

API Guide

ANT-38500



For further product information and downloads please visit
www.antrica.com

Table of Contents

Table of Contents	1
1. Overview	4
1.1 About This Document	4
1.2 Syntax and Usage	4
2. API Groups	6
2.1 General	6
2.2 ReadParam	7
2.2.1. Read page (readpage)	7
2.2.2. Read Parameter (readparam)	7
2.3 WriteParam	8
2.3.1. Write Parameter (writeparam)	8
2.4 GetState	11
2.4.1. Get input state (getinput)	11
2.4.2. Get rate state (getrate)	12
2.5 SetState	13
2.5.1. Set Output (setoutput)	13
2.6 SendPTZ	15
2.6.1. Send PTZ command (sendptz)	15
2.7 SerialPort	24
2.7.1. Serial port command (serialport)	24
2.7.2. Serial port access through TCP channel	28
2.8 SearchData	28
2.8.1. Get Root command (getroot)	28
2.8.2. Get List command (getlist)	30
2.8.3. Delete File command (deletefile)	31

2.8.4. Retrieve file.....	32
2.9 DrawOSD (Supported model only)	33
2.9.1. Draw Text OSD Command (drawtext)	33
2.9.2. Draw Bitmap OSD Command (drawbitmap)	34
2.9.3. Draw Object OSD Command (drawobject)	36
2.9.4. Erase Area Command (erase).....	38
2.9.5. OSD Image Command (osdimage).....	40
2.9.6. Update OSD Command (update).....	41
2.9.7. Delete Bitmap (delete).....	42
2.9.8. Upload Bitmap (HTTP POST)	43
2.10 SendAudio	44
2.11 UploadFile	44
2.11.1. Upload Upgrade file (HTTP POST)	44
2.11.2. Upload Config file (HTTP POST).....	44
2.11.3. Upload IVS License file (HTTP POST).....	45
2.11.4. Upload user-defined Bitmap OSD file (HTTP POST).....	45
3. Capture Groups.....	47
3.1 Capturing One JPEG Still Image	47
3.2 Receiving Sequential JPEG Stream	47
Appendix A. PTZ Coordinate Range.....	50
A.1 Truen IP zoom/PTZ camera	50
Appendix B. Continuous PTZ operation.....	51
Appendix C. Privacy Mask Setting.....	52
C.1 Finding mask drawing type from camera	52
C.2 Type1.....	52
C.3 Type2.....	54
C.4 Type3.....	55
C.5 Type4.....	56



1. Overview

1.1 About This Document

This document specifies the external HTTP-based application programming interface of Antrica cameras and video servers. The term 'server' or 'system' in this document refers to an IP camera or video server.

The HTTP-based interface provides the functionality for reading and writing internal parameters and sending commands such as PTZ control.

1.2 Syntax and Usage

HTTPAPI command should have the following syntax which is the standard way of specifying a URL according to RFC 2396.

```
http://<server>/httpapi/<api_group_name>?action=<action_name>
&<parameter>=<value> [&<parameter>=<value>...]
```

Spaces and other reserved characters (";", "/", "?", ":", "@", "&", "=", "+", ",", and "\$") within a <parameter> or a <value> must be replaced with %<ASCII hex>. For example, in the string My camera, the space must be replaced with %20, My%20camera.

Only HTTP GET method should be used.

The server's reply has the following syntax.

```
HTTP/1.0 <HTTP code> <HTTP text>\r\n
Content-Type: text/plain\r\n
\r\n
<Content>\r\n
```

Standard HTTP status codes are used for <HTTP code>, and it always returns "200 OK". If the command is rejected or failed, <Content> includes the error code.

2. API Groups

2.1 General

Each API belongs to one of 10 API groups which are summarized in the following table.

API Group	Action Name	Description
ReadParam	readpage readparam	Read settings in a server
WriteParam	writeparam	Change settings in a server
GetState	getinput getrate getdisk getupgrade getlog	Get states of a server. States include sensor ports, video loss state, motion detection state, various statistics, recording state etc.
SetState	setoutput	Set states of a server's ports such as relay.
SendPTZ	sendptz	Send PTZ commands to a server
SerialPort	serialport	Read or write data from/to a server's serial port.
SearchData	getroot	Retrieve the information of root folder
	getlist	Retrieve the file list of each date
	deletefile	Delete files.
SendAudio		Send audio data to a server
UploadFile		Upload files such as Upgrade file, Config file, IVS License file.
System test		Send command for system diagnostics

2.2 ReadParam

2.2.1. Read page (readpage)

Read all parameters in a page. Refer another document, **Configuration Parameters.pdf**, for descriptions on all available pages and parameters. In fact, a page corresponds to the page in web-based setup.

Parameters	Values	Description
page	<string> system, video,	

Only one page can be read at a time.

Example) To read video page

<pre>http://<server>/httpapi/ReadParam?action=readpage&page=video</pre>
<pre>HTTP/1.0 200 OK\r\n Content-Type: text/plain\r\n \r\n VID_PREFERENCE=0\r\n VID_RESOLUTION=1\r\n VID_FRAMERATE=3\r\n VID_QUALITY=2\r\n</pre>

2.2.2. Read Parameter (readparam)

Read one or more selected parameters. Parameters belonging to different

pages can be read in one command.

Parameters	Values	Description
Any setup parameter	<int> 0	

Up to 48 parameters can be read in one command.

Example) To read system ID and system version

<pre>http://<server>/httpapi/ReadParam?action=readparam&SYS_SYS TEMID=0&SYS_VERSION=0</pre>
<pre>HTTP/1.0 200 OK\r\n Content-Type: text/plain\r\n \r\n SYS_SYSTEMID=My Video Server\r\n SYS_VERSION=V1.102J\r\n</pre>

2.3 WriteParam

2.3.1. Write Parameter (writeparam)

Set the value of a parameter.

Parameters	Values	Description
page	<string> system, video,	
channel	<int>1 ~ 4	Multi channel server only
configset	<int>1 ~ 4	If multi config supported
Any setup parameter	Parameter dependent	

More than one parameter can be set in one command. However, those

parameters should belong to the same page. Page parameter can be omitted. The 'channel' parameter can be located at any position in the multi channel video server. In this case, all parameters after 'channel' are affected until the next 'channel' parameter is found.

Usage of 'configset' is similar to that of 'channel'. Please refer to the example below.

Example) To change System ID and System Mode (with specifying the page)

<pre>http://<server>/httpapi/WriteParam?action=writeparam&page=system&SYS_SYSTEMID=My%20system&SYS_SYSTEMMODE=0</pre>
<pre>HTTP/1.0 200 OK\r\nContent-Type: text/plain\r\n\r\nOK\r\n</pre>

Example) To change System ID and System Mode (without specifying the page)

<pre>http://<server>/httpapi/WriteParam?action=writeparam&SYS_SYSTEMID=My%20system&SYS_SYSTEMMODE=0</pre>
<pre>HTTP/1.0 200 OK\r\nContent-Type: text/plain\r\n\r\nOK\r\n</pre>

Example) To change the resolution and frame rate of each channel in 4 channel video server.

<pre>http://<server>/httpapi/WriteParam?action=writeparam&chann</pre>

```
e1=1&VID_RESOLUTION=1&VID_FRAMERATE=0&channel=2&VID_RESOLUTION=2&VID_FRAMERATE=3&channel=3&VID_RESOLUTION=1&VID_FRAMERATE=1&channel=4&VID_RESOLUTION=1&VID_FRAMERATE=0
```

```
HTTP/1.0 200 OK\r\n  
Content-Type: text/plain\r\n  
\r\n  
OK\r\n
```

Example) To change the Day&Night Mode of each config set in the camera which supports 4 config set

```
http://<server>/httpapi/WriteParam?action=writeparam&configset=1&CAM_SONY_DN=1&configset=2&CAM_SONY_DN=0&configset=3&CAM_SONY_DN=1&configset=4&CAM_SONY_DN=1
```

```
HTTP/1.0 200 OK\r\n  
Content-Type: text/plain\r\n  
\r\n  
OK\r\n
```

Depending on the parameters, the system may reboot or connections to clients may be disconnected. Such parameter-specific operation of the server is specified in the description of each parameter if any.

Parameters such as SYS_REBOOT and SYS_RESET which cause special operation of the server can't be used with other configuration parameters.

Example) To reboot the server

```
http://<server>/httpapi/WriteParam?action=writeparam&SYS_R
```

```
EBOOT=0
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
OK\r\n
```

Up to 48 parameters can be set in one command.

2.4 GetState

2.4.1. Get input state (getinput)

Get state of device ports, video input etc.

Parameters	Values	Description
GIS_SENSOR1 GIS_SENSOR2 GIS_SENSOR3 GIS_SENSOR4	<int> 0 or 1	0 : Not detected 1 : Detected
GIS_VIDEOLOSS1 GIS_VIDEOLOSS2 GIS_VIDEOLOSS3 GIS_VIDEOLOSS4	<int> 0 or 1	0 : Video detected 1 : Video loss
GIS_MOTION1 GIS_MOTION2 GIS_MOTION3 GIS_MOTION4	<int> 0 or 1	0 : No motion 1 : Motion detected

GIS_BUZZER	<int> 0 or 1	0 : Deactivated 1 : Activated
GIS_ALARM1 GIS_ALARM2 GIS_ALARM3 GIS_ALARM4	<int> 0 or 1	0 : Deactivated 1 : Activated
GIS_RECORD1 GIS_RECORD2 GIS_RECORD3 GIS_RECORD4	<int> 0 or 1	0 : Not recording 1 : Recording

More than one parameter can be read in any order in one command.

Example) To read the state of sensor port and others

```

http://<server>/httpapi/GetState?action=getinput&GIS_SENSOR1=0&GIS_VIDEOLOSS1=0&GIS_MOTION1=0
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
GIS_SENSOR1=1\r\n
GIS_VIDEOLOSS1=0\r\n
GIS_MOTION1=0\r\n

```

2.4.2. Get rate state (getrate)

Retrieve rate(statistics) information.

Parameters	Values	Description
------------	--------	-------------

GRS_VENCFRAME1	<int> 0 ~ 30	Video encoding frame rate (fps)
GRS_VENCBITRATE1	<int> 0 ~	Video encoding bitrate (kbps)
GRS_VDECFRAME1	<int> 0 ~ 30	Video decoding frame rate (fps)
GRS_VDECBITRATE1	<int> 0 ~	Video decoding bitrate (kbps)
GRS_AENCBITRATE1	<int> 0 ~	Audio encoding bitrate (kbps)
GRS_ADECBITRATE1	<int> 0 ~	Audio decoding bitrate (kbps)
GRS_STXRATE1 GRS_STXRATE2	<int> 0 ~	Current Tx Data rate of serial port (bps)
GRS_SRXRATE1 GRS_SRXRATE2	<int> 0 ~	Current Rx Data rate of serial port (bps)

Example) To retrieve frame rate and bitrate of video channel 1

<pre>http://<server>/httpapi/GetState?action=getrate&GRS_VENCFRAME1=0&GRS_VENCBITRATE1=0</pre>
<pre>HTTP/1.0 200 OK\r\n Content-Type: text/plain\r\n \r\n GRS_VENCFRAME1=30\n GRS_VENCBITRATE1=1920\n</pre>

2.5 SetState

2.5.1. Set Output (setoutput)

Set the state of devices or ports in the server.

Parameters	Values	Description
------------	--------	-------------

OUT_BUZZER	<int> 0 or 1	0 : Deactivate 1 : Activate
OUT_ALARM1 OUT_ALARM2 OUT_ALARM3 OUT_ALARM4	<int> 0 or 1	0 : Deactivate 1 : Activate
BTN_UP BTN_DOWN BTN_LEFT BTN_RIGHT BTN_SET	<int> 0 or 1	0 : Deactivate 1 : Activate
FASTUPDATE_CH1_PRI FASTUPDATE_CH2_PRI FASTUPDATE_CH3_PRI FASTUPDATE_CH4_PRI FASTUPDATE_CH1_SEC FASTUPDATE_CH2_SEC FASTUPDATE_CH3_SEC FASTUPDATE_CH4_SEC FASTUPDATE_CH1_SEC2 FASTUPDATE_CH1_SEC3	<int> 0	Run fast update of the H.264 stream

Example) To turn on the buzzer

<pre>http://<server>/httpapi/SetState?action=setoutput&OUT_BUZZER=0</pre>
<pre>HTTP/1.0 200 OK\r\n Content-Type: text/plain\r\n</pre>

```
\r\n
Ok\r\n
```

Example) To run fast update of the primary channel #1

```
http://<server>/httpapi/SetState?action=setoutput&FASTUPDA
TE_CH1_PRI=0
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Ok\r\n
```

2.6 SendPTZ

2.6.1. Send PTZ command (sendptz)

Send PTZ commands to the server.

Parameters	Values	Description
PTZ_CHANNEL	<int> 1	Video channel number. In case of 1 channel video server(TCS-200) or IP camera, it is 1 always. When not set, default channel is assumed (i.e. 1)
PTZ_MOVE	<string> stop continue home irisopen	PTZ operation by PTZ_MOVE stops immediately regardless of timeout value when 'stop' command is sent.

	irisclose	'continue' is used to continue previously set PTZ operation and set the timeout value again.
	<string>, <int> left right up down zoomin zoomout focusnear focusfar	<int> value ranging from -1 to 10 specifies the speed of the operation. When it is not set or it is set to -1, default speed is used.
	<string>,<int>,<int> leftup leftdown rightup rightdown	Two <int> values ranging from -1 to 10 specify pan and tilt values respectively. When it is not set or it is set to -1, default speed is used. Omitting only one value is not allowed.
PTZ_TIMEOUT	<int> 10 ~ 120000	Can be used in combination with PTZ_MOVE. It specifies the timeout of move operation in millisecond unit. While PTZ operation is working by PTZ_MOVE

		command, it stops when either timeout happens or stop command is received. Default value of PTZ_TIMEOUT is 800.
PTZ_LIGHT	<int> 0 or 1	0 : Light on 1 : Light off
PTZ_POWER	<int> 0 or 1	0 : Power on 1 : Power off
PTZ_AUTOPAN	<int> 0 or 1	0 : Autopan stop 1 : Autopan start
PTZ_IRISAUTO	<int> 1	1 : Enable iris auto
PTZ_PATROL	<int> 0 or 1	0 : Patrol stop 1 : Patrol start
PTZ_WIPER	<int> 0 or 1	0 : Wiper stop 1 : Wiper start
PTZ_FOCUSAUTO	<int> 1	1 : Enable focus auto
PTZ_MENUON	<int> 1	1 : Menu enable
PTZ_MENUOFF	<int> 1	1 : Menu disable
PTZ_MENUENTER	<int> 1	1 : Enter key on menu mode
PTZ_MENUESC	<int> 1	1 : ESC key on menu mode
PTZ_PRESETSET	<int> 0 ~	Set preset position with preset number
PTZ_PRESETCLR	<int> 0 ~	Clear preset position with preset number
PTZ_PRESETGOTO	<int> 0 ~	Go to preset position with preset number

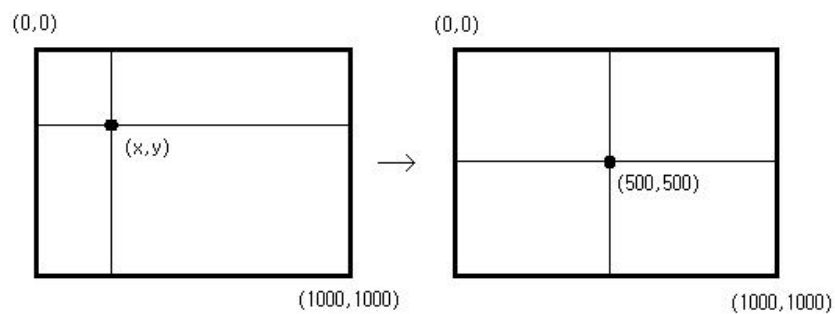
PTZ_ABSOLUTEPOSITION	<int>,<int>,<int>,<int>	Move to absolute position of Pan, Tilt, Zoom and Focus. The range of each value follows the specification of PT device. If some are set to -1, it will maintain current position.
PTZ_RELATIVEPOSITION	<int>,<int>,<int>,<int>	Move to relative position of Pan, Tilt, Zoom and Focus. The range of each value follows the specification of PT device.
PTZ_ABSOLUTEPOSITIONWITHSPEED	<int>,<int>,<int>,<int>,<int>	Move to absolute position of Pan, Tilt, Zoom and Focus with speed. The range of each value follows the specification of PT device. If some are set to -1, it will maintain current position. The speed can be 1 ~ 255.
PTZ_RELATIVEPOSITIONWITHSPEED	<int>,<int>,<int>,<int>,<int>	Move to relative position of Pan, Tilt, Zoom and Focus with speed. The range of each value follows the specification of PT device. The speed can be 1 ~ 255.

PTZ_GETPOSITION	<int> 0	Read current position of Pan, Tilt, Zoom and Focus. -1 comes when failed to read or when not supported by the PT device
PTZ_GETZOOMPOSITION	<int> 0	Read current position of Zoom. -1 comes when failed to read or when not supported by the PT device
PTZ_CENTERING	<int>,<int>	Control pan and tilt to view the clicked position (x, y) on the center of the window.
PTZ_CENTERINGWITHZOOM	<int>,<int>	Same as PTZ_CENTERING except that zoom is also controlled to 2x magnitude.
PTZ_DRAGZOOM	<int>,<int>,<int>,<int>	PTZ is controlled to view the region of (x1, y1, x2, y2) coordinate.
PTZ_POSITIONRESTORE	<int> 0	PTZ position is restored the position before one of the following operation: PTZ_CENTERING, PTZ_CENTERINGWITHZOOM,
PTZ_SETMASK	<int>,<int>,<int>,<int>,<int>	Masknum, x1, y1, x2, y2 See Appendix C

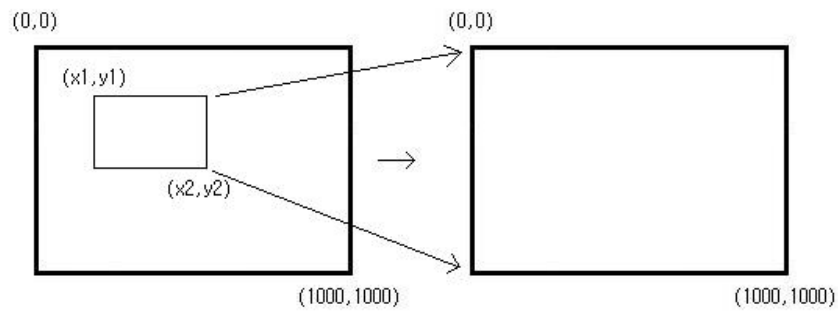
Only PTZ_CHANNEL or PTZ_TIMEOUT can be used in combination with other parameters.

Two PTZ_MOVE can be used in one command only when one is the one of pan&tilt command(left, right, up, down, leftup, leftdown, rightup and rightdown) and the other is zoom command(zoomin and zoomout).

Regardless of the video encoding resolution and zoomed state, coordinate system in PTZ_CENTERING and PTZ_CENTERINGWITHZOOM is (0, 0, 1000, 1000) where top-left corner is (0, 0) and bottom-right corner is (1000, 1000). The following figure shows that (x, y) position which is typically by mouse click operation is move to the center of the screen.



The same coordinate is used in PTZ_DRAGZOOM command: (0, 0, 1000, 1000). Dragged region is zoomed to full size as shown in the following figure.



PTZ_CENTERING, PTZ_CENTERINGWITHZOOM, PTZ_DRAGZOOM, and PTZ_POSITIONRESTORE are supported only by some specific PTZ camera models.

Example) To pan-left the camera for the first video channel for 500ms with speed 4

```

http://<server>/httpapi/SendPTZ?action=sendptz&PTZ_CHANNEL=1&PTZ_MOVE=left,4&PTZ_TIMEOUT=500
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Ok\r\n

```

Example) To pan-right, tilt-up and zoom-in the camera for the first video channel with each speed.

```

http://<server>/httpapi/SendPTZ?action=sendptz&PTZ_CHANNEL=1&PTZ_MOVE=rightup,4,5&PTZ_MOVE=zoomin,3
HTTP/1.0 200 OK\r\n

```

```
Content-Type: text/plain\r\n\r\nOk\r\n
```

Example) To turn on the power of the camera (the first video channel)

```
http://<server>/httpapi/SendPTZ?action=sendptz&PTZ_POWER=1\r\n\r\nHTTP/1.0 200 OK\r\n\r\nContent-Type: text/plain\r\n\r\nOk\r\n
```

Example) To move the camera for the first video channel to preset position 23

```
http://<server>/httpapi/SendPTZ?action=sendptz&PTZ_CHANNEL=1&PTZ_PRESETGOTO=23\r\n\r\nHTTP/1.0 200 OK\r\n\r\nContent-Type: text/plain\r\n\r\nOk\r\n
```

Example) To move the camera to absolute Pan/Tilt position of (1000, 2000).

```
http://<server>/httpapi/SendPTZ?action=sendptz&PTZ_ABSOLUTEPOSITION=1000,2000,-1,-1\r\n\r\nHTTP/1.0 200 OK\r\n\r\nContent-Type: text/plain\r\n\r\nOk\r\n
```

Example) To read the camera position.

```
http://<server>/httpapi/SendPTZ?action=sendptz&PTZ_GETPOSITION=0
```

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
PTZ_GETPOSITION=1000,2000,655,-1\r\n
```

Example) To move (200,300) to center position.

```
http://<server>/httpapi/SendPTZ?action=sendptz&PTZ_CENTERING=200,300
```

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Ok\r\n
```

Example) To move to drag box (100,100)~(200,300).

```
http://<server>/httpapi/SendPTZ?action=sendptz&PTZ_DRAGZOOM=100,100,200,300
```

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Ok\r\n
```

2.7 SerialPort

2.7.1. Serial port command (serialport)

Send or receive serial data through a serial port of a server.

Parameters	Values	Description
COM_OPEN	<int> 1 ~ 300	Notify the server that a serial port will be used. Session timeout value of max 300 sec is specified. When succeeded, the server returns an ID of the session which can be used for succeeding commands. This ID should be used together with COM_PORT always.
COM_CLOSE	<int> id	Close a session which will not be used anymore.
COM_KEEPALIVE	<int> id	Send keep-alive to keep a session of ID id. Session timeout value is reset to the value set by COM_OPEN.
COM_PORT	<int> 1 ~	Specify a serial port to used.
COM_ID	<int> id	Specify the ID of a session to be used for read or write operation.
COM_WRITE	<hexstring>	Hex coded bytes with values of 0~9, a~f, A~F. Writes the specified data string to the selected serial port. Max 256의 hex codes(i.e. 128 bytes) can be

		written at one command.
COM_READ	<int> 1 ~ 128	Read byte stream from selected serial port.
COM_TIMEOUT	<int> 0 ~ 5000	Specify 'read timeout' in millisecond unit which means how long it should wait before return. This parameter is used together with COM_READ only. When set to 0, it returns immediately after reading stored data. Default value is 0

The first step to use a serial port for reading or writing data is to get admission to use it with COM_OPEN. COM_OPEN creates an internal session and returns the ID of the session, which is to be used for read/write operation. The system starts to buffer data received to the corresponding port after the ID(session) is allocated. Up to 1024 bytes can be buffered, and the oldest data are discarded first when overflow happens. The ID is invalidated automatically when the timeout of the session specified with COM_OPEN expires. COM_KEEPLIVE is used to make the ID valid continuously. So, it is necessary to send COM_KEEPLIVE in an interval shorter than the session timeout specified with COM_OPEN.

Example) To allocate ID to use serial port 1

<pre>http://<server>/httpapi/SerialPort?action=serialport&COM_OPEN=60&COM_PORT=1</pre>
<pre>HTTP/1.0 200 OK\r\n</pre>

```
Content-Type: text/plain\r\n\r\nCOM_ID=2\r\n
```

Example) To send keep-alive to use the session of ID 2 continuously

```
http://<server>/httpapi/SerialPort?action=serialport&COM_KEEPAKIVE=2\r\n\r\nHTTP/1.0 200 OK\r\nContent-Type: text/plain\r\n\r\nOk\r\n
```

Example) To close a session of ID 2

```
http://<server>/httpapi/SerialPort?action=serialport&COM_CLOSE=2\r\n\r\nHTTP/1.0 200 OK\r\nContent-Type: text/plain\r\n\r\nOk\r\n
```

Allocated ID is used for reading or writing data through a serial port.

Example) To write 8 bytes data to the session of ID 2.

```
http://<server>/httpapi/SerialPort?action=serialport&COM_ID=2&COM_WRITE=08AF010030D04EEF\r\n\r\nHTTP/1.0 200 OK\r\nContent-Type: text/plain\r\n
```

```
\r\n
Ok\r\n
```

COM_TIMEOUT can be combined with COM_READ. When data become available within the timeout, it returns with the data. Otherwise it returns after the timeout expires.

Example) To write 8 bytes data to the session of ID 2 and read 4 bytes

```
http://<server>/httpapi/SerialPort?action=serialport&COM_ID=2&COM_WRITE=08AF010030D04EEF&COM_READ=4&COM_TIMEOUT=1000
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
COM_READ=AABBCCDD\r\n
```

It is possible to use COM_WRITE without allocating ID with COM_OPEN. That is, writing to serial port can be done without allocating a session. In this case, COM_PORT should be specified instead of COM_ID.

Example) To write 8 bytes data to serial port 1

```
http://<server>/httpapi/SerialPort?action=serialport&COM_PORT=1&COM_WRITE=08AF010030D04EEF
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Ok\r\n
```

2.7.2. Serial port access through TCP channel

Alternative way to access a serial port from client is to use a TCP channel through which bidirectional serial data are sent and received. The TCP port can be configured on **Device->Serial** setup page of web viewer. Any free port can be used, but it is recommended to use 2281. Client application can establish a TCP connection to the port and send and receive any byte streams as if the port is a serial port. Typically PTZ control commands can be sent and responses are received using this function.

COM1 (RS-232 Port)

Protocol: RS-232

Bitrate: 9600 bps

Data Bit: 8 Bits

Parity: None

Stop Bit: 1 Bits

Pass Through TCP Port: 2281 (0=Disable)

2.8 SearchData

2.8.1. Get Root command (getroot)

Retrieve the information(Sub folder list) of root folder in the edge storage(USB stick or SD card attached to the camera or encoder).

Parameters	Values	Description
getpagenum	Don't care	Retrieve the number of pages consisting of sub folders under the root folder.
page	<int> 1 ~	Retrieve the list of sub folders in a page.

The root folder consists of sub folders. Each sub folder contains recorded files

for a specific date. Sub folder has the name in YYYY_MM_DD format (eg: 2012_11_30). Sub folder list is retrieved in page unit. Each page can contain max 50 items. Newer sub folders come earlier in the list. If the number of total sub folders is 80, the list of files is divided into two pages. The first page contains 50 sub folders and the second page contains 30 sub folders.

When only getpagenum is included in the parameter, only the number of total pages is returned and the list itself is not returned.

When page parameter is omitted or set to 0, the first page (page 1) is assumed. For other invalid page value, "No Data" is returned.

Example) To get total page number

<pre>http://<server>/httpapi/SearchData?action=getroot&getpagenum=0</pre>
<pre>HTTP/1.0 200 OK\r\nContent-Type: text/plain\r\n\r\nTotalPage=4\r\n</pre>

Example) To retrieve sub folder list of page 2

<pre>http://<server>/httpapi/SearchData?action=getroot&page=2</pre>
<pre>HTTP/1.0 200 OK\r\nContent-Type: text/plain\r\n\r\n2012_11_11\r\n2012_11_10\r\n</pre>

```

2012_11_09\n
.....
2012_10_17\n
\r\n

```

2.8.2. Get List command (getlist)

Retrieve the list of files under a sub folder.

Parameters	Values	Description
date	<string>	Specify a sub folder name.
getpagenum	Don't care	Retrieve the number of pages of files under a specific sub folder.
page	<int> 1 ~	Retrieve the list of files for a page.

getlist action should contain date parameter which specifies a sub folder. A sub folder contains recorded files. List of files is retrieved in page unit whose max size is 50. Newer files come earlier in the list.

When only getpagenum is included in the parameter, only the number of total pages is returned and the list itself is not returned. When page parameter is omitted or set to 0, the first page (page 1) is assumed. For other invalid page value, "No Data" is returned.

Example) To get the number of total pages

```

http://<server>/httpapi/SearchData?action=getlist&date=201
2_11_30&getpagenum=0
HTTP/1.0 200 OK\r\n

```

```
Content-Type: text/plain\r\n\r\nTotalPage=14\r\n
```

Example) To retrieve file list of page 8

```
http://<server>/httpapi/SearchData?action=getlist&date=2012_11_11&page=8\r\n\r\nHTTP/1.0 200 OK\r\n\r\nContent-Type: text/plain\r\n\r\n\r\n/data/2012_11_11/121111_105900_110000.avi      5.48M\r\n/data/2012_11_11/121111_105800_105900.avi    5.40M\r\n/data/2012_11_11/121111_105700_105800.avi    6.15M\r\n.....\r\n/data/2012_11_11/121111_103800_103900.avi    5.65M\r\n\r\n
```

2.8.3. Delete File command (deletefile)

Delete file(s) in sub folder(s).

Parameters	Values	Description
date	<string>	Specify a sub folder name.
filename	<string>	Specify files under a sub folder

deletefile action is used to delete one or more recorded files. date parameter should be used to specify a sub folder before filename parameter.

One command contains multiple combinations of date parameter and filename

parameter. One date parameter can have multiple succeeding file name parameters.

“.avi” extension in file name parameter can be omitted. A sub folder is deleted automatically when all files under the folder are deleted.

Example) To delete files

```
http://<server>/httpapi/SearchData?action=deletefile&date=
2012_11_30&filename=121111_105900_110000&filename=121111_1
05900_110000&date=2012_11_29&filename=120033_135900_140000
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Ok\r\n
```

2.8.4. Retrieve file

After getting filename list, you can use standard HTTP GET method for getting the file.

If a file is represented in the file list as:

```
/data/2012_11_11/121111_105900_110000.avi
```

Then the URL for getting the file is:

```
http://<server>/data/2012_11_11/121111_105900_110000.avi
```

2.9 DrawOSD (Supported model only)

2.9.1. Draw Text OSD Command (drawtext)

Draw Text OSD.

Parameters	Values	Description
TEXT_STARTX	<int> 0 ~ 1000	Permillage
TEXT_STARTY	<int> 0 ~ 1000	Permillage
TEXT_FONTSIZE	<int> 12 ~ 84	
TEXT_FONTCOLOR	<hex> 0 ~ FFFFFFFF	RGB Hex Value
TEXT_STRING	<string>	Max 64Bytes
TEXT_ALPHA	<int> 0 ~ 255	When not set, default alpha blending value is 255.
TEXT_BORDERLINECOLOR	<hex> 0 ~ FFFFFFFF	RGB Hex Value When not set, No border Line.

Example) Set drawtext1 (TEXT_ALPHA, TEXT_BORDERLINECOLOR default)

```
http://<server>/httpapi/DrawOSD?action=drawtext&TEXT_STARTX=0&TEXT_STARTY=500&TEXT_FONTSIZE=36&TEXT_FONTCOLOR=0000FF&TEXT_STRING=TESTSTRING
HTTP/1.1 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
OK\r\n
```

Example) Set drawtext2

```
http://<server>/httpapi/DrawOSD?action=drawtext&TEXT_STARTX=0&TEXT_STARTY=500&TEXT_FONTSIZE=36&TEXT_FONTCOLOR=0000FF&TEXT_STRING=TESTSTRING&TEXT_ALPHA=255&TEXT_BORDERLINE_COLOR
```

=FFFF00
HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n \r\n OK\r\n

Example) Wrong setting and Error list.

http://<server>/httpapi/DrawOSD?action=drawtext&TEXT_STARTX=0&TEXT_STARTY=500&TEXT_FONTSIZE=36
HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n \r\n ERROR:DRAW_TEXT Need mandatory value.\r\n
Error List - ERROR:TEXT_OOO Out of range. - ERROR:DRAW_TEXT Need mandatory value.

2.9.2. Draw Bitmap OSD Command (drawbitmap)

Draw Bitmap OSD.

Parameters	Values	Description
BITMAP_INDEX	<int> 0 ~ 15	Uploaded Bitmap Index.
BITMAP_STARTX	<int> 0 ~ 1000	Permillage
BITMAP_STARTY	<int> 0 ~ 1000	Permillage
BITMAP_ALPHA	<int> 0 ~ 255	When not set, default alpha blending value is 255.

It supports only 24bit bitmap.

Example) Set drawbitmap1 (BITMAP_ALPHA default)

```
http://<server>/httpapi/DrawOSD?action=drawbitmap&BITMAP_INDEX=0&BITMAP_STARTX=50&BITMAP_STARTY=50
```

```
HTTP/1.1 200 OK\r\nContent-Type: text/plain\r\n\r\nOK\r\n
```

Example) Set drawbitmap2

```
http://<server>/httpapi/DrawOSD?action=drawbitmap&BITMAP_INDEX=0&BITMAP_STARTX=50&BITMAP_STARTY=50&BITMAP_ALPHA=127
```

```
HTTP/1.1 200 OK\r\nContent-Type: text/plain\r\n\r\nOK\r\n
```

Example) Wrong setting and Error list.

```
http://<server>/httpapi/DrawOSD?action=drawbitmap&BITMAP_INDEX=0&BITMAP_STARTX=50&BITMAP_STARTY=50&BITMAP_ALPHA=259
```

```
HTTP/1.1 200 OK\r\nContent-Type: text/plain\r\n\r\nERROR:BITMAP_ALPHA Out of range.\r\n
```

Error List

- ERROR:BITMAP_OOO Out of range.
- ERROR:DRAW_BITMAP No Bitmap.

- ERROR:DRAW_BITMAP Need mandatory value.

2.9.3. Draw Object OSD Command (drawobject)

Draw Object(Line, Circle, Box) OSD.

Parameters	Values	Description
OBJECT_TYPE	<int> 0 or 1 or 2	0 : Line 1 : Circle 2 : Box
OBJECT_STARTX	<int> 0 ~ 1000	Permillage
OBJECT_STARTY	<int> 0 ~ 1000	Permillage
OBJECT_ENDX	<int> 0 ~ 1000	Permillage
OBJECT_ENDY	<int> 0 ~ 1000	Permillage
OBJECT_INNERCOLOR	<hex> 0 ~ FFFFFFFF	RGB Hex Value. When not set, Inner color is transparent.
OBJECT_ALPHA	<int> 0 ~ 255	When not set, default alpha blending value is 255.
OBJECT_BORDERLINECOLOR	<hex> 0 ~ FFFFFFFF	When not set, default borderline color value is FFFFFFF.
OBJECT_THICKSIZE	<int> 0 ~	0 : No Borderline 1 ~ : Borderline thick size.

When OBJECT_TYPE is 0(Line), OBJECT_THICKSIZE cannot be set as 0.

When OBJECT_INNERCOLOR is default, OBJECT_THICKSIZE cannot be set as 0.

Example) Set drawobject(Line)

```
http://<server>/httpapi/DrawOSD?action=drawobject&OBJECT_TYPE=0&OBJECT_STARTX=0&OBJECT_STARTY=0&OBJECT_ENDX=1000&OBJECT_ENDY=1000&OBJECT_INNERCOLOR=0&OBJECT_ALPHA=254&OBJECT_BORDERLINECOLOR=FFFFFF&OBJECT_THICKSIZE=5
```

```
HTTP/1.1 200 OK\r\n  
Content-Type: text/plain\r\n  
\r\n  
OK\r\n
```

Example) Set drawobject(Circle)

```
http://<server>/httpapi/DrawOSD?action=drawobject&OBJECT_TYPE=1&OBJECT_STARTX=100&OBJECT_STARTY=100&OBJECT_ENDX=900&OBJECT_ENDY=300&OBJECT_INNERCOLOR=0&OBJECT_ALPHA=254&OBJECT_BORDERLINECOLOR=FFFFFF&OBJECT_THICKSIZE=5
```

```
HTTP/1.1 200 OK\r\n  
Content-Type: text/plain\r\n  
\r\n  
OK\r\n
```

Example) Set drawobject(Box)

```
http://<server>/httpapi/DrawOSD?action=drawobject&OBJECT_TYPE=2&OBJECT_STARTX=300&OBJECT_STARTY=300&OBJECT_ENDX=500&OBJECT_ENDY=500&OBJECT_INNERCOLOR=0&OBJECT_ALPHA=254&OBJECT_BORDERLINECOLOR=FFFF&OBJECT_THICKSIZE=5
```

```
HTTP/1.1 200 OK\r\n  
Content-Type: text/plain\r\n
```

```
\r\n
OK\r\n
```

Example) Wrong setting and Error list.

```
http://<server>/httpapi/DrawOSD?action=drawobject&OBJECT_TYPE=0&OBJECT_STARTX=0&OBJECT_STARTY=0&OBJECT_ENDX=1000&OBJECT_ENDY=1000&OBJECT_ALPHA=254&OBJECT_BORDERLINECOLOR=FFFFFF&OBJECT_THICKSIZE=0
HTTP/1.1 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
ERROR:DRAW_OBJECT When ObjectType is 0, thickSize cannot be set as 0.\r\n
Error List
- ERROR:OBJECT_OOO Out of range.
- ERROR:DRAW_OBJECT When ObjectType is 0, thickSize cannot be set as 0.
- ERROR:DRAW_OBJECT When InnerColor is default, thicksize cannot be set as 0 .
- ERROR:DRAW_OBJECT Need mandatory value.
- ERROR:DRAW_OBJECT Draw fail.
```

2.9.4. Erase Area Command (erase)

Erase OSD.

Parameters	Values	Description
ERASE_AREA	<int>,<int>,<int>,<int>	Start X Position (0 ~ 1000)

		Start Y Position (0 ~ 1000) Width (0 ~ 1000) Height (0 ~ 1000)
--	--	--

Example) Erase all area.

http://<server>/httpapi/DrawOSD?action=erase&ERASE_AREA=0,0,1000,1000
HTTP/1.1 200 OK\r\nContent-Type: text/plain\r\n\r\nOk\r\n

Example) Erase specific area.(Start Position(300,300), End Position(500, 500))

http://<server>/httpapi/DrawOSD?action=erase&ERASE_AREA=300,300,200,200
HTTP/1.1 200 OK\r\nContent-Type: text/plain\r\n\r\nOk\r\n

Example) Wrong setting and Error list.

http://<server>/httpapi/DrawOSD?action=erase&ERASE_AREA=300,300,200,1200
HTTP/1.1 200 OK\r\nContent-Type: text/plain\r\n\r\nERROR:ERASE_AREA Out of range\r\n

Error List

- ERROR:ERASE_AREA Out of range.
- ERROR:ERASE_AREA Need mandatory value.

2.9.5. OSD Image Command (osdimage)

Get / Set Current OSD Image.

Parameters	Values	Description
OSDIMAGE_GET	<int>,<int>,<int>,<int>	Start X Position (0 ~ 1000) Start Y Position (0 ~ 1000) Width (1 ~ 1000) Height (1 ~ 1000)
OSDIMAGE_SET	<int>,<int>	Start X Position (0 ~ 1000) Start Y Position (0 ~ 1000)

OSDIMAGE_GET function saves OSD Image data of setting area to memory.

OSDIMAGE_SET function draws OSD Image data which is saved on the memory in the screen.

OSD Image data saved on memory maintains until camera restarts and It is renewed if OSDIMAGE_GET function is newly called.

Example) Get OSD Image

http://<server>/httpapi/DrawOSD?action=osdimage&OSDIMAGE_GET=0,0,500,500
HTTP/1.1 200 OK\r\nContent-Type: text/plain\r\n\r\n

Ok\r\n

Example) Set OSD Image

http://<server>/httpapi/DrawOSD?action=osdimage&OSDIMAGE_SET=300,300,0,0,0

HTTP/1.1 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Ok\r\n

Example) Wrong setting and Error list.

http://<server>/httpapi/DrawOSD?action=osdimage&OSDIMAGE_SET=300,3100

HTTP/1.1 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
ERROR:OSDIMAGE_SET Out of range\r\n

Error List

- ERROR:OSDIMAGE_GET Out of range.
- ERROR:OSDIMAGE_SET Out of range.
- ERROR:OSDIMAGE_GET Need mandatory value.
- ERROR:OSDIMAGE_SET No OSDImage
- ERROR:OSDIMAGE_SET Need mandatory value.

2.9.6. Update OSD Command (update)

Update OSD.

Parameters	Values	Description
UPDATE_OSD	<int>	Don't care

It draws OSD which is set up by drawtext, drawbitmap and drawobject on the screen.

Example) Update OSD.

http://<server>/httpapi/DrawOSD?action=update&UPDATE_OSD=0
HTTP/1.1 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
OK\r\n

2.9.7. Delete Bitmap (delete)

Delete uploaded bitmap file.

Parameters	Values	Description
DELETE_BITMAP	<int>	Uploaded Bitmap Index (0 ~ 15)

Example) Delete bitmap file

http://<server>/httpapi/DrawOSD?action=delete&DELETE_BITMAP=1
HTTP/1.1 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
OK\r\n

Example) Wrong delete setting

```
http://<server>/httpapi/DrawOSD?action=delete&DELETE_BITMAP=17
```

```
HTTP/1.1 200 OK\r\n
```

```
Content-Type: text/plain\r\n
```

```
\r\n
```

```
ERROR:DELETE_BITMAP Wrong bitmap index\r\n
```

2.9.8. Upload Bitmap (HTTP POST)

It supports only 24bit bitmap.

Bitmap can be uploaded by sending below form-data(multipart/form-data) to

<http://<server>/uploadfile.html>

```
<form>
```

```
  <input type="file" name="DrawOSDBitmap" id="DrawOSDBitmap"/>
```

```
  <input type="text" name="DrawOSDBitmapIndex" id="DrawOSDBitmapIndex"/>
```

```
</form>
```

DrawOSDBitmapIndex range : 0 ~ 15.

Full Capacity of bitmap can be uploaded(already uploaded and will upload bitmap) is 300Kbytes.

Example) Uploading bitmap by using "Curl".

```
curl -F "DrawOSDBitmap=@<bitmap_file_path_and_name>" -F "DrawOSDBitmapIndex=<OSDIndex>"
```

```
http://<server>/uploadfile.html
```

2.10 SendAudio

Audio stream from client to server can be delivered through a TCP channel initiated by the client. The listen port of the server is configured using AUD_RECVPORT. Default value of this port is 2280. After the channel is established, the client can send audio data without any header information through the channel. That is, pure G.711 u-law data are to be streamed to the server.

2.11 UploadFile

2.11.1. Upload Upgrade file (HTTP POST)

Upgrade file can be uploaded by sending below form-data(multipart/form-data) to <http://<server>/progress.html>

```
<form>
  <input name="upgrade" type="file"/>
</form>
```

If uploading file is completed, firmware upgrade process will start.

Example) Upload Upgrade file by using "Curl".

```
curl -X POST -H "Expect:" -H "Cache-Control: no-cache" -H "Content-Type: multipart/form-data;" -F
"upgrade=@<upgrade_file_path_and_name>" http://<server>/progress.html
```

2.11.2. Upload Config file (HTTP POST)

Config file can be uploaded by sending below form-data(multipart/form-data) to <http://<server>/restore.html>

```
<form>
  <input name="config" type="file"/>
```

```
</form>
```

If uploading file is completed, config restore process will start.

Example) Upload Config file by using "Curl".

```
curl -X POST -H "Expect:" -H "Cache-Control: no-cache" -H "Content-Type: multipart/form-data;" -F  
"config=@<config_file_path_and_name>" http://<server>/restore.html
```

2.11.3. Upload IVS License file (HTTP POST)

IVS License file can be uploaded by sending below form-data(multipart/form-data) to http://<server>/upload_ivsLic.html

```
<form>  
  <input name="ivsLic" type="file"/>  
</form>
```

If uploading file is completed, new IVS license will be applied.

Example) Upload IVS License file by using "Curl".

```
curl -X POST -H "Expect:" -H "Cache-Control: no-cache" -H "Content-Type: multipart/form-data;" -F  
"ivsLic=@<IVS_License_file_path_and_name>" http://<server>/upload_ivsLic.html
```

2.11.4. Upload user-defined Bitmap OSD file (HTTP POST)

Bitmap OSD file can be uploaded by sending below form-data(multipart/form-data) to <http://<server>/uploadfile.html>

```
<form>  
  <input name="bitmaposd" type="file"/>  
</form>
```

If uploading file is completed, new bitmap file will be applied.

Example) Upload Bitmap OSD file by using "Curl".

```
curl -X POST -H "Expect:" -H "Cache-Control: no-cache" -H "Content-Type: multipart/form-data;" -F  
"bitmaposd=@<Bitmap_OSD_file_path_and_name>" http://<server>/uploadfile.html
```

3. Capture Groups

(Dual streaming models support only).

3.1 Capturing One JPEG Still Image

JPEG images can be received using HTTP API.

Capture command should have the following syntax.

```
http://<server>/capture/ch<ch_num>.jpg[?quality=<value>]
```

This JPEG image is created with a resolution of primary encoding whenever it is requested.

The range of quality is 10 to 100, and regarded 80 if omitted.

100 is the best.

Example) To get one jpeg image of channel #3 and quality is 80.

```
http://<server>/capture/ch3.jpg?quality=80
```

```
HTTP/1.0 200 OK\r\n
Content-Type: image/jpeg\r\n
Content-Length: 20482\r\n
\r\n
<JPEG Image Data>\r\n
```

3.2 Receiving Sequential JPEG Stream

It is possible to receive series of JPEG images from the secondary encoding module through an HTTP connection. The video algorithm of the secondary encoding module should be set to MJPEG.

The syntax is very similar to that for getting a still image.

```
Secondary 1:  
http://<server>/capture/ch<ch_num>.mjpg  
  
Secondary 2:  
http://<server>/capture/ch<ch\_num>\_2.mjpg  
  
Secondary 3:  
http://<server>/capture/ch<ch\_num>\_3.mjpg
```

In this case user can not specify the quality value. The resolution and quality of the JPEG images follow the setting of secondary video encoding. JPEG streaming is terminated when the HTTP channel is closed.

Example) To get jpeg stream of channel #2.

```
http://<server>/capture/ch2.mjpg  
HTTP/1.0 200 OK\r\n  
Content-Type: multipart/x-mixed-replace;boundary=framebou  
ndary\r\n  
\r\n  
--frameboundary\r\n  
Content-Type: image/jpeg\r\n  
Content-Length: 16532\r\n  
\r\n  
<Jpeg Image Data>\r\n  
--frameboundary\r\n  
Content-Type: image/jpeg\r\n
```

```
Content-Length: 17236\r\n
```

```
\r\n
```

```
<JPEG Image Data>\r\n
```

```
--frameboundary\r\n
```

```
.
```

```
.
```

```
.
```

Appendix A. PTZ Coordinate Range

A.1 IP zoom/PTZ camera

- ◆ Pan: 0 ~ 35999 (0 ~ 360°)
- ◆ Tilt: 0 ~ 9000 (0 ~ 90°)
- ◆ Zoom: (follows Pelco-D recommendation)
 - The position is given as a ratio based on the device's Zoom Limit setting.
The position is calculated as follows:
Position = (desired_zoom_position / zoom_limit) * 65535
Where desired_zoom_position and zoom_limit are given in units of magnification.
 - Example: Given that the zoom limit of the device's camera is X18, calculate the value needed to set the zoom position to X5:
Position = (5 / 18) * 65535 = approximately 18204
 - Zoom limit: optical zoom x digital zoom
Digital zoom => Sony: x12, Samsung: x12, Panasonic: x20, Hitachi: N/A
- ◆ Speed: 1 ~ 10 (10: the fastest)
 - Range of zoom control speed may depend on the zoom module used. For example, SONY zoom module support the range of 8 values. In such case, range 1 ~ 10 is mapped to actually supported range.

Appendix B. Continuous PTZ operation

Frequently ask questions

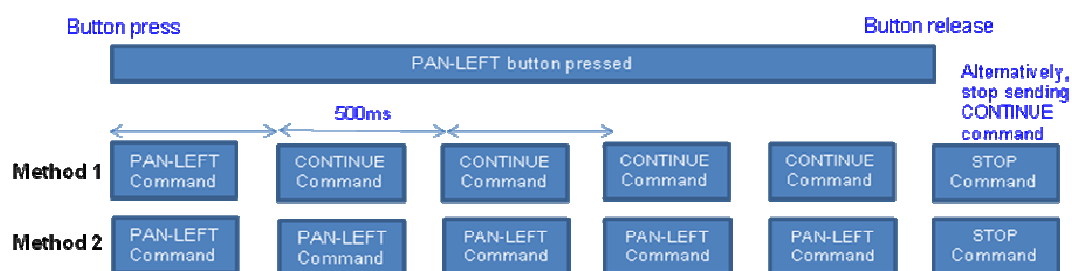
- Why pan(tilt or zoom) command starts panning but it stops very soon after a short move?
- How to achieve continues move while a button for pan/tilt/zoom is pressed?

Reason and solution

PTZ system is designed to work in the following way:

- If a command is sent once, the operation stops automatically in about 800ms (default timeout) even without STOP command. This is to protect the camera from the situation of missing STOP command.

In order to achieve continuous move, it is necessary to send CONTINUE command(or original PTZ command) periodically in 500ms interval (in fact, in the interval less than configured timeout value).



Appendix C. Privacy Mask Setting

C.1 Finding mask drawing type from camera

Antrica cameras have several types of the privacy mask. The mask drawing style can be found using SYS_MODULE_TYPE and SYS_PTZ_TYPE.

SYS_PTZ_TYPE	SYS_MODULE_TYPE			
	10, 12, 16	11	1, 2, 3, 4, 6, 9, 13, 15	5
0	Type1	Type1	X	Type4
1	Type2	X	Type2	X
7	X	X	Type3	X

Example) To read SYS_MODULE_TYPE and SYS_PTZ_TYPE

```
http://<server>/httpapi/ReadParam?action=readparam&SYS_MODULE_TYPE=0&SYS_PTZ_TYPE=0
```

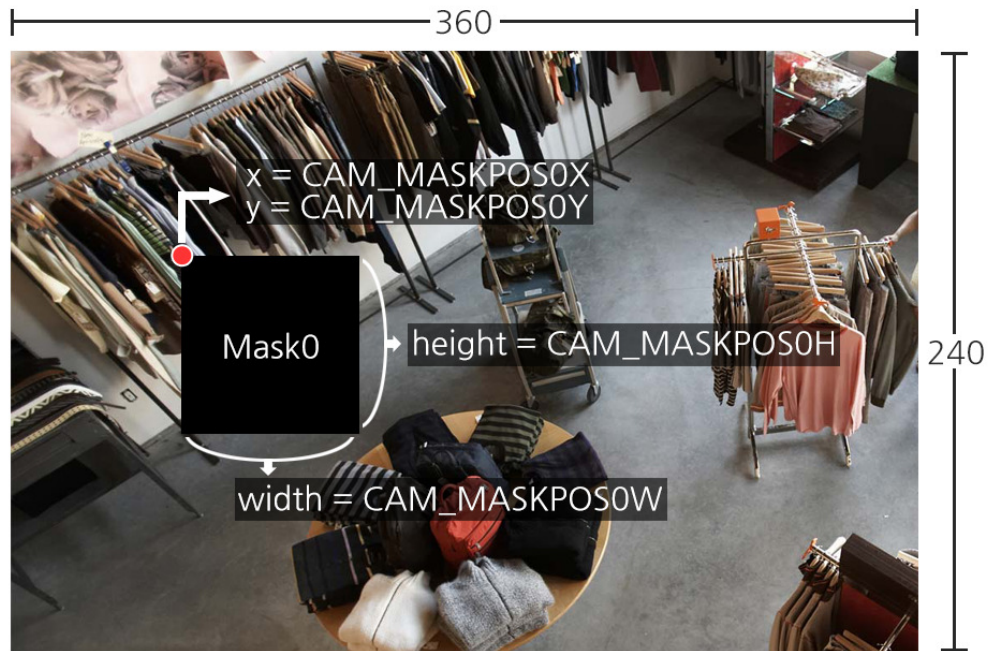
```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
SYS_MODULE_TYPE=12\r\n
SYS_PTZ_TYPE=1\r\n
```

C.2 Type1

- Num of Mask : 4
- Using WriteParam API, set CAM_MASKPOSnX, CAM_MASKPOSnY, CAM_MASKPOSnW and CAM_MASKPOSnH. 'n' can be 0 ~ 3 for each

mask number.

- CAM_MASKPOSnX and CAM_MASKPOSnW can be 0~360.
- CAM_MASKPOSnX + CAM_MASKPOSnW <= 360
- CAM_MASKPOSnY and CAM_MASKPOSnH can be 0~240.
- CAM_MASKPOSnY + CAM_MASKPOSnH <= 240
- CAM_MASKPOSnW or CAM_MASKPOSnH set to 0, the mask will be erased.



Example) To draw mask0 at (100, 100) with size (200, 50)

```
http://<server>/httpapi/WriteParam?action=writeparam&CAM_M  
ASKPOS0X=100&CAM_MASKPOS0Y=100&CAM_MASKPOS0W=200&CAM_MASKP  
OS0H=50
```









```

HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
OK\r\n

```

C.3 Type2

- Num of Mask : 4
- Using SendPTZ API, set mask areas interactively.
- Each operations are defined PTZ_COMMAND=value.

	Mask-1	Mask-2	Mask-3	Mask-4
New	160	161	162	163
Display on/off	176	177	178	179
Display all off	184			
	196	197	198	199
	204	205	206	207
	212	213	214	215
	220	221	222	223
	192	193	194	195
	200	201	202	203
	208	209	210	211
	216	217	218	219

Example) To create mask-1 at center position

```

http://<server>/httpapi/SendPTZ?action=sendptz&PTZ_CHANNEL
=1&PTZ_COMMAND=160

```

```

HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
OK\r\n

```

Example) To move left of mask-2

```





http://<server>/httpapi/SendPTZ?action=sendptz&PTZ_CHANNEL
=1&PTZ_COMMAND=213
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
OK\r\n

```

C.4 Type3

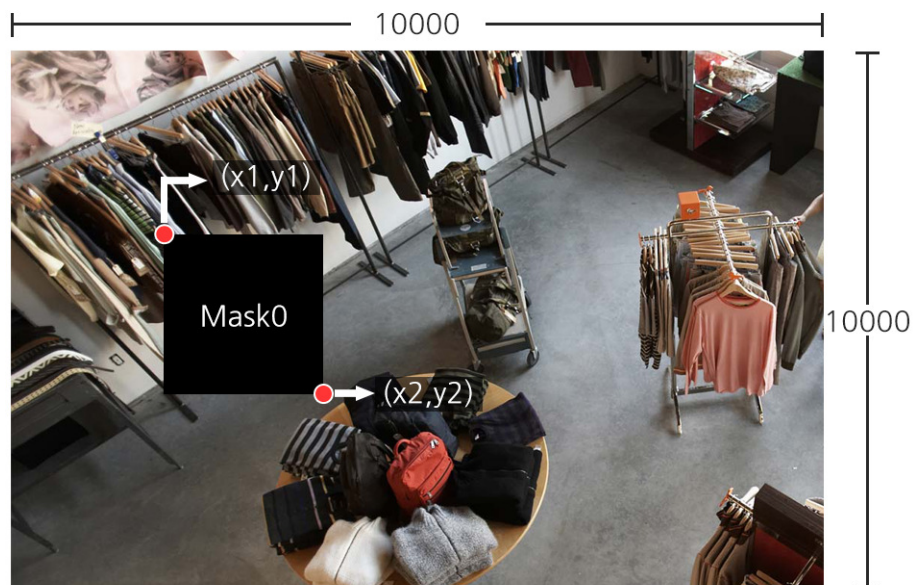
- Num of Mask : 8
- Using SendPTZ API, set mask areas interactively.
- Each operations are defined PTZ_COMMAND=value.

	Mask-1	Mask-2	Mask-3	Mask-4	Mask-5	Mask-6	Mask-7	Mask-8
New	160	161	162	163	164	165	166	167
Search	168	169	170	171	172	173	174	175
Display on/off	176	177	178	179	180	181	182	183
Display all off	184							

	192	193	194	195	196	197	198	199
	200	201	202	203	204	205	206	207
	208	209	210	211	212	213	214	215
	216	217	218	219	220	221	222	223

C.5 Type4

- Num of Mask : 8
- Using SendPTZ API, set mask areas interactively.
- When drawing mask, used PTZ_SETMASK=masknum,x1,y1,x2,y2
 - masknum : 0 ~ 7 (Mask-1 ~ Mask-8)
 - x1, y1, x2, y2 : 0 ~ 10000
 - x1 < x2, y1 < y2



- Additional operations are defined PTZ_COMMAND=value.

	Mask-1	Mask-2	Mask-3	Mask-4	Mask-5	Mask-6	Mask-7	Mask-8
Color	160	161	162	163	164	165	166	167
Transparency	184	185	186	187	188	189	190	191
Display Mode	176	177	178	179	180	181	182	183

- Color changed (1 of 16 colors can be selected)
- Transparency changed (25% -> 50% -> 100%)
- Display Mode changed (Inversion->Mosaic->Color->off)

Example) To draw mask-1 with a width and height of 200 at the center position

```

http://<server>/httpapi/SendPTZ?action=sendptz&PTZ_CHANNEL
=1&PTZ_SETMASK=0,400,400,600,600
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
OK\r\n

```