	The abstract progress.
+UID	string	O	The identity of the person, max string length is 31.
+GroupID	string	O	The identity of the face group, max string length is 63.
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=FaceFeatureAbstract;action=Start;index=0;data={
	"Infos": [{ "State": "Progress",
	"Progress": 30,
	"UID": "20005",
	"GroupID": "10000"
	}, {…}, …]
	}
	--<*boundary*>

16.1.3 Video Analyse Event

16.1.3.1 [Event] LeftDetection

Table 16-38

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect some object left, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Object	object	R	The object that left.
+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=LeftDetection;action=pulse;index=0;data={
	"Object": { "BoundingBox": [2992,1136,4960,5192] }
	}
	--<*boundary*>		

16.1.3.2 [Event] TakenAwayDetection

Table 16-39

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect some object was taken away, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Object	object	R	The object that was taken away.

+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=TakenAwayDetection;action=pulse;index=0;data={
	"Object": { "BoundingBox": [2992,1136,4960,5192] }
	}
	--<*boundary*>		

16.1.3.3 [Event] WanderDetection

Table 16-40

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect some object was wandering, send this event		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Objects	Array<object>	R	The objects that was wandering.
+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
Tracks	Array<Array<Array<int>>>	O	The object wandering tracks, array of polyline, one polyline for one object, polyline is array of points, point is array of two int, x's value and y's value. Coordinate remap to 0 — 8192.
DetectRegion	Array<Array<int>>	R	The detection region, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=WanderDetection;action=start;index=0;data={
	"Objects": [{ "BoundingBox": [2992,1136,4960,5192] },
	{ "BoundingBox": [4392,4136,6960,6512] },
	{…}, …],
	"Tracks": [[[1202, 576], [1456, 863], [1921, 1204], [2341, 1823], [3512, 2314]],
	[[2214, 3412], [3153, 3674], [4512, 4213]],
	[…], …],
	"DetectRegion": [[1292,3469], [6535,3373], …]
	}		

	--<*boundary*>

16.1.3.4 [Event] StayDetection

Table 16-41

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect some object was stay, send this event		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Object	object	O	The object that was stay.
+BoundingBox	Array<int>	O	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
Objects	array<object>	O	If detect several object, store in this array.
+BoundingBox	Array<int>	O	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
DetectRegion	Array<Array<int>>	O	The detection region, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
AreaID	int	O	The area id, begin from 1, if omit, means single area.
PresetID	int	O	The preset id, if omit, means preset is unknown.
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=StayDetection;action=start;index=0;data={
	"Object": { "BoundingBox": [2992,1136,4960,5192] },
	"Objects": [{ "BoundingBox": [2992,1136,4960,5192] },
	{ "BoundingBox": [4392,4136,6960,6512] },
	{…}, …],
	"DetectRegion": [[1292,3469], [6535,3373], …],
	"AreaID" : 2
	}
	--<*boundary*>		

16.1.3.5 [Event] HumanTrait

Table 16-42

Usage	Refer to "4.4.3 Subscribe to Snapshot "for how to subscribe event
Description	When detect a human trait, send this event.
[Event Params] (key=value format)	

Name	Type	R/O	Param Description
Events	array<object>	O	Event info array.
+EventBaseInfo	object	R	Base info of event.
++Code	String	R	Event Code. It should be **HumanTrait.**
++Action	String	R	Event Action. It can be: "Start", "Stop", "Pulse".
++Index	int	O	The channel index relate to this event, start from 0.
+HumanAttributes	object	O	The human attributes.
++BoundingBox	Array<int>	O	The detected human bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
++Sex	string	O	Sex, can be "Man", "Woman", "Unknown".
++Age	int	O	Age.
++Angle	int	O	Angle, 0: unknown, 1: front, 2: side, 3: back.
++CoatColor	string	O	Coat color, can be: "White", "Orange", "Pink", "Black", "Red", "Yellow", "Gray", "Blue", "Green", "Purple", "Brown", "Sliver", "Darkviolet", "Maroon", "Dimgray", "Whitesmoke", "Darkorange", "Mistyrose", "Tomato", "Olive", "Gold", "Darkolivegreen", "Chartreuse", "Greenyellow", "Forestgreen", "Seagreen", "Chartreuse", "Deepskyblue", "Cyan", "Other".
++CoatType	int	O	Coat type, 0: unknown, 1: long sleeve, 2: short sleeve.
++TrousersColor	string	O	Trousers color, value can be that of CoatColor.
++TrousersType	int	O	Trousers type, 0: unknown, 1: long pants, 2: short pants, 3: skirt.
++HasHat	int	O	Has hat or not, 0: unknown, 1: not has hat, 2: has hat.
++HasBag	int	O	Has bag or not, 0: unknown, 1: not has bag, 2: has bag.
++HasUmbrella	int	O	Has umbrella or not, 0: unknown, 1: not has umbrella, 2: has umbrella.
++Bag	int	O	Bag type, 0: unknown, 1: handbag, 2: shoulder bag, 3: knapsack, 4: draw-bar box.
++UpperPattern	int	O	Upper clothes pattern, 0: unknown, 1: pure color, 2: stripe, 3: pattern, 4: gap, 5: grid.
++HairStyle	int	O	Hair style, 0: unknown, 1: long hair, 2: short hair, 3: ponytail, 4: updo, 5: hiddened.
++Cap	int	O	Cap style, 0: unknown, 1: normal cap, 2: helmet.
+FaceAttributes	object	O	If the human's face can be detected, find it's attributes.
++BoundingBox	Array<int>	O	The detected face bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
++Sex	string	O	Sex, it can be "Man", "Woman", "Unknown".
++Age	int	O	Age.

++Feature	array<string>	O	Face feature, can be some of the following : "WearGlasses", "SunGlasses", "NoGlasses", "Smile", "Anger", "Sadness", "Disgust", "Fear", "Surprise", "Neutral", "Laugh", "Happy", "Confused", "Scream".
++Eye	int	O	Eye status, 0: not detected, 1: close eye, 2: open eye.
++Mouth	int	O	Mouth status, 0: not detected, 1: close mouth, 2: open mouth.
++Mask	int	O	Mask status, 0: not detected, 1: not wearing mask, 2: wearing mask.
++Beard	int	O	Beard status, 0: not detected, 1: no beard, 2: has beard.
++Glass	Int	O	Glasses status, 0: unknown, 1: not wearing, 2: normal Glasses, 3: sun glasses, 4: black frame glasses.
[Example]			

Event	--<*boundary*> Content-Type: text/plain Content-Length: <*length*> Events[0].EventBaseInfo .Code=HumanTrait Events[0].EventBaseInfo .Action=Pulse Events[0].EventBaseInfo .Index=0 Events[0].HumanAttributes.BoundingBox[0]=1341 Events[0].HumanAttributes.BoundingBox[1]=2451 Events[0].HumanAttributes.BoundingBox[2]=4513 Events[0].HumanAttributes.BoundingBox[3]=4135 Events[0].HumanAttributes.Sex=Man Events[0].HumanAttributes.Age=30 Events[0].HumanAttributes.CoatColor=White Events[0].HumanAttributes.CoatType=1 Events[0].HumanAttributes.TrousersColor=Black Events[0].HumanAttributes.TrousersType=1 Events[0].HumanAttributes.HasHat=1 Events[0].HumanAttributes.HasBag=2 Events[0].FaceAttributes.BoundingBox[0]=1341 Events[0].FaceAttributes.BoundingBox[1]=2451 Events[0].FaceAttributes.BoundingBox[2]=4513 Events[0].FaceAttributes.BoundingBox[3]=4135 Events[0].FaceAttributes.Sex=Man Events[0].FaceAttributes.Age=30 Events[0].FaceAttributes.Feature[0]=Smile Events[0].FaceAttributes.Eye=2 Events[0].FaceAttributes.Mouth=1 Events[0].FaceAttributes.Glass=1 --<*boundary*> Content-Type: image/jpeg Content-Length: <*image size*> <Jpeg image data> --<*boundary*>

16.1.3.6 [Event] CrossLineDetection

Table 16-43

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect some object cross the line, send this event.		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Object	object	R	The object that cross the line.
+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right

			—bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
Objects	array<object>	O	If detect several object, store in this array.
+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right —bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
DetectLine	Array<Array<int>>	R	The detection line, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
Direction	string	O	The crossline direction, can be : “LeftToRight”, “RightToLeft”, “Any”.
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=CrossLineDetection;action=pulse;index=0;data={
	"Object": { "BoundingBox": [2992,1136,4960,5192] },
	"DetectLine": [[1292,3469], [6535,3373], …],
	"Direction" : "LeftToRight"
	}
	--<*boundary*>		

16.1.3.7 [Event] CrossRegionDetection

Table 16-44

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect some object cross the region, send this event		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Object	object	R	The object that cross the region.
+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left-top point, x's value of right —bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
Objects	array<object>	O	If detect several object, store in this array.
+BoundingBox	Array<int>	R	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right —bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
DetectRegion	Array<Array<int>>	R	The detection region, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.

Action	string	R	The cross action, can be : "Appear", "Disappear", "Cross","Inside"
Direction	string	O	The cross direction, valid on if the “Action” is “Cross”, can be : "Enter", "Leave", "Both".
[Example]			

Event	--<boundary>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=CrossRegionDetection;action=pulse;index=0;data={
	"Object": { "BoundingBox": [2992,1136,4960,5192] },
	"DetectRegion": [[1292,3469], [6535,3373], …],
	“Action” : “Cross”,
	"Direction" : "Enter"
	}
	--<*boundary*>		

16.1.3.8 [Event] QueueStayDetection

Table 16-45

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect the queue stay time too long, send this event		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
Object	object	O	The object that was stay.
+BoundingBox	Array<int>	O	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
Objects	array<object>	O	If detect several object, store in this array.
+BoundingBox	Array<int>	O	The detected object bounding box, 4 interge, refer to x's value of left—top point, y's value of left—top point, x's value of right—bottom point, y's value of right—bottom point. Coordinate remap to 0 — 8192.
DetectRegion	Array<Array<int>>	O	The detection region, the first array is point list, max item is 20, the second array is point, must be two int, means x and y value, coordinate remap to 0 — 8192.
AreaID	int	O	The area id, begin from 1, if omit, means single area.
PresetID	int	O	The preset id, if omit, means preset is unknown.
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=QueueStayDetection;action=start;index=0;data={
	"Object": { "BoundingBox": [2992,1136,4960,5192] },
	"Objects": [{ "BoundingBox": [2992,1136,4960,5192] },
	{ "BoundingBox": [4392,4136,6960,6512] },
	{…}, …],
	"DetectRegion": [[1292,3469], [6535,3373], …],
	"AreaID" : 2
	}
	--<*boundary*>

16.1.3.9 [Event] QueueNumDetection

Table 16-46

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When detect the queue people number exceed limit, send this event		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
ManList	Array<object>	R	The people info list.
+BoundingBox	Array<int>	R	The detected people bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
+Stature	int	R	The people's stature, unit is cm.
AreaID	int	O	The area id, begin from 1, if omit, means single area.
PresetID	int	O	The preset id, if omit, means preset is unknown.
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: <*length*>
	
	Code=QueueNumDetection;action=pulse;index=0;data={
	"ManList": [{ "BoundingBox": [2992,1136,4960,5192], "Stature": 170 },
	{ "BoundingBox": [4392,4136,6960,6512], "Stature": 175 },
	{…}, …],
	"AreaID" : 2
	}
	--<*boundary*>		

16.2 Intelligent Traffic APIs

16.2.1 Traffic Record

16.2.1.1 Insert Traffic BlackList/RedList Record

Table 16-47

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=insert&name=<**RecordName**>		
Method	GET		
Description	Insert traffic BlackList/Red record.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The record table name, "**TrafficBlackList**" for BlackList and "**TrafficRedList**" for RedList.
PlateNumber	string	R	The number of car plate, max string length is 31. It must be unique.
MasterOfCar	string	O	The car owner, max string length is 15
PlateColor	string	O	Plate color, max string length is 31, ex: "Yellow", "Blue", … etc.
PlateType	string	O	Plate type, max string length is 31.
VehicleType	string	O	Vehicle type, max string length is 31.
VehicleColor	string	O	Vehicle color, max string length is 31. ex: "Yellow", "Blue", … etc.
BeginTime	string	O	Begin time, ex: "2010-05-25 00:00:00".
CancelTime	string	O	Cancel time, ex: "2010-06-25 00:00:00".
AuthorityList	object	O	Authority list, only valid for "**TrafficRedList**" table.
+OpenGate	bool	O	Authority to open the gate.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
RecNo	int	R	The new record's id, return -1 if the device handles asynchronously.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insert&name=TrafficBlackList&PlateNumber=AC00001&MasterOfCar=ZhangSan&PlateColor=Yellow&VehicleColor=Blue&BeginTime=2011-01-01%2012:00:00&CancelTime=2011-01-10%2012:00:00		
Response	RecNo=12345		

16.2.1.2 Update Traffic BlackList/RedList Record

Table 16-48

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=update&name=<**RecordName**>
Method	GET

Description	Update Traffic BlackList/Red record. **Note:** Besides action, name, recno, there should be at least one more parameter to be updated.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The record table name, "**TrafficBlackList**" for BlackList and "**TrafficRedList**"for RedList.
recno	int	R	record id
PlateNumber	string	R	The number of car plate, max string length is 31. It must be unique.
MasterOfCar	string	O	The car owner, max string length is 15
…<other param>	-	-	…<See above insert command for other params of the record. They are all optional.>
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=update&name=TrafficBlackList&recno=12345&PlateNumber=AC00001&MasterOfCar=ZhangSan&PlateColor=Yellow&VehicleColor=Blue&BeginTime=2011-01-01%2012:00:00&CancelTime=2011-01-10%2012:00:00		
Response	OK		

16.2.1.3 Remove Traffic BlackList/RedList Record

Table 16-49

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=remove&name=<**RecordName**>		
Method	GET		
Description	Remove Traffic BlackList/Red record.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The record table name, "**TrafficBlackList**"for BlackList and "**TrafficRedList**"for RedList.
recno	int	R	The record id.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=remove&name=TrafficBlackList&recno=12345		
Response	OK		

16.2.1.4 Find Traffic BlackList/RedList Record

Table 16-50

URL	http://<*server*>/cgi-bin/recordFinder.cgi?action=find&name=<**RecordName**>

Method	GET		
Description	Find Traffic BlackList/RedList record.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The record table name, "**TrafficBlackList**"for BlackList and "**TrafficRedList**"for RedList.
count	int	O	Max result to return, default is 1024.
StartTime	string	O	The start of the record's CreateTime.
EndTime	string	O	The end of the record's CreateTime.
condition	object	O	Search condition.
+PlateNumber	string	O	Car plate number, max string length is 47.
+PlateNumberVague	string	O	Car plate number substring, match any car plate number that contain this substring, max string length is 47.
+PlateNumberVagueGroup	array<string>	O	Car plate number substring array, max string length is 47.
+QueryCount	int	O	Query count, default is 1000.
+QueryResultBegin	int	O	Begin number in the result set, default is 0.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
totalCount	int	O	Total record num that find.
found	int	O	Record num that returned.
records	array<object>	R	The records that returned.
+RecNo	int	R	Record id.
+CreateTime	int	R	The create time of record.
+PlateNumber	string	R	Car plate number.
+MasterOfCar	string	O	Car owner.
+…<other param>	-	-	…<see above insert command for other params of the record>
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=TrafficBlackList&condition.PlateNumber=AC00001&StartTime=123456700&EndTime=123456800&count=100		
Response	totalCount=1000 found=100 records[0].RecNo=12345 records[0].CreateTime=123456789 records[0].PlateNumber=AC00001 records[0].MasterOfCar=ZhangSan … records[1].RecNo=13579 records[1].CreateTime=123456799 records[1].PlateNumber=AC00001 records[1].MasterOfCar=LiSi …		

16.2.2 Intelligent Traffic Event

16.2.2.1 [Event] TrafficJunction

Table 16-51

Usage	Refer to "4.4.3 Subscribe to Snapshot"for how to subscribe event		
Description	When detects vehicle passing, send this event.		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	R	Event info array.
+EventBaseInfo	object	R	Base info of event.
++Code	String	R	Event Code. It should be **TrafficJunction.**
++Action	String	R	Event Action. It can be "Start", "Stop" or "Pulse".
++Index	int	O	The channel index relate to this event.
+GroupID	int	O	The id of event group.
+CountInGroup	int	O	Event count in the event group.
+IndexInGroup	int	O	The index of this event in the event group, start from 1.
+Lane	int	O	Lane number, start from 0.
+TriggerType	int	O	Trigger type. It can be: 0—car detector, 1—radar, 2—video.
+Speed	int	O	Vehicle speed, unit is km/hour.
+Vehicle	Object	O	The information of vehicle object.
++BoundingBox	Array<int>	R	The detected car bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
++Text	string	O	Vehicle logo.
++SubText	string	O	Vehicle sub logo.
++SubBrand	int	O	Vehicle sub brand index.
++BrandYear	int	O	Vehicle brand year index.
+TrafficCar	object	O	Traffic Car info.
++RecNo	int	R	The record id.
++PlateNumber	string	R	Car plate number.
++PlateType	string	O	Plate type.
++PlateColor	string	O	Plate color, ex: "Yellow", "Blue", … etc.
++VehicleColor	string	O	Vehicle color, ex: "Yellow", "Blue", … etc.
++BoundingBox	Array<int>	R	The detected plate bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
++Country	string	O	Country info. Max string length is 19.
++Speed	int	O	Vehicle speed, unit is km/hour.
++Event	string	O	The event info, ex: "TrafficJunction".
+CommInfo	object	O	Traffic event common info.
++Seat	array<object>	O	Vehicle front seat info.

+++Type	string	O	Front seat type. It can be: "Main""Slave".
+++Status	array<string>	O	Some driver status. It can be some of the following: "Smoking", "Calling".
+++SunShade	string	O	Sunshade status. It can be: "Unknow", "WithSunShade", "WithoutSunShade".
+++ShadePos	Array<int>	O	The detected sun shade bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.
+++SafeBelt	string	O	SafeBelt status. It can be: "Unknow", "WithSafeBelt", "WithoutSafeBelt".

[Example]

Event

```
--<boundary>
Content-Type: text/plain
Content-Length: <length>

Events[0].EventBaseInfo.Code=TrafficJunction
Events[0].EventBaseInfo.Action=Pulse
Events[0].EventBaseInfo.Index=0
Events[0].GroupID=123
Events[0].CountInGroup=3
Events[0].IndexInGroup=1
Events[0].Lane=0
Events[0].Vehicle.BoundingBox[0]=1341
Events[0].Vehicle.BoundingBox[1]=2451
Events[0].Vehicle.BoundingBox[2]=4513
Events[0].Vehicle.BoundingBox[3]=4135
Events[0].Vehicle.Text=Audi
Events[0].Vehicle.SubText=A6L
Events[0].Vehicle.SubBrand=5
Events[0].Vehicle.BrandYear=2
Events[0].TrafficCar.RecNo=123
Events[0].TrafficCar.PlateNumber=AC00003
Events[0].TrafficCar.PlateColor=Yellow
Events[0].TrafficCar.VehicleColor=Blue
Events[0].TrafficCar.BoundingBox[0]=1341
Events[0].TrafficCar.BoundingBox[1]=2451
Events[0].TrafficCar.BoundingBox[2]=4513
Events[0].TrafficCar.BoundingBox[3]=4135
Events[0].TrafficCar.Country=China
Events[0].CommInfo.Seat[0].Type=Main
Events[0].CommInfo.Seat[0].Status[0]=Smoking
Events[0].CommInfo.Seat[0].SunShade=WithSunShade
Events[0].CommInfo.Seat[0].ShadePos[0]=2021
Events[0].CommInfo.Seat[0].ShadePos[1]=3041
Events[0].CommInfo.Seat[0].ShadePos[2]=2151
Events[0].CommInfo.Seat[0].ShadePos[3]=3661
```

	Events[0].CommInfo.Seat[0].SafeBelt=WithoutSafeBelt … --<*boundary*> Content-Type: image/jpeg Content-Length: <image size> <Jpeg image data> --<*boundary*>

16.2.2.2 [Event] TrafficRetrograde

When detects vehicle retrograde, send this event.

Event params is the same as **TrafficJunction**, except for event Code is **TrafficRetrograde.**

16.2.2.3 [Event] TrafficJam

Table 16-52

Usage	Refer to "4.4.3 Subscribe to Snapshot"for how to subscribe event		
Description	When detects traffic jam, send this event.		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	R	Event info array.
+EventBaseInfo	object	R	Base info of event.
++Code	String	R	Event Code. It should be **TrafficJam.**
++Action	String	R	Event Action. It can be "Start", "Stop" or "Pulse".
++Index	int	O	The channel index relate to this event.
+GroupID	int	O	The id of event group.
+CountInGroup	int	O	Event count in the event group.
+IndexInGroup	int	O	The index of this event in the event group, start from 1.
+Lane	int	O	Lane number, start from 0.
+StartJaming	int	O	Start jam time, UTC seconds.
+AlarmInterval	int	O	Alarm interval, unit is second.
+JamLenght	int	O	Jam length, percentage of the lane.
+JamRealLength	int	O	Jam real length, unit is metre.
[Example]			
Event	--<*boundary*> Content-Type: text/plain Content-Length: <length> Events[0].EventBaseInfo.Code=TrafficJam Events[0].EventBaseInfo.Action=Pulse Events[0].EventBaseInfo.Index=0 Events[0].GroupID=123 Events[0].CountInGroup=3 Events[0].IndexInGroup=1		

	Events[0].Lane=0 Events[0].StartJaming=123456789 Events[0].AlarmInterval=180 Events[0].JamLenght=70 Events[0].JamRealLength=120 … --<*boundary*> Content-Type: image/jpeg Content-Length: <image size> <Jpeg image data> --<*boundary*>

16.2.2.4 [Event] TrafficUnderSpeed

When detects vehicle under speed, send this event.

Event params is the same as **TrafficJunction**, except for event Code is **TrafficUnderSpeed**, and add following params:

Table 16-53

[Extra Event Params] (key=value format)			
Name	Type	R/O	Param Description
+SpeedLimit	array<int>	O	Speed limit, 2 integer, min speed and max speed.
+UnderSpeeding Percentage	int	O	Percentage of under speed.

16.2.2.5 [Event] TrafficOverSpeed

When detects vehicle over speed, send this event.

Event params is the same as **TrafficJunction**, except for event Code is **TrafficOverSpeed**, and add following params:

Table 16-54

[Extra Event Params] (key=value format)			
Name	Type	R/O	Param Description
+SpeedLimit	array<int>	O	Speed limit, 2 integer, min speed and max speed.
+SpeedingPercentage	int	O	Percentage of over speed.

16.2.2.6 [Event] TrafficPedestrain

Table 16-55

Usage	Refer to "4.4.3 Subscribe to Snapshot"for how to subscribe event		
Description	When detects human in the lane, send this event		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	R	Event info array.

+EventBaseInfo	object	R	Base info of event.
++Code	String	R	Event Code. It should be **TrafficPedestrain.**
++Action	String	R	Event Action. It can be "Start", "Stop" or "Pulse".
++Index	int	O	The channel index relate to this event.
+GroupID	int	O	The id of event group.
+CountInGroup	int	O	Event count in the event group.
+IndexInGroup	int	O	The index of this event in the event group, start from 1.
+Lane	int	O	Lane number, start from 0.
+Vehicle	object	O	Actually this is the human that detected, not vehicle.
++Category	string	R	Must be "**Passerby**", means this actually is a human.
++BoundingBox	Array<int>	R	The detected car bounding box, 4 interge, refer to x's value of left-top point, y's value of left-top point, x's value of right-bottom point, y's value of right-bottom point. Coordinate remap to 0 — 8192.

[Example]

Event

--<*boundary*>
Content-Type: text/plain
Content-Length: <length>

Events[0].EventBaseInfo.Code=TrafficPedestrain
Events[0].EventBaseInfo.Action=Pulse
Events[0].EventBaseInfo.Index=0
Events[0].GroupID=123
Events[0].CountInGroup=3
Events[0].IndexInGroup=1
Events[0].Lane=0
Events[0].Vehicle.Category=Passerby
Events[0].Vehicle.BoundingBox[0]=1341
Events[0].Vehicle.BoundingBox[1]=2451
Events[0].Vehicle.BoundingBox[2]=4513
Events[0].Vehicle.BoundingBox[3]=4135
…
--<*boundary*>
Content-Type: image/jpeg
Content-Length: <image size>

<Jpeg image data>
--<*boundary*>

16.2.2.7 [Event] TrafficParking

When detects vehicle illegal parking, send this event.

Event params is the same as **TrafficJunction**, except for event Code is **TrafficParking**, and remove param "**speed**", add following params:

Table 16-56

[Extra Event Params] (key=value format)

Name	Type	R/O	Param Description
+StartParking	int	O	The start time of parking, UTC seconds.
+AlarmInterval	int	O	The alarm interval.
+ParkingAllowed Time	int	O	The allowed time of parking.

16.2.3 Traffic Flow

16.2.3.1 [Event] TrafficFlowStat

Table 16-57

Usage	Refer to "4.9.17 Subscribe to Event Message"for how to subscribe event		
Description	When traffic flow trigger the rule, send this event		
[Event Params] (JSON format)			
Name	Type	R/O	Param Description
FlowStates	array<object>	R	Traffic flow info, each object in list is traffic flow info about one lane.
+Lane	int	R	Lane number, start from 0.
+Flow	int	R	Traffic flow number.
+Period	int	R	Traffic stat time, unit is minute.
+PeriodByMili	int	O	Traffic stat time, unit is millisecond, value should between 0 and 59999.
+DrivingDirection	array<string>	O	Driving direction, should be an array of three strings: ● 1st string: direction, can be: "**Approach**", "**Leave**" ● 2nd string: Approach position name ● 3rd string: Leave position name
[Example]			

Event	--<*boundary*>
	Content-Type: text/plain
	Content-Length: xxxx
	
	Code=TrafficFlowStat;action=Pulse;index=0;data={
	"FlowStates": [{ "Lane": 0,
	"Flow": 50,
	"Period": 5,
	"DrivingDirection": ["Approach", "Hangzhou", "Shanghai"]
	}, {…}, …]
	}
	--<*boundary*>		

16.2.3.2 Find Traffic Flow History

Table 16-58

URL	http://<*server*>/cgi-bin/recordFinder.cgi?action=find&name=TrafficFlow
Method	GET

Description	Find traffic flow history record.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The name of record table. It should be "**TrafficFlow**".
count	int	O	Max result to return, default is 1024.
StartTime	string	O	The start of the record's CreateTime.
EndTime	string	O	The End of the record's CreateTime.
condition	object	O	Search condition.
+Channel	int	O	Video channel index which starts from 0.
+Lane	int	O	Lane index, starts from 0.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
totalCount	int	O	Total record num that find.
found	int	O	Record num that returned.
records	array<object>	R	The records that returned.
+RecNo	int	R	Record id.
+CreateTime	int	R	The create time of record.
+StatisticsTime	int	R	Traffic flow statistics time, UTC seconds.
+Period	string	R	Statistics period, unit is second.
+Channel	int	R	Video channel index which starts from 0.
+Lane	int	R	Lane index, starts from 0.
+Vehicles	int	R	Total vehicle num.
+AverageSpeed	float	O	Average speed of the vehicle, -1 means no vehicle, 0 means congestion.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=TrafficFlow&condition.Channel=0&condition.Lane=0&StartTime=123456700&EndTime=123456800&count=100		
Response	totalCount=1000 found=100 records[0].RecNo=12345 records[0].CreateTime=123456789 records[0].StaticsticsTime=123456789 records[0].Period=300 records[0].Channel=0 records[0].Lane=0 records[0].Vehicles=250 records[0].AverageSpeed=25.4 … records[1].RecNo=13579 records[1].CreateTime=123456799 records[1].StaticsticsTime=123456799 records[1].Period=300 records[1].Channel=0 records[1].Lane=0 records[1].Vehicles=220		

	records[1].AverageSpeed=21.8 …

16.3 Access Control APIs

16.3.1 Access User

16.3.1.1 Add Access User

Table 16-59

URL	http://<*server*>/cgi-bin/Attendance.cgi?action=addUser		
Method	GET		
Description	Add an access user. (This api is supported by attendance machine.)		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
UserID	string	R	The user's id.
UserName	string	O	The user's name.
CardNo	string	O	The user's card number.
Password	string	O	The user's password.
[Response Params] (OK)			
[Example]			
Request	GET http://<server>/cgi-bin/Attendance.cgi?action=addUser&UserID=102&UserName=Zhang San&Password=123456		
Response	OK		

16.3.1.2 Modify Access User

Table 16-60

URL	http://<*server*>/cgi-bin/Attendance.cgi?action=modifyUser		
Method	GET		
Description	Modify an access user. (This api is supported by attendance machine.) **Note:** You should provide at lease one optional param to update.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
UserID	string	R	The user's id.
UserName	string	O	The user's name.
CardNo	string	O	The user's card number.
Password	string	O	The user's password.
[Response Params] (OK)			
[Example]			
Request	GET http://<server>/cgi-bin/Attendance.cgi?action=modifyUser&UserID=102&UserName=Zhang San&Password=123456		

Response	OK

16.3.1.3 Delete Access User

Table 16-61

URL	http://<*server*>/cgi-bin/Attendance.cgi?action=deleteUser		
Method	GET		
Description	Delete an access user. (This api is supported by attendance machine.)		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
UserID	string	R	The user's id.
[Response Params] (OK)			
[Example]			
Request	GET http://<*server*>/cgi-bin/Attendance.cgi?action=deleteUser&UserID=102		
Response	OK		

16.3.1.4 Find Access User

- Find access user by UserID

Table 16-62

URL	http://<*server*>/cgi-bin/Attendance.cgi?action=getUser		
Method	GET		
Description	Get an access user's information. (This api is supported by attendance machine.)		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
UserID	string	R	The user's id
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
UserInfo	object	R	The user's info
+UserID	string	R	The user's id
+UserName	string	O	The user's name
+CardNo	string	O	The user's card number
+Password	string	O	The user's password
[Example]			
Request	GET http://<server>/cgi-bin/Attendance.cgi?action=getUser&UserID=102		

Response	{ "UserInfo": {
	"UserID": 102,
	"UserName": "ZhangSan",
	"Password": "123456"
	}		

- Find all access user

Table 16-63

URL	http://<*server*>/cgi-bin/Attendance.cgi?action=findUser
Method	GET
Description	Get all information of access user.

[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Offset	int	R	The offset of the user record table
Count	int	R	The count of user info record, max is 100.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
Total	int	R	Total user num
UserInfo	array<object>	R	The user's info
+UserID	string	R	The user's id
+UserName	string	O	The user's name
+CardNo	string	O	The user's card number
+Password	string	O	The user's password
[Example]			
Request	GET http://<server>/cgi-bin/Attendance.cgi?action=findUser&Offset=0&Count=100		

Response	{ "Total": 1000,
	"UserInfo": [{
	"UserID": 102,
	"UserName": "ZhangSan",
	"Password": "123456"
	}, { … } …]
	}		

16.3.1.5 Add Access User Face

Table 16-64

URL	http://*<server>*/cgi-bin/FaceInfoManager.cgi?action=add		
Method	POST		
Description	Add an access user face.		
[Request Params] (JSON format in body)			
Name	Type	R/O	Param Description
UserID	string	R	The user's id.
Info	object	R	The user's info.
+UserName	string	O	The user's name.
+RoomNo	array<string>	O	The user's room num list, max array size is 32, max string size is 12. (This param is not supported by access control.)
+FaceData	array<string>	O	The user's face feature, encode with base64, max array size is 20, and max string size is 2k. **Note**: There must be at least one between FaceData and PhotoData.
+PhotoData	array<string>	O	The user's face photo, encode with base64, max array size is 5, and max string size is 200k. **Note**: There must be at least one between FaceData and PhotoData.
[Response Params] (OK)			
[Example]			
Request	POST http://<server>/cgi-bin/FaceInfoManager.cgi?action=add		

	Content-Type: application/json
	Content-Length: <length>
	
	{ "UserID": "102",
	"Info": {
	"UserName": "ZhangSan",
	"RoomNo": ["301", "303", …],
	"FaceData": ["xxxx", "xxxx", …],
	"PhotoData": ["yyyy", "yyyy", …]
	}
	}

Response	OK

16.3.1.6 Modify Access User Face

Table 16-65

URL	http://<*server*>/cgi-bin/FaceInfoManager.cgi?action=update		
Method	POST		
Description	Update an access user face.		
[Request Params] (JSON format in body)			
Name	Type	R/O	Param Description
UserID	string	R	The user's id.
Info	object	R	The user's info.
+UserName	string	O	The user's name.
+RoomNo	array<string>	O	The user's room num list, max array size is 32, max string size is 12. (This param is not supported by access control.)
+FaceData	array<string>	O	The user's face feature, encode with base64, max array size is 20, and max string size is 2k. **Note**: There must be at least one between FaceData and PhotoData.
+PhotoData	array<string>	O	The user's face photo, encode with base64, max array size is 5, max string size is 200k. **Note**: There must be at least one between FaceData and PhotoData.
[Response Params] (OK)			
[Example]			
Request	POST http://<server>/cgi-bin/FaceInfoManager.cgi?action=update Content-Type: application/json Content-Length: <length> { "UserID": "102", "Info": { "UserName": "ZhangSan", "RoomNo": ["301", "303", …], "FaceData": ["xxxx", "xxxx", …], "PhotoData": ["yyyy", "yyyy", …]		

	} }
Response	OK

16.3.1.7 Delete Access User Face

Table 16-66

URL	http://<*server*>/cgi-bin/FaceInfoManager.cgi?action=remove		
Method	GET		
Description	Delete an access user face.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
UserID	string	R	The user's id.
[Response Params] (OK)			
[Example]			
Request	GET http://<server>/cgi-bin/FaceInfoManager.cgi?action=remove&UserID=102		
Response	OK		

16.3.1.8 Find Access User Face

- Start to find access user face

Table 16-67

URL	http://<*server*>/cgi-bin/FaceInfoManager.cgi?action=startFind		
Method	GET		
Description	Start to find access user face.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Condition	object	O	Search condition.
+UserID	string	O	The user's id.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
Token	int	O	The token of this search, use this token to get result and stop search.
Total	int	O	Result num, return 0 if not found.
[Example]			
Request	GET http://<server>/cgi-bin/FaceInfoManager.cgi?action=startFind		
Response	{ "Token": 1234, "Total": 20 }		

- Get the find result

Table 16-68

URL	http://<*server*>/cgi-bin/FaceInfoManager.cgi?action=doFind
Method	GET

Description	Get the find result.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Token	int	R	The token of this search, use this token to get result and stop search.
Offset	int	R	Offset in the result record set, range is [0, Total – 1].
Count	int	R	Count of result to get.
[Response Params] (JSON format)			
Name	Type	R/O	Param Description
Info	array<object>	R	The face information of user.
+UserID	string	R	The user's id.
+MD5	array<string>	O	The user's face photo's MD5 hash string, max array size is 5, max string size is 33.
[Example]			
Request	GET http://<server>/cgi-bin/FaceInfoManager.cgi?action=doFind&Token=1234&Offset=0&Count=20		
Response	{ "Info": [{ "UserID": "102", "MD5": ["xxxx", "xxxx", …] }, { … }, …] }		

- Stop the find session

Table 16-69

URL	http://<*server*>/cgi-bin/FaceInfoManager.cgi?action=stopFind		
Method	GET		
Description	Stop the find session.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
Token	int	R	Token for this search, use this token to get result and stop search.
[Response Params] (OK)			
[Example]			
Request	GET http://<server>/cgi-bin/FaceInfoManager.cgi?action=stopFind&Token=1234		
Response	OK		

16.3.1.9 Add Access User Card and Fingerprint

- Add Access user card only

Table 16-70

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCard
Method	GET
Description	Insert access user card info.
[Request Params] (key=value format in URL)	

Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be "**AccessControlCard**".
CardName	string	R	Access user card name, max string length is 32.
CardNo	string	R	Access user card number.
UserID	string	R	The user's id.
CardStatus	int	R	Card status, 0 means normal, and abnormal status can be sum of following: ● 1 << 0: report the loss of the card ● 1 << 1: the card is withdrawed ● 1 << 2: the card is frozen ● 1 << 3: the card is arrearage ● 1 << 4: the card exceeds the time limit ● 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts.
CardType	int	O	The card type: 0 : Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blacklist Card, 5: Stress Card, 0xff: Mother Card
Password	string	O	The Access card's password. (This param is not supported by video talk device)
Doors	array<int>	O	The index of the doors that this card can open. (This param is not supported by video talk device)
TimeSections	array<int>	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)
VTOPosition	string	O	VTO position number. (This param is not supported by access control device)
ValidDateStart	string	O	The start of valid date, format is "**yyyyMMdd hhmmss**".
ValidDateEnd	string	O	The end of valid date, format is "**yyyyMMdd hhmmss**".
IsValid	bool	O	Is the card still valid. (This param is not supported by video talk device)

[Response Params] (key=value format)			
Name	Type	R/O	Param Description
RecNo	int	R	The record id.

[Example]	
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insert&name=AccessControlCard&CardName=ZhangSan&CardNo=12345&UserID=102&CardStatus=0&CardType=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&VTOPosition=01018001&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811
Response	RecNo=12345

- Add access user card with fingerprint

Table 16-71

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=insertEx&name=AccessControlCard
Method	POST
Description	Insert access user card and fingerprint. (This api is not supported by video talk device)

[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be "**AccessControlCard**".
CardName	string	R	Access user card name, max string length is 32.
CardNo	string	R	Access user card number.
UserID	string	R	The user's id.
CardStatus	int	R	Card status, 0 means normal, and abnormal status can be sum of following: ● 1 << 0: report the loss of the card ● 1 << 1: the card is withdrawed ● 1 << 2: the card is frozen ● 1 << 3: the card is arrearage ● 1 << 4: the card exceeds the time limit ● 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts.
CardType	int	O	The card type: 0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blacklist Card, 5: Stress Card, 0xff: Mother Card
Password	string	O	The Access card's password.
Doors	array<int>	O	The index of the doors that this card can open.
TimeSections	array<int>	O	The index of the time sections of each door that this card can open.
VTOPosition	string	O	VTO position number. (This param is not supported by access control device)
ValidDateStart	string	O	The start of valid date, format is "**yyyyMMdd hhmmss**".
ValidDateEnd	string	O	The end of valid date, format is "**yyyyMMdd hhmmss**".
IsValid	bool	O	Is the card still valid.
FingerprintPacket	object	O	The fingerprint packet info.
+Length	int	O	The length of each fingerprint packet.
+Count	int	O	The count of fingerprint packet.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
RecNo	int	R	The record id.
[Example]			

Request	POST http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=insertEx&name=AccessControlCard&CardName=ZhangSan&CardNo=12345&UserID=102&CardStatus=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&VTOPosition=01018001&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811&FingerprintPacket.Length=500&FingerprintPacket.Count=3 Content-Type: application/octet-stream Content-Length: <length> <fingerprint packet binary data>
Response	RecNo=12345

16.3.1.10 Modify Access User Card and Fingerprint

- Update Access user card only

Table 16-72

URL	http://*<server>*/cgi-bin/recordUpdater.cgi?action=update&name=AccessControlCard		
Method	GET		
Description	Update access user card info. **Note:** You should provide at lease one optional param to update.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be "**AccessControlCard**".
recno	int	R	The record id.
CardName	string	O	Access user card name, max string length is 31.
CardNo	string	O	Access user card number.
UserID	string	O	The user's id.
…<other param>	—	—	…<See above insert command for other params of the record, They are all optional.>
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=update&name=AccessControlCard&recno=12345&CardName=ZhangSan&CardNo=12345&UserID=102&CardStatus=0&CardType=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811		
Response	OK		

- Update access user card and fingerprint

Table 16-73

URL	http://*<server>*/cgi-bin/recordUpdater.cgi?action=updateEx&name=AccessControlCard
Method	POST
Description	Update access user card info and fingerprint. (This api is not supported by video talk device) **Note:** You should provide at lease one optional param to update.

[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be "**AccessControlCard**".
recno	int	R	The record id.
CardName	string	O	Access user card name, max string length is 32.
CardNo	string	O	Access user card number.
UserID	string	O	The user's id.
…<other param>			…<See above insert command for other params of the record. They are all optional.>
FingerprintPacket	object	O	The fingerprint packet info.
+Length	int	O	The length of each fingerprint packet.
+Count	int	O	The count of fingerprint packet.
[Response Params] (OK)			
[Example]			
Request	POST http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=updateEx&name=AccessControlCard&recno=12345&CardName=ZhangSan&CardNo=12345&UserID=102&CardStatus=0&CardType=0&Password=123456&Doors[0]=1&Doors[1]=3&Doors[2]=5&ValidDateStart=20151022%20093811&ValidDateEnd=20151222%20093811&FingerprintPacket.Length=500&FingerprintPacket.Count=3 Content-Type: application/octet-stream Content-Length: <*length*> <fingerprint packet binary data>		
Response	OK		

16.3.1.11 Delete Access User Card and Fingerprint

- Delete Access user card and fingerprint record by recno

Table 16-74

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=remove&name=AccessControlCard		
Method	GET		
Description	Remove the access user card and fingerprint record. (The video talk device can not delete fingerprint data.)		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be "**AccessControlCard**".
recno	int	R	The record id.
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=remove&name=AccessControlCard&recno=12345		

Response	OK

- Delete all Access user card and fingerprint records

Table 16-75

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=clear&name=AccessControlCard		
Method	GET		
Description	Remove all the access user card and fingerprint records. (The video talk device can not delete fingerprint data.)		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be "**AccessControlCard**".
[Response Params] (OK)			
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=clear&name=AccessControlCard		
Response	OK		

16.3.1.12 Find Access User Card and Fingerprint

- Find Access user card by condition

Table 16-76

URL	http://<*server*>/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCard		
Method	GET		
Description	Find Access user card record by condition.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Record Name, should be "**AccessControlCard**".
count	int	O	Max result to return, default is 1024.
StartTime	string	O	The start of the record's CreateTime.
EndTime	string	O	The End of the record's CreateTime.
condition	object	O	Search condition.
+CardNo	string	O	Access user card number.
+UserID	string	O	The user's id.
+IsValid	bool	O	The access card valid or not.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
totalCount	int	O	The total record num that find.
found	int	O	The record num that returned.
records	array<object>	R	The records that returned.
+RecNo	int	R	The record id.
+CreateTime	int	R	The create time of record.
+CardName	string	R	The access user card name, max string length is 32.
+CardNo	string	R	The access user card number.
+UserID	string	R	The user's id.

+CardStatus	int	R	Card status, 0 means normal, and abnormal status can be sum of following: ● 1 << 0: report the loss of the card ● 1 << 1: the card is withdrawed ● 1 << 2: the card is frozen ● 1 << 3: the card is arrearage ● 1 << 4: the card exceeds the time limit ● 1 << 5: the card is arrearage but still can open the door, and there will be voice prompts.
+CardType	int	O	The card type: 0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blacklist Card, 5: Stress Card, 0xff: Mother Card
+Password	string	O	The Access card's password. (This param is not supported by video talk device)
+Doors	array<int>	O	The index of the doors that this card can open. (This param is not supported by video talk device)
+TimeSections	array<int>	O	The index of the time sections of each door that this card can open. (This param is not supported by video talk device)
+VTOPosition	string	O	VTO position number. (This param is not supported by access control device)
+ValidDateStart	string	O	The start of valid date, format is "**yyyyMMdd hhmmss**".
+ValidDateEnd	string	O	The end of valid date, format is "**yyyyMMdd hhmmss**".
+IsValid	bool	O	Is the card still valid. (This param is not supported by video talk device)
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCard&condition.UserID=103&StartTime=123456700&EndTime=123456800&count=100		

Response	totalCount=1000 found=100 records[0].RecNo=12345 records[0].CreateTime=123456789 records[0].CardName=ZhangSan records[0].CardNo=300 records[0].UserID=103 records[0].CardStatus=0 records[0].CardType=0 records[0].Doors[0]=1 records[0].Doors[1]=3 records[0].Doors[2]=5 records[0].VTOPosition=01018001 records[0].ValidStart=20151022 093811 records[0].ValidEnd=20151222 093811 records[0].IsValid=true … records[1].RecNo=13579 records[1].CreateTime=123456799 records[1].StaticsticsTime=123456799 records[1].CardName=ZhangSan records[1].CardNo=302 records[1].UserID=103 records[1].CardStatus=0 records[1].CardType=0 records[1].Doors[0]=2 records[1].Doors[1]=4 records[1].Doors[2]=6 records[1].VTOPosition=01018002 records[1].ValidStart=20151022 093811 records[1].ValidEnd=20151222 093811 records[1].IsValid=true …

- Find Access user card by recno

Table 16-77

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=get&name=AccessControlCard		
Method	GET		
Description	Find Access user card record by recno.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Record Name, should be "**AccessControlCard**".
recno	int	R	The record id.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
record	object	R	The record that returned.
+RecNo	int	R	The record id.

+CreateTime	int	R	The create time of record.
+CardName	string	R	The access user card name, max string length is 32.
+CardNo	string	R	The access user card number.
+UserID	string	R	The user's id.
…<other param>	—	—	…<See above find command for other params of the record.>
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=get&name=AccessControlCard&recno=4		
Response	record.RecNo=4 record.CreateTime=123456789 record.CardName=ZhangSan record.CardNo=300 record.UserID=103 record.CardStatus=0 record.CardType=0 record.Doors[0]=1 record.Doors[1]=3 record.Doors[2]=5 record.VTOPosition=01018001 record.ValidStart=20151022 093811 record.ValidEnd=20151222 093811 record.IsValid=true …		

- Find Access user card and fingerprint by recno

Table 16-78

URL	http://<*server*>/cgi-bin/recordUpdater.cgi?action=getEx&name=AccessControlCard		
Method	GET		
Description	Find Access user card and fingerprint record by recno. (This api is not supported by video talk device)		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Record Name, should be "**AccessControlCard**".
recno	int	R	The record id.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
record	object	R	The record that returned.
+RecNo	int	R	The record id.
+CreateTime	int	R	The create time of record.
+CardName	string	R	The access user card name, max string length is 32.
+CardNo	string	R	The access user card number.
+UserID	string	R	The user's id.
…<other param>	—	—	…<See above find command for other params of the record.>
+FingerprintPacket	object	O	The fingerprint packet info.
++Length	int	O	The length of each fingerprint packet.
++Count	int	O	The count of fingerprint packet.

[Example]	
Request	GET http://192.168.1.108/cgi-bin/recordUpdater.cgi?action=getEx&name=AccessControlCard&recno=4
Response	HTTP/1.1 200 OK Server: Device/1.0 Content-Type: multipart/x-mixed-replace; boundary=<*boundary*> Content-Length: <length> --<*boundary*> Content-Type: text/plain Content-Length: <length> record.RecNo=4 record.CreateTime=123456789 record.CardName=ZhangSan record.CardNo=300 record.UserID=103 record.CardStatus=0 record.CardType=0 record.Doors[0]=1 record.Doors[1]=3 record.Doors[2]=5 record.VTOPosition=01018001 record.ValidStart=20151022 093811 record.ValidEnd=20151222 093811 record.IsValid=true … record.FingerprintPacket.Length=500 record.FingerprintPacket.Count=3 --<*boundary*> Content-Type: application/octet-stream Content-Length: <length> <fingerprint packet binary data> --<*boundary*>--

16.3.1.13 Get the Total Number of Records of Access User Card and Fingerprint

Table 16-79

URL	http://<*server*>/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCard
Method	GET
Description	Get the total number of records of the access user card record.

[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	Access user card and fingerprint record name, should be "**AccessControlCard**".
[Response Params] (key=value)			
count	int	R	The total number of records.
[Example]			
Request	GET http://192.168.1.108/cgi-bin/recordFinder.cgi?action=getQuerySize&name=AccessControlCard		
Response	count=150		

16.3.2 Access control

16.3.2.1 [Event] AccessControl

Table 16-80

Usage	Refer to "4.4.3 Subscribe to Snapshot"for how to subscribe event		
Description	When user trying to open the door, send this event.		
[Event Params] (key=value format)			
Name	Type	R/O	Param Description
Events	array<object>	R	Event info array
+EventBaseInfo	object	R	Base info of event
++Code	String	R	Event Code. It should be **AccessControl**
++Action	String	R	Event Action. It can be: "Start", "Stop", "Pulse"
++Index	int	O	The channel index relate to this event.
+RecNo	int	R	The record id
+Name	string	O	The name of the door
+Type	string	R	The event type. It can be: "**Entry**", "**Exit**"
+Status	int	O	Open door result, can be: 0—failed, 1—success If this param does not exist, that means success.
+Method	int	R	Open door method, can be: 0: by password 1: by access card 2: by access card and then password 3: by password and then access card 6: by fingerprint 15: by face recognition
+CardNo	string	O	Card number if the door is opened by card
+UserID	string	R	The user id
[Example]			
Event	--<*boundary*> Content-Type: text/plain Content-Length: <length>		

	Events[0].EventBaseInfo.Code=TrafficJam Events[0].EventBaseInfo.Action=Pulse Events[0].EventBaseInfo.Index=0 Events[0].RecNo=123 Events[0].Name=Door1 Events[0].Type=Entry Events[0].Status=1 Events[0].Method=1 Events[0].CardNo=09DDAABB Events[0].UserID=101 … --<*boundary*> Content-Type: image/jpeg Content-Length: <image size> <Jpeg image data> --<*boundary*>

16.3.2.2 Find AccessControl Record

Table 16-81

URL	http://<*server*>/cgi-bin/recordFinder.cgi?action=find&name=<**RecordName**>		
Method	GET		
Description	Find AccessControl record.		
[Request Params] (key=value format in URL)			
Name	Type	R/O	Param Description
name	string	R	The record table name. It should be "**AccessControlCardRec**".
count	int	O	Max result to return, default is 1024.
StartTime	string	O	The start of the record's CreateTime.
EndTime	string	O	The end of the record's CreateTime.
condition	object	O	Search condition.
+CardNo	string	O	Access user card number.
[Response Params] (key=value format)			
Name	Type	R/O	Param Description
totalCount	int	O	Total record num that find.
found	int	O	Record num that returned.
records	array<object>	R	The records that returned.
+RecNo	int	R	Record id.
+CreateTime	int	R	Record create time, UTC time.
+CardNo	int	R	Access user card number.
+CardName	string	O	Access user card name, max string length is 31.
+CardType	int	O	The card type: 0: Normal Card, 1: VIP Card, 2: Visitor Card, 3: Patrol Card, 4: Blacklist Card, 5: Stress Card, 0xff: Mother Card

+Password	string	O	The Access card's password.
+UserID	string	R	The user's id.
+Type	string	R	The event type. It can be: "**Entry**", "**Exit**".
+Status	int	O	Open door result. It can be: 0—failed, 1—success. If this param does not exist, that means success.
+Method	int	R	Open door method. It can be: ● 0: by password ● 1: by access card ● 2: by access card and then password ● 3: by password and then access card ● 6: by fingerprint ● 15: by face recognition
+Door	int	O	The index of the door. (This param is not supported by video talk device)
+ReaderID	string	O	The access user card ID of reader. (This param is not supported by video talk device)
+ErrorCode	int	O	The error code, valid only when Status is 0.
+URL	string	O	The picture's URL, max string length is 127. (This param is not supported by access control device)
+RecordURL	string	O	The record video's URL, max string length is 127. (This param is not supported by access control device)

[Example]

Request	GET http://192.168.1.108/cgi-bin/recordFinder.cgi?action=find&name=AccessControlCardRec&StartTime=123456700&EndTime=123456800&condition.CardNo=12001&count=100
Response	totalCount=1000 found=100 records[0].RecNo=12345 records[0].CreateTime=123456789 records[0].CardNo=12001 records[0].CardName=ZhangSan records[0].UserID=ZhangSan records[0].Type=Entry records[0].Method=1 … records[1].RecNo=13579 records[1].CreateTime=123456799 records[1].CardNo=12001 records[1].CardName=ZhangSan records[1].UserID=ZhangSan records[1].Type=Exit records[1].Method=1 …