

CGI Command for Network Camera

Ver.1.0.5
Aug.13 2008

This document is intended as a guide for application developers and describes how to use scripting in CGI Command of Network Camera. The information is provided “as is” without warranty of any kind and is subject to change without notice. And also reserves the right to revise the content of this document at any time without prior notice.

REVISION HISTORY

Date	Release	Author	Description
Aug.24 2007	1.0.0	KW Lee	<ul style="list-style-type: none"> • Firstly Release
Aug.30.2007	1.0.1	KW Lee	<ul style="list-style-type: none"> • Remove unused cgi
Dec.14.2007	1.0.2	Jack Lou	<ul style="list-style-type: none"> • Add [System], [Image.MPEG], [Image.MJPEG], [Audio] to param.cgi • Modified param.cgi [ImageSource.IO.Sensor], [Network.Wireless], [Network.DNSUpdate] for param.cgi
Jan.14.2008	1.0.3	Jack Lou	<ul style="list-style-type: none"> • Add ptctrl.cgi • Add [3GPP], [FTP], [NetStorage], [External.Storage] ,[GPIO], [PanTilt] to param.cgi
Arl.21.2008	1.0.4	Jack Lou	<ul style="list-style-type: none"> • Errata: correct examples in ipfilter.cgi and ptctrl.cgi
Aug.13.2008	1.0.5	Jason Chiu	<ul style="list-style-type: none"> • Add standard server push CGI spec
			<ul style="list-style-type: none"> •

This CGI Command document specifies the method of communication with the Network Camera for controlling camera functions as well as for getting and setting internal parameter values, which helps the application integrators develop software applications more easily.

1. Add, update, remove and list parameters and their values

Method: GET/POST

Syntax:

[http://<servername>/cgi/<group>/param.cgi?action=<value>&group=<value>\[¶meter=<value>\[¶meter=<value>...\]\]](http://<servername>/cgi/<group>/param.cgi?action=<value>&group=<value>[¶meter=<value>[¶meter=<value>...]])

with the following parameters and values

for the full parameter list please refer to Appendix A

<parameter>=<value>	Value	Description
action=<string>	update , list , alllist , add, remove	specifies the action to take. Depending on this parameter, various parameters may be set, as described in the following sections. Add and remove: Only applicable for dynamic parameter groups such as the event parameters.
group=<string>	<string>	Specifies the group.
name=<string>	<string>	Specifies the parameter name.
<parameter name>=<string>	<string>	Specifies the parameter value.

Example:

(1) **update :** Update a parameter

<http://myserver/cgi/admin/param.cgi?action=update&group=Network&IPAddress=192.168.0.10>

return:

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

	Content-Length: nnnn\r\n \r\n Network.IPAddress=192.168.0.10\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: nnnn\r\n \r\n Request failed: <error message>\r\n

(2) **list** : List a parameter

<http://myserver/cgi/admin/param.cgi?action=list&group=Network&name=IPAddress>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: nnnn\r\n \r\n Network.IPAddress=192.168.0.10\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: nnnn\r\n \r\n Request failed: <error message>\r\n

(3) **List** : List a group parameters

<http://myserver/cgi/admin/param.cgi?action=list&group=Network>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n [Network] Network.IPAddress=192.168.0.10\r\n Network.MACAddress=192.168.0.10\r\n\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n

	\r\n Request failed: <error message>\r\n
--	---

(4) **alllist : List all parameter**

<http://myserver/cgi/admin/param.cgi?action=alllist>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n [Network] Network.IPAddress=192.168.0.10 Network.MACAddress=192.168.0.10 [System date] ...\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

2. Add, modify and delete users

Add a new user with password and group membership, modify the information and remove a user.

Note: This request requires admin access (admin authorization).

Method: GET/POST

Syntax:

[http://<servername>/cgi/admin/pwdgrp.cgi?<parameter>=<value>\[&<parameter>=<value>...\]](http://<servername>/cgi/admin/pwdgrp.cgi?<parameter>=<value>[&<parameter>=<value>...])

with the following parameters and values

<parameter>=<value>	Value	Description
action=<string>	add, update, remove , get	add = create a new user account. update = change account information of specified parameters if the account exists. remove = remove an existing account if it

		exists. get = get a list of the users which belong to each group defined.
user=<string>	<string>	The user account name.
pwd=<string>	<string>	The unencrypted password of the account.
grp=<string>	administrator, operator, viewer.	An existing primary group name of the account..
<parameter name>=<string>	<string>	An existing primary group name of the account.

Example:

2.1 Create a new administrator account.

<http://myserver/cgi/admin/pwdgrp.cgi?action=add&user=paul&pwd=foo&grp=admin>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n OK\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

2.2 List groups and users.

<http://myserver/cgi/admin/pwdgrp.cgi?action=get>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n administrator="root,stanley,..." operator="jack,brian,..." viewer="angus,becky,..."\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

3. Factory default

3.1.Factory default

Reload factory default. All parameters except BootProto, IPAddress, SubnetMask, Broadcast and Router are set to their factory default values.

Note: This requires administrator access (administrator authorization).

Method: GET

Syntax:

<http://<servername>/cgi/admin/factorydefault.cgi>

Example:.

<http://myserver/cgi/admin/factorydefault.cgi>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n OK\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

3.2.Hard factory default

Reload factory default. All parameters are set to their factory default value.

Note: This requires administrator access (administrator authorization).

Method: GET

Syntax:

<http://<servername>/cgi/admin/hardfactorydefault.cgi>

Example:

<http://<servername>/cgi/admin/hardfactorydefault.cgi>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n OK\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

4. Firmware upgrade

Upgrade the firmware version.

Note: This requires administrator access (administrator authorization).

Method: POST

Syntax: <http://<servername>/cgi/admin/firmwareupgrade.cgi?type=<value>>

<parameter>=<value>	Value	Description
type=<string>	normal,factorydefault,overwrite	normal = Upgrade and restore old settings. factorydefault = Upgrade and discard all settings. overwrite = ignore the modelname checking(only can be used in manufacture) default = normal

Example:


```

POST /cgi/admin/firmwareupgrade.cgi?type=normal HTTP/1.1\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="firmware.bin"; filename="firmware.bin"\r\n
Content-Type: application/octet-stream\r\n
\r\n
<firmware file content>
--AsCg5y\r\n

```

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n OK\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

5. Restart server

Restart server.

Note: This requires administrator access (administrator authorization).

Method: GET

Syntax:

<http://<servername>/cgi/admin/restart.cgi>

Example:

<http://<servername>/cgi/admin/restart.cgi>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n OK\r\n
---------	---

failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n
--------	--

6. Server report

This CGI request generates and returns a server report. This report is useful as an input when requesting support.

The report includes product information, parameter settings and system logs.

Note: This requires administrator access (administrator authorization).

Method: GET

Syntax:

<http://<servername>/cgi/admin/serverreport.cgi>

Example:

<http://<servername>/cgi/admin/serverreport.cgi>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Server Name:XXXXXX IP Address:XXXXXX MAC Address:XXXXXX Model Number:XXXXXX Product Number:XXXXXX H/W Version:XXXXXX F/W Version:XXXXXX ROM Version:XXXXXX PT Support:Yes IO Trigger Support:Yes\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

7. System logs

Get system log information.

Note: This requires administrator access (administrator authorization).

Note: The response is product/release-dependent.

Method: GET

Syntax:

<http://<servername>/cgi/admin/systemlog.cgi>

Example:

<http://<servername>/cgi/admin/systemlog.cgi>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n 2007/1/25 11:43:30 Power ON. 2007/1/25 12:00:40 192.168.3.103 johnson login. \r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

8. System date and time

Get or set the system date and time.

Method: GET

Syntax:

[http://<servername>/cgi/admin/date.cgi?action=<value>\[&<parameter>=<value>...\]](http://<servername>/cgi/admin/date.cgi?action=<value>[&<parameter>=<value>...])

with the following parameters and values

<parameter>=<value>	Value	Description
action=<string>	get,set	get = get the current date and time. set = set the current date and time.
year=<string>	<string>	Current year.
month=<string>	<string>	Current month.
day=<string>	<string>	Current day.

hour=<string>	<string>	Current hour.
minute=<string>	<string>	Current minute.
second=<string>	<string>	Current second.
timezone=<string>	0-62 (Appendix B)	<p>Specifies the time zone that the new date and/or time is given in. The camera translates the time into local time using whichever time zone has been specified through the web configuration. If omitted the new date and/or time is assumed to be in local time.</p> <p>Note: Requires that daylight saving time (DST) is turned off, and that the time mode of the camera is not to synchronize with an NTP server or with the computer time.</p> <p>Currently only GMT is considered valid input. The rest of the time zones are subject to future expansion.</p>

Example:

1. Get the date.

<http://myserver/cgi/admin/date.cgi?action=get>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n 2007/1/20 15:20:25\r\n
---------	---

failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n
--------	--

b.Set the date.

<http://myserver/cgi/admin/date.cgi?action=set&year=2005&month=4&day=3>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n ok\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

9.Snapshot

Request a jpeg image.

Method: GET

Syntax:

<http://<servername>/cgi/jpg/image.cgi>

Example:

<http://myserver/cgi/jpg/image.cgi>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n <image data>\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n

	Request failed: <error message>\r\n
--	-------------------------------------

10.MJpeg

Returns a multipart image stream with the default resolution and compression as defined in the system configuration.

Method: GET

Syntax: <http://<servername>/cgi/mjpg/mjpeg.cgi>

Example:

<http://myserver/cgi/mjpg/mjpeg.cgi>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: multipart/ mixed;boundary=myboundary\r\n \r\n --myboundary\r\n Content-Length: <content length>\r\n \r\n <Payload Header> <Payload Data> --myboundary\r\n Content-Length: <content length>\r\n \r\n <Payload Header> <Payload Data>
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

10.1 Payload Header Format

Parameter	Size(Byte)	Description
Counter	4	payload counter .
Size	4	payload size .

Type	4	'J' for JPG, 'A' for audio, 0 for I frame, 1-30 for p frame .
Second	4	Time stamp
Microsecond	4	Time stamp
Format	4	<p>for audio</p> <p>Bit0-3 type</p> <p>2: PCM</p> <p>3: uLaw</p> <p>4: AMR</p> <p>Bit4-5 sample size</p> <p>0: 8bits</p> <p>1: 16bits</p> <p>Bit6-8 sample rate</p> <p>0: 8K</p> <p>1: 11K</p> <p>2: 24K</p> <p>3: 44K</p> <p>4: 48K</p> <p>Bit9-10 channel</p> <p>1: 1 channel</p> <p>2: 2 channels</p> <p>Bit11-31 packet length in minisecs.</p> <p>For Video:</p> <p>Bit0-3 type</p> <p>0:JPEG</p> <p>1:MPEG4</p> <p>Bit4-7 scale</p> <p>0: VGA</p> <p>1:QVGA</p> <p>2:QQVGA</p> <p>Bit8-13 frame rate (1-30)</p> <p>Bit14-21 IP ratio (1-30)</p> <p>Bit22-25 quality level (0 - 4)</p> <p>Bit26-31 reserved</p>
Status	4	Bits 0 – 7: window1 motion level (0 – 255)

		Bits 8 – 15: window2 motion level (0 – 255) Bits 16 – 23: window3 motion level (0 – 255) Bit 24: trigger input 1 status (1:high/0:low) Bit 25: trigger input 2 status (1:high/0:low) Bit 26: trigger output 1 status (1:high/0:low) Bit 27: trigger input 2 status (1:high/0:low) Bits 28 – 31: reserved
--	--	--

11.MPEG4

Returns a multipart MPEG4 image stream with the default resolution and compression as defined in the system configuration

Method: GET

Syntax: <http://<servername>/cgi/mpeg4/mpeg4.cgi>

Example:

<http://myserver/cgi/mpeg4/mpeg4.cgi>

return: The same as MJPEG except payload data is MPEG4

12.Motion Detection

Method: GET/POST

Syntax: [http://<servername>/cgi/motion/md.cgi?action=<value>?<parameter>=<value> \[&<parameter>=<value>...\]](http://<servername>/cgi/motion/md.cgi?action=<value>?<parameter>=<value> [&<parameter>=<value>...])

parameter	Value	Description
action=<String>	add,remove,update,list,listall	add = create a new motion detect window. update = change window information of

		<p>specified parameters if the window exists.</p> <p>remove = remove an existing motion detect window</p> <p>list = list a window information.</p> <p>listall = list all window information</p>
id	0-1	The id of motion detect window.
enable	0/1	Whether to enable this motion detect window
name=<String>	any string <16 charecters	The name of motion detect window.
top=<int>	0-479	The top value of motion detect window..
bottom=<int>	0-479	The bottom value of motion detect window.
left=<int>	0-639	The left value of motion detect window.
right=<int>	0-639	The right value of motion detect window.
sensitivity=<int>	0-255	The sensitivity level of motion detect window.

Example:

1. Add a Motion Detection window:

<http://myserver/cgi/motion/md.cgi?action=add&id=0>

2. Add a new Motion Detection window with specified values:

<http://myserver/cgi/motion/operator/param.cgi?action=add&id=0&enable=1>

[&name=Motion2&top=50&bottom=70&left=50&right=85](http://myserver/cgi/motion/operator/param.cgi?action=add&id=0&enable=1&name=Motion2&top=50&bottom=70&left=50&right=85)

3. Remove a Motion Detection window

<http://myserver/cgi/motion/md.cgi?action=remove&name=Motion1>

4. Update the Motion Detection parameters

<http://myserver/cgi/motion/md.cgi?action=update&id=0&enable=1&name=Motion2>

[&top=150&bottom=200](http://myserver/cgi/motion/md.cgi?action=update&id=0&enable=1&name=Motion2&top=150&bottom=200)

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n OK\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

5. List one motion detection window information

<http://myserver/cgi/motion/md.cgi?action=list&id=1>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n id=1 enable=1 name=motion1 top=0 bottom=100 left=0 bottom=100 sensitivity=100\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

6. List all Motion Detection windows information

<http://myserver/cgi/motion/md.cgi?action=listall>

return:

succeed	HTTP/1.1 200 OK\r\n
---------	---------------------

	Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n id=0 enable=0 name=motion1 top=0 bottom=100 left=0 right=100 sensitivity=100 id=1\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

13.Audio

Request an audio stream.

13.1 Audio query

Query if the audio is available

Method: GET

Syntax: <http://<servername>/cgi/query/query.cgi?<parameter>>

Parameter	Value	Description
Listen	0-2	Query the listening status (Server to PC) 0: free to connect 1: microphone not available 2: device is disable
talk	0-2	Query the PC-talk status (PC to server) 0: free to connect 1: speaker in use , not available 2: device is disable

Example: GET <http://myserver/cgi/query/query.cgi?listen>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n listen=1\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

Example2: GET <http://myserver/cgi/query/query.cgi?talking>

return:

succeed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n talking=2\r\n
failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

13.2 Audio (Camera to PC)

Returns a multipart audio stream

Method: GET

Syntax: <http://<servername>/cgi/audio/audio.cgi?type=<value>>

parameter	Value	Description
Type	PCM	Request linear PCM audio

Example: <http://myserver/cgi/audio/audio.cgi?type=PCM>

return:

succeed	HTTP/1.1 200 OK\r\n\r\nContent-Type: multipart/ mixed;boundary=myboundary\r\n\r\n--myboundary\r\n\r\nContent-Length: <content length>\r\n\r\n<Payload Header>\r\n<Payload Data>\r\n--myboundary\r\n\r\nContent-Length: <content length>\r\n\r\n<Payload Header>\r\n<Payload Data>\r\n....
failed	HTTP/1.1 503 Service Not Available\r\n\r\nContent-Length: 0\r\n\r\n

13.3 Audio (PC to Camera)

Send multipart audio stream to camera (PC talk)

Method: POST

Syntax: <http://<servername>/cgi/audio/audio.cgi?type=PCM>

parameter	Value	Description
Type	PCM	Send linear PCM audio to camera

Example: POST <http://myserver/cgi/audio/audio.cgi?type=PCM>

Request Format:

Content-Type: multipart/ mixed;boundary=myboundary\r\n\r\n--myboundary\r\n\r\nContent-Length: <content length>\r\n\r\n<Payload Header>
--

```

<Payload Data>

--myboundary\r\n

Content-Length: <content length>\r\n

\r\n
<Payload Header>
<Payload Data>

....

```

Return:

Succeed	(None: camera will continue to receive requests until PC disconnect)
Failed	HTTP/1.1 403 Forbidden\r\n Content-Type: text/plain\r\n Content-Length: 0\r\n \r\n

14. IP filter

The requests specified in the IP filter section are supported by products that support IP address filtering.

Method: Get/Post

Syntax:

[http://<servername>/cgi/admin/ipfilter.cgi?action=<value>\[&iprange=<value>\]](http://<servername>/cgi/admin/ipfilter.cgi?action=<value>[&iprange=<value>])

with the following parameters and values

parameter	Value	Description
action=<string>	add, remove,removeall,update list	Specifies the action to take. add = Add new IP address (or addresses). remove = remove an entry in the IP address List. removeall = Remove all IP addresses. The IP address filtering function will automatically be disabled. update = Update settings for the IP address filtering function. list = List the settings for the IP address filtering function.
iprange=<ip range>	<IP addresses range>	The addresses denied passing through the

	192.168.0.1-192.168.0.100, 192.168.1.1-192.168.1.10, . . .	filter..
--	--	----------

Example:

1.Remove an entry in the list of addresses.

<http://myserver/cgi/admin/ipfilter.cgi?action=remove&iprange=10.13.10.12-10.13.10.100>

2.Add 10.13.10.12 -10.13.10.100 to the list of addresses which will be denied access to the device.

<http://myserver/cgi/admin/ipfilter.cgi?action=add&iprange=10.13.10.12-10.13.10.100>

3.Remove all IP addresses and automatically disable the IP address filtering function

<http://myserver/cgi/admin/ipfilter.cgi?action=removeall>

Return:

A successful add, remove, removeall, or update.	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n OK\r\n
A successful List	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n 1.1.1.1-1.1.1.2\r\n 2.2.2.2-2.2.2.3\r\n 3.3.3.3-3.3.3.4\r\n 1.1.1.1-1.1.1.2\r\n\r\n
Failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n

	\r\n Request failed: <error message>\r\n
--	---

15. Pan & Tilt

Making the Pan & Tilt commands ,

Also check the [PanTilt] in the param.cgi to make additional pan&tilt configuration

15.1 Pan Tilt and position related

Method: Get/Post

Syntax:

[http:// <servername>/cgi/admin/ptctrl.cgi?action=move&Cmd=<value>](http://<servername>/cgi/admin/ptctrl.cgi?action=move&Cmd=<value>)

with the following parameters and values

parameter	Value	Description
action=<string>	move	Specifies the action to take. .
Cmd	A String(up down left right home stop)	PanTilt move command .
	A String(panscan)	Do Pan scan command
	A String(autopatrol)	Do Auto Patrol command
	A String(calibration)	Adjust PT motor
	A String(Position1 Position2 Position3 Position4	Move to preset position

	Position5	
	Position6	
	Position7	
	Position8)	

Example:

1. Move up

<http://myserver/cgi/admin/ptctrl.cgi?action=move&Cmd=up>

2. Make a pan scan

<http://myserver/cgi/admin/ptctrl.cgi?action=move&Cmd=panscan>

3. Go to position 5

<http://myserver/cgi/admin/ptctrl.cgi?action=move&Cmd=Position5>

Return:

A successful	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n
Failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

15.2 Pan & Tilt moving with specified steps

Method: Get/Post

Syntax:

[http:// <servername>/cgi/admin/ptctrl.cgi?action=movedegree&Cmd=<value>&Degree=<value>](http://<servername>/cgi/admin/ptctrl.cgi?action=movedegree&Cmd=<value>&Degree=<value>)

with the following parameters and values

parameter	Value	Description
action=<string>	movedegree	Specifies the action to take. .
Cmd	A String(up	PanTilt move command . .

	down left right)	
Degree	A number	Move units

Example:

1.Move up 30 steps

<http://myserver/cgi/admin/ptctrl.cgi?action=movedegree&Cmd=up&Degree=30>

2.Move right 40 steps

<http://myserver/cgi/admin/ptctrl.cgi?action=movedegree&Cmd=right&Degree=40>

Return:

A successful	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n
Failed	HTTP/1.1 200 OK\r\n Content-Type: text/plain\r\n Content-Length: <content length>\r\n \r\n Request failed: <error message>\r\n

16. HTTP code and meanings

HTTP code	HTTP text	Description
200	OK	The request has succeeded, but an application error can still occur, which will be returned as an application error code.
204	No Content	The server has fulfilled the request, but there is no new information to send back.
302	Moved Temporarily	The server redirects the request to the URI given in the Location header.
400	Bad Request	The request had bad syntax or was impossible to fulfill.

401	Unauthorized	The request requires user authentication or the authorization has been refused.
404	Not Found	The server has not found anything matching the request.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource.
500	Internal Error	The server encountered an unexpected condition that prevented it from fulfilling the request.
503	Service Unavailable	The server is unable to handle the request due to temporary overload.

17. MJpeg (Standard server push)

Returns a standard server push motion jpeg stream with the default resolution and compression as defined in the system configuration

Syntax: <http://<servername>/cgi/mjpg/mjpg.cgi>

Example:

Please add this example html code in web page.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<TITLE>MAIN page of VIEW</TITLE>
</HEAD>
<BODY>
<img src = " http://<servername>/cgi/mjpg/mjpg.cgi "/>
</BODY>
</HTML>
```

return: Standard server push motion jpeg stream

Appendix A

A1. Security level

Security level	Description
0	Unprotected, but it is not possible to access the camera from outside without at least view rights.
1	View access right needed to access the parameter.
4	Operator access right needed to access the parameter.
6	Administrator access right needed to access the parameter.
7	Internal parameters that can only be changed by firmware applications or by root, editing the configuration files directly.

A2. Parameter Group

[General]

Parameter name	Default value	Valid values	Security level (get/set)	Description
Brand	*	A string (Auto generated)	0/7	The brand of the product.
ProdFullName	*	A string (Auto generated)	0/7	The full name of the product.
ProdNbr	*	A string (Auto generated)	0/7	The product number.
ProdType	*	Video Server, Network Camera, Network Video Recorder (Auto generated)	0/7	The product type.
ProdBuild	*	A number	0/7	The product build

* The default value for these parameters is product specific.

ProdType: 0-> Video Server 1-> Network Camera 2->Network Video Recorder

[Hardware]

Parameter name	Default value	Valid values	Security level (get/set)	Description
MPEG4	*	0/1	0/7	Mpeg4 stream is available or not.
MJPEG	*	0/1	0/7	Motion-Jpeg stream is available or not
MicIn	*	0/1	0/7	Audio stream from device to PC.
AudioOut	*	0/1	0/7	Audio from PC to device
PT	*	0/1	0/7	Pan-Tilt model
Zoom	*	0/1	0/7	Zoom Lens available
NightSensor	*	0/1	0/7	Night sensor available
Wireless	*	0/1	0/7	Wireless available
RS485	*	0/1	0/7	RS-485 protocol support
TriggerIn	*	0-number	0/7	Numbers of trigger in
TriggerOut	*	0-number	0/7	Numbers of trigger-out
MotionDetect	*	0/1	0/7	Motion detection is available
USB	*	0/1	0/7	Usb slot is available

* The default value for these parameters is product specific.

[System.Info]

Parameter name	Default value	Valid values	Security level (get/set)	Description
ServerName	*	A string	0/6	Server name.
ServerLocation	*	A string	0/6	Server location
MACAddress	*	xx:xx:xx:xx:xx:xx	0/7	MAC address
ModelName	*	A string	0/7	Model name
FWVersion	*	X.Y.Z (ex:1.0.0)	0/7	Firmware version
FWBuild	*	A number	0/7	Firmware build number

[Network]

Parameter name	Default value	Valid values	Security level (get/set)	Description
BootProto	static	dhcp, static pppoe	6/6	IP address assignment method.

Broadcast	192.168.0.255	An IP address	6/7	Broadcast address. Used to disseminate information to several recipients simultaneously.
DefaultRouter	192.168.0.1	An IP address	6/6	Default router/gateway used for connecting devices attached to different networks and network segments.
DNSServer1	0.0.0.0	An IP address	6/6	Primary Domain Name System server.
DNSServer2	0.0.0.0	An IP address	6/6	Secondary Domain Name System server.
InterfaceSelectMode ¹	auto	auto,	6/7	Interface select mode. Defines how the device chooses which network interface to use. In auto mode the wireless interface will be used when a wired network connection cannot be detected. In

				wired mode only the wired interface will be used, regardless of its' status.
IPAddress	192.168.0.30	An IP address	6/6	IP Address. The physical address of the device on the network.
Media	auto	auto,	6/7	Media type on the network.
SubnetMask	255.255.255.0	An IP address	6/6	Subnet mask. Divides the network.

*Need reboot to take effect

[Network.PPPoE]

Parameter name	Default value	Valid values	Security level (get/set)	Description
UserName		A string	6/6	The user name for the pppoe server .
Password		A string	6/6	The password for the pppoe server .

[Network.DNSUpdate]

Parameter name	Default value	Valid values	Security level (get/set)	Description
Provider			6/6	DDNS provider.
DNSName		<absolute domain name> or	6/6	The name entered here will be associated with the product's IP address in the

		<i>< domain name relative to root></i>		DNS server. An example of a DNS name is product.example.com.
Enabled	no	yes, no	6/6	Enable/disable dynamic DNS service.
UserName		A String	6/6	
Password		A String	6/6	
ProviderList		A String	6/7	List all DDNS Provider. Each provider is separated by a semicolon [format] www.provider1.com: www.provider2.org: www.provider3.net...

[Network.UPnP]

Parameter name	Default value	Valid values	Security level (get/set)	Description
Enabled	yes	yes, no	6/6	Enables Universal Plug and Play.

FriendlyName	<product name> - <serial number>	A string	6/7	The name of the UPnP device.
--------------	---	----------	-----	---------------------------------

[Network.Wireless]

Parameter name	Default value	Valid values	Security level (get/set)	Description
ESSID		A string, up to	6/6	The name of the

		32 ASCII characters are accepted		wireless network the device is configured for. The name must be exactly the same as used in the wireless access point, or the connection will not be established. ESSID is called SSID by some vendors.
Mode	managed	managed, ad-hoc	6/6	The type of wireless network
Enabled	yes	yes, no	6/6	Enable/disable wireless
Channel	Product	1..14	6/6	Wireless channel(used in ad-hoc mode)

Parameter name	Default value	Valid values	Security level (get/set)	Description
Method	WPA-PSK	WPA-PSK WPA2-PSK	6/6	Security method.
Enabled	no	yes, no	6/7	Show Enable/disable WPA-PSK security. WPA2-PSK
Encryption		TKIP AES	6/6	Encryption mode
GenerationMethod	psk-phrase	psk-phrase, psk-hex	6/7	Defines which of the parameters below that is used as pre-shared key.
Passphrase		A string, 8 - 63 ascii chars	6/6	The passphrase must match the passphrase in the wireless access point. When passphrase is used, the parameter GenerationMethod must be "psk-phrase".
Key		64 Hex chars	6/6	The key must match the key in the wireless access point. When key is used, the parameter GenerationMethod must be "psk-hex".

Parameter name	Default value	Valid values	Security level (get/set)	Description
Authentication	open	open, shared	6/6	The WEP authentication method.
Enabled	no	yes, no	6/6	Enable/disable WEP encryption over the wireless link.
KeyLength	128	64, 128	6/6	Set the strength of the wireless encryption.
GenerationMethod	manual	manual, ASCII,	6/6	This parameter must tell which of the parameters below that is used and how to interpret them. Manual hex always works, while ASCII and passphrase are easier to remember.
Key1		A string	6/6	The keys must match the keys in the wireless access point.
Key2		A string	6/6	
Key3		A string	6/6	
Key4		A string	6/6	
ActiveKey	1	1 ... 4	6/6	Which key to use when transmitting. The key should be changed every now and then, to increase

				the security of the wireless connection.
--	--	--	--	---

[NetworkLED]

Parameter name	Default value	Valid values	Security level (get/set)	Description
Usage	on	on, off	6/6	Network LED behavior. on = LED is allowed to lit. off = LED is not allowed to lit.

[SMTP]

Parameter name	Default value	Valid values	Security level (get/set)	Description
FromEmail		An e-mail address	6/6	E-mail address of the individual or device from which the e-mail is sent.
MailServer1		An IP address or a host name	6/6	Primary mail server.
MailServer1		An IP address or a host name	6/6	Primary mail server.
EmailTo		A string	6/6	E-mail address.
EmailTo2		A string	6/6	E-mail address.

[SMTP.Authentication]

Parameter name	Default value	Valid values	Security level (get/set)	Description
Enabled	no	yes, no	6/6	Enable/disable authentication.
UserName		A string	6/6	The user name for the mail server or the POP server.
Password		A string	6/6	The password for the mail server or the POP server.
WeakestMethod	Login	Login	6/7	The weakest method allowed for mail server login.

[FTP]

Parameter name	Default value	Valid values	Security level (get/set)	Description
FtpServer		An IP address or a host name	6/6	Ftp server address.
Port	21	1..65535	6/6	Ftp server port.
UserName		A string	6/6	UserName to login the Ftp server.
Password		A string	6/6	Password to login the Ftp server.
Path		A string	6/6	Login path.
Passive	yes	yes no	6/6	yes: Transfer data in passive mode. no: in active mode

[NetStorage]

Parameter name	Default value	Valid values	Security level (get/set)	Description
SambaServer		An IP address or a host name	6/6	Samba server address.
Share		A string	6/6	Samba share folder name
Path		A string	6/6	Login path.
UserName		A string	6/6	UserName to login the samba server.
Password		A string	6/6	Password to login the samba server.

Anonymous	no	yes no	6/6	Anonymous access to samba server
SplitBy	size	size time	6/6	Split the file by size or by recording time
SplitBySize	Product	12..640	6/6	When 'SplitBy' is set to 'size', this value should be set to indicate the threshold size(MB) of the recording file being splitted
SplitByTime	Product	1..60	6/6	When 'SplitBy' is set to 'time', this value should be set to indicate the threshold time (minutes) of the recording file being splitted
DiskFull	Product	stop recycle	6/6	When the samba server storage is full ,set to 'stop' will stop further recording. And set to 'recycle' will delete the oldest folder to restore the capacity of the server.

[External.Storage]

Parameter name	Default value	Valid values	Security level (get/set)	Description
Mount		yes no	6/6	yes: the external storage has already mounted no: external storage is not mounted or is not existed. Note that only 'no' can be sent while doing set method
TotalSpace		Digits	6/7	Total capacity of the external storage (MB).
FreeSpace		digits	6/7	Free capacity for the external storage (MB).
DiskFull	Product	stop recycle	6/6	When the external storage is full ,set to 'stop' will stop further recording. And set to 'recycle' will delete the oldest folder to restore the capacity of the server.

[Time]

Parameter name	Default value	Valid values	Security level (get/set)	Description
ServerDate		A date	6/6	The date (yyyy-mm-dd) when the device's time was set manually or synchronized with the computer.
ServerTime		A time	6/6	The time (hh:mm:ss) when the device's time was set manually or synchronized with the computer.
SyncSource	Product dependent	NTP, manually	6/6	The source to synchronize the time with; NTP or manually
TimeZone	22	0-62 (refer to Appendix	6/6	Time zone.

		B)		
--	--	----	--	--

[Time.NTP]

Parameter name	Default value	Valid values	Security level (get/set)	Description
Server	0.0.0.0	An IP address or a host name	6/6	The NTP server to connect to when synchronizing the time in the device.
Update	06:00:00	A time (06:00:00 12:00:00 24:00:00)	6/6	Time interval (hh:mm:ss) between connections to the NTP server.

[ImageSource.IO.Sensor]

Parameter name	Default value	Valid values	Security level (get /set)	Description
AutoSlowShutter	(*)	on, off	4/4	on = allows camera to use slower shutter in low light

				environments. off = keep a fixed shutter speed.
Brightness	(*)	0 ... 100	4/4	The image brightness.
ColorLevel	(*)	0 ... 100	4/4	The image color level.
Contrast	(*)	0 ... 100	4/4	The image contrast.
Exposure	(*)	auto, flickerfree50, flickerfree60	4/4	The image exposure.
NightVision	(*)	auto, none	4/4	Will increase sensitivity in low light environments when on.
Mirror	off	on, off	4/4	Mirror the image horizontally
OverlayDate	off	on, off	4/4	Put datetime overlay on the video
OverlayOpaque	off	on, off	4/4	on=overlay text with opaque background off=overlay text with transparent background

(*) model dependent

[Properties.Audio]

Parameter name	Default value	Valid values	Security level (get/set)	Description
Audio		yes, no	0/7	The product has audio support.
Talk		Yes, no	0/7	
Format	PCM	PCM	0/7	The supported formats separated by commas, e.g.pcm.

[System]*

Parameter name	Default value	Valid values	Security level (get/set)	Description
HTTPPort	80	1..65535	6/7	HTTP port number.
RTSPPort	554	1..65535	6/7	RTSP port number.

(*) need reboot to take effect

[Image.MJPEG] *

Parameter name	Default value	Valid values	Security level (get	Description
----------------	---------------	--------------	---------------------	-------------

			/set)	
Resolution	Product	0..2	4/4	MJPEG stream resolution 0: 640x480 1: 320x240 2: 160x120
Framerate	30	1-25,30	4/4	MJPEG stream frame rate per second
Quality	Product	0-4	4/4	MJPEG stream quality, 0 is lowest, 4 is highest

(*) need reboot to take effect

[Image.MPEG] *

Parameter name	Default value	Valid values	Security level (get /set)	Description
Resolution	Product	160x120,320x240, 640x480	4/4	MPEG4 stream resolution 0: 640x480 1: 320x240 2: 160x120
Framerate	30	1-25,30	4/4	MPEG4 stream frame rate per second
Quality	Product	0-4	4/4	MPEG4 stream quality, 0 is lowest, 4 is highest

(*) need reboot to take effect

[Audio]

Parameter name	Default value	Valid values	Security level (get/set)	Description
MicEnable	Yes	yes, no	6/6	Enable the Microphone input
SpeakerEnable	Yes	yes, no	6/6	Enable the speaker output
SpeakerVolume	Product	0..99		Speaker volume

[GPIO]

Parameter name	Default value	Valid values	Security level (get/set)	Description
TriggerIn#	On	A string (on, off)	#/6	Set Trigger Input on / off Ex: TriggerIn0 = on or off
TriggerOut#	Low	A string (high, low)	#/6	Set Trigger Output Status high / low. Ex: TriggerOut0= high or low

TriggerInStatus#	On	A string (on, off)	6/#	Get Trigger Input on / off Status Ex: http:// <servername>/cgi/admin/param.cgi?action=list&group=GPIO&name=TriggerInStatus0
TriggerOutStatus#	Low	A string (high, low)	6/#	Get Trigger Output Status Ex: http:// <servername>/cgi/admin/param.cgi?action=list&group=GPIO&name=TriggerOutStatus0

Note: The # is replaced with a number which based on the numbers of input/output pins starting from zero. e.g. TriggerIn0 , TriggerIn1 , TriggerIn2.

cs1300/cs1310 model support 1 TriggerIn , 1 TriggerOut.

[PanTilt]

Parameter name	Default value	Valid values	Security level (get/set)	Description
Enable	by model	A string (yes, no)	6/7	Pan/Tilt support msg
PanSpeed	slow	A string (slow, medium fast)	6/6	Pan Speed.

TiltSpeed	slow	A string (slow, medium fast)	6/6	Tilt Speed
PanScanSpeed	slow	A string (slow, medium fast)	6/6	Pan Scan Speed
PanStep	1	A number(0~20)	6/6	Pan Steps(degree)
TiltStep	1	A number(0~20)	6/6	Tile Steps(degree)
PatrolStayTime	1	A number(1~999),	6/6	Stay Time(secs) between each preset position
Position1	Position1	A string	6/6	Name of preset Position1.
Position2	Position2	A string	6/6	Name of preset Position2.
Position3	Position3	A string	6/6	Name of preset Position3.
Position4	Position4	A string	6/6	Name of preset Position4.
Position5	Position5	A string	6/6	Name of preset Position5.
Position6	Position6	A string	6/6	Name of preset Position6.
Position7	Position7	A string	6/6	Name of preset Position7.
Position8	Position8	A string	6/6	Name of preset Position8.

Appendix B

TimeZone List

Index	Zone
0	(GMT-12:00) Eniwetok, Kwajalein
1	(GMT-11:00) Midway Island, Samoa
2	(GMT-10:00) Hawaii
3	(GMT-09:00) Alaska
4	(GMT-08:00) Pacific Time(US & Canada); Tijuana
5	(GMT-07:00) Arizona
6	(GMT-07:00) Mountain Time(US & Canada)
7	(GMT-06:00) Central Time(US & Canada)
8	(GMT-06:00) Mexico City, Tegucigalpa
9	(GMT-06:00) Saskatchewan
10	(GMT-05:00) Bogota, Lima, Quito
11	(GMT-05:00) Eastern Time(US & Canada)
12	(GMT-05:00) Indiana(East)
13	(GMT-04:00) Atlantic Time(Canada)
14	(GMT-04:00) Caracas, La Paz
15	(GMT-04:00) Santiago
16	(GMT-03:30) Newfoundland
17	(GMT-03:00) Brasilia
18	(GMT-03:00) Buenos Aires, Georgetown
19	(GMT-02:00) Mid-Atlantic
20	(GMT-01:00) Azores, Cape Verde Is.
21	(GMT) Casablanca, Monrovia
22	(GMT) Greenwich Mean Time : Dublin, Edinburgh, Lisbon, London
23	(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
24	(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague
25	(GMT+01:00) Brussels, Copenhagen, Madrid, Paris, Vilnius
26	(GMT+01:00) Sarajevo, Skopje, Sofija, Warsaw, Zagreb
27	(GMT+02:00) Athens, Istanbul, Minsk
28	(GMT+02:00) Bucharest
29	(GMT+02:00) Cairo
30	(GMT+02:00) Harare, Pretoria
31	(GMT+02:00) Helsinki, Riga, Tallinn

32	(GMT+02:00) Israel
33	(GMT+03:00) Baghdad, Kuwait, Riyadh
34	(GMT+03:00) Moscow, St. Petersburg, Volgograd
35	(GMT+03:00) Nairobi
36	(GMT+03:30) Tehran
37	(GMT+04:00) Abu Dhabi, Muscat
38	(GMT+04:00) Baku, Tbilisi
39	(GMT+04:30) Kabul
40	(GMT+05:00) Eekaterinburg
41	(GMT+05:00) Islamabad, Karachi, Tashkent
42	(GMT+05:30) Bombay, Calcutta, Madras, New Delhi
43	(GMT+06:00) Almaty, Dhaka
44	(GMT+06:00) Colombo
45	(GMT+07:00) Bangkok, Hanoi, Jakarta
46	(GMT+08:00) Beijing, Chongqing, Hong Kong, Urumqi
47	(GMT+08:00) Perth
48	(GMT+08:00) Singapore
49	(GMT+08:00) Taipei
50	(GMT+09:00) Osaka, Sapporo, Tokyo
51	(GMT+09:00) Seoul
52	(GMT+09:00) Yakutsk
53	(GMT+09:30) Adelaide
54	(GMT+09:30) Darwin
55	(GMT+10:00) Brisbane
56	(GMT+10:00) Canberra, Melbourne, Sydney
57	(GMT+10:00) Guam, Port Moresby
58	(GMT+10:00) Hobart
59	(GMT+10:00) Vladivostok
60	(GMT+11:00) Magadan, Solomon Is., New Caledonia
61	(GMT+12:00) Auckland, Wellington
62	(GMT+12:00) Fiji, Kamchatka, Marshall Is.