Network IP Camera Application Programming Interface (NIPCA)

Ver. 1.9.6

Document History

Version	Date	Comment
0.99a	2007-11-09	Focus on the configuration settings.
0.99b	2007-11-20	Add the Valid values. Add RS-485 commands. Refine all document.
0.99c	2007-11-21	Modify HTTP status codes, basic info, datetime DST, motion detection.
0.99d	2007-12-14	2.3: Modify HTTP status codes description. Add descriptions and examples of parameters and values. 3.3.4 3.3.5: Modify flicker, autoexposure for sensor_info.cgi sensor.cgi. 3.4.8: Modify upnpav, upnpcp for upnp.cgi. 5.1.3: Remove redundant penable, tenable, zenable for ptz_step.cgi. 5.1.5: Add p,t,z position for ptz_preset_list.cgi. 6.1.2: Add keep_alive for notify_stream.cgi.
0.99e	2007-12-26	Add the 3.5 event handling.
1.0	2008-01-14	3.1.3 Modify the request to /users/verify.cgi 3.1.6 3.1.7 Modify method and offset. Add the ID of the table dynamic DNS service providers.
1.1	2008-01-31	2.1 Fix the POST Content-Type to application/x-www-form-urlencoded. Add the ACS Stream Header.
1.2	2008-02-21	2.1 Fix the POST parameters.
1.3	2008-05-26	3.5.1 change the definition of macro block size. 3.3.4 add hue, autoexposure, autogainctrl 3.3.5 add hue, autogainctrl 3.6.8 add delaytime
1.4	2008-05-16	3.1.1 get basic information: add ‘nipca’ item 3.1.1 add ‘videoout’ item 3.3.4 add ‘inputsize’ and ‘videooutformat’ item. 3.3.5 add ‘color’ item. 3.5.1 add more actions like cifs_rec, cifs_shot… 3.5.4 add more actions and the field prerecord and postrecord. 3.5.15, 3.5.16 add keep space item. 4.1.7 add get video stream of associated profile 6.1.1, 6.1.2 add mdv# item 7.1 change url of rtsp: mpeg4 -> mp4 mjpeg -> jpeg add /live# url

1.5	2008-10-17	3.3.1, 3.3.2 add vprofileformat for video stream cgi 4.1.7 3.5.3, 3.5.4 add sw_input event 3.5.3, 3.5.4 change ‘actions’ and ‘action’ keyword to be ‘handlers’ and ‘handler’ 5.3 add an software event trigger function 4.1.5, 4.1.6 revise the part of format description 5.2.1 unify the speed range to 1-10 4.1.8 add put audio upstream (two-way audio talk) 4.1.9 add H.264 streaming cgi url Add 7.1.1-7.1.3 to support customized url entry of RTSP live stream. 3.4.1 add httpexternalport, rtspport and rtspexternalport.
1.6	2009-02-17	5.3.1 add ‘trigger’ item to indicate that client want to turn this event on or off. 3.3.1, 3.3.2, 4.1.7 Refine the definition of vprofileformat 5.1.1 add item “customizedhome” to indicate whether camera support 5.1.9 function or not. 5.1.9 add PTZ home manifest command. 3.1.13 add ‘reset sensor to default configuration’ function 3.3.5, 3.3.6 add videoinformat for some video server models 3.3.14, 3.3.15 add IR LED setting functions 3.3.16, 3.3.17 add ICR setting functions
1.7	2009-06-10	3.3.18, 3.3.19 add authentication control for live video and snapshot 3.1.1 add field ‘product’, ‘brand’ to basic information 5.1.10, 5.1.11 add Auto Patrol/Auto Pan for PTZ control function 5.1.12 Configure auot patrol preset sequence order. 3.6.5 add example html to submit firmware to ip camera. 4.1.2 add ‘profileid’ optional parameter 4.1.3 change this section to MPEG-4 elementary stream CGI. 4.1.5, 4.1.6, 4.1.9 refine the description of parameter ‘profileid’ 4.1.7 modify the format of stream with profile M-JPEG (replaced by ACVS wrapped stream) 4.1.11 add audio profile stream CGI 8.2 Add two more frame type for ACVS header. 3.3.4, 3.3.5, 3.3.6 add sharpness 3.4.14, 3.5.15 add wireless strength function and wireless site survey 3.7 SD card operations added
1.8	2009-09-30	4.1.8 Modify audio uploading method 3.7 Update whole SD card section. 3.5.11 Add enable, prefix and interval field. 3.5.13 Add enable field. 3.5.15 Add enable field.
1.9	2010-4-22	5.1.13 get, set the type of focus function: auto focus or manual focus 5.1.14 adjust the focus manually, focus in or focus out from current position Remove unsupport list. Fix wording and wrong statements.
1.9.1	2010-7-7	3.5.3. Newly add get Recorder action 3.5.4. Newly add set Recorder action 3.5.5. Newly add get Snapshot action 3.5.6. Newly add set Snapshot action 3.5.7. Newly add get Alarm out action 3.5.8. Newly add set alarm out action

1.9.2	2010-8-4	3.4.15. Newly add get HTTPS configuration 3.4.16. Newly add set HTTPS configuration
1.9.3	2010-10-31	3.3.2. 3.3.3.Modify quality not only for MJPEG and Newly add qualitymodes
1.9.4	2011-7-01	3.3.2. Newly add get sensor output configuration 3.3.3. Newly add set sensor output configuration 3.3.1. Modify the resolutions and vprofileres# description. 3.3.4. Modify the resolution description. 3.3.6. Add flicker, mirror, flip, color 5. Add /ptz/ directory to PTZ privilege group. 5.1.15., 5.1.16. Add get/set PTZ privilege group. 3.3.1. Add parameters: cur_micvol, cur_speakervol 5.3. PTDC Pan/Tilt get information 5.4. PTDC Pan/Tilt set information 5.5. PTDC Pan/Tilt other Parts 5.6. PTDC Zoom/Focus/Focus Type get information 5.7. PTDC Zoom/Focus/Focus Type set information
1.9.5	2011-11-22	3.3.4 Newly add get video type. 3.3.5 Newly add set video type. 3.3.18 Newly add set IR LED. 3.3.19 Newly add get ICR(Infrared Cut filter Removal) settings. 3.3.20 Newly add set ICR(Infrared Cut filter Removal) settings. 3.3.23 Newly add query Privacy mask information 3.3.24 Newly add get Privacy mask 3.3.25 Newly add set Privacy mask 3.4.17 Newly add list all the IP access list. 3.4.18 Newly add add, delete acces IP. 3.6.10 Newly add get Privacy mode configuration. 3.6.11 Newly add set Privacy mode configuration. 3.6.12 Newly add get TV output. 3.6.13 Newly add eet TV output. 3.6.14 Newly add get DC power. 3.6.15 Newly add set DC power. 3.6.16 Newly add get device timestamp. 3.6.17 Newly add set device timestamp. 4.1.10 Newly add get audio MS-ADPCM stream 4.1.11 Newly add get audio MU-LAW stream 4.1.12 Newly add get audio AAC stream 4.1.13 Newly add get audio A-LAW stream 5.1.17 Newly add query focus information. 5.1.18 Newly add get the current position of focus. 5.1.19 Newly add set absolutely position of focus. 5.1.20 Newly add fine-tune focus automatically. 5.1.21 Newly add PTZ direction of movement. 5.8.1 Newly add get the current digital PTZ position. 5.8.2 Newly add add, delete or goto a digital PTZ preset. 5.8.3 Newly add move digital PTZ absolutely. 5.8.4 Newly add move digital PTZ relatively. 5.8.5 Newly add digital PTZ autopan. 5.8.6 Newly add digital PTZ sequence. 3.1.1 Add field “focus” “pir” “irc” and “ir” to basic information. 3.3.1 Add “resolutionlist#”, “frameratelist#” fields. 3.3.6, 3.3.7 Add “viewwindow” fields.

		3.3.8 Add “wds” “exposuremode” , “maxshutter”, “minshutter”. “maxgain”, “noisereduction”, “wdrlevel” fields. 3.3.9, 3.3.10 Add “wds” “exposuremode” , “maxshutter”, “minshutter”. “maxgain”, “noisereduction”, “wdrenable”, “wdrlevel” fields. 3.3.11, 3.3.12 Add audio codec format value: MU-LAW, A-LAW. 3.5.1, 3.5.2 Add “percentage” “pir” fields. 3.5.4 Add “fileLenMin” field for setting recording file size by minute(s). 3.7.1 Add status values. 3.7.2 Specify response format. 6.1.1, 6.1.2 Add “irled” “autofocusbusy” field. 8.2 Add audio data format value of ACS audio header to support AAC and A-LAW audio codec format. 3.5.3, 3.5.4, 3.5.5 Errata correction. Refine TimeZone and Day Light Saving Time as below 1. TimeZone2: from “Midway Island, Samoa” to “Samoa” 2. TimeZone14: from “Bogota, Lima, Quito, Rio Branco” to “Bogota, Lima, Quito” 3. TimeZone17: from “Caracas, La Paz” to “La Paz, Georgetown” 4. TimeZone22: from “Buenos Aires, Georgetown” to “Buenos Aires” 5. TimeZone30: from “Casablanca, Monrovia, Reykjavik” to “Monrovia, Reykjavik” 6. TimeZone41: from “GMT+02:00” to “GMT+03:00” 7. TimeZone43: from “GMT+02:00” to “GMT+01:00” 8. TimeZone47: from “GMT+03:00” to “GMT+04:00” 9. TimeZone55: from “GMT+05:00 Ekaterinburg” to “GMT+06:00 Yekaterinburg” 10. TimeZone61: from “(GMT+06:00) Almaty, Novosibirsk” to “(GMT+07:00) Novosibirsk” 11. TimeZone63: from “GMT+07:00” to “GMT+08:00” 12. TimeZone67: from “GMT+08:00 Irkutsk, Ulaan Bataar” to “GMT+09:00 Irkutsk” 13. TimeZone70: from “GMT+09:00” to “GMT+10:00” 14. TimeZone80: from “(GMT+11:00) Magadan, Solomon Is., New Caledonia” to “(GMT+12:00) Magadan” 15. TimeZone81: from “Fiji, Kamchatka, Marshall Is.” To “Fiji” 16. TimeZone83: from “Nuku’alofa” to “Nukualofa” 17. TimeZone84: newly add “(GMT-04:30) Caracas” 18. TimeZone85: newly add “(GMT+11:00) Solomon Is., New Caledonia” 19. TimeZone86: newly add “(GMT) Casablanca” 20. TimeZone87: newly add “(GMT+08:00) Ulaanbaatar”
1.9.6	2014-3-13	Refine whole document (description and indent) 2.1 Add url encode description. 2.2 Add xml encode description. 2.3 Add descriptions of HTTP status code 307 and 409. 3.1.1 Add parameter “mic, led, td, playing_music” to basic information. 3.1.5 Update Value Definition, Add parameter utcdate and utctime. 3.1.6 Update Value Definition 3.1.7 Add interface Get Camera Capability. 3.2.1 Add Value Listing Sequence 3.2.2 Add parameter newname. 3.2.4 Add Value Listing Sequence

		3.3.1 Add Value Listing Sequence & Update Value Definition Add parameter codeclist#, qualitylist# , vbitratelist# and vban. 3.3.8 Add Value Listing Sequence 3.3.11~3.3.12 Add parameter “bit_depth” 3.3.13 Add Value Listing Sequence & Update Definition 3.3.15 Add Value Listing Sequence & Update Definition 3.3.19 Add parameter “light_threshold_list” and “light_threshold” 3.3.26 Add interface “Get Thermal Detection” 3.3.27 Add interface “Set Thermal Detection” 3.3.28 Add interface “Get IR LED Illumination Distance” 3.3.29 Add interface “Set IR LED Illumination Distance” 3.3.30 Add interface “Get List of Default System Music” 3.3.31 Add interface “Play Music from Default System Music” 3.3.32 Add interface “Stop Play Music” 3.3.33 Add interface “Get Audio Player Mode” 3.4.1 Update Value Definition 3.4.3 Update Value Definition 3.4.5 Update Value Definition 3.4.11 Update Value Definition 3.4.19 Add Wireless Get AP-Mode Setting 3.4.20 Add Wireless Set AP-Mode 3.4.21 Add interface “Get Wireless STA Status” 3.5 Modify chapter title. 3.5.1 Add parameter pir_sensitivity. 3.5.3 ~ 3.5.4 Add parameter fileFormat, bySound. extraLight 3.5.5 Add parameter smtpEncrypt, ftpInterval, ftpIgnore, smtpInterval, smtpIgnore. bySound, extraLight 3.5.6 Add parameter smtpPort1, smtpEncrypt, ftpInterval, ftpIgnore, smtpInterval, smtpIgnore, bySound, extraLight. Refine description of continues and schedule parameters. 3.5.7 Add parameter bySound 3.5.8 Add parameter out1BySound. 3.5.9 Add interface “Get Sound Detection” 3.5.10 Add interface “Set Sound Detection” 3.5.11 Add interface “Get Environmental Sound Level” 3.6.5 Add interface “Get Led Mode” 3.6.6 Add interface “Set Led Mode” 3.6.7 Re-index 3.6.5(Firmware Upgrade) as 3.6.7 3.6.8 Re-index 3.6.6(Reboot camera) as 3.6.8 3.6.9 Re-index 3.6.7(Reset All Configuration to Factory Default) as 3.6.9 3.6.10 Re-index 3.6.8(Get RS-485 Settings) as 3.6.10, Update Value Definition 3.6.11 Re-index 3.6.9(Set RS-485 Setting) as 3.6.11 3.6.12 Re-index 3.6.10(Get Privacy Mode Setting) as 3.6.12 3.6.13 Re-index 3.6.11(Set Privacy Mode Setting) as 3.6.13 3.6.14 Re-index 3.6.12(Get TV Output) as 3.6.14 3.6.15 Re-index 3.6.13(Set TV Output) as 3.6.15 3.6.16 Re-index 3.6.14(Get DC Power) as 3.6.16 3.6.17 Re-index 3.6.15(Set DC Power) as 3.6.17 3.6.18 Re-index 3.6.16(Get Device Timestamp) as 3.6.18 3.6.19 Re-index 3.6.17(Set Device Timestamp) as 3.6.19 3.7.6 Add interface “Upload A File to SD Card”

		3.7.7 Add interface “Get List of Music from SD Card” 3.7.8 Add interface “Play Music from SD Card” 3.8 Add new interfaces for camera log 3.8.1 Add get syslog setting 3.8.2 Add set syslog setting 3.8.3 Add get event log 3.9 Add new interfaces for fisheye operation 3.9.1 Add interface “Query Mount Type Information” 3.9.2 Add interface “Get Current Mount Type” 3.9.3 Add interface “Set Mount Type” 3.9.4 Add interface “Query Fisheye Display Mode Information of Live Video” 3.9.5 Add interface “Get Fisheye Display Mode of Live Video” 3.9.6 Add interface “Set Fisheye Display Mode of Live Video” 3.9.7 Add interface “Query Fisheye Split Window Information” 3.9.8 Add interface “Get Current Fisheye Split Window” 3.9.9 Add interface “Set Fisheye Split Window” 4.1.5 Refine interface description 4.1.6 Refine interface description 4.1.7 Add interface speager2.cgi and dgtalkie.cgi 4.1.15 Add interface dgtalkie_info.cgi 4.1.16 Add interface “Get Transport Stream Audio/Video Streaming” 5.1.5 Add Value Listing Sequence, remove list order request. 5.1.6 Update Value Definition 5.1.12 Refine description, modify GET to GET/POST 5.1.17 ~ 5.1.20 Duplicate interfaces to path /ptz 5.6.7 CGI name error – get_focus_type 5.6.9 Title error – Get Zoom Step Accuracy 5.6.12 Title error – Get Home Zoom Step 5.8.1 Add Parameters ‘profileid’ 5.8.3 CGI name error – GET /config/digital_ptz_move.cgi 6.1.1 Add parameter pir, audio_detected, audio_detect_val, speaker_occupied, mic_muted, td, tpC, tpF, playing_music, white_light_led. Remove parameter “mdetecting”. Duplicate interface to path /users. Remove status “off” from parameter storagefull and storagefail. 6.1.2 Synchronize field with 6.1.1 notify.cgi, keep information the same. Remove parameter “usbstatus” because of incorrect define. Remove status “off” from parameter storagefull and storagefail. Duplicate interface to path /users. 7.1.3 Refine interface description. 8.1 Add/Modify time zone list. 8.3 Add Appendix “Fisheye Display Mode Schematic Information”

Contents

1. Overview

Network IP Camera Access Application Programming Interface (NIPCA-API) is a HTTP-based API of IP camera product. By using NIPCA API, users can write their own programs to configure camera, access multimedia streaming and control facilities.

Most NIPCA API use query string, INI or XML format for transporting HTTP-base message. But some of them may transport binary data between each other, such as get streaming, download file or others. We will describe the general HTTP request format in the following chapter.

For streaming API, it will output streaming data to client directly. The streaming output format depends on each streaming API of IP camera. Different model may have different output format. This document only provides a general entry point to let the IP camera output streaming via a permanent HTTP connection.

Furthermore, we also provide RTSP interface. User can also access the streaming through RTSP interface.

1.1. API Versions

Because some models produced early than the first version of NIPCA released. NIPCA may not apply to these kinds of models. We may also publish further version in the future. Due to this reason, there may have some difference between different versions. However, all of our products shall provide API version information with every firmware version.

1.2. Valid Values

The following valid values are used in this document:

Values	Description
Integer	Any number between -2147483647 (-2^{31}-1) and 2147483647 (2^{31}-1).
m ... n	Any number between number m and number n.
#	A number equals or greater than 0
String	Any string encoded by UTF-8
IP address	IP address format sting made up of four decimal numbers, each range from 0 to 255, separate by dots. Example: 192.168.0.90
MAC Address	“MAC Address” is a string that made up of six groups of two-digit hexadecimal numbers, separate by colons. Example: 00:40:8C:CD:00:00
Time	A time format string made up of hours, minutes and seconds, separate by colons like hh:mm:ss. Example: 23:01:14
Date	A date format string made up of year, month and day, separate by dashes like yyyy-mm-dd. Example: 2004-02-16
<value 1>, <value 2>, <value 3>, ...	Enumeration. Only when given values are valid.
<italic string>	Every italic strings inside brackets including the brackets should be replaced by proper values.

2. HTTP Interface

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. NIPCA use the HTTP protocol to be an interface for transport data and message between user and IP camera. The IP camera provides a service to receive HTTP requests and respond messages to clients.

This chapter will describe the formats of request and response, also comprising different request formats of NIPCA. Although IP camera supports both of HTTP/1.0 and HTTP/1.1, we strongly recommend using HTTP/1.1 rather than HTTP/1.0 because of compatibility consideration. For more information about HTTP/1.1, please refer to RFC 2616.

2.1. Request Messages

For query information of IP camera, please use the following syntax:

```
GET http://<camera name>/<CGI-URL>?<parameter>=<value> HTTP/1.1<CRLF>
Authorization: <authentication info><CRLF>
Host: <camera ip-adress><CRLF>
<CRLF>
```

Where,

<CGI-URL>: A URL of CGI. For example, get basic information is "/common/info.cgi".

<value>: Value assign to parameter, should be "URL encode"

<authentication info>: basic or digest. (Authorization is optional for some CGIs)

<CRLF>: **Carriage Return** and **Line Feed** (\r\n).

In order to set values in the IP camera, user may use HTTP method GET, the syntax like below:

```
GET http://<camera name>/<CGI-URL>?<parameter>=<value>[&<parameter>=<value>...] HTTP/1.1<CRLF>
Authorization: <authentication info> <basic-cookie><CRLF>
Host: <camera ip-adress><CRLF>
<CRLF>
```

Or HTTP method POST, the syntax like below:

```
POST http://<camera name>/<CGI-URL> HTTP/1.1<CRLF>
Authorization: <authentication info> <basic-cookie><CRLF>
Host: <camera ip-adress><CRLF>
Content-Type: application/x-www-form-urlencoded<CRLF>
Content-Length: <body length><CRLF>
<CRLF>
<parameter>=<value>[&<parameter>=<value>]
```

Where,

<body length>: length of entity body.

<parameter>: This field will be described in the following chapters. Valid characters only include alphabets ([A-Za-z]), digits ([0-9]) and underline(_). There is no such restriction for *<value>*. The content part of the post message should be encoded with "url-encoding" function.

2.2. Response Messages

While IP Camera receives request message from user, it will execute related action, and respond result message to user.

Most of NIPCA functions use following two kinds of response formats - INI or XML format. Please refer to the example below:

(However, some NIPCA functions may response binary data if necessary.)

INI format response:

```
HTTP/1.1 <HTTP code> <HTTP text><CRLF>
Content-Type: text/plain<CRLF>
Content-Length: <body length><CRLF>
<CRLF>
<parameter>=<values><CRLF>
...
```

XML format response:

```
HTTP/1.1 <HTTP code> <HTTP text><CRLF>
Content-Type: text/xml<CRLF>
Content-Length: <body length><CRLF>
<CRLF>
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="<CGI_NAME>"?>
<root>
<common>
...
</common>
<config>
...
</config>
</root>
```

Please note that if response in XML format, it should execute XML encode before respond to client.

2.3. Response Status Codes

The API status codes are defined below.

Table 1: HTTP Status Codes

HTTP code	HTTP text	Description
200	OK	Standard response for successful HTTP requests, but an application error may occur, please refer to each CGI response.
307	Temporary Redirect	The requested resource resides temporarily under different URI.
400	Bad Request	The request cannot be fulfilled due to bad syntax.
401	Unauthorized	Authentication is required and has failed or has not yet been provided.
404	Not Found	The required API is not supported for this IP camera.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource
500	Internal Server Error	IP camera encountered an internal error or the API cannot get the correct status.
503	Service Unavailable	IP camera is unable to handle the request due to temporary overload.

3. Configuration API

The CGIs under folder "/config" can only be accessed by administrators. Most of the CGIs are one-shot commands, which only return current configurations and status of IP camera and terminated. If user needs to monitor camera status for a long time, please use 6.1.2 notify_stream.cgi instead.

3.1. Device Information

3.1.1. Get Basic Information

Request:

GET /common/info.cgi

(No authentication required.)

Response:

Name	Value	Description
model	String	Model name
product	String	Product name of camera
brand	String	Brand name
version	String	Version number of firmware
build	String	Firmware build number
nipca	String	Version number of NIPCA supported (e.g. 1.2, 1.4)
name	String	Camera name (hostname).
location	String	Camera location
macaddr	MAC address	Media access control address
ipaddr	IP address	IP address of current active network interface. (Please note that this will not be an IP address of PPPoE.)
netmask	IP address	Subnet mask
gateway	IP address	The gateway address (or default gateway) is a router interface connected to the local network, which sends packets out of the local network.
wireless	yes, no	Wireless is enable/disable. (Omitted if the IP camera doesn't have a wireless.)
ptz	P, T, Z	Only show supported Pan or Tilt or Zoom. For example, ptz=P,T (Omitted if IP camera does not have PTZ)
focus	yes, no	Focus function is enabled or not. (Omitted if IP camera doesn't have a focus function.)
inputs	#	The number of Alarm IN.
outputs	#	The number of Alarm OUT.
speaker	yes, no	Support "speaker" or not
videoout	yes, no	Support "video out" or not
pir	yes, no	Support "PIR" (Passive Infrared Sensor) or not
icr	yes, no	Support "ICR" (Infrared Cut Filter Removal) or not.
ir	yes, no	Support "IR" (Infrared) or not
mic	yes, no	Support "microphone" or not.
led	yes,no	Support "special purpose led" or not.
td	yes,no	Support "thermal detection" or not.
playing_music	yes,no	Support "playing music" or not.

3.1.2. Quickly Verify User

Request:
GET /users/verify.cgi

Response:

Name	Value	Description
group	String	Group name of specified user in HTTP Authorization header field.

If authorization fails, IP Camera will return HTTP/1.1 401 Unauthorized.

3.1.3. Get Camera Info

Request:
GET /config/camera_info.cgi

Response:

Name	Value	Description
name	String	Camera name (hostname)
location	String	Camera location

3.1.4. Set Camera Info

Request:
GET/POST /config/camera_info.cgi

Parameters:

Please refer to the table 3.1.3

Response:
Please refer to the table 3.1.3

3.1.5. Get System Date and Time

Request:
GET /config/datetime.cgi

Response:

Name	Value	Description
method	0, 1	0: disable ntpd 1: enable ntpd
timeserver	Host or IP address	Host or IP address of NTP server.
timezone	#	Time zone ID, please refer to 8.1 Table 1 **Time Zone**
utcdate	Date	yyyy-mm-dd (UTC Date)
utctime	Time	hh:mm:ss (UTC Time)
date	Date	yyyy-mm-dd (Local Date)
time	Time	hh:mm:ss (Local Time)
dstenable	no, yes	Disable or enable DST (Daylight Saving Time)
dstauto	no, yes	Set DST automatically
offset	Time	The amount of time that how long is clock should be turned back/forward (hh:mm). According by DST.
starttime	m.w.d/hh:mm:ss	The time when DST should be enabled in the format m.w.d/hh:mm:ss day d (0 ... 6) of week w (1 ... 5) of month m (1 ... 12). d=0 is a Sunday.
stoptime	m.w.d/hh:mm:ss	Indicate the time, which should disable DST. The format is as same as above.

3.1.6. Set System Date and Time

Request:

GET/POST /config/datetime.cgi

Parameters:

Name	Value	Description
method	0, 1, 2	0: disable ntpd 1: enable ntpd 2: manual setting, date and time are required.
timeserver	Host or IP address	Host or IP address of NTP server
timezone	1 ... 83	Time zone ID, please refer to 8.1 Table 1 **Time Zone**
date	Date	yyyy-mm-dd
time	Time	hh:mm:ss
dstenable	no, yes	Disable/enable DST (Daylight Saving Time)
dstauto	no, yes	Set DST automatically
offset	Time	The amount of time that how long is clock should be turned back/forward (hh:mm). According by DST.
starttime	m.w.d/hh:mm:ss	The time when DST should be enabled in the format m.w.d/hh:mm:ss day d (0 ... 6) of week w (1 ... 5) of month m (1 ... 12). d=0 is a Sunday.
stoptime	m.w.d/hh:mm:ss	Indicate the time, which should disable DST. The format is 是 as same as above.

Response:

Please refer to the table 3.1.5.

3.1.7. Get Camera Capability

Request:

GET /users/capability.cgi

Response:

Name	Value	Description
speaker_codec	pcm,ms_adpcm,aac,amr,ulaw,mp3	Acceptable encode audio format. Each items separated by ","(Comma) without space. Example: speaker_codec=pcm,ms_adpcm,aac,…etc
speaker_data_length	String	Lists of acceptable audio data length, 1024 (bytes) is preferred. Each date length (integer number) separated by ","(comma) without blank. Example: speaker_data_length=256,512,1024,2048,…etc (Relative to 8.2 ACS Audio header field "ulDataLength")
speaker_smaple_rate	String	Acceptable audio sample rate, separate by ","(Comma) without blank. For example: sample_rate=8000,16000
speaker_bit_depth	String	Acceptable audio bit depth(8, 16, 24), separate by ","(Comma) without blank. For example: bit_depth=8,16

3.2. Users and Groups

3.2.1. Get Users

Request:

GET /config/user_list.cgi

Parameters:

None or

name=<username>

Response:

If no request parameter

Name	Value	Description
users	#	The total number of users.
<username> ...	<group name> ...	For example, admin=administrator It will display all user names line by line. (Order by admin, 0~9, A~Z, a~z)

Else if request parameter is name

Name	Value	Description
group	String	The group, which this user belong to.

3.2.2. Add or Modify User

Request:
GET/POST /config/user_mod.cgi

Parameters:

Name	Value	Description
name	String	Original user name
newname	String	New user name. If not given, user name will not change.
password	String	Base64 encoded password
group	String	The group, which this user belong to.

Response:
Please refer to the table above.

3.2.3. Delete Users

Request:
GET/POST /config/user_del.cgi

Parameters:

name =<username1>,<username2>, ...

Allow delete multi-users at once.

Response:
name=<username1>,<username2>, ...

3.2.4. Get Groups

Request:
GET /config/group_list.cgi

Parameters:

None or

name=<groupname>

Response:
If no request parameter

Name	Value	Description
groups		The total number of groups.
<groupname> ...	<user1>, ...	For example, Administrator=admin,root (Order by 0~9, A~Z, a~z) It will display group names and users line by line. (Order by 0~9, A~Z, a~z)

Else if request parameter is name

Name	Value	Description
user	<user1>, ...	User names (Order by 0~9, A~Z, a~z)
privilege	ptz, outputs, speaker, mic, video, notify	List permissions of this group. (Order by ptz, outputs, speaker, mic, video, notify)

3.3. Video, Sensor and Audio

3.3.1. Query Stream Information

Users are able to get parameter values from IP camera. Some parameters are optional and display only if IP Camera supported.

Request:

GET /config/stream_info.cgi

GET /users/stream_info.cgi (accessible by all viewers groups)

Response:

Name	Value	Description
videos	MPEG4, MJPEG, H264	List of available video codecs. For example: videos=MPEG4,MJPEG
codeclist#	c1, c2, c3,…	Available video codec list of current profile#
audios	PCM, ADPCM, AMR, AAC	List of available audio codecs. For example: audios=PCM
resolutions	<width>x<height>,…	List of available video resolutions. For example: resolutions=640x480,320x240,160x120 when the current sensor output is VGA. (Order from High to Low)
resolutionlist# (Optional)	<width>x<height>,…	Available resolution of current profile #. "#" is a number from 1 to the count of profiles. (Order from High to Low)
vbitrates	b1, b2, b3, …	List of available bitrate (kbps) For example: vbitrates=600,800,1000 (Order from Low to High)
vbitratelist#	b1, b2, b3, …	Available bitrate list of current profile#
goplengths (Optional)		List of available GOP lengths.
qualitymodes	CBR, Fixquality	List of available quality mode. For example: qualitymodes = CBR, Fixquality
framerates	f1, f2, f3, …	List of available frame rates. (Order from Low to High)
frameratelist# (Optional)	f1, f2, f3, …	Available frame rate of current profile #. "#" is a number from 1 to the count of profiles. (Order from Low to High)
qualities	q1, q2, q3, …	List of available quality. (Order from Low to High)
qualitylist#	q1, q2, q3, …	Available quality list of current profile#
asamplerates	Integer	List of audio sample rates (kHz)
abitrates	Integer	List of audio bitrate (kbps)
micvol	m1, m2, m3, ...	Available range of microphone volume is from v1 to v2. For example: micvol=0,10,20,…,90,100 (Order from Low to High)
cur_micvol	Integer	Current microphone volume. (0: also mean disabled.)
speakervol	s1, s2, s3, ...	Range of speaker volume. (Order from Low to High)
cur_speakervol	Integer	Current speaker volume. (0: also mean disabled.)
vprofileformat	<Ver#>	Current version is '1.5': This value indicates whether camera support /video/video.cgi or not. Please refer to 4.1.7
vprofilenum	#	Total number of available video streams.
vprofile#	<codec name>	Video profile # ("#" is a number from 1 to the count of profiles)
vprofileurl#	String	URL address of video profile #
vprofileres#	<width>x<height>	Resolution size of video profile#. Resolution size depends on the current sensor output.
aprofilenum	#	Total number of available audio streams.
aprofile#	<codec name>	Audio profile # ("#" is a number from 1 to the count of profiles)

aprofileurl#	String	URL address of audio profile#
vban	String	String indicates video forbidden combination of video profile setting, base on original setting support list. The vban can indicate the forbidden option include codecs, resolutions, frame rate, quality and bitrate. vban=[ban_item]:[ban_item] ban_item=profile_num|codec|resoluction|frame_rate|quality|bit_rate Each forbidden option separated by "|"(Vertical bar) and each ban item separated by ":"(Colon). Option content separated by ","(Comma) without space, and only indicate ban range from low to high, small to large. If there are no limitation of option, using "-"(Dash) to ignore. Please also note that same profile_num may have more than one ban item. Base on original setting list, if there is no any limitation for video profile setting. vban should not be provided. **For example:** vban=3|MJPEG|1024x768|6,30|-|-:4|-|1024x768|-|-|1024,4096 Above info indicate profile3 forbidden using MJPEG with resolution 1024x768 and frame rate in 6 to 30. Profile 4 forbidden using any codec with resolution 1024x768 with quality from 1024 to 4096. However, profile 3 using MJPEG with resolution 1024x768 and frame rate is under 6 is allowed, and profile 4 only forbidden combination, which using resolution 1024x768 with bit rate locate from 1024 to 4096. This mean only all combination matched in vban ban_item are forbidden. Otherwise are allowed.

3.3.2. Get Sensor Output

Request:

GET /config/sensor_output.cgi

Response:

Name	Value	Description
supports	<sensor output name>,<sensor output name>,.....	List of Available sensor output. For example: supports=VGA,HDTV,4VGA
current	<sensor output name>	This value indicates current sensor output. For example: current=4VGA. Users are able to change current sensor output result. Such as **vprofilere#** fields of stream_info.cgi and **resolution** field of video.cgi.

3.3.3. Set Sensor Output

Request:

GET/POST /config/sensor_output.cgi

Parameters:

current=<sensor output name>

Response:

Please refer to the table 3.3.2

3.3.4. Get Video Type

Request:

GET /config/video_type.cgi

Response:

Name	Value	Description
profilenumber	Integer	“1” is single-stream “2” is dual-stream “3” is triple-stream. User cannot set profilenumber whether profile has been used in the event. (Or recording).
aspectratio	String	4:3 and 16:9

3.3.5. Set Video Type

Request:

GET/POST /config/video_type.cgi

Parameters:

Please refer to the table 3.3.4

Response:

Please refer to the table 3.3.4

3.3.6. Get Video Configuration

Request:

GET /config/video.cgi

Parameters:

profileid=<video profile number>

Response:

Name	Value	Description
vprofileformat	Ver#	Current version is ‘1.5’: This value indicates whether camera support /video/video.cgi or not. Please refer to 4.1.7
profileid	#	Profile number ("#" is a number from 1 to the count of profiles)
resolution	<width>x<height>	Resolution depends on current sensor output. Before user set the resolution of profile, user needs to get list of available video resolutions by using stream_info.cgi.
viewwindow	<width>x<height>	Digital PTZ resolution.
bitrate	Integer	In kbit/s
codec	MPEG4, MJPEG, H264	A video codec
goplength	Integer	MPEG GOP length.
qualitymode	CBR, Fixquality	Quality mode
framerate	1 ... 30	Frame rate per second (fps)
quality	# (0-100)	Available quality

3.3.7. Set Video Configuration

Request:

GET/POST /config/video.cgi

Parameters:

Please refer to the table 3.3.6

Response:

Please refer to the table 3.3.6

3.3.8. Sensors Information

Request:

GET /config/sensor_info.cgi

Response: (Only display parameters which are supported)

Name	Value	Description
brightness	b1...b2 or b1,b2,b3	Available brightness range or enumeration (Order from Low to High)
contrast	c1...c2 or c1,c2,c3	Available contrast range or enumeration (Order from Low to High)
saturation	s1...s2 or s1,s2,s3	Available saturation range or enumeration (Order from Low to High)
hue	h1...h2 or h1,h2,h3	Available hue range or enumeration (Order from Low to High)
whitebalance	auto, fixed_indoor, fixed_outdoor, fixed_fluor, disabled	Available white balances list
maxexposuretime	m1...m2 or m1,m2,m3	Range of the maximum exposure time from $1/m_1$ to $1/m_2$ second or enumeration. (Order from High to Low)
backlightcomp	yes, no	Has backlight compensation
noisereduction	off, low, high	List noise reduction level.
autoexposure	yes, no	Indicate whether camera support auto exposure function
autogainctrl	yes, no	Indicate whether camera support auto gain control.
inputsize	<width>x<height>	Dimension of sensor size.
videooutformat	auto: auto detect. ntsc: NTSC pal: PAL pal-m:PAL M pal-n: PAN-N	For the cameras which has an analog video output connector, this field indicates the format of the video signal
sharpness	s1...s2 or s1,s2,s3	Available sharpness range or enumeration. (Order from Low to High)
flicker	auto, 50, 60	Anti flicker. (auto ,50 or 60 Hz)
mirror	off, on	Disable/enable image flip horizontally
flip	off, on	Disable/enable image flip vertically
color	yes, no	Select color mode or B/W mode
wds	off, on	Disable/enable WDS
exposuremode	Auto, Indoor, Outdoor, Night, Moving, Low_noise, Customize1, Customize2, Customize3	List of available exposure mode.
maxshutter	s1...s2 or s1,s2,s3	Range of the shutter from 1/m1 to 1/m2 second or enumeration. (Order from High to Low)
minshutter	s1...s2 or s1,s2,s3	Range of the shutter from 1/m1 to 1/m2 second or enumeration. (Order from High to Low)
maxgain	g1...g2 or g1,g2,g3	Available gain range or enumeration of exposure mode. (Order from Low to High)
noisereduction	off, low, high, n1...n2 or n1,n2,n3	Available noise reduction range or enumeration. (Order from Low to High)
wdrlevel	w1…w2 or w1,w2,w3	Available wdrlevel range or enumeration. (Order from Low to High)

3.3.9. Get Sensors Configuration

Please call 3.3.6.sensor_info.cgi to get enumeration values of relative parameters.

Request:
GET /config/sensor.cgi

Response: (Only display parameters which are supported)

Name	Value	Description
brightness	Integer	Brightness setting
contrast	Integer	Contrast setting
saturation	Integer	Saturation setting
hue	Integer	Hue setting
whitebalance	auto, fixed_indoor, fixed_outdoor, fixed_fluor, disabled	White balance setting
flicker	auto, 50, 60	Anti flicker. (auto, 50 or 60 Hz)
autoexposure	yes, no	Enable/Disable auto exposure
maxexposuretime	#	The divisor of maximum exposure time (1/# second). For example: If the maximum exposure time is 1/10, then the value of this field is 10.
backlightcomp	yes, no	Backlight compensation. When background is very bright, enable this setting can make darker objects appear more clear in foreground
noisereduction	off, low, high	Noise reduction level.
mirror	off, on	Disable/enable image flip horizontally
flip	off, on	Disable/enable image flip vertically
autogainctrl	yes, no	Enable/disable auto gain control function
color	yes, no	Select color mode or B/W mode
videoinformat	auto: auto detect. ntsc: NTSC pal: PAL pal-m:PAL M pal-n: PAN-N	For video server, input analog video signal could be one of many video formats, such as NTSC or PAL. In order to let video server recognize the format of video input signal, sensor module should be configured to match the format.
sharpness	Integer	Sharpness setting
wds	off, on	Disable/enable WDS
exposuremode	Auto, Indoor, Outdoor, Night, Moving, Low_noise, Customize1, Customize2, Customize3	Exposure mode.
maxshutter	#	Divisor of maximum shutter (1/# second). For example: If the maximum shutter is 1/10, then the value of this field is 10.
minshutter	#	Divisor of minimum shutter (1/# second). For example: If the minimum shutter is 1/10, then the value of this field is 10.
maxgain	Integer	Maximum gain.
noisereduction	off, low, high or integer	Noise reduction level.
wdrenable	off, on	Disable/enable WDR.
wdrlevel	Integer	WDR level.

3.3.10. Set Sensors Configuration

Request:
GET/POST /config/sensor.cgi

Parameters:

Please refer to the table 3.3.8 and 3.3.9 to set the valid values.

Response:
Please refer to the table 3.3.9

3.3.11. Get Audio Configuration

Request:

GET /config/audio.cgi

Parameters:

profileid=<audio profile number>

Response:

Name	Value	Description
profileid	#	Audio profile number (# is a number from 1 to the count of profiles)
codec	PCM, ADPCM, AMR, AAC, MU-LAW, A-LAW	Audio codec
samplerate	Integer	Audio Sample Rate. (in kHz)
channel	1, 2	Audio channel number.
bitrate	Integer	Audio output bitrate. (in kbit/s)
bit_depth	Integer	Audio output bit depth

3.3.12. Set Audio Configuration

Request:

GET/POST /config/audio.cgi

Parameters:

Please refer to the table 3.3.11

Response:

Please refer to the table 3.3.11

3.3.13. Get Microphone

Request:

GET /config/mic.cgi

Response:

Name	Value	Description
enable	no, yes	Microphone is disable/enable
volume	Integer	Microphone volume. Please refer to table 3.3.1 "micvol"

3.3.14. Set Microphone

Request:

Please refer to the table 3.3.13

Response:

Please refer to the table 3.3.13

3.3.15. Get Speaker

Request:

GET /config/speaker.cgi

Response:

Name	Value	Description
enable	no, yes	Speaker is disable/enable
volume	Integer	Speaker volume. Please refer to table 3.3.1 "speakervol"

3.3.16. Set Speaker

Request:
GET/POST /config/speaker.cgi

Parameters:

Please refer to the table 3.3.15

Response:
Please refer to the table 3.3.15

3.3.17. Reset Sensor to Default Configuration

Request:
GET/POST /config/sensor_reset.cgi

Parameters:

reset=go

Response:

Name	Value	Description
reset	yes, fail	Result of sensor reset

3.3.18. Set IR LED

Request:
GET/POST /config/irled.cgi

Parameters:

Name	Value	Description
mode	on, off	Turn on/off IR LED.

Response:
Please refer to above table

3.3.19. Get ICR (Infrared Cut Filter Removal) Setting

Request:
GET/POST /config/icr.cgi

Response:

Name	Value	Description
mode	day, night, auto, manual, schedule	Indicate current ICR mode. (day, night, auto, manual or schedule mode)
starttime	Time	Start time of schedule (In 24hr format "hh:mm", available only if mode=schedule) For example: 07:30 means 7:30 am. For example: 19:30 means 7:30 pm.
endtime	Time	End time of schedule (In 24hr format "hh:mm", available only if mode=schedule) For example: 07:30 means 7:30 am. For example: 19:30 means 7:30 pm.
light_threshold_list	0, 5, 10, …, 95, 100	Read only. Available list for light_threshod, only the values in the list are available for light_threshold setting. For example: light_threshold_ist=15,20,25,30,35,40,45,50,55,60,65,70,75,80,85
light_threshold	Integer	Light sensor threshold (sensitivity), please refer to light_threshold_list for available setting value.

3.3.20. Set ICR (Infrared Cut Filter Removal) Setting

Request:

Please refer to the table 3.3.19

Response:

Please refer to the table 3.3.19

3.3.21. Get Stream Authentication Setting

Request:

GET /config/stream_auth.cgi

Response:

Name	Value	Description
livevideo	on, off	Indicate whether it needs authentication to get live video stream.
snapshoturl	on, off	Indicate whether it needs authentication to get a snapshot.

3.3.22. Set Stream Authentication Setting

Request:

GET/POST /config/stream_auth.cgi

Parameters:

Please refer to the table 3.3.21

Response:

Please refer to the table 3.3.21

Note: If the value of 'livevideo' is off, then authentication for snapshot URL will be turned off automatically.

3.3.23. Query Privacy Mask Information

Request:

GET /config/privacymask_info.cgi

Response:

Name	Value	Description
maxnum	#	Maximum of privacy mask.
maxarea	<width>x<height>	Privacy mask range of the maximum space.

3.3.24. Get Privacy Mask

Request:

GET /config/privacymask.cgi

Response:

Name	Value	Description
enable#	no, yes	Disable/enable privacy mask window #.
area#	String	Privacy mask window # in the format x,y,w,h x,y is the coordinate. 0,0 means left top position. w,h is the width and height of the window.

3.3.25. Set Privacy Mask

Request:

GET/POST /config/privacymask.cgi

Parameters:

Please refer to the table 3.3.24

Response:

Please refer to the table 3.3.24

3.3.26. Get Thermal Detection Configuration

Request:

GET /config/thermal_detection.cgi

Response:

Name	Value	Description
enable	no, yes	Disable/enable thermal detection.
high_detect	on, off	Enable/Disable high temperature detection.
low_detect	on, off	Enable/Disable low temperature detection.
high_threshold	Integer	High temperature threshold.
low_threshold	Integer	Low temperature threshold.
unit	C,F	Celsius or Fahrenheit unit.

3.3.27. Set Thermal Detection Configuration

Request:

GET/POST /config/thermal_detection.cgi

Parameters:

Please refer to the table 3.3.26

Response:

Please refer to the table 3.3.26 Get Thermal Detection Configuration

3.3.28. Get IR LED Illumination Distance

Request:

GET /config/ir_ill_dist.cgi

Response:

Name	Value	Description
illdistance	Integer	IR illumination distance level from close to far, different model may allow different range list of level. range from 1(closest) to 10(utmost)
illdistance_list	1,2,…,10	Allow illumination list. For example, if model only allow setting distance level 5 and 10. illdistance_list=5,10

3.3.29. Set IR LED Illumination Distance

Request:

GET/POST /config/ir_ill_dist.cgi

Parameters:

Name	Value	Description
illdistance	Integer	IR illumination distance from close to far, range from 1(closest) to 10(utmost). Please check illdistance_list for available value.

Response:

Please refer to the table 3.3.28 Get Thermal Detection Configuration

3.3.30. Get List of Default System Music

Request:

GET /music/sys_music_list.cgi

Response:

Name	Value	Description
num	Integer	Number of items (file or folder) in indicated page
items	String	**Attributes of deleted items** <name>: name of file or folder. **Use ':' (colon) to divide two items.** **Example:** items=music01 : music02…

3.3.31. Play Music from Default System Music

Request:

GET/POST /music/sys_music_play.cgi

Parameters:

Name	Value	Description
file	String	File name of music file
loop	yes,no	Looping music or not. (Loop and timer cannot be given at the same time)
shuffle	yes,no	Shuffle music or not.
timer	Time	hh:mm:ss. (loop and timer cannot be given at the same time)

Ps: If enable shuffle, this cgi should play all music, not only specific file.

Response:

Please refer to the table above.

3.3.32. Stop Play Music

Stop play music immediately, no response.

Request:

GET /music/music_stop.cgi

3.3.33. Get Audio Player Mode

Request:

GET /music/audio_player_mode.cgi

Parameters:

Response:

Name	Value	Description
triggerby	manual, always, schedule, motion, sound	Trigger by what scenario. Manual control, always/schedule play or motion/sound detect play.

3.4. Network

3.4.1. Get Network Configuration

Request:

GET /config/network.cgi

Response:

Name	Value	Description
dhcp	off, on	Disable/Enable dynamic IP address assignment
ip	IP address	IP address of static IP setting
netmask	IP address	Subnet mask of static IP setting
gateway	IP address	Default gateway of static IP setting
dns1	IP address	Primary DNS server of static IP setting
dns2	IP address	Secondary DNS server of static IP setting
pppoe	off, on	Using PPPoE or not
pppoeuser	String	PPPoE user name
pppoepass	String	PPPoE password
ddns	off, on	Disable/enable dynamic DNS service
ddnsprovider	String	ID of the provider, Please refer to **8.1 Table 2 Dynamic DNS Service Providers**
ddnshost	String	DDNS host name
ddnsuser	String	DDNS user name
ddnspass	String	DDNS password
upnp	off, on	Disable/enable UPnP
httpport	1 ... 65535	TCP port number for HTTP
httpexternalport	1 ... 65535	External port number of UPnP NAT router, which is mapping to HTTP service port of camera
rtspport	1 ... 65535	Port number of RTSP service
rtspexternalport	1 ... 65535	External port number for UPnP NAT router, which is mapping to RTSP service port of camera

3.4.2. Set Network Configuration

Request:

GET/POST /config/network.cgi

Parameters:

Please refer to the table 3.4.1

Response:

Please refer to the table 3.4.1

3.4.3. Get PPPoE Setting

Request:

GET /config/pppoe.cgi

Response:

Name	Value	Description
pppoe	off, on	Disable/enable PPPoE
user	String	PPPoE user name
pass	String	PPPoE password

3.4.4. Set PPPoE

Request:

GET/POST /config/pppoe.cgi

Parameters:

Please refer to the table 3.4.3

Response:

Please refer to the table 3.4.3

3.4.5. Get DDNS Setting

Request:

GET /config/ddns.cgi

Response:

Name	Value	Description
ddns	off, on	Disable/enable dynamic DNS service
provider	String	ID of the provider. Please refer to **8.1 Table 1 Dynamic DNS Service Providers**
host	String	DDNS host name
user	String	DDNS user name
pass	String	DDNS password

3.4.6. Set DDNS

Request:

GET/POST /config/ddns.cgi

Parameters:

Please refer to the table 3.4.5

Response:

Please refer to the table 3.4.5

3.4.7. Get UPnP Information

Request:

GET /config/upnp.cgi

Response:

Name	Value	Description
upnpav	off, on	Disable/enable UPnP AV server.
upnpcp	off, on	Disable/enable UPnP CP port forward

3.4.8. Set UPnP information

Request:

GET /config/upnp.cgi

Parameters:

Please refer to the table 3.4.7

Response:

Please refer to the table 3.4.7

3.4.9. Get TCP Port Number for HTTP

Request:

GET /config/httpport.cgi

Response:

Name	Value	Description
httpport	1 ... 65535	TCP port number of HTTP service

3.4.10. Set TCP Port Number for HTTP

Request:

GET/POST /config/httpport.cgi

Please refer to the table 3.4.9

Response:

Please refer to the table 3.4.9

3.4.11. Get System Wireless

Request:

GET /config/wireless.cgi

Response:

Name	Value	Description
enable	off, on	Disable/enable wireless
mode	managed, ad-hoc	Type of wireless network to associate with, managed (using an access point) or ad-hoc (not using an access point).
essid	String	ESSID
chpatterns	String	Pattern of available wireless channels. (Read-only.) 1111000011110000 means channel 1,2,3,4,9,10,11,12 are available.
channel	1 ... 16	Wireless channel
auth	open, shared, WPA-PSK, WPA2-PSK	Indicate authentication method. (Open system, shared key, WPA-PSK or WPA2-PSK)
encryption	none, WEP, TKIP, AES	When auth is open: none, WEP. When auth is shared: WEP When auth is WPA-PSK or WPA2-PSK: TKIP, AES
format	hex, ASCII	Only used in WEP encryption
keylength	64, 128	WEP key length (bits)
activekey	1 ... 4	Which WEP key to be used when transmitting.
key1		"key#" must match the keys in the wireless access point. That could be in hex format or in ASCII format. Hex: the string must be exactly 10 hex characters for 64-bit WEP and 26 hex characters for 128-bit WEP. (Hex chars are 0123456789ABCDEF) ASCII: The string must be exactly 5 characters for 64-bit WEP and 13 characters for 128-bit WEP.
key2		
key3		
key4		
passphrase	String	WPA passphrase

3.4.12. Set System Wireless

Request:

GET/POST /config/wireless.cgi

Parameters:

Please refer to the table 3.4.11

Response:

Please refer to the table 3.4.11

3.4.13. Get Current Wireless Connection Condition

Request:

GET /config/wlansignal.cgi

Response:

Name	Value	Description
signal	0…100	Current wireless channel signal strength

3.4.14. Execute Wireless Site Survey

Request:

GET /config/wlansurvey.cgi

Response: (1 site)

Name	Value	Description
ssid	String	SSID
signal	0…100	The signal strength indicator of wireless AP.
mode	Ad-hoc infrastructure	Wireless mode
channel	1 ... 16	Wireless channel
auth	open, shared, WPA-PSK, WPA2-PSK	Indicate which authentication method is. (Open system, shared key, WPA-PSK or WPA2-PSK)
encryption	none, WEP, TKIP, AES	When auth is open: none, WEP. When auth is shared: WEP When auth is WPA-PSK or WPA2-PSK: TKIP, AES

Note:
Each wireless AP (access point) has several attributes such as the above table. "ssid" is the first attribute of any wireless AP. Camera will output these attributes of all found wireless AP in sequence.

3.4.15. Get HTTPS Configuration

Request:

GET/POST /config/https.cgi

Response:

Name	Value	Description
https_enable	no, yes	HTTPS function is disabled or enabled
https_only	no, yes	HTTP function is disabled or enabled (Indicate whether only enabled HTTPS function or not)
certificate_country	String	Country name for self-signed certificate. (2 letter code)
certificate_state	String	State or province name for self-signed certificate
certificate_locality	String	Locality name for self-signed certificate
certificate_organization	String	Organization name for self-signed certificate
certificate_organization_unit	String	Organizational unit name for self-signed certificate
certificate_common_name	String	Common name for self-signed certificate
certificate_validity	1 ... 65535	Number of days (validity) for self-signed certificate

3.4.16. Set HTTPS Configuration

Request:

GET/POST /config/https.cgi

Parameters:

Please refer to the table 3.4.15

Response:

Please refer to the table 3.4.15

3.4.17. List All the IP Access List

Request:

GET /config/acces_list.cgi

Response:

Name	Value	Description
maxallow	Integer	Maximum number of accesses.
allowlist	<IP address ~ IP address>, …	List all of the allow IP range.
maxdeny	An integer	Maximum number of accesses.
denylist	<IP address~IP address>, …	List all of the deny IP range.

3.4.18. Add, Delete Access IP

Request:

GET /config/acces.cgi

Parameters:

Name	Value	Description
type	allow, deny	Select type of setting.
act	add, del	Select action to perform content.
range	IP address ~ IP address	This parameter applies only if **"act=add".**
index	Integer	This parameter applies only if **"act=del"**. According to this index value in the IP access list to delete the corresponding position. The minimum index value is 0.

Response:

Please refer to the table above

If setting parameters "range" success. It will return an IP address of original setting value, else will return "empty string" (For example: range=\r\n).
If setting parameters "index" success. It will also return index value, else will return "-1"
(For example: index=-1\r\n).

3.4.19. Get Wireless AP-Mode Setting

Some specific IP Camera may have wireless AP-mode. This kind of cameras has partial functions of access-point. User may associate to camera as a wireless client to do further operation of camera.

Request:
GET /config/wireless_ap.cgi

Response:

Name	Value	Description
enable	off, on	Disable/enable wireless AP mode
essid	String	ESSID
security	none,wpa,wpa2	Indicate security method. One and only one method will be return. (Not all of them).
key	String	"key" must match the key in the wireless access point.

3.4.20. Set Wireless AP-Mode

Some specific IP Camera may have wireless AP-mode. This kind of cameras has partial functions of access-point. User may associate to camera as a wireless client to do further operation of camera.

Request:
GET/POST /config/wireless_ap.cgi

Response:

Name	Value	Description
enable	off, on	Disable/enable wireless AP mode
essid	String	ESSID
security	none,wpa,wpa2	Indicate security method. For AP mode, Camera may only allow some setting method (Not all of them).
key	String	"key" must match the key in the wireless access point.

3.4.21. Get Wireless STA Status.

Query status of wireless station NIC. “wlan_sta_status.cgi” will respond enablement and association status of wireless station(such as STA mode or AP-client mode). However, please be aware of if using WEP as security method, the association status cannot be reliable because even user setup wireless with incorrect setting. Client still can associate to AP.

Request:
GET /common/wlan_sta_status.cgi

Response:

Name	Value	Description
enable	off, on	Disable/enable wireless station
associated	yes, no	Associate to AP or not.

3.5. Motion/Sound Detection

3.5.1. Get Motion Detection

There are two possible types of motion detection depend on IP Camera hardware design:

1. Macro Block Type
2. Window Type

Request:
GET /config/motion.cgi

Response:
Macro Block Type:

Name	Value	Description
enable	no, yes	Disable/enable motion detection
mbmask	Hex string	The macro block mask hex string of the native screen resolution, which is calculated linearly from left to right, top to bottom. The size of a macro block depends on the video resolution. However, no matter which resolution of video is, the number of macro block is always 40x30. There is 40 blocks of width and 30 block of height in extension of the video.
sensitivity	0 ... 100	Sensitivity
percentage	0…100	Percentage
pir	no, yes	Disable/enable PIR
pir_sensitivity	0, 50, 100	Sensitivity level: Low(0), Medium(50), High(100)

Window Type:

Name	Value	Description
totalnum	#	Numbers of total motion detection window. (Read-only)
sensitivity	0 ... 100	Sensitivity
percentage	0…100	Percentage
enable#	no, yes	Disable/enable motion detection window #
mdw#	String	Motion detection window # in the format x,y,w,h x,y is coordinate. 0,0 means the left top position. w,h is width and height of window.
pir	no, yes	Disable/enable PIR
pir_sensitivity	0, 50, 100	Sensitivity level: Low(0), Medium(50), High(100)

3.5.2. Set Motion Detection

Request:
GET/POST /config/motion.cgi

Parameters:

Please refer to the table 3.5.1

Response:
Please refer to the table 3.5.1

3.5.3. Get Recorder Action

Request:
GET /cgi/admin/recorder.cgi

Response: (Represented by XML**)**

```
<config>
    <record>
        <enable>0</enable>
        <profileID>0</profileID>
        <continuous>0</continuous>
        <prerecord>0</prerecord>
        <postrecord>0</postrecord>
        <keepSpace>100</keepSpace>
        <fileFormat>0</fileFormat>
        <fileLenMin>1</fileLenMin>
        <recycle>0</recycle>
        <recordTo>
            <toSamba>
                <enable>0</enable>
                <anonymous>1</anonymous>
                <user></user>
                <password></password>
                <server></server>
                <shareFolder></shareFolder>
            </toSamba>
            <toUSB>
                <enable>1</enable>
            </toUSB>
        </recordTo>
        <schedule>
            <enable>0</enable>
            <profileName>Record</profileName>
            <item01>0,0,0,1,0,0</item01>
            <item02>1,0,0,2,0,0</item02>
            <item03>2,0,0,3,0,0</item03>
            <item04>3,0,0,4,0,0</item04>
            <item05>4,0,0,5,0,0</item05>
            <item06>5,0,0,6,0,0</item06>
            <item07>6,0,0,0,0,0</item07>
            <itemSize>7</itemSize>
        </schedule>
        <triggerBy>
            <byMotion>0</byMotion>
            <bySound>0</bySound>
            <byIn1>0</byIn1>
            <byIn2>0</byIn2>
        </triggerBy>
        <extraLight>0</extraLight>
    </record>
</config>
```

Where <fileFormat> define as below:
0: avi
1: mp4

3.5.4. Set Recorder Action

Request:

GET/POST /cgi/admin/recorder.cgi

Parameters:

Name	Value	Description
recordEnable	0 or 1	Disable(0)/enable(1) recording action
profileID	#	Id of recording profile.
prerecord	#	Pre-recording time in second (0~15)
postrecord	#	Post-recording time in second (0~15)
keepSpace	#	Keep 32~999999 Megabyte for hard disk or SD card
fileFormat	0 or 1	AVI format(0) or MP4 format(1)
fileLenMin	1 ... 60	Separate recording file in 1 … 60 minute(s). Please reference to model support table for definition of limitation of maximum value.
recycle	0 or 1	Disable(0)/enable(1) recording recycle while the CIFS disk or USB is full
toUSB	0 or 1	Disable(0)/enable(1) record to USB or Samba server. Both of these two parameters must be given and only one of them can be enable simultaneously.
toSamba		
byMotion	0 or 1	Disable(0)/enable(1) events to trigger recording. All of these parameters must be given and all of them can be enable simultaneously. Please also not that, parameter “byMotion” or “bySound” may have number of mechanisms, like video motion and PIR. When user sets byMotion enable, no matter which mechanism trigger (video motion or PIR) will cause recording start.
bySound		
byIn1		
byIn2		
continuous	0 or 1	Disable(0)/enable(1) record continuous or by schedule. Both of these two parameters must be given and only one of them can be enable simultaneously.
schedule		
anonymous	0 or 1	0: login as account, 1: login as anonymous
user	String	Maximum length of string is 30 characters. Allow symbols =>!#$%()*+,-./:;=?@[\\]^_`{|}~
password	String	Maximum length of string is 30 characters. Allow symbols => !#$%()*+,-./:;=?@[\\]^_`{|}~
server	String	Maximum length of string is 30 characters. Allow symbols => _-.
shareFolder	String	Maximum length of string is 30 characters. Allow symbols => _-
item01	String [A:0…6],[B:0…23],[C:0…59],[D:0…6],[E:0…23],[F:0…59]	A: start day (0:Sunday … 6:Saturday) B: start hour C: start minute D: end day (0:Sunday … 6:Saturday) E: stop hour F: stop minute For example: 1,1,1,1,2,2 means schedule record from Monday AM:01:01 to Monday AM:02:02 and 1,0,0,2,0,0 mean schedule record for a whole day of Monday. Note: Item number depend on "itemSize"
item02		
item03		
item04		
item05		
item06		
item07		
itemSize	#	Save # number of schedule items (0~7),
extraLight	0 or 1	Disable(0)/enable(1) extra Light, such as white Light LED Only supporting model need this. An optional parameter.

Response:

Please refer to the table 3.5.3. The following extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Checking cgi failed, please check the setting rules
	3. updateFailed	Setting cgi failed, please check the setting rules

3.5.5. Get Snapshot Action

Request:

GET /cgi/admin/adv_snapshot_cont.cgi

Response: (Represented by XML)

```
<config>
    <mail>
        <smtpEnable1>0</smtpEnable1>//
        <smtpServer1></smtpServer1>
        <smtpPort1>25</smtpPort1>
        <smtpUser1></smtpUser1>
        <smtpPass1></smtpPass1>
        <receiver1></receiver1>
        <sender1></sender1>
        <smtpInterval>600</smtpInterval>
        <smtpIgnoreTime>30</smtpIgnoreTime>
        <smtpEncrypt>0</smtpEncrypt>
    </mail>
    <ftp>
        <ftpEnable1>0</ftpEnable1>
        <ftpServer1></ftpServer1>
        <ftpPort1>21</ftpPort1>
        <ftpUser1></ftpUser1>
        <ftpPass1></ftpPass1>
        <folder1></folder1>
        <ftpInterval>1</ftpInterval>
        <ftpIgnoreTime>10</ftpIgnoreTime>
        <passive1>1</passive1>
        <fixFile1>DCS-2102</fixFile1>
    </ftp>
    <snapshot>
        <enable>0</enable>
        <continuous>0</continuous>
        <schedule>
            <enable>0</enable>
            <profileName>Snapshot</profileName>
            <item01>0,0,0,1,0,0</item01>
            <item02>1,0,0,2,0,0</item02>
            <item03>2,0,0,3,0,0</item03>
            <item04>3,0,0,4,0,0</item04>
            <item05>4,0,0,5,0,0</item05>
            <item06>5,0,0,6,0,0</item06>
            <item07>6,0,0,0,0,0</item07>
            <itemSize>7</itemSize>
        </schedule>
        <triggerBy>
            <byMotion>0</byMotion>
            <bySound>0</bySound>
            <byIn1>0</byIn1>
            <byIn2>0</byIn2>
        </triggerBy>
        <snapTo>
            <toFtp>1</toFtp>
            <toSmtp>1</toSmtp>
        </snapTo>
        <extraLight>0</extraLight>
    </snapshot>
</config>
```

Where <smtpEncrypt> **define as below :**

0: None
1: SSL-TLS
2: STARTTLS

3.5.6. Set Snapshot Action

Request:

GET/POST /cgi/admin/adv_snapshot_cont.cgi

Parameters:

Name	Value	Description
enable	0 or 1	Disable(0)/enable(1) snapshot function
toFTP	0 or 1	Disable(0)/enable(1) snapshot and send to FTP or SMTP. All of these parameters must be given, all of them can be enable simultaneously.
toSmtp		
Continuous	0 or 1	Disable(0)/enable(1). Both of these two parameters must be given and only one of them can be enable simultaneously.
schedule		
byMotion	0 or 1	Disable(0)/enable(1) events (multi-selection) to trigger snapshot. All of these parameters must be given and all of them can be enable simultaneously. Please also not that, parameter “byMotion” or “bySound” may have number of mechanisms, like video motion and PIR. When user sets byMotion enable, no matter which mechanism trigger (video motion or PIR) will cause snapshot start.
bySound		
byIn1		
byIn2		
smtpEnable1	0 or 1	Disable(0)/enable(1) snapshot to SMTP1
smtpServer1	String	SMTP Server host name or IP address. Maximum length of string is 60 characters. Allow symbols => _-.
smtpPort1	#	SMTP server port (1~65535)
smtpUser1	String	SMTP user name Maximum length of string is 30 characters Allow symbols => !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
smtpPass1	String	SMTP password Maximum length of string is 30 characters, Allow symbols => !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
receiver1	String	Destination email address, Maximum length of string is 60 characters. Allow symbols => !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
sender1	String	Source email address Maximum length of string is 60 characters. Allow symbols => !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
smtpInterval	#	Time interval of sending mail in seconds. (1~86400)
smtpIgnoreTime	#	Ignore seconds between even occur. (1~86400)
smtpEncrypt	0, 1, 2	0(None); 1(SSL-TLS); 2(STARTTLS)
ftpEnable1	0 or 1	Disable(0)/enable(1) snapshot to FTP1
ftpServer1	String	FTP server host name or IP address. Maximum length of string is 60 characters. Allow symbols => _-.
ftpPort1	#	FTP port (1~65535)
ftpUser1	String	FTP login user name. Maximum length of string is 30 characters. Allow symbols => !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
ftpPass1	String	FTP password. Maximum length of string is 30 characters Allow symbols => !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
folder1	String	FTP initial path. Maximum length of string is 30 characters. Not allow symbols => ~!@#$%^&()+{}`=[];',/|*
ftpInterval	#	Time interval of continue snapshot to FTP in second. (1~86400)
ftpIgnoreTime	#	Ignore seconds between event occur. (1~86400)
passive1	0 or 1	Disable(0)/enable(1) FTP passive mode
fixFile1	String	Prefix name of pictures. Maximum length of string is 30 characters. Not allow symbols =>~!@#$%^&()+{}`=[];',/\\|*
item01	String [A:0…6],[B:0…23],[C:0…59],[D:0…6],[E:0…23],[F:0…59]	A: start day (0:Sunday … 6:Saturday) B: start hour C: start minute D: end day (0:Sunday … 6:Saturday)
item02		
item03		

item04		E: stop hour F: stop minute For example: 1,1,1,1,2,2 means schedule snapshot from Monday AM:01:01 to Monday AM:02:02 and 1,0,0,2,0,0 mean schedule snapshot for a whole day of Monday. Note: Item number depend on itemSize
item05		
item06		
item07		
itemSize	#	Save # number of schedule items (0~7) For example: itemSize = 3, so item01~item03's value will be saved
extraLight	0 or 1	Disable(0)/enable(1) extra Light, such as white Light LED Only supporting model need this. An optional parameter.

Response:

Please refer to the table 3.5.5. The following extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Checking cgi failed, please check the setting rules
	3. updateFailed	Setting cgi failed, please check the setting rules

3.5.7. Get Alarm Out Action

Request:

GET /cgi/admin/adv_do.cgi

Response: (Represented by XML**)**

```
<config>
        <digitalOutput>
                <toOut1>
                        <enable>0</enable>
                        <triggerBy>
                                <byMotion>0</byMotion>
                                <bySound>0</bySound>
                                <byIn1>0</byIn1>
                                <byIn2>0</byIn2>
                        </triggerBy>
                </toOut1>
        </digitalOutput>
</config>
```

3.5.8. Set Alarm Out Action

Request:

GET/POST /cgi/admin/adv_do.cgi

Parameters:

Name	Value	Description
toOut1	0 or 1	Alarm to output 1
out1ByMotion	0 or 1	Disable(0)/enable(1) events (multi-selection) to trigger alarm out1
out1BySound		
out1ByIn1		
out1ByIn2		

Response:

Please refer to the table 3.5.7. The following extra result tag will also be presented:

Result: (Represented by XML)	<config> <result> <code>ok</code> </result> </config>	
Result code	1. ok	Success
	2. invalidParameter	Checking cgi failed, please check the setting rules
	3. updateFailed	Setting cgi failed, please check the setting rules

3.5.9. Get Sound Detection

Request:

GET /config/audio_detection.cgi

Response:

Name	Value	Description
enable	yes, no	Enable/disable sound detection
sensitivity	Integer	Sensitivity level of sound detection, from 0 ~ 100.

3.5.10. Set Sound Detection

Request:

GET/POST /config/audio_detection.cgi

Parameters:

Name	Value	Description
enable	yes, no	Enable/disable sound detection
sensitivity	Integer	Sensitivity level of sound detection, from 0 ~ 100.

Response:

Please refer to the table above.

3.5.11. Get Environmental Sound Level

Request:

GET /users/env_sound_lv.cgi

Response:

Name	Value	Description
audio_detect_val	Integer	Decibel level of environmental sound.

3.6. System Tools

3.6.1. Get Digital Input/Output

This CGI is a one-shot command, which only return current configurations and status of IP camera and then terminated. If user needs to monitor camera status for a long time, please use 6.1.2 notify_stream.cgi instead.

Request:
GET /config/io.cgi

Response: (Only display when inputs and outputs are supported)

Name	Value	Description
in1	off, on	Digital input set 1
in2	off, on	Digital input set 2
in3	off, on	Digital input set 3
in4	off, on	Digital input set 4
out1	off, on	Digital output set 1
out2	off, on	Digital output set 2
out3	off, on	Digital output set 3
out4	off, on	Digital output set 4

3.6.2. Set Digital Output

Only digital outputs are allowed to be set. Inputs are read-only.

Request:
GET/POST /config/io.cgi

Parameters:

Name	Value	Description
out1	off, on	Digital output set 1
out2	off, on	Digital output set 2
out3	off, on	Digital output set 3
out4	off, on	Digital output set 4

Response:
Please refer to the table above.

3.6.3. Get LED

Request:
GET /config/led.cgi

Response:

Name	Value	Description
led	on, off	Enable/disable special purpose LED.

3.6.4. Set LED

Set LED status on/off, only workable when LED status is in manual mode. Please refer to 3.6.5 Get LED Mode.

Request:
GET/POST /config/led.cgi

Parameters:

Please refer to the table 3.6.3

Response:
Please refer to the table 3.6.3

3.6.5. Get LED Mode

Request:

GET /config/led_mode.cgi

Response:

Name	Value	Description
mode	manual, always_on, always_off, by_motion, by_in1, by_hot, by_cold	String described what kind of event control and trigger special purpose led.

3.6.6. Set LED Mode

Request:

GET/POST /config/led_mode.cgi

Parameters:

Please refer to the table 3.6.5

Response:

Please refer to the table 3.6.5

3.6.7. Firmware Upgrade

Request:

POST /config/firmwareupgrade.cgi

The firmware file content is provided in HTTP body according to the format described in RFC 1867.

The HTTP body will create automatically by browser if it use HTML form with input type "file".

```
Example:
POST /config/firmwareupgrade.cgi HTTP/1.0\r\n
Content-Type: multipart/form-data; boundary=AsCg5y\r\n
Content-Length: <content length>\r\n
\r\n
--AsCg5y\r\n
Content-Disposition: form-data; name="update.bin"; filename="update.bin"\r\n
Content-Type: application/octet-stream\r\n
\r\n
<firmware file content>
\r\n
--AsCg5y--\r\n
```

Response:

Name	Value	Description
upgrade	ok, fail	Upgrade result is successful or fail

Note:
User is able to transport firmware to IP camera by using web browsers. (IE, Firefox, Chrome, Safari … etc)
In order to do this, user can write a HTML file with form architecture to post firmware file to camera. For example:

```
<!-- saved from url=(0022)http://internet.e-mail -->
<html>

<head>
<script language="JavaScript" type="text/javascript">
function sendUpdate()
{
    var updateForm = document.updateForm;
    document.updateForm.action = "http://" + camip.value + "/config/firmwareupgrade.cgi";
    updateForm.submit();
}
</script>
</head>

<body>
Input camera ip (ex. 192.168.1.1): <input name="camip" type="text" id="camip" value=""/>
<form enctype="multipart/form-data" method="post" action="" name="updateForm">
    Choose firmware file: <input name="upload" type="file" id="upload" value=""/>
    and click
<input name="submit6" value="commit" type="button" onclick="sendUpdate()"/>
</form>
</body>

</html>
```

3.6.8. Reboot Camera

Request:

GET/POST /config/system_reboot.cgi

Parameters:

reboot=go

Response:

Name	Value	Description
reboot	yes, fail	Reboot result is successful or fail

3.6.9. Reset All Configurations to Factory Default

Request:

GET/POST /config/system_reset.cgi

Parameters:

reset=go

Response:

Name	Value	Description
reset	yes, fail	Reset result is successful or fail

3.6.10. Get RS-485 Settings

Request:

GET /config/rs485.cgi

Response:

Name	Value	Description
enable	no, yes	Disable/enable RS-485
proto	custom, Dyna, Lilin, Lilin2, PelcoD, PelcoP	Protocol type
devid	#	Device ID of the RS-485 slave device. Dyna: 1 ... 223 Lilin: 1 ... 64 Lilin2: 0 … 255 PelcoD: 1 ... 255 PelcoP: 1 ... 32 Custom: not applicable
baudrate	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	Custom baud rate
databits	7, 8	Custom data bits
parity	None, Even, Odd	Custom parity
stopbits	1, 2	Custom stop bits
home		Custom home command
up		Custom up command
down		Custom down command
left		Custom left command
right		Custom right command
stop		Custom stop command
stoppattern	String	Whether use the custom stop command for custom command 1, 2, 3, 4 0101 means custom command 2 and 4 with stop command.
cmdname1	String	Custom command1 name
cmdname2	String	Custom command2 name
cmdname3	String	Custom command3 name
cmdname4	String	Custom command4 name
cmdstr1	String	Custom command1 string
cmdstr2	String	Custom command2 string
cmdstr3	String	Custom command3 string
cmdstr4	String	Custom command4 string
delaytime	Integer	Interval of time between two consecutive commands. (In millisecond, for example, 300 mean 300 milliseconds).

3.6.11. Set RS-485 Settings

Request:

GET/POST /config/rs485.cgi

Parameters:

Please refer to the table 3.6.10.

Response:

Please refer to the table 3.6.10.

3.6.12. Get Privacy Mode Settings

Request:

GET /config/privacy.cgi

Response:

Name	Value	Description
enable	no, yes	Disable/enable Privacy mode
manual	on, off	Turn on/off the privacy mode

3.6.13. Set Privacy Mode Settings

Request:

GET/POST /config/ privacy.cgi

Parameters:

Please refer to the table 3.6.12.

Response:

Please refer to the table 3.6.12.

3.6.14. Get TV Output

Request:

GET /config/tvoutput.cgi

Response:

Name	Value	Description
out	on, off	Disable/enable this cable connection into the TV output adapter.

3.6.15. Set TV Output

Request:

GET/POST /config/tvoutput.cgi

Parameters:

Please refer to the table 3.6.14.

Response:

Please refer to the table 3.6.14.

3.6.16. Get DC Power

Request:

GET /config/dcpower.cgi

Response:

Name	Value	Description
mode	on, off, auto, schedule	Disable/enable the camera’s DC power port.
starttime	Time	Start time of schedule (In 24hr format "hh:mm", only when mode=schedule) For example: 07:30 means 7:30 am. For example: 19:30 means 7:30 pm.
endtime	Time	Start time of schedule (in 24hr format "hh:mm", only when mode=schedule) For example: 07:30 means 7:30 am. For example: 19:30 means 7:30 pm.

3.6.17. Set DC Power

Request:

GET/POST /config/dcpower.cgi

Parameters:

Please refer to the table 3.6.16.

Response:

Please refer to the table 3.6.16.

3.6.18. Get Device Timestamp

Request:

GET /config/timestamp.cgi

Response:

Name	Value	Description
enable	on, off	Disable/enable the camera’s time stamp (label).
showtime	on, off	Disable/enable the camera’s time stamp (time).
label	String	Time stamp on the image.

3.6.19. Set Device Timestamp

Request:

GET/POST /config/timestamp.cgi

Parameters:

Please refer to the table 3.6.18

Response:

Please refer to the table 3.6.18

3.7. SD Card Operation

3.7.1. Get Information of SD Card

This CGI is a one-shot request. User can use this request to get summary information of SD card, which is inserted in the camera. For getting up-to-date status of SD card, user should use notify_stream.cgi request (6.1.2) instead of pooling this cgi frequently.

Request:

GET /config/sdcard.cgi

Response: (Only display parameters which are supported)

Name	Value	Description
status	ready, protected, full, invalid, over_capacity, need_reinitialize, formatting, recording	Current status of SD Card
total	Integer	Total size of SD Card in KB
used	Integer	Used size of SD Card in KB
free	Integer	Free size of SD Card in KB
picture	Integer	How many sub-folders in picture folder
video	Integer	How many sub-folders in video folder

3.7.2. Format SD Card

Request:

GET /config/sdcard_format.cgi

Parameters:

Name	Value	Description
format	go	Do format or query SD status.

Response:

Please refer to the table 3.7.1

3.7.3. List Items of SD Card

Request:

GET /config/sdcard_list.cgi

Parameters:

Name	Value	Description
type	picture, video	File type (picture or video)
path	String	Path of file.
page	Integer	This command would list files of particular page. User is able to list particular page.
pagesize	Integer	How many files in a page. Value=5, 10, 15, 20, 50 or 100.

Response:

Name	Value	Description
sd_status	ready, invalid, protected, full	Status of SD Card.
type	picture, video	File type (picture or video)
path	String	Indicate current examining path
page	Integer	This command would list items of a page. User can indicate which page want to be list. For example: If SD card has 40 items (file or folder) in `path` folder and user would like to list files in page 1, and also assume each page can list 20 items. The request URL would be: GET /config/sdcard_list.cgi?type=picture&path=/20080229/00&page=1&pagesize=20 Where: type=picture: list items in path 'picture' path=/20080229/00: list the items which in path 'picture/20080229/00' page=1: list the items in page 1 pagesize=20: indicate that there are 20 items in each page Camera will response the fist 20 items which locate in path 'picture/20080229/00'
pagesize	Integer	How many items (file or folder) in a page. See more details in previous parameter.
total_file	Integer	Total number of these items (file or folder)
total_page	Integer	Total page of these items (file or folder)
num	Integer	Number of items (file or folder) in indicated page
items	String	**Attributes of listed items** <name>|<type>|<recording type>|<size or num>**:** **Where:** name: name of file or folder. type: indicate this ‘name’ is a file or a folder, f(file) or d(folder)； recording type: d (digital input) or m (motion) or n（normal） size or num: file’s size(type=f), number of files in the folder(type=d)。 **Use ‘|’(pipe or vertical bar) to separate each of attribute. Use ‘:’ (colon) to divide two items.** **Example:** items=20090826|d|n|100:20090826_000000.avi|f|n|512:20090825|d|n|50:…

3.7.4. Download Files of SD Card

Request:

GET /config/sdcard_download.cgi

Parameters:

Name	Value	Description
type	picture, video	File type (picture or video)
path	String	Path of file
file	String	File name. It could be gotten through request /config/sdcard_list.cgi. Attribute type must be 'f ' (only allow download "file").

Response: (when file is available)

```
HTTP/1.0 200 OK<CRLF>
Content-Type: application/octet-stream<CRLF>
Content-Length: <size of file><CRLF>
<CRLF>
<Binary data of file>
```

Response: (when file is not available)

Name	Value	Description
path	String	Path of file
file	String	File name.
result	Integer	Status of downloading action. 1: File is not exists 2: Target is a folder, can't be download

3.7.5. Delete Files of SD Card

Request:

GET /config/sdcard_delete.cgi

Parameters:

Name	Value	Description
type	picture, video	File type (picture or video)
path	String	Path of file
name	<file 1>[:<file 2>: …]	Indicate which files are going to be deleted. File name could be gotten through request /config/sdcard_list.cgi. Use ':'(colon) as separate character.

Response:

Name	Value	Description
num	Integer	Number of items (file or folder) which in 'name' (deleting list).
path	String	Path of file
sd_status	ready, invalid, protected	Status of SD Card. "invalid" and "protected" indicate delete operation failure.
items	String	**Attributes of deleted items** <name>|<type>|<status>**:** **Where:** name: name of file or folder. type: indicate file type of "name", f(file) or d(folder) or n(unknown); status: the status of deleting action of indicated item (file or folder), <filename>. 0: item has deleted successfully 1: item does not exist 2: item has not deleted **Use '|'(pipe or vertical bar) to separate each of attribute. Use ':' (colon) to divide two items.** **Example:** items=20090826|d|0:20090826_000000.avi|n|1: 20090826_101200.avi |f|2:… Where: 20090826: this item is a directory and has deleted successfully; 20090826_000000.avi: this item does not exists; 20090826_101200.avi: this item is a file and has not deleted.

3.7.6. Upload A File to SD Card

This CGI allow users upload music files to SD card into folder "/music", the parameter "path" is base on root folder "/music" of SD card. And different camera may support different music encoding format, please make sure the encode type is supported for play.

At this CGI, the file content is provided in HTTP body according to the format described in RFC 1867.
The HTTP body will create automatically by browser if it use HTML form with input type "file", Please refer to 3.6.7 Firmware upgrade for more detail information.

Request:

POST /music/sdcard_upload.cgi

Parameters:

Name	Value	Description
path	String	Destination path. Root path "/" is mapping to folder "/music/" of SD card.
filename	String	Target file name

Response:

Name	Value	Description
sd_status	ready, protected, full, invalid, over_capacity, need_reinitialize, formatting, recording	Current status of SD Card
total	Integer	Total size of SD Card in KB
used	Integer	Used size of SD Card in KB
free	Integer	Free size of SD Card in KB
path	String	Destination path.

3.7.7. Get List of Music from SD Card

This CGI will only list supported encoding format files.

Request:

GET /music/sdcard_music_list.cgi

Parameters:

Name	Value	Description
path	String	Path of file. Root path "/" is mapping to folder "/music/" of SD card.
page	Integer	This command would list files of particular page. User is able to list particular page.
pagesize	Integer	How many files in a page. Value=5, 10, 15, 20, 50 or 100.

Response:

Name	Value	Description
sd_status	String	Status of SD Card.
path	String	Indicate current examining path. Root path "/" is mapping to folder "/music/" of SD card.
page	Integer	This command would list items of a page. User can indicate which page want to be list. For example: If SD card has 40 items (file or folder) in 'path' folder and user would like to list files in page 1, and also assume each page can list 20 items. The request URL would be: GET config/sd_music_list.cgi?path=/Music&page=1&pagesize=20 Where: path=/Music: list the items which in path '/Music' page=1: list the items in page 1 pagesize=20: indicate that there are 20 items in each page Camera will response the fist 20 items which locate in path '/Music'

pagesize	Integer	How many files in a page.
total_file	Integer	Total number of these items (file or folder)
total_page	Integer	Total page of these items (file or folder)
num	Integer	Number of items (file or folder) in indicated page
items	String	**Attributes of items** <name>|<type>**:** **Where:** name: name of file or folder. type: indicate file type of “name”, f(file) or d(folder) or n(unknown); **Use ‘|’(pipe or vertical bar) to separate each of attribute. Use ‘:’ (colon) to divide two items.** **Example:** items=music01|f : music02|f : folder00|d :…

3.7.8. Play Music from SD Card

Play music file from folder "/music" of SD card, it is able to stop playing by using /music/music_stop.cgi.
Please also reference "3.3.32 Stop Play Music".

Request:

GET/POST /music/sdcard_play.cgi

Parameters:

Name	Value	Description
path	String	Path of file. Root path "/" is mapping to folder "/music/" of SD card.
file	String	File name of music file
loop	yes,no	Looping file/folder or not. (Loop and timer cannot be given at the same time)
shuffle	yes,no	Shuffle folder or not.
timer	Time	hh:mm:ss. (loop and timer cannot be given at the same time)

Response:

Name	Value	Description
sd_status	String	Status of SD Card.
path	String	Path of file.
file	String	File name of music file
total_file	Integer	Total files in current play list.
loop	yes,no	Looping file/folder or not. (Loop and timer cannot be given at the same time)
shuffle	yes,no	Shuffle folder or not.
timer	Time	hh:mm:ss. (loop and timer cannot be given at the same time)

Ps: If enable shuffle, this cgi should play all music, not only specific file.

3.8. Camera Log

3.8.1. Get Syslog Setting

3.8.1 and 3.8.2 specifies an interface to suit the syslog protocol. Camera will send syslog message to syslog server according to the setting.

Request:

GET /config/syslog.cgi

Response:

Name	Value	Description
enable	off, on	Disable/Enable syslog function
logserver	IP or hostname	Log server IP address or host name
logserverport	Port number	Default is 514, otherwise the port number should between 1 ~65535
duplicate	off, on	Off do not allow duplicate message, on allow duplicate message
priority	0…6	Default is 6: INFO 0: EMERG 1: ALERT 2: CRIT 3: ERR 4: WARNING 5: NOTICE 6: INFO

3.8.2. Set Syslog Setting

3.8.1 and 3.8.2 specifies an interface to suit the syslog protocol. Camera will send syslog message to syslog server according to the setting.

Request:

GET/POST /config/syslog.cgi

Parameters:

Please refer to the table 3.8.1

Response:

Please refer to the table 3.8.1.

3.8.3. Get Event Log

User can get event log from Camera through event_log.cgi.

Request:

GET /config/event_log.cgi

Response:

```
HTTP/1.0 200 OK\r\n
...
<Event Type> <Time Stamp> <Event Message>
<Event Type> <Time Stamp> <Event Message>
...
```

Example:

```
sd 2013-01-01 04:04:53 Video Clip 20130101_040430.avi by Sound Detection to E-mail is OK.
md 2013-01-01 02:21:55 Snapshot 20130101_022150.jpg by Motion Detection to SD Card is OK.
th 2013-01-01 03:44:30 Snapshot 20130101_034429.jpg by Hot Thermal Detection to SD Card is OK.
tc 2013-01-01 03:44:30 Snapshot 20130101_034429.jpg by Cold Thermal Detection to SD Card is OK.
di 2013-01-01 04:01:03 Recording 20130101_040042.mp4 by Digital Input to SD Card is OK
```

Note:

sd = Sound Detection
md = Motion Detection
th = Hot Thermal Detection
tc = Cold Thermal Detection
di = Digital Input

3.9. Fisheye Operation

3.9.1. Query Mount Type Information

Request:

GET /config/mounttype_info.cgi

Response:

Name	Value	Description
typelist	<mount type 1>,< mount type 2>,< mount type 3>,…	List all of available mount types, separated by ","(Comma)

3.9.2. Get Current Mount Type

Request:

GET /config/mounttype.cgi

Response:

Name	Value	Description
type	<mount type>	Current mount type of fisheye

3.9.3. Set Mount Type

Request:

GET/POST /config/mounttype.cgi

Parameters:

Please refer to the table 3.9.2

Response:

Please refer to the table 3.9.2

3.9.4. Query Fisheye Display Mode Information of Live Video

Request:

GET /config/displaymode_info.cgi

Response:

Name	Value	Description
mode	<mode name 1>,<mode name 2>,<mode name 3>,…	List all of available display modes. More information, please refer to Appendix 8.3

3.9.5. Get Fisheye Display Mode of Live Video

Request:
GET /config/displaymode.cgi

Response:

Name	Value	Description
mode	<mode name>	The fisheye current display mode. More information, please refer to Appendix 8.3

3.9.6. Set Fisheye Display Mode of Live Video

Request:
GET/POST /config/displaymode.cgi

Parameters:

Please refer to the table 3.9.5

Response:
Please refer to the table 3.9.5

3.9.7. Query Fisheye Split Window Information

Request:
GET /config/fisheyewindow_info.cgi

Response:

Name	Value	Description
windowlist	<Integer>,<Integer>,<Integer>,…	List all of available window number list.

3.9.8. Get Current Fisheye Split Window

Request:
GET /config/fisheyewindow.cgi

Response:

Name	Value	Description
id	Integer	Current fisheye split window.

3.9.9. Set Fisheye Split Window

Request:
GET/POST /config/fisheyewindow.cgi

Parameters:

Please refer to the table 3.9.8

Response:
Please refer to the table 3.9.8

4. Streaming

4.1. Live Streaming URL

Live streaming CGIs provide developers to download/upload streaming through HTTP/HTTPS protocol. Normally, when client send a request to server, server will respond HTTP header and streaming data (streaming data may contain ACS header) to client side. We strongly recommend the client side to close the connection directly without any postfix date if connection close successfully or been interrupted. However, some CGIs implemented by web server and may return second HTTP header in non-standard way when connection close. In order to handle this case, client side should able to treat this kind of case as connection close without problem.

4.1.1. Get a JPEG image

Returns JPEG image with default resolution and compression as defined in the configuration.

Request:

GET /image/jpeg.cgi

Response:

```
HTTP/1.0 200 OK\r\n
Content-Type: image/jpeg\r\n
Content-Length: <image size>\r\n
\r\n
<JPEG image data>\r\n
```

4.1.2. Get MJPEG Video Stream

Returns multipart image stream with default resolution and compression as defined in the configuration. The content type is “multipart/x-mixed-replace”. Each image ends with a boundary string <boundary>.

Request:

`GET /video/mjpg.cgi`

Parameters:

Name	Value	Description
profileid (Optional.)	#	If "profileid" does not supplied, the URL will output one of stream profile that is matching assign format (MJPEG).

Response:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=<boundary>\r\n
\r\n
--<boundary>\r\n
Content-Type: image/jpeg\r\n
Content-Length: <image size>\r\n
\r\n
<JPEG image data>\r\n
--<boundary>\r\n
Content-Type: image/jpeg\r\n
Content-Length: <image size>\r\n
\r\n
<JPEG image data>\r\n
--<boundary>\r\n
```

4.1.3. Get MPEG-4 Elementary Video Stream

Returns MPEG-4 elementary stream with assigned profile id, which defined in the configuration. The content type is “video/MP4V-ES”. Please refer to INAN MIME Media Types.

Request:

```
GET /video/MP4V-ES.cgi
```

Parameters:

Name	Value	Description
profileid (Optional)	#	If “profileid” does not supplied, the URL will output one of stream profile that is matching assign format (MPEG4 elementary stream).

Response:

```
HTTP/1.0 200 OK\r\n
Content-Type: video/MP4V-ES\r\n
\r\n
<MPEG-4 ISO/IEC 14496-2 elementary stream>
```

4.1.4. Get MPEG-4 Video Stream

Return MPEG-4 video stream. The video data header, please refer to the ACS Stream Header.

Request:

GET /video/ACVS.cgi

Parameters:

Name	Value	Description
profileid (Optional)	#	If “profileid” does not supplied, the URL will output one of stream profile that is matching assign format (ACVS/MPEG4 stream).

Response:

```
HTTP/1.0 200 OK\r\n
Content-type: video/ACVS\r\n
\r\n
<ACAS Video Stream Data>
```

Where <ACAS Video Stream Data> is defined as below:

```
<ACS_VideoHeader>
<MPEG4 Raw Data>
<ACS_AudioHeader>
<MPEG4 Raw Data>
...
```

<ACS_VideoHeader> is defined in **8.2 Advanced IP-Camera Stream (ACS) Header.**
<MPEG4 Raw Data> is raw data of MPEG4 video stream.

4.1.5. Get Audio Stream

This ACAS CGI will response PCM stream from profileid#, if profileid# does not use PCM as its encode method, it should close connection directly. Please refer to ACS Stream Header for more audio data header information.

Request:

GET /audio/ACAS.cgi

Parameters:

Name	Value	Description
profileid (Optional.)	#	If “profileid” does not supplied, the URL will output one of stream profile that is matching assign format (ACAS/PCM stream).

Response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/ACAS\r\n
\r\n
<ACAS Audio Stream Data>
```

Where <ACAS Audio Stream Data> is defined as below:

```
<ACS_AudioHeader>
<Audio Raw Data>
<ACS_AudioHeader>
<Audio Raw Data>
...
```

<ACS_AudioHeader> is defined in in **8.2 Advanced IP-Camera Stream (ACS) Header**.
<Audio Raw Data> is raw data of audio stream. The format of this data depends on <ACS_AudioHeader>.

4.1.6. Get Profile Video Stream

Return video stream, which associated with a specific profile. Video stream format depends on the compression type of video in that profile. For more information, please read the note below.

Request:

GET /video/video.cgi

Parameters:

Name	Value	Description
profileid (Optional)	# (1 to the count of video profile)	If “profileid” does not supplied, the URL will output the stream of default profile (profile id = 1).

Response:

```
HTTP/1.0 200 OK\r\n
Content-type: video/ACVS\r\n
\r\n
<video streaming data>
```

Note:
`<video streaming data>`:
If compression type of the designated profile is motion-JPEG (MJPEG), the stream format is different with as multipart format stream. Video stream of motion-JPEG profile wrapped by ACVS header.
On the other hand, if the compression type is MPEG4 (or H264 or other advanced compression methods), the output format follows the ACVS standard format (See Appendix 8.2).
This URL is available only if the value of item ‘vprofileformat’ in 3.3.1 or 3.3.2 is equal to or greater than 1.0

4.1.7. Put Audio Upstream (two-way audio talk)

There are three CGI functions to put audio upstream. speaker.cgi is not encouraged, please using speaker2.cgi or dgtalkie.cgi instead if possible. Please also note that user may only allow have one upstream simultaneously.

1. /dev/speaker.cgi
2. /dev/speaker2.cgi
3. /dev2/dgtalkie.cgi

- **For /dev/speaker.cgi :**

/dev/speaker.cgi is kind of obsolete function of NIPCA, we strongly recommend using /dev/speaker2.cgi instead. However, NIPCA will keep support speaker.cgi to make sure prior products, which follow prior NIPCA version can still work well.

There are two request functions for using /dev/speaker.cgi. The first request is “verification request”, and the other is called “uploading request”.

While uploading audio data from client to server (camera), the client may run into some situations instead of successfully keep upload audio data. For example, if another client is uploading audio, server will disconnect other connection when other client try to start upload audio. On the other hand, if client send command with wrong authentication information, the server will also reject this request.

For the reasons above, client should use verification request to check is authentication information correct before upload audio stream. In other words, if a user has past the verification request, but it still get disconnecting after uploading request. It is because of the other connection exist.

Verification request:

Request object:

```
HEAD /dev/speaker.cgi?client=<MAC address of client side>
```

Request header:

```
Authorization: Basic <base64 encode(username:password)>\r\n
Content-Type: audio/ACAS\r\n
Content-length: 0\r\n
\r\n
```

Response of verification request:
If the authorization is verified, the camera should return 200 OK to indicate client side can keep uploading request:

```
HTTP/1.0 200 OK\r\n
```

If the authorization failed, the camera would return HTTP error code to indicate client side should stop the uploading request, for example:

```
HTTP/1.0 401 Unauthorized\r\n
```

Uploading request:

Request object:

```
POST /dev/speaker.cgi?client=<MAC address of client side>
```

Request header:

```
Authorization: Basic <base64 encode(username:password)>\r\n
Content-Type: audio/ACAS\r\n
Content-length: 4\r\n
Connection: Keep-Alive\r\n
\r\n
```

Request body:

```
<Random 4CC>

(wait for 2 sec.)

<AAH>
<1K audio data>
<AAH>
<1K audio data>
<AAH>
<1K audio data>
```

```
...
```

Where:

```

<Random 4CC>:
4-byte random character code.

(wait for 2 sec.):
After received 4-bytes random character code, client should wait for 1 sec before
sending more audio data.

<AAH>:
The header of AAH defined as follow:

typedef struct _ACS_AudioHeader
{
unsigned long ulHdrID;               //Header ID  = 0xF6010000
unsigned long ulHdrLength;         // sizeof(ACS_AudioHeader)
unsigned long ulDataLength;        // audio data length
unsigned long ulSequenceNumber; // sequence number
unsigned long ulTimeSec;             //sample time stamp
unsigned long ulTimeUSec;          // sample time stamp
unsigned long ulDataCheckSum;   // not used…
unsigned short usFormat;             //  0x00000010  S16 LE
unsigned short usChannels;          // 1 channel
unsigned short usSampleRate;       // 8000   hz
unsigned short usSampleBits;        // 16   bits
unsigned long ulReserved;           //
}ACS_AudioHeader, *PACS_AudioHeader;

<1K audio data>:
Audio data acquired by client side in the format specified by <AAH> header
```

Response of uploading request:

There is no response for uploading request.

- ## For /dev/speaker2.cgi

/dev/speaker2.cgi is based on speaker.cgi. It is a refined version to make sure this interface can meet HTTP 1.1 RFC.

Please refer to /dev/speaker.cgi above for more detail information.

The main different between speaker2.cgi and speaker.cgi list below:

1. HTTP header Content-Length should reflect to real byte length of upload audio data and less then 2GB.
2. HEAD request of speaker.cgi is unnecessary, please issue POST request directly.
3. <Random 4CC> at the first body is also unnecessary, remove this part.
4. <wait for 2 sec> is also unnecessary, remove this part.
5. Data size of ACS Audio Header (ulDataLength) should be fixed at 1024.

Uploading request:

Request object:

```
POST /dev/speaker2.cgi?client=<MAC address of client side>
```

Request header:

```
Authorization: Basic <base64 encode(username:password)>\r\n
Content-Type: audio/ACAS\r\n
Content-length: <audio data length>\r\n
Connection: Keep-Alive\r\n
\r\n
```

Request body:

```
<AAH>
<1K audio data>
<AAH>
<1K audio data>
<AAH>
<1K audio data>
...
```

- ## For /dev2/dgtalkie.cgi

/dev2/dgtalkie.cgi is different from speaker.cgi and speaker2.cgi. Please treat dgtalkie.cgi as another kind of way to upload audio stream.

There are two goals of dgtalkie.cgi:

1. Allow talker execute theirs action as soon as possible and easily use without interruption.
2. Allow failure be acknowledged at least.

Major Change:

1. Allow POST request to be established as soon as possible.
2. Define all control signals as packages in the POST body.
3. For every success action, client will not receive any status code from server.
4. Once any package failed, server will return status code and close connection immediately. Client should keep reading status code when posting data to server to avoid missing receive status code. When receive failed status code, client should close the connection and reconnect to dgtalkie.cgi

Request Definition:

POST /dev2/dgtalkie.cgi

The Post body contains two types of packages, **Control Signals** and **Audio Data**.

1) Packages for control Signals
   - Heart beat
   - Occupy / Release
   - Talk Begin / Talk End
   - End of Session

```
typedef struct _ACS_Control
{
unsigned long ulHdrID;               //Header ID  = 0xF7010000
unsigned long ulHdrLength;         // sizeof(ACS_Control)
unsigned long ulDataLength;        // Should always be 4 bytes for control code
unsigned long ulSequenceNumber; // sequence number
unsigned long ulTimeSec;             //sample time stamp
unsigned long ulTimeUSec;          // sample time stamp
unsigned long ulDataCheckSum;   // As same as the control code
unsigned long uControlCode;
}ACS_Control, *PACS_Control;
```

Control Code list:

- 0x01: Heart beat
  If there is no continuous data post to camera, client should send hear beat to camera periodically in every 30 seconds. If there are no any packages in POST body more than 1 minute, camera will close the connection.
- 0x02: Occupy
  Client requests an occupation to camera's talking line. If failed (already occupied), a "HTTP 409 Conflict" will be responded and the connection will be closed.
- 0x03: Release
  Client requests for release occupation.
- 0x04: Talk Begin
  Client requests sending audio packages. Audio packages will be sent directly following this control package. Camera should mute the microphone and play audio packages consequently.
- 0x05: Talk End
  Client requests stop sending audio packages. No more audio package will be sent following this control package.

- 0x06: End of Session

  Client requests finish POST connection. Camera will return "HTTP 200 OK" and close the connection.
- 0x07: Talk begin without mute

  Client requests sending audio packages. Audio packages will be sent directly following this control package.

2) Packages for Audio Data

The Audio Data package same as speaker.cgi and spaeker2.cgi, Header is defined as below and preferred bytes length of each audio payload is 1K(1024):

```
typedef struct _ACS_AudioHeader
{
unsigned long ulHdrID;               //Header ID  = 0xF6010000
unsigned long ulHdrLength;         // sizeof(ACS_AudioHeader)
unsigned long ulDataLength;        // audio data length, 1024 bytes is preferred.
unsigned long ulSequenceNumber; // sequence number
unsigned long ulTimeSec;             //sample time stamp
unsigned long ulTimeUSec;          // sample time stamp
unsigned long ulDataCheckSum;   // not used…
unsigned short usFormat;              //  0x00000010  S16 LE
unsigned short usChannels;          // 1 channel
unsigned short usSampleRate;       // 8000   hz
unsigned short usSampleBits;        // 16   bits
unsigned long ulReserved;           //
}ACS_AudioHeader, *PACS_AudioHeader;
```

Application Notes for Client Developers:

1. Request for POST connection and sending heart beat immediately when application lunch.
2. Keep-Alive/Content-Length Requirements for Digest Authentication
   I. The first request without digest authentication should be sent with Content-Length 0(without any POST body), and try to reuse the connection if camera response "HTTP 401 Unauthorized" with "Connection: Keep-Alive"
   II. The nonce of Authentication should be kept unless a "stale=true" response is received.
   III. The followed requests with digest authentication should be sent with Content-Length as 2GB(2,000,000,000)
   IV. All responses from camera (expect request with Content-Length as 0) include "failure status code" or "HTTP 200 OK" when end of session request. The "Connection: close" will be added in HTTP header and camera will close the connection.

4.1.8. Get H264 Video Stream

Return H264 video stream. Please refer to ACS Stream Header for more information about video data header.

Request:

GET /video/ACVS-H264.cgi

Parameters:

Name	Value	Description
profileid (Optional.)	#	If “profileid” does not supplied, the URL will output one of stream profile that match assign format (ACVS/H264 stream).

Response:

```
HTTP/1.0 200 OK\r\n
Content-type: video/ACVS\r\n
\r\n
<ACAS Video Stream Data>
```

Where <ACAS Video Stream Data> is defined as below:

```
<ACS_VideoHeader>
<H264 Raw Data>
<ACS_AudioHeader>
<H264 Raw Data>
...
```

`<ACS_VideoHeader>` is defined in **8.2 Advanced IP-Camera Stream (ACS) Header.**
`<MPEG4 Raw Data>` is raw data of MPEG4 video stream.

4.1.9. Get Audio WAVE Stream

Return audio stream in WAV format.

Request:
GET /audio/x-wav.cgi

Parameters:

Name	Value	Description
sec	#	Duration of audio streaming. 0: (default) indicate maximum duration 1-120000: indicate the duration in second

Response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/x-wav\r\n
\r\n
<wave format data>
```

<wave format data> is a standard Microsoft wave file format.
Please refer to **MIME: audio/x-wav**.

4.1.10. Get Audio MS-ADPCM Stream

Return audio stream in MS-ADPCM format.

Request:
GET /audio/ACAS-MSADPCM.cgi

Response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/ACAS\r\n
\r\n
<ACAS Audio Stream Data>
```

Where <ACAS Audio Stream Data> is defined as below:

```
<ACS_AudioHeader>
<Audio Raw Data>
<ACS_AudioHeader>
<Audio Raw Data>
...
```

<ACS_AudioHeader> is defined in **8.2 Advanced IP-Camera Stream (ACS) Header**.
<Audio Raw Data> is raw data of audio stream. The format of this data depends on <ACS_AudioHeader>.

4.1.11. Get Audio MU-LAW Stream

Return audio stream in MU-LAW format.

Request:
GET /audio/ACAS- ULAW.cgi

Response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/ACAS\r\n
\r\n
<ACAS Audio Stream Data>
```

Where <ACAS Audio Stream Data> is defined as below:

```
<ACS_AudioHeader>
<Audio Raw Data>
<ACS_AudioHeader>
<Audio Raw Data>
...
```

<ACS_AudioHeader> is defined in **8.2 Advanced IP-Camera Stream (ACS) Header**.
<Audio Raw Data> is raw data of audio stream. The format of this data depends on <ACS_AudioHeader>.

4.1.12. Get Audio AAC Stream

Return audio stream in AAC format.

Request:
GET /audio/ACAS- AAC.cgi

Response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/ACAS\r\n
\r\n
<ACAS Audio Stream Data>
```

Where <ACAS Audio Stream Data> is defined as below:

```
<ACS_AudioHeader>
<Audio Raw Data>
<ACS_AudioHeader>
<Audio Raw Data>
...
```

`<ACS_AudioHeader>` is defined in **8.2 Advanced IP-Camera Stream (ACS) Header**.
`<Audio Raw Data>` is raw data of audio stream. The format of this data depends on <ACS_AudioHeader>.

4.1.13. Get Audio A-LAW Stream

Return audio stream in A-LAW format.

Request:
GET /audio/ACAS- ALAW.cgi

Response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/ACAS\r\n
\r\n
<ACAS Audio Stream Data>
```

Where <ACAS Audio Stream Data> is defined as below:

```
<ACS_AudioHeader>
<Audio Raw Data>
<ACS_AudioHeader>
<Audio Raw Data>
...
```

`<ACS_AudioHeader>` is defined in **8.2 Advanced IP-Camera Stream (ACS) Header**.
`<Audio Raw Data>` is raw data of audio stream. The format of this data depends on <ACS_AudioHeader>.

4.1.14. Get Profile Audio Stream

Return audio stream associated with a specific profile. The audio stream format(encode method) depends on the compression type of audio in that profile. Please read note below.

Request:
GET /audio/audio.cgi

Parameters:

Name	Value	Description
profileid (Optional)	# (1 to the count of audio profile)	If “profileid” does not supplied, the URL will output the stream of default profile (profile id = 1).

Response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/ACAS\r\n
\r\n
<audio streaming data>
```

Note:
`<audio streaming data>` is audio stream wrapped by ACAS header.

4.1.15. Get Dgtalkie Information

Query information of dgtalkie.cgi. Please refer to 4.1.7.

Request:

GET /common/dgtalkie_info.cgi

Response:

Name	Value	Description
dgtalkie	yes,no	Support dgtalkie or not
supports	String	Supported command list. Separate by ","(Comma) Example: supports=0x01,0x02,0x03,0x04,0x05,0x06,0x07

4.1.16. Get Transport Stream Audio/Video Streaming

Return streaming in MPEG transport stream format. Transport Stream is a standard format for transmission and storage of audio, video and program and System Information Protocol date. It is a container format encapsulating packetized elementary streams. For more information, please refer to ISO/IEC standard documents.

Request:

GET /video/mpegts.cgi

Response:

```
HTTP/1.0 200 OK\r\n
Content-Type: video/mpeg\r\n
Content-Length: <image size>\r\n
\r\n
<Transport Stream Data>\r\n
```

5. Camera Control API

5.1. Remote Control

5.1.1. Query PTZ Information

Request:

GET /config/ptz_info.cgi
GET /ptz/ptz_info.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Response: (Only display parameters which are supported)

Name	Value	Description
pmax	Integer	Maximum degree of pan position
pmin	Integer	Minimum degree of pan position
tmax	Integer	Maximum degree of tilt position
tmin	Integer	Minimum degree of tilt position
zmax	Integer	Maximum degree of zoom position
zmin	Integer	Minimum degree of zoom position
customizedhome	no, yes	Indicate is camera can use "customized home" function. Please refer to section 5.1.9

5.1.2. Get Current PTZ Position

Request:

GET /config/ptz_pos.cgi

GET /ptz/ptz_pos.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Response: (Only display parameters which are supported)

Name	Value	Description
p	Integer	Pan position
t	Integer	Tilt position
z	Integer	Zoom position

5.1.3. Get PTZ Movement Size in a Step

Request:

GET /config/ptz_step.cgi

GET /ptz/ptz_step.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Response: (Only display parameters which are supported)

Name	Value	Description
pstep	Integer	Pan movement size in a step
tstep	Integer	Tilt movement size in a step
zstep	Integer	Zoom movement size in a step

5.1.4. Set PTZ Movement Size in a Step

User can specify any of the parameters users want to set.

Request:

GET/POST /config/ptz_step.cgi

Parameters:

Please refer to the table 5.1.3.

Response:

Please refer to the table 5.1.3

5.1.5. List All PTZ Presets

Request:

GET /config/ptz_preset_list.cgi

GET /ptz/ptz_preset_list.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Response:

Name	Value	Description
presets	<preset name1>, ...	All presets name.

<preset name1>
...
	<p>,<t>,<z>
	...
		Position of the preset name line by line.
		For example,
		door1=100,0
		gate1=-20,-100

5.1.6. Add, Delete or Goto a PTZ Preset

Administrator are able to Add, Delete or Goto a PTZ preset point, however if user request the cgi under /ptz/, user may only allow using "Goto", restriction of folder /ptz depend on model.

Request:

GET/POST /config/ptz_preset.cgi

GET/POST /ptz/ptz_preset.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Parameters:

Name	Value	Description
name	String	Preset name
act	add del go	Add current position to the preset name Delete the preset (only available at /config/ dir) Go to the preset

Response:

Please refer to the table above.

5.1.7. Move PTZ Absolutely

Request:

GET/POST /config/ptz_move.cgi

GET/POST /ptz/ptz_move.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Parameters:

Name	Value	Description
p	Integer	Pans the device relate to the (0,0,0) position.
t	Integer	Tilts the device relate to the (0,0,0) position.
z	Integer	Zooms the device relate to the (0,0,0) position.

Response:

Please refer to the table above. If target position is out of boundary, response will return actual position. (Absolute position)

5.1.8. Move PTZ Relatively

Request:

GET/POST /config/ptz_move_rel.cgi

GET/POST /ptz/ptz_move_rel.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Parameters:

Name	Value	Description
p	-32 ... 32	Pans the device relate to current position.
t	-32 ... 32	Tilts the device relate to current position.
z	-32 ... 32	Zooms the device relate to current position.

Response:
Please refer to the table above. If target position is out of boundary, response will return the actual movement relate to p, t, z values.

5.1.9. Get, Set, Goto, Reset PTZ Customized Home Position

Administrator are able to Get, Set, Goto or Reset PTZ home position, however if user request the cgi under /ptz/, user may only allow using "Goto", restriction of folder /ptz depend on model.

Request:
GET/POST /config/ptz_home.cgi

GET/POST /ptz/ptz_home.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Parameters:

Name	Value	Description
act	get set go reset	Get current home position. This is the default value. Set current home position (only available at /config/ directory) Go to home position Reset home position to factory default. (Only available at /config/ directory)
p	Integer	(Only for act=set) Pans the device relative to the default (0,0,0) position.
t	Integer	(Only for act=set) Tilts the device relative to the default (0,0,0) position.
z	Integer	(Only for act=set) Zooms the device relative to the default (0,0,0) position.

Response:
Return current home position.

Name	Value	Description
p	Integer	Pan position
t	Integer	Tilt position
z	Integer	Zoom position

Note:
If none of any parameters are given, the action is equal to ‘act=get’.
If parameters p,t,z are given but the parameter ‘act’ is not ‘set’, it will be ignored by camera.

5.1.10. Auto Patrol

Request:
GET/POST /config/auto_patrol.cgi

GET/POST /ptz/auto_patrol.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Parameters:

Name	Value	Description
act	go [continue] [stop]	Run PTZ’s auto patrol. act=go means run auto patrol function for one time. act=[continue] means begin continuous patrol mode. act=[stop] means stop patrol.

Response:
Please refer to the table above.

Note:
The item enclosed by [ ] means optional value. That optional value can be used only in some particular models.

5.1.11. Auto Pan

Request:

GET/POST /config/auto_pan.cgi

GET/POST /ptz/auto_pan.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Parameters:

Name	Value	Description
act	go [continue] [stop]	Run PTZ's auto pan. act=go means run auto pan function for one time act=[continue] means begin continuous pan mode. act=[stop] means stop pan action.

Response:

Please refer to the table above.

Note:
The item enclosed by [ ] means optional value. That optional value can be used only in some particular models.

5.1.12. Configure Sequence Order of Presets for Auto Patrol

Request:

GET/POST /config/config_auto_patrol.cgi

Parameters:

Name	Value	Description
presets	<preset name1>, <preset name2>,...	A sequence of preset positions. The CGI auto_patrol function will move camera PTZ preset by this sequence. Maximum count of presets in this sequence is 20. Note: If "presets" is not given, the camera will list current sequence.

Response:

Please refer to the table above.

If preset positions more than 20, only the first 20 presets in the sequence will be kept by camera.

5.1.13. Get, Set the Type of Focus Function (Auto Focus or Manual Focus)

Request:

GET/POST /config/focus_type.cgi
GET/POST /ptz/focus_type.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Parameters:

Name	Value	Description
act	get set	Get current focus type. Set the focus type

type	Integer	The type of focus function (only for act=set)
		type=1: auto focus
		type=0: manual focus

Response:

Name	Value	Description

type	Integer	The type of focus function
		type=1: auto focus
		type=0: manual focus

5.1.14. Adjust the Focus Manually, Focus Near or Focus Far from Current Position

Request:

GET/POST /config/manual_focus.cgi
GET/POST /ptz/manual_focus.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Parameters:

ç	Value	Description
step	Integer (-32…32)	The step to adjust focus more near/far. The positive value means focus far (focus on distant object), and the negative value means focus near (focus on closer object).

Response:

Please refer to the table above.
If target position out of boundary, response will return the actual step performed.
If current focus type is not manual focus, this CGI does nothing and returns "step=0"

5.1.15. Get PTZ Control Privilege Groups

This command allows users to query the list of privilege groups which have privilege to access /ptz/ directory, control PTZ and focus of camera.

Request:

GET /config/ptz_privilege.cgi

Response:

Name	Value	Description
groups	String	List of the groups which have privilege to access /ptz/ directory, control PTZ and focus of camera. Administrator group are always accessible to /ptz/ directory, not matter is listed in this parameter or not. The groups of the list are separated by comma.

5.1.16. Set PTZ Control Privilege Groups

This command allows user to configure the list of privilege groups which could access to /ptz/ directory, control PTZ and focus of camera.

Request:

GET POST /config/ptz_privilege.cgi

Parameters:

Please refer to the table 5.1.15.

Response:

Please refer to the table 5.1.15.

5.1.17. Query Focus Information

Request:

GET /config/focus_info.cgi

GET /ptz/focus_info.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Response:

Name	Value	Description
max	Integer	Maximum position of focus.
min	Integer	Minimum position of focus.

5.1.18. Get Current Focus Position

Request:

GET /config/focus_pos.cgi

GET /ptz/focus_pos.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Response:

Name	Value	Description
focus	An integer	Focus position.

5.1.19. Set Absolutely Focus Position

Request:

GET/POST /config/focus_pos.cgi

GET/POST /ptz/focus_pos.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Parameters:

Please refer to the table 5.1.18.

Response:

Please refer to the table 5.1.18

5.1.20. Fine-tune Focus Automatically

Request:

GET/POST /config/focus_act.cgi

GET/POST /ptz/focus_act.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Parameters:

Name	Value	Description
autofocus	yes	Fine tune focus automatically.

Response:

Name	Value	Description
autofocus	yes, fail	Indicate action is successful or fails.

5.1.21. PTZ Direction of Movement

Request:

GET/POST /config/ptz_direction.cgi
GET/POST /ptz/ptz_direction.cgi (Please refer to 5.1.15 to check accessible PTZ privilege group.)

Parameters:

Name	Value	Description
direction	String	Use “up”、“down”、”left”、”right”、”upleft”、”upright”、”downleft”、”downright”、”stop”、”zoomwide”、”zoomtele”、”zoomstop”.
speed	Integer	Speed control for PTZ (1…10)

5.2. Via RS-485

5.2.1. Execute RS-485 Commands

Request:

GET/POST /config/rs485_do.cgi

Parameters:

Name	Value	Description
direction	0-13	0: wide (zoom out) with stop. 1: up with stop 2: tele (zoom in) with stop 3: left with stop 4: home (only for custom command) 5: right with stop 6: focus far with stop 7: down with stop 8: focus near with stop 9: 10: custom command 1 11: custom command 2 12: custom command 3 13: custom command 4
speed	Integer	Speed control for up, down, left, right. (1 – 10) (Includes: Dyna, Lilin, Lilin2, PelcoD, PelcoP)

5.3. PTDC Pan/Tilt Get Information

5.3.1. Get Pan/Tilt Position

Get current Pan/Tilt position degree.

Request:
GET /cgi/ptdc.cgi?command=get_pos

Response: (Represented by XML)

```
<config>
	<posX>0</posX>
	<posY>0</posY>
</config>
```

5.3.2. Get Pan/Tilt Position by Step

Get current Pan/Tilt position step.

Request:
GET /cgi/ptdc.cgi?command=get_pos_step

Response: (Represented by XML)

```
<config>
	<posXStep>0</posXStep>
	<posYStep>0</posYStep>
</config>
```

5.3.3. Get Pan/Tilt Boundary

Get Pan/Tilt position boundary degree.

Request:
GET /cgi/ptdc.cgi?command=get_boundary

Response: (Represented by XML)

```
<config>
	<boundaryMaxX>164</boundaryMaxX>
	<boundaryMinX>-164</boundaryMinX>
	<boundaryMaxY>83</boundaryMaxY>
	<boundaryMinY>-30</boundaryMinY>
</config>
```

5.3.4. Get Pan/Tilt Boundary by Step

Get Pan/Tilt position boundary by step.

Request:
GET /cgi/ptdc.cgi?command=get_step_boundary

Response: (Represented by XML)

```
<config>
	<boundaryMaxStepX>18944</boundaryMaxStepX>
	<boundaryMinStepX>-18944</boundaryMinStepX>
	<boundaryMaxStepY>14528</boundaryMaxStepY>
	<boundaryMinStepY>-5360</boundaryMinStepY>
</config>
```

5.3.5. Get Pan/Tilt Accuracy

Get current valid Pan/Tilt minimum movement and precision degree.

Request:
GET /cgi/ptdc.cgi?command=get_pt_accuracy

Response: (Represented by XML)

```
<config>
    <panMinMovement>0.14</panMinMovement>
    <panPrecision>0.14</panPrecision>
    <tiltMinMovement>0.18</tiltMinMovement>
    <tiltPrecision>0.09</tiltPrecision>
</config>
```

Note:
1. xxxMinMovement:
User has to set the minimum degree in every movement via Pan and/or Tilt to make sure the physical PT’s movement is visible.

2. xxxPrecision:
The movement unit, user has to set the degree as multiple times of xxxPrecision for PT hardware precision consideration.

5.3.6. Get Pan/Tilt Accuracy by Step

Get current valid Pan/Tilt minimum movement and precision steps.

Request:
GET /cgi/ptdc.cgi?command=get_pt_step_accuracy

Response: (Represented by XML)

```
<config>
    <panMinMovementStep>16</panMinMovementStep>
    <panPrecisionStep>16</panPrecisionStep>
    <tiltMinMovementStep>32</tiltMinMovementStep>
    <tiltPrecisionStep>16</tiltPrecisionStep>
</config>
```

Note:
xxxMinMovementStep:
User has to set the minimum steps in every movement via Pan and/or Tilt to make sure the physical PT’s movement is visible.
xxxPrecisionStep:
The movement unit, user has to set the steps as multiple times of PrecisionStep for PT hardware precision consideration.

5.3.7. Get Pan/Tilt View Angle

Get Pan/Tilt view angle degree in live view.

Request:
GET /cgi/ptdc.cgi?command=get_view_angle

Response: (Represented by XML)

```
<config>
    <viewAngleHorizontal>51</viewAngleHorizontal>
    <viewAngleVertical>39</viewAngleVertical>
</config>
```

5.3.8. Get Pan/Tilt View Step

Get Pan/Tilt view angle degree in live view.

Request:
GET /cgi/ptdc.cgi?command=get_view_step

Response: (Represented by XML)

```
<config>
        <viewStepHorizontal>5865</viewStepHorizontal>
        <viewStepVertical>6747</viewStepVertical>
</config>
```

5.3.9. Get Pan/Tilt Preset Positions

Get current Pan/tilt Preset Positions’ information.

Request
GET /cgi/ptdc.cgi?command=get_preset_positions

Response: (Represented by XML)

```
<config>
        <size>2</size>
        <presetName0>1</presetName0>
        <presetX0>0</presetX0>
        <presetY0>30</presetY0>
        <presetZ0>1</presetZ0>
        <presetF0>0</presetF0>
        <presetFType0>1</presetFType0>
        <presetName1>2</presetName1>
        <presetX1>-60</presetX1>
        <presetY1>30</presetY1>
        <presetZ1>1</presetZ1>
        <presetF1>0</presetF1>
        <presetFType1>1</presetFType1>
</config>
```

Note:

X,Y	Unit: degree (pan/tilt)
Z	Unit: optical mag (zoom)
F	Unit: Step (focus)
FT	0: manual 1: auto (focus type)

5.3.10. Get Pan/Tilt/Zoom Hardware Information

Get current Pan/Tilt/Zoom/Focus hardware information.

Request

GET /cgi/ptdc.cgi?command=get_ptz_hw_info

Response: (Represented by XML)

```
<config>
    <pan>
          <inAngle>
               <viewAngleHorizontal>45</viewAngleHorizontal>
               <boundaryMaxX>164</boundaryMaxX>
               <boundaryMinX>-164</boundaryMinX>
               <panMinMovement>0.14</panMinMovement>
               <panPrecision>0.14</panPrecision>
          </inAngle>
          <inStep>
              <viewStepHorizontal>5175</viewStepHorizontal>
               <boundaryMaxStepX>18944</boundaryMaxStepX>
               <boundaryMinStepX>-18944</boundaryMinStepX>
               <panMinMovementStep>16</panMinMovementStep>
               <panPrecisionStep>16</panPrecisionStep>
          </inStep>
    </pan>
    <tilt>
        <inAngle>
               <viewAngleVertical>38</viewAngleVertical>
               <boundaryMaxY>83</boundaryMaxY>
               <boundaryMinY>-30</boundaryMinY>
               <tiltMinMovement>0.18</tiltMinMovement>
               <tiltPrecision>0.09</tiltPrecision>
          </inAngle>
          <inStep>
              <viewStepVertical>6574</viewStepVertical>
              <boundaryMaxStepY>14528</boundaryMaxStepY>
              <boundaryMinStepY>-5360</boundaryMinStepY>
              <tiltMinMovementStep>32</tiltMinMovementStep>
              <tiltPrecisionStep>16</tiltPrecisionStep>
          </inStep>

    </tilt>
    <zoom>
        <inMag>
               <boundaryMaxZoom>10</boundaryMaxZoom>
               <boundaryMinZoom>1</boundaryMinZoom>
               <zoomMinMovement>0.09</ zoomMinMovement>
               <zoomPrecision>0.09</ zoomPrecision>
        </inMag>
        <inStep>
               <zoomMaxStep>1174</ zoomMaxStep>
               <zoomMinStep>0</zoomMinStep>
               <zoomMinMovementStep>1</zoomMinMovementStep>
               <zoomPrecisionStep>1</zoomPrecisionStep>
        </inStep>
    </zoom>
    <focus>
          <boundaryMaxFocus>1038</boundaryMaxFocus>
          <boundaryMinFocus>194</boundaryMinFocus>
          <focusMinMovementStep>1</focusMinMovementStep>
          <focusPrecisionStep>1</focusPrecisionStep>
    </focus>
</config>
```

Note:
Pan/Tilt Unit: degree
Zoom:
    inMag Unit: optical mag
    inStep Unit: step
Focus Unit: step

5.3.11. Get Pan/Tilt Home Position

Get current Pan/Tilt home position.

Request
GET /cgi/ptdc.cgi?command=get_home_pos

Response: (Represented by XML)

```
<config>
	<homePosX>0</homePosX>
	<homePosY>0</homePosY>
</config>
```

5.3.12. Get Pan/Tilt Patrol Speed

Get patrol speed.

Request
GET /cgi/ptdc.cgi?command=get_patrol_speed

Response: (Represented by XML)

```
<config>
	<speed>5</speed>
</config>
```

5.3.13. Get Pan/Tilt Wait Time

Get patrol waiting time (waitTime).

Request
GET /cgi/ptdc.cgi?command=get_wait_time

Response: (Represented by XML)

```
<config>
	<waitTime>5</waitTime>
</config>
```

Note:
Unit: second
When patrol to a turning point, it will stop and wait for "waitTime" seconds

5.4. PTDC Pan/Tilt set information

5.4.1. Set Pan/Tilt Position

Set Pan/Tilt absolute position in degree unit.

Request:

GET/POST /cgi/ptdc.cgi?command=set_pos&posX=100&posY=30

Parameters:

Name	Value	Description
posX	float	The movement degree base on Zoom mag 1x
posY	float	The movement degree base on Zoom mag 1x

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	posX, posY parameters incorrect
	3. failure	Error command type
	4. oob	Out of boundary

5.4.2. Set Pan/Tilt Relative Position

Set Pan/Tilt relative position in degree unit.

Request:

GET/POST /cgi/ptdc.cgi?command=set_relative_pos&posX=100&posY=30

Parameters:

Name	Value	Description
posX	float	The movement degree base on Zoom mag 1x
posY	float	The movement degree base on Zoom mag 1x

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	posX, posY parameters incorrect
	3. failure	Error command type
	4. oob	Out of boundary

5.4.3. Set Pan/Tilt Position by Step

Set Pan/Tilt absolute position in step unit.

Request:

GET/POST /cgi/ptdc.cgi?command=set_pos_step&posXStep=100&posYStep=30

Parameters:

Name	Value	Description
posXStep	#	The movement step does not relate to Zoom mag 1x
posYStep	#	The movement step does not relate to Zoom mag 1x

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	posXStep, posYStep parameters incorrect.
	3. failure	Error command type
	4. oob	Out of boundary

5.4.4. Set Pan/Tilt Relative Position by Step

Set Pan/Tilt relative position in step unit.

Request:

GET/POST /cgi/ptdc.cgi?command=set_relative_pos_step&posXStep=100&posYStep=30

Parameters:

Name	Value	Description
posXStep	#	The movement step does not relate to Zoom mag 1x
posYStep	#	The movement step does not relate to Zoom mag 1x

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	posXStep, posYStep parameters incorrect
	3. failure	Error command type
	4. oob	Out of boundary

5.4.5. Set Home

Set current Pan/Tilt/Zoom/Focus/FocusType as home position.

Request:

GET/POST /cgi/ptdc.cgi?command=set_home

Response:

A result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.4.6. Restore Default Home

Set home position to factory default.

Request:

GET/POST /cgi/ptdc.cgi?command=restore_home

Response:

A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.4.7. Set Patrol Speed

Set patrol speed. (single_pan/pan_patrol/single_patrol/user_patrol).

Request:

GET/POST /cgi/ptdc.cgi?command=set_patrol_speed&speed=3

Parameters:

Name	Value	Description
speed	#	Patrol speed (1,2,3…)

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	"speed" parameter incorrect
	3. failure	Error command type
	4. oob	Out of boundary

5.4.8. Set Patrol Waiting Time

Set the waiting time to patrol action.

Request:

GET/POST /cgi/ptdc.cgi?command=set_wait_time&waitTime=3

Parameters:

Name	Value	Description
waitTime	#	Waiting time between 0 ~ 3600 sec

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	"waitTime" parameter incorrect
	3. failure	Error command type
	4. oob	Out of boundary

5.5. PTDC Pan/Tilt Other Parts

5.5.1. Calibration

Execute Pan/Tilt calibration, and then move to current home position.

Request:
GET/POST /cgi/ptdc.cgi?command=calibration

Response:
A result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.2. Single Pan

Execute one operation of Pan/Tilt patrol along the pan direction.

Request:
GET/POST /cgi/ptdc.cgi?command=single_pan

Response:
A result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.3. Pan Patrol

Execute Pan/Tilt patrol along the pan direction for user-defined times.

Request:
GET/POST /cgi/ptdc.cgi?command=pan_patrol

Response:
A result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.4. Single Patrol

Execute one operation of Pan/Tilt patrol depend on user-defined route.

Request:

GET/POST /cgi/ptdc.cgi?command=single_patrol

Response:

A result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.5. User Patrol

Execute Pan/Tilt patrol depend on user-defined route and times.

Request:

GET/POST /cgi/ptdc.cgi?command=user_patrol

Response:

A result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.6. Stop Patrol

The soft stop patrol operation. (It will not stop immediately until reach a turning point)

Request:

GET/POST /cgi/ptdc.cgi?command=stop_patrol

Response:

A result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.7. Stop P/T

The hard stop P/T operation. (Pan/Tilt will stop immediately)

Request:

GET/POST /cgi/ptdc.cgi?command=stop

Response:

A result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.8. Go Home

Execute Pan/Tilt operation to home position:

Request:

GET/POST /cgi/ptdc.cgi?command=go_home

Response:

A result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.9. Goto Preset Position

Go to preset point as user definition. (Zero base. Index 0 is the first point as user definition instead of index1)

Request:

GET/POST /cgi/ptdc.cgi?command=goto_preset_position&index=0

or

GET/POST /cgi/ptdc.cgi?command=goto_preset_position&presetName=aaa

Response:

A result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	"index" or “presetName” parameter incorrect.
	3. failure	Error command type

5.6. Get PTDC Zoom/Focus/Focus-Type Information

5.6.1. Get Zoom Boundary

Get Zoom magnification boundary, return 1 represent as 1x mag, 10 represent as 10x mag.

Request:
GET /cgi/ptdc.cgi?command=get_zoom_boundary

Response: (Represented by XML)

```
<config>
      <boundaryMaxZoom>10</boundaryMaxZoom>
      <boundaryMinZoom>1</boundaryMinZoom>
</config>
```

Note:
Unit: mag

5.6.2. Get Zoom Mag

Get current Zoom magnification.

Request:
GET /cgi/ptdc.cgi?command=get_zoom_mag

Response: (Represented by XML)

```
<config>
      <zoomMag>5.5</zoomMag>
</config>
```

Note:
The return value above represent the current optical magnification is 5.5x

5.6.3. Get Zoom Boundary by Step

Get Zoom step boundary.

Request:
GET /cgi/ptdc.cgi?command=get_zoom_step_boundary

Response: (Represented by XML)

```
<config>
      <zoomMaxStep>1174</ zoomMaxStep>
      <zoomMinStep 0</ zoomMinStep>
</config>
```

5.6.4. Get Zoom Step

Get current Zoom step.

Request:
GET /cgi/ptdc.cgi?command=get_zoom_step

Response: (Represented by XML)

```
<config>
      <zoomStep>1174</zoomStep>
</config>
```

Note:
Unit: step
The return value above represent the current optical step is 1174

5.6.5. Get Focus Boundary

Get Focus boundary.

Request:
GET /cgi/ptdc.cgi?command=get_focus_boundary

Response: (Represented by XML)

```
<config>
	<boundaryMaxFocus>1024</boundaryMaxFocus>
	<boundaryMinFocus>194</boundaryMinFocus>
</config>
```

Note:
Unit: step
Return value above represent the range of available focus step.

5.6.6. Get Focus Step

Get current Focus step.

Request:
GET /cgi/ptdc.cgi?command=get_focus_step

Response: (Represented by XML)

```
<config>
	<focusStep>1000</focusStep>
</config>
```

5.6.7. Get Focus Type

Get current Focus Type.

Request:
GET /cgi/ptdc.cgi?command=get_focus_type

Response: (Represented by XML)

```
<config>
	<focusType>1</focusType>
</config>
```

Note:
focusType 0:manual focus 1:auto focus

5.6.8. Get Zoom Accuracy (by Magnification)

Get current Zoom valid minimum movement and precision magnification.

Request:
GET /cgi/ptdc.cgi?command=get_zoom_accuracy

Response: (Represented by XML)

```
<config>
	<zoomMinMovement>0.09</zoomMinMovement>
	<zoomPrecision>0.09</zoomPrecision>
</config>
```

Note:
Unit: mag
xxxMinMovement:
User has to set the minimum mag in every movement via Zoom to make sure the physical Zoom’s movement is visible.
xxxPrecision:
The movement unit, user has to set the mag as multiple times of xxxPrecision for Zoom hardware precision consideration.

5.6.9. Get Zoom Accuracy (by Step)

Get current Zoom valid minimum movement and precision steps.

Request:
GET /cgi/ptdc.cgi?command=get_zoom_step_accuracy

Response: (Represented by XML)

```
<config>
      <zoomMinMovementStep>1</zoomMinMovementStep>
      <zoomPrecisionStep>1</zoomPrecisionStep>
</config>
```

Note:
Unit: step
xxxMinMovementStep:
User has to set the minimum steps in every movement via Zoom to make sure the physical Zoom’s movement is visible.
xxxPrecisionStep:
The movement unit, user has to set the steps as multiple times of PrecisionStep for Zoom hardware precision consideration.

5.6.10. Get Focus Accuracy

Get current focus valid minimum movement and precision steps.

Request:
GET /cgi/ptdc.cgi?command=get_focus_accuracy

Response: (Represented by XML)

```
<config>
      <focusMinMovementStep>1</focusMinMovementStep>
      <focusPrecisionStep>1</focusPrecisionStep>
</config>
```

Note:
Unit: step
xxxMinMovementStep:
User has to set the minimum steps in every movement via focus to make sure the physical focus’s movement is visible.
xxxPrecisionStep:
The movement unit, user has to set the steps as multiple times of PrecisionStep for focus hardware precision consideration.

5.6.11. Get Home Zoom Mag

Get current home Zoom magnification.

Request:
GET /cgi/ptdc.cgi?command=get_home_zoom_mag

Response: (Represented by XML)

```
<config>
      <homeZoomMag>5.5</homeZoomMag>
</config>
```

Note:
Unit: mag
The return value above represent the home optical mag is 5.5x

5.6.12. Get Home Zoom Step

Get current home Zoom step.

Request:
GET /cgi/ptdc.cgi?command=get_home_zoom_step

Response: (Represented by XML)

```
<config>
	<homeZoomStep>1174</homeZoomStep>
</config>
```

Note:
Unit: step
The return value above represent the home optical step is 1174

5.6.13. Get Home Focus Step

Get current home focus step.

Request:
GET /cgi/ptdc.cgi?command=get_home_focus_step

Response: (Represented by XML)

```
<config>
	<homeFocusStep>1000</homeFocusStep>
</config>
```

Note:
Unit: step
The return value above represent the home focus step is 1000

5.6.14. Get Home Focus Type

Get current home focus type.

Request:
GET /cgi/ptdc.cgi?command=get_home_focus_type

Response: (Represented by XML)

```
<config>
	<homeFocusType>1</homeFocusType>
</config>
```

Note:
focusType = 0: manual

focusType = 1: auto focus

5.7. Set PTDC Zoom/Focus/Focus-Type Information

5.7.1. Set Zoom Mag

Set absolute Zoom position in magnification unit.

Request:

GET/POST /cgi/ptdc.cgi?command=set_zoom&zoom_mag=5.5

Parameters:

Name	Value	Description
zoom_mag	float	Zoom mag

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	"zoom_mag" parameters incorrect
	3. failure	Error command type
	4. oob	Out of boundary

5.7.2. Set Relative Zoom Mag

Set relative Zoom position in magnification unit.

Request:

GET/POST /cgi/ptdc.cgi?command=set_relative_zoom&zoom_mag=2.5

Parameters:

Name	Value	Description
zoom_mag	float	Zoom mag

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	"zoom_mag" parameters incorrect
	3. failure	Error command type
	4. oob	Out of boundary

Note:
Unit: mag
The example request is going to increase 2.5x mag to current Zoom mag.
For example: original mag is 5.5x, and then goes to 8.0x after this command.

5.7.3. Set Zoom Step

Set absolute Zoom position in step unit.

Request:

GET/POST /cgi/ptdc.cgi?command=set_zoom_step&zoom_step=50

Parameters:

Name	Value	Description
zoom_step	#	Zoom step

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	"zoom_step" parameters incorrect
	3. failure	Error command type
	4. oob	Out of boundary

5.7.4. Set Relative Zoom Step

Set relative Zoom position in step unit.

Request:

GET/POST /cgi/ptdc.cgi?command=set_relative_zoom_step&zoom_step=25

Parameters:

Name	Value	Description
zoom_step	#	Zoom step

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	"zoom_step" parameters incorrect
	3. failure	Error command type
	4. oob	Out of boundary

Note:
Unit: step
Sample request is going to increase 25 steps to current zoom step.
For example: Original zoom step is 50, and then goes to 75 after this command.

5.7.5. Set Focus Step

Set absolute focus position in step unit.

Request:

GET/POST /cgi/ptdc.cgi?command=set_focus&focus_step=500

Parameters:

Name	Value	Description
focus_step	#	Focus steps

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	"focus_step" parameters incorrect
	3. failure	Error command type
	4. oob	Out of boundary

5.7.6. Set Relative Focus Step

Set relative Focus position in step unit.

Request:

GET/POST /cgi/ptdc.cgi?command=set_relative_focus_step&focus_step=25

Parameters:

Name	Value	Description
focus_step	#	Focus step

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	"focus_step" parameters incorrect
	3. failure	Error command type
	4. oob	Out of boundary

Note:
Unit: step
Sample request is going to increase 50 steps to current Focus step.
For example: Original Focus step is 250, and then goes to 300 after this command.

5.7.7. Set Focus Type

Set Focus type as manual/auto mode:

Request:

GET/POST /cgi/ptdc.cgi?command=set_focus_type&focus_type=0

Parameters:

Name	Value	Description
focus_type	0 or 1	Focus Type

Response:

An extra result tag will also be presented:

Result: (Represented by XML)	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Need correct focus_type parameter
	3. failure	Error command type
	4. oob	Out Of Boundary

Note:
focus_type=0 : manual

focus_type=1 : auto focus

5.8. Digital PTZ Control

5.8.1. Get Current Digital PTZ Position

Request:

GET /config/digital_ptz_pos.cgi

Parameters:

Name	Value	Description
profileid	#	Profile number (# is a number from 1 to count of profiles)

Response:

Name	Value	Description
profileid	#	Profile number (# is a number from 1 to count of profiles)
p	Integer	The digital pan position.
t	Integer	The digital tilt position.
z	Integer	The digital zoom position.

5.8.2. Add, Delete or Goto a Digital PTZ Preset

Request:

GET /config/digital_ptz_preset.cgi

Parameters:

Name	Value	Description
profileid	#	Profile number(# is a number from 1 to count of profiles)
act	add, del, go	Action type.
name	String	Digital preset name

Response:

Please refer to the table above. If the movement is out of boundary, response will return actual absolute position.

5.8.3. Move Digital PTZ Absolutely

Request:

GET /config/digital_ptz_move.cgi

Parameters:

Name	Value	Description
profileid	#	Profile number (# is a number from 1 to count of profiles)
p	Integer	Pans the device relative to the (0,0,0) position
t	Integer	Tilts the device relative to the (0,0,0) position
z	Integer	Zooms the device relative to the (0,0,0) position.

Response:

Please refer to the table above. If the movement is out of boundary, response will return the actual absolute position.

5.8.4. Move Digital PTZ Relatively

Request:

GET /config/digital_ptz_move_rel.cgi

Parameters:

Name	Value	Description
profileid	#	Profile number (# is a number from 1 to count of profiles)
p	Integer	Pans the device relative to the current position
t	Integer	Tilts the device relative to the current position
z	Integer	Zooms the device relative to the current position.

Response:

Please refer to the table above. If the movement is out of boundary, response will return actual relative p, t, z values it moved.

5.8.5. Digital PTZ Auto-pan

Request:

GET /config/digital_ptz_autopan.cgi

Parameters:

Name	Value	Description
profileid	#	Profile number (# is a number from 1 to count of profiles)
act	go, stop	Action type.

Response:

Please refer to the table above. If the movement is out of boundary, response will return the actual relative p, t, z values it moved.

5.8.6. Digital PTZ Sequence

Request:

GET /config/digital_ptz_sequence.cgi

Parameters:

Name	Value	Description
profileid	#	Profile number (# is a number from 1 to count of profiles)
act	add, del, go, stop	Action type.
name	String	Only for “act=add”
index	Integer	From 0 to max
time	Integer	Only for “act=add”

Response:

Please refer to the table above. If the movement is out of boundary, response will return the actual relative p, t, z values it moved.

6. Notification API

6.1. Camera Status Notification

Please note that "notify.cgi" and "notify_stream.cgi" have two paths for HTTP GET request. The first way is under /config/ and second is under /users. Both of their behaviors are totally the same. The only difference is when CGIs in folder /config need administrator privilege, and in folder /users only need user privilege. The notify.cgi and notify_stream.cgi under /config are obsolete, and using them under folder /users are encouraged.

6.1.1. Get Notification Status

This CGI is a one-shot command, which only return current status of IP camera. If user wants to monitor camera status for a long time, please using 6.1.2. notify_stream.cgi instead.

Request:

GET /config/notify.cgi

or

GET /users/notify.cgi

Response:

Name	Value	Description
md#	on, off	Event motion detection # is triggered or not.
mdv#	Integer	Percentage of motion detected by camera. Degree of motion. Range from 0~100.
pir	on, off	PIR is trigged or not
input#	on, off	Event input # is triggered or not.
storagefull	on	Event storage full. (When free storage space less than user definition.)
storagefail	on	Event storage fail. (When storage cannot be accessed successfully)
recording	on, off	Status is recording
snapshooting	on, off	Status is taking snapshot
output#	on, off	Status of output # is on or off
vsignal	on, off	Status of video signal is on or lost
speaker	on, off	Status of speaker is on or off
mic	on, off	Status of microphone is on or off
irled	on, off	Status of IR LED is on or off
led	on, off	Status of special purpose LED is on or off
audio_detected	on, off	Audio detection triggered or not.
audio_detect_val	Integer	Decibel level from audio detection.
speaker_occupied	on, off	Speaker status is occupied or not. (half-duplex)
mic_muted	on, off	Microphone is muted or not
td	disable,normal,hot,cold	Thermal detection status.
tpC	Integer	Celsius temperature
tpF	Integer	Fahrenheit temperature
playing_music	on, off	Playing music or not.
autofocusbusy	yes, no	Status of autofocus is on or off
cameraname	<camera name>	Camera name
white_light_led	on, off	Status of White Light LED is lit on or off

6.1.2. Get Notification Stream

Request:

GET /config/notify_stream.cgi

or

GET /users/notify_stream.cgi

Response:

Client side should keep receive notification information from camera. The information includes all available events or status as following table. The notification information only generate when event or status changed. If there is no changed event or status being reported within 30 seconds, a special tag: “`keep_alive`” will be sent to the client side.

Name	Value	Description
md#	on, off	Event motion detection # is triggered or not.
mdv#	Integer	Percentage of motion detected by camera. Degree of motion. Range from 0~100.
pir	on, off	PIR is trigged or not
input#	on, off	Event input # is triggered or not.
storagefull	on	Event storage full. (When free storage space less than user definition.)
storagefail	on	Event storage fail. (When storage cannot be accessed successfully)
recording	on, off	Status is recording
snapshooting	on, off	Status is taking snapshot
output#	on, off	Status of output # is on or off
vsignal	on, off	Status of video signal is on or lost
speaker	on, off	Status of speaker is on or off
mic	on, off	Status of microphone is on or off
irled	on, off	Status of IR LED is on or off
led	on, off	Status of Special purpose LED is on or off
audio_detected	on, off	Audio detection triggered or not.
audio_detect_val	Integer	Decibel level from audio detection.
speaker_occupied	on, off	Speaker status is occupied or not. (half-duplex)
mic_muted	on, off	Microphone is muted or not
td	disable,normal,hot,cold	Thermal detection status.
tpC	Integer	Celsius temperature
tpF	Integer	Fahrenheit temperature
playing_music	on, off	Playing music or not.
autofocusbusy	yes, no	Status of autofocus is on or off
cameraname	<camera name>	Camera name
white_light_led	on, off	Status of White Light LED is lit on or off

7. RTSP API

The Real Time Streaming Protocol (RTSP) is a network protocol designed for getting audio and video streaming data provided by a media server. The IP camera can act as a media server and stream real time audio and video. By using RTSP request, a client application can get streaming data from IP camera. For more information about RTSP protocol, please refer to RFC 2326.

7.1. Live Streaming

7.1.1. Get URL Entry of Specified Profile

Request:

GET /config/rtspurl.cgi

Parameters:

profileid=<video profile number>

Response:

Name	Value	Description
profileid	#	Profile number (# is a number from 1 to the count of profiles)
urlentry	<entry of video profile>	URL entry of associated video stream profile

7.1.2. Set Video Configuration

Request:

GET/POST /config/rtspurl.cgi

Parameters:

Please refer to the table 7.1.1

Response:

Please refer to the table 7.1.1

7.1.3. Get Live Video

The requested URI of an IP camera stream data can be described by following:

rtsp://<server ip>/<urlentry>

Users are able to get RTSP stream on PC or any other RTSP player.

Where <urlentry> is a URI entry associated with one of the video profile.

The value can be gotten by calling **/config/rtspurl.cgi** (Please refer to 7.1.1)

NOTE:

Due to our camera allow users to change the URL entry of each video profile, the following RTSP URLs are obsoleted.

User should use 7.1.1 to get RTSP stream URL.

~~rtsp://<server ip>/mp4~~

(Get video and audio stream with MPEG-4 video format)

~~rtsp://<server ip>/jpeg~~

(Get video (and audio) stream with MJPEG video format)

~~**rtsp://<server ip>/3gpp**~~

(Get video (and audio) stream with MPEG-4 video format from 3GPP compliant device.)

~~**rtsp://<server ip>/live#**~~

The # is a number from 1 to the count of video profile.

For example: using **rtsp://192.168.1.1/live1** to get the stream of video profile number 1.

7.2 RTSP Methods:

Please use RTSP method "OPTIONS" to get the other methods supported by the IP camera.

OPTIONS: Report the methods supported by the IP camera.

8. Appendix

8.1. Table Used in NIPC

Table 1: Time Zone

ID	Time zone
1	(GMT-12:00) International Date Line West
2	(GMT-11:00) Samoa
3	(GMT-10:00) Hawaii
4	(GMT-09:00) Alaska
5	(GMT-08:00) Pacific Time (US & Canada)
6	(GMT-08:00) Tijuana, Baja California
7	(GMT-07:00) Chihuahua, La Paz, Mazatlan
8	(GMT-07:00) Mountain Time (US & Canada)
9	(GMT-07:00) Arizona
10	(GMT-06:00) Central America
11	(GMT-06:00) Guadalajara, Mexico City, Monterrey
12	(GMT-06:00) Saskatchewan
13	(GMT-06:00) Central Time (US & Canada)
14	(GMT-05:00) Bogota, Lima, Quito
15	(GMT-05:00) Eastern Time (US & Canada)
16	(GMT-05:00) Indiana (East)
17	(GMT-04:00) La Paz, Georgetown, San Juan
18	(GMT-04:00) Atlantic Time (Canada)
19	(GMT-04:00) Santiago
20	(GMT-04:00) Manaus
21	(GMT-03:30) Newfoundland
22	(GMT-03:00) Buenos Aires
23	(GMT-03:00) Brasilia
24	(GMT-03:00) Greenland
25	(GMT-03:00) Montevideo
26	(GMT-02:00) Mid-Atlantic
27	(GMT-01:00) Azores
28	(GMT-01:00) Cape Verde Is.
29	(GMT) Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London
30	(GMT) Monrovia, Reykjavik
31	(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague
32	(GMT+01:00) West Central Africa
33	(GMT+01:00) Sarajevo, Skopje, Warsaw, Zagreb
34	(GMT+01:00) Brussels, Copenhagen, Madrid, Paris
35	(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
36	(GMT+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius
37	(GMT+02:00) Athens, Bucharest, Istanbul
38	(GMT+02:00) Beirut
39	(GMT+02:00) Harare, Pretoria
40	(GMT+02:00) Cairo
41	(GMT+03:00) Minsk, Kaliningrad
42	(GMT+02:00) Amman
43	(GMT+01:00) Windhoek
44	(GMT+02:00) Jerusalem
45	(GMT+03:00) Baghdad
46	(GMT+04:00) Moscow, St. Petersburg, Volgograd
47	(GMT+04:00) Tbilisi

ID	Time zone
48	(GMT+03:00) Nairobi
49	(GMT+03:00) Kuwait, Riyadh
50	(GMT+03:30) Tehran
51	(GMT+04:00) Baku
52	(GMT+04:00) Abu Dhabi, Muscat
53	(GMT+04:00) Yerevan
54	(GMT+04:30) Kabul
55	(GMT+06:00) Yekaterinburg
56	(GMT+05:00) Islamabad, Karachi, Tashkent
57	(GMT+05:30) Chennai, Kolkata, Mumbai, New Delhi
58	(GMT+05:30) Sri Jayawardenepura
59	(GMT+05:45) Kathmandu
60	(GMT+06:00) Astana, Dhaka
61	(GMT+07:00) Novosibirsk
62	(GMT+06:30) Yangon (Rangoon)
63	(GMT+08:00) Krasnoyarsk
64	(GMT+07:00) Bangkok, Hanoi, Jakarta
65	(GMT+08:00) Beijing, Chongqing, Hong Kong, Urumqi
66	(GMT+08:00) Taipei
67	(GMT+09:00) Irkutsk
68	(GMT+08:00) Perth
69	(GMT+08:00) Kuala Lumpur, Singapore
70	(GMT+10:00) Yakutsk
71	(GMT+09:00) Osaka, Sapporo, Tokyo
72	(GMT+09:00) Seoul
73	(GMT+09:30) Adelaide
74	(GMT+09:30) Darwin
75	(GMT+10:00) Hobart
76	(GMT+10:00) Brisbane
77	(GMT+11:00) Vladivostok
78	(GMT+10:00) Canberra, Melbourne, Sydney
79	(GMT+10:00) Guam, Port Moresby
80	(GMT+12:00) Magadan
81	(GMT+12:00) Fiji
82	(GMT+12:00) Auckland, Wellington
83	(GMT+13:00) Nukualofa
84	(GMT-04:30)Caracas
85	(GMT+11:00) Solomon Is., New Caledonia
86	(GMT) Casablanca
87	(GMT+08:00) Ulaanbaatar
88	(GMT-04:00) Asuncion
89	(GMT-04:00) Cuiaba
90	(GMT-03:00) Cayenne, Fortaleza
91	(GMT-03:00) Salvador
92	(GMT+02:00) Damascus
93	(GMT+02:00) Nicosia
94	(GMT+04:00) Port Louis

Table 2: Dynamic DNS Service Providers

ID	provider URIs
	www.ez-ip.net
	www.penguinpowered.com
	members.dhs.org
dyndns	members.dyndns.org
	www.3322.org
	update.ods.org
	cgi.tzo.com
	members.easydns.com
	api.easydns.com
	www.justlinux.com
	www.dyns.cx
	dup.hn.org
	www.zoneedit.com
	ipv6tb.he.net

8.2. Advanced IP-Camera Stream (ACS) Header

Multimedia header:

ACS Audio header	ACS Video header

typedef struct _ACS_AudioHeader
{
unsigned long ulHdrID; //Header ID
unsigned long ulHdrLength;
unsigned long ulDataLength;
unsigned long ulSequenceNumber;
unsigned long ulTimeSec;
unsigned long ulTimeUSec;
unsigned long ulDataCheckSum;
unsigned short usFormat;
unsigned short usChannels;
unsigned short usSampleRate;
unsigned short usSampleBits;
unsigned long ulReserved;
}ACS_AudioHeader, *PACS_AudioHeader;
	typedef struct _ACS_VideoHeader
	{
	unsigned long ulHdrID; //Header ID
	unsigned long ulHdrLength;
	unsigned long ulDataLength;
	unsigned long ulSequenceNumber;
	unsigned long ulTimeSec;
	unsigned long ulTimeUSec;
	unsigned long ulDataCheckSum;
	unsigned short usCodingType;
	unsigned short usFrameRate;
	unsigned short usWidth;
	unsigned short usHeight;
	unsigned char ucMDBitmap;
	unsigned char ucMDPowers[3];
	}ACS_VideoHeader, *PACS_VideoHeader

Description:

The byte order of this header is little-endian.

Common part:

ulHdrID: Special id for identifying ACS header. For audio: the value of this id is 0xF6010000 (since our header is in little-endian so the byte array of this id is ’00 00 01 F6’). For video the value is 0xF5010000.

ulHdrLength: Header length. (32 bytes in current version)

ulDataLength: Payload data length.

ulSequenceNumber: Sequence number.

ulTimeSec: Time stamp in sec since 1970/01/01 00:00.

ulTimeUSec: Microsecond part of time stamp

ulDataCheckSum: Store last 4 bytes of payload data.

Audio part:

usFormat: Audio data format. The possible value:

AFMT_MS_ADPCM: 0

AFMT_MU_LAW: 1

AFMT_A_LAW: 2

AFMT_IMA_ADPCM: 4

AFMT_U8: 8

AFMT_S16_LE: 0x10 /* Little endian signed 16*/

AFMT_S16_BE: 0x20 /* Big endian signed 16 */

AFMT_S8: 0x40

AFMT_U16_LE: 0x80 /* Little endian U16 */

AFMT_U16_BE: 0x100 /* Big endian U16 */

AFMT_MPEG: 0x200 /* MPEG (2) audio */

AFMT_AC3: 0x400

AFMT_AMR: 0x800

AFMT_AAC: 0x1000

AFMT_ALAW: 0x2000

usChannels: Audio channels number: mono(1) or stereo(2).

usSampleRate: Sample rate.

usSampleBits: Bits count per sample.

ulReseverd: Reserved.

Video part:

usCodingType: Encoding type of frame. The possible values are:

VFCT_IVOP (MPEG4): 0

VFCT_PVOP (MPEG4): 1

VFCT_JPEG: 5

VFCT_H264_IFRM: 10

VFCT_H264_PFRM: 11

usFrameRate: Frames per second.

usWidth: The width of frame dimension

usHeight: The height of frame dimension

ucMDBitmap: The height of frame dimension

ucMDPowsers[3]: The height of frame dimension

Extension header:

We propose extensive header for dealing with other information attaching with multimedia stream. Instead of appending this kind of information to multimedia stream, it will save more bandwidth utilization.

Table: Alphanetworks IP-Camera streaming (ACS) extension header:

ACS extension header

```
typedef struct _ACS_ExtentHeader
{
     unsigned long ulHdrID; // ’00 00 01 FE’
     unsigned long ulHdrLength;
     unsigned char pbyReserved[96];
} ACS_ExtentHeader, *PACS_ExtentHeader
```

Description:

The extension header is interleaved within the video stream or audio stream when information required by client.

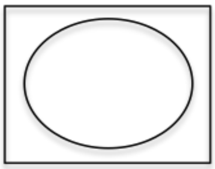
`ulHdrID`: Special id for identifying ACS header. 0xFE010000.

`ulHdrLength`: Header length. (32 bytes in current version)

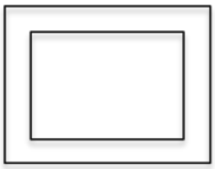
`pbyReserved[96]`: To be defined.

8.3. Fisheye Display Mode Schematic Information

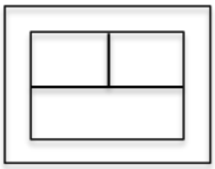
1O:



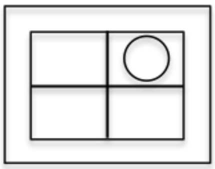
1R



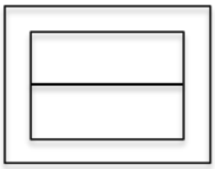
1P2R



1O3R



2P



4R