IP Camera Open API, HTTP - Interface Specification

Content

DOCUMENT HISTORY

Version	Date	Supported Firmware	Release Notes
1.00	2007-May-14	V3.0.0.0	Initial version
1.01	2007-Sep-27		Check un-implement function
1.02	2007-Oct-01		Update ptzconfig.cgi
1.03	2007-Oct-11	V3.0.2.0	Implement I/O - input.cgi and output.cgi
1.04	2007-Dec-03	V3.0.2.1571	Implement videostatus.cgi
1.05	2007-Dec-18	V3.0.2.1631	Implement receive.cgi and imagesize.cgi
1.06	2007-Dec-28	V3.0.2.1691	Revise Document
2.00	2008-May-22	V5.0.0.0	video.cgi support resolution info. **h264**
2.01	2008-Sep-26	V5.0.0.2565	Add port.cgi

2.02	2008-Oct-26	V5.0.0.2667	Modify PTZ.cgi
2.03	2009-Mar-02	V5.0.0.3006	Modify OSD.cgi
2.04	2009-Jun-06	V5.0.0.3270	Implement motion data
2.05	2009-Sep-25	V5.0.0.3507	Modified PTZ control
2.06	2009-Oct-08	V5.0.0.3551	Support ptz.cgi?rawdata=<string>
3.00	2009-Dec-10	V5.x.1.3714	API re-define.
3.01	2010-Jun-10	V6	Still image support profilename parameter.
3.02	2012-Feb-16	V6.M	Support overlay image
3.03	2012-May-21	V6	Support continuous recoding list
3.03	2012-Jul-20	V6 and V5	Force Trigger
3.04	2012-Jul-24	V6	Add 3.13 Apple Push Notification Service (APNS)
3.05	2012-Aug-3	V6	Add audio deteciton
3.06	2012-Aug-3	V6	Add fisheye control
3.07	2012-Oct-18	V6.S	Add face detection
3.08	2012-Dec-18	V6.M	Add Camera Tampering
3.09	2013-Sep-27	V6.S	Modify face detection. Add Cross Line, Objection Detection,and VMD.
3.10	2014-Apr-2	V6.M, V6.N, V6.J, V6.L	Add UTF8OSDEnabled and UTF8OSDStr parameters in setosd.cgi
3.11	2015-Jan-13	V6.S	Add iSCSI

---

1. REFERENCES

HTTP protocol

- Hypertext Transfer Protocol -- HTTP/1.0

2. INTERFACE SPECIFICATION

2.1 CGI URL syntax

Syntax:

```
http://<servername>/<subdir>[/<subdir>...]/<cgi>.<ext>
```

```
[?<parameter>=<value>[&<parameter>=<value>...]]
```

Example: List the Network parameters.

```
http://<servername>/config.cgi?action=list&group=Network
```

2.2 HTTP Server responses

The built-in Web server uses the standard HTTP status codes.

Return:

```
HTTP/1.0 <HTTP code> <HTTP text>\r\n
```

with the following 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.

Example: Request includes invalid file names.

```
HTTP/1.0 404 Not Found\r\n
```

3. API GROUPS

3.1 General

The requests specified in the General section are supported by all video products with firmware version 3.00 and above.

3.1.1 Add, update, remove and list parameters and their values

Note:

- The URL must follow the standard way of writing a URL, (RFC 2396: Uniform Resource Identifiers (URI) Generic Syntax); that is, 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 will have to be replaced with %20, My%20camera.

Method: GET/POST

Syntax:

```
http://<servername>/config.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

Alias Syntax:

```
http://<servername>/param.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
action=<string>	add, remove, update or list	Specifies the action to take. Depending on this parameter, various parameters may be set, as described in the following sections.

3.1.1.1 List parameters

Syntax:

```
http://<servername>/config.cgi?action=list
[&<parameter>=<value>...]
```

Alias Syntax:

```
http://<servername>/param.cgi?action=list
[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
group=<string>[,<string>...]	<group![.name]>[,<group![.name]>...]	Returns the value of the camera parameter named <group>.<name>. If <name> is omitted, all the parameters of the <group> are returned. The camera parameters must be entered exactly as they are named in the camera or video server. Wildcard (*) can be used when listing parameters. See example below. If this parameter is omitted, all parameters in the device are returned.
responseformat	rfc	Get the HTTP response format according to standard. Response format: HTTP/1.0 200 OK\r\n Content-Type: text/plain\r\n \r\n <parameter pair>

Example: List the Network parameters.

```
http://myserver/config.cgi?action=list&group=Network
```

Alias:

```
http://myserver/param.cgi?action=list&group=Network
```

3.1.1.2 List output format

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\n
\n
<parameter pair>
where <parameter pair> is
<parameter>=<value>\n
[ <parameter pair> ]
```

Example: Network query response.

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\n
\n
Network.IPAddress=10.13.12.36\n
Network.!SubnetMask=255.255.255.0\n
```

If the CGI request includes an invalid parameter value, the server returns an error message. Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\n
\n
# Error: <description>\n
```

3.1.1.3 Update parameters

Syntax:

```
http://<servername>/config.cgi?action=update[&<parameter>=<value>...]
```

Alias Syntax:

```
http://<servername>/param.cgi?action=update[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
<string>=<string>	<group.name>=<value>	Assigns <value> to the parameter <group.name>. The <value> must be URL-encoded when it contains non-alphanumeric characters. The camera parameters must be entered exactly as named in the camera or the video server.

Example: Set the default image resolution to 320x240 pixels.

```
http://myserver/config.cgi?action=update&Image.I0.Resolution=320x240
```

Alias:

```
http://myserver/param.cgi?action=update&Image.I0.Resolution=320x240
```

3.1.1.4 Add parameters

Note: Only applicable for dynamic parameter groups such as the event parameters.

Syntax:

```
http://<servername>/config.cgi?action=add[&<parameter>=<value>...]
```

Alias Syntax:

```
http://<servername>/param.cgi?action=add[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
template=<string>	<template>	Use the specified <template> when creating the new group. The template is a file, which describes all parameters for this group. Depending on the product, different templates are available, please check the Release notes of the firmware version. See examples below.
group=<string>	Event, PTZ.PresetPos, GuardTour, GuardTour.G#.Tour	Specifies the parent group. The parent group defines where in the parameter structure the new group will be created. For example, if adding an event (template=event) and specify group=Event the new group will be available as Event.E<number>. Where <number> is the unique number for the group (see return values below). The character before <number> is generated from the last section of the group name. E.g. Event will generate the character E and Event.Notification will generate the character N.
<string>=<string>	<group.name>=<value>	Set a parameter in the newly created group. As the group number is not known before the group is created, the id-number is simply left out, see the examples below. The new group number is created dynamically and can be any number. This is why all parameters are specified to set without any group number. The base path to the parameter is specified as <group>.<uppercase first letter of group>.<parameter name>.

Example: Create a new event under the group Event and set the name to MyEvent and the Enabled parameter to yes.

```
http://myserver/config.cgi?action=add&group=Event
    &template=event&Event.E.Name=!MyEvent&Event.E.Enabled=yes
```

Alias:

```
http://myserver/param.cgi?action=add&group=Event
    &template=event&Event.E.Name=!MyEvent&Event.E.Enabled=yes
```

Example: A listing of the new group will output the following.

```
Event.E0.Name=!MyEvent
Event.E0.Type=T
Event.E0.Enabled=yes
Event.E0.Active=no
Event.E0.Priority=1
Event.E0.Image=1
Event.E0.HWInputs=xxxx
Event.E0.SWInput=
Event.E0.Weekdays=1111111
Event.E0.Starttime=00:00
Event.E0.Duration=0
Event.E0.ImageURLSettingsEnabled=no
Event.E0.ImageURLSettings=
```

```
Event.E0.IncludePreTrigger=no
Event.E0.PreTriggerSize=0
Event.E0.PreTriggerInterval=1000
Event.E0.PreTriggerIntervalUnit=s
Event.E0.PreTriggerUnit=s
Event.E0.IncludePostTrigger=no
Event.E0.PostTriggerSize=0
Event.E0.PostTriggerInterval=1000
Event.E0.PostTriggerIntervalUnit=s
Event.E0.PostTriggerUnit=s
Event.E0.IncludeBestEffort=no
Event.E0.BestEffortInterval=1000
Event.E0.BestEffortDuration=0
Event.E0.BestEffortIntervalUnit=s
Event.E0.BestEffortDurationUnit=s
Event.E0.FileName=image.jpg
Event.E0.Suffix=%y-%m-%d_%H-%M-%S-%f
Event.E0.MaxSequenceNumber=-100
```

Note that in this example the id is E0. This can be any number, depending on if other events were added before. Parameters that are not specified in the request will have their default values, as specified in the configuration file.

3.1.1.5 Remove parameters

Note: Only applicable for dynamic parameter groups such as the event parameters.

Syntax:

```
http://<servername>/config.cgi?action=remove[&<parameter>=<value>...]
```

Alias Syntax:

```
http://<servername>/param.cgi?action=remove[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
group=<string>[,<string>...]	<group>[,<group>]	Deletes the specified <group>

Example: Delete event group E2 and E4.

```
http://myserver/config.cgi?action=remove&group=Event.E2,Event.E4
```

Alias:

```
http://myserver/param.cgi?action=remove&group=Event.E2,Event.E4
```

3.1.1.6 Add/Remove? server responses

These actions produce one of the following server responses:

Return: A successful add.

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

Return: A successful remove.

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

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

Example: Add new event entry and set the specified name.

```
http://myserver/config.cgi?action=add&group=Event&template=event&Event.E.Name=!MyEvent
```

Alias:

```
http://myserver/param.cgi?action=add&group=Event&template=event&Event.E.Name=!MyEvent
```

Response:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\n
\n
E7 OK\r\n
```

Example: Delete an event entry.

```
http://myserver/config.cgi?action=remove&group=Event.E7
```

Alias:

```
http://myserver/param.cgi?action=remove&group=Event.E7
```

Response:

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

3.1.2 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 root access (root authorization).

Method: GET/POST

Syntax:

```
http://<servername>/usrgrp.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

Alias Syntax:

```
http://<servername>/pwdgrp.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
action=<string>	add, update, remove or 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, a non-existing user name. Valid characters are a thru z, A thru Z and 0 thru 9.
pwd=<string>	<string>	The unencrypted password of the account. Valid characters are a thru z, A thru Z and 0 thru 9.
grp=<string>	administrator, viewer	An existing primary group name of the account.
sgrp=<string>	ptz	existing secondary group names of the account.

Example: Create a new administrator account.

```
http://myserver/usrgrp.cgi?action=add&user=joe&pwd=foo&grp=administrator&sgrp=ptz
```

Alias:

```
http://myserver/pwdgrp.cgi?action=add&user=joe&pwd=foo&grp=administrator&sgrp=ptz
```

Example: Change the password of an existing account.

```
http://myserver/usrgrp.cgi?action=update&user=joe&pwd=bar
```

Alias:

```
http://myserver/pwdgrp.cgi?action=update&user=joe&pwd=bar
```

Example: Remove an account.

```
http://myserver/usrgrp.cgi?action=remove&user=joe
```

Alias:

```
http://myserver/pwdgrp.cgi?action=remove&user=joe
```

Example: List groups and users.

```
http://myserver/usrgrp.cgi?action=get
```

Alias:

```
http://myserver/pwdgrp.cgi?action=get
```

3.1.3 Factory default

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

Note: This requires administrator access (administrator authorization).

Method: GET

Syntax:

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

3.1.4 Hard factory default

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

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

Method: GET

Syntax:

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

3.1.5 Backup

Download backup.bin into PC.

Note: This requires ROOT access (administrator authorization).

Method: GET

Syntax:

```
http://<servername>/backup.cgi
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: application/octet-stream\r\n
Content-Disposition: attachment; filename=backup.bin\r\n
\r\n
<file content of backup.bin>
```

3.1.6 Restore

Upload a unit specific backup previously created by the backup.cgi.

Note: This requires administrator access (administrator authorization).

Method: POST

Syntax:

```
http://<servername>/restore.cgi
```

The file content is provided in the HTTP body according to the format given in RFC 1867. The body is created automatically by the browser if using HTML form with input type "file".

Example: Upload of backup, where "\r\n" has been omitted in the HTTP body.

```
POST /restore.cgi? HTTP/1.0\r\n
Content-Type: multipart/form-data; boundary=AaBo3x\r\n
Content-Length: <content length>\r\n
\r\n
--AaBo3x\r\n
Content-Disposition: form-data; name="backup.bin";
filename="backup.bin"\r\n
Content-Type: application/octet-stream\r\n
\r\n
<file content of backup.bin>
\r\n
--AaBo3x--\r\n
```

3.1.7 Firmware upgrade

Upgrade the firmware version.

Note: This requires administrator access (administrator authorization).

Method: POST

Syntax:

```
http://<servername>/firmwareupgrade.cgi
```

The file content is provided in the HTTP body according to the format given in RFC 1867. The body is created automatically by the browser if using HTML form with input type "file".

Example:

```
POST /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="firmware.pck"; filename="firmware.pck"\r\n
Content-Type: application/octet-stream\r\n
\r\n
<firmware file content>
\r\n
--AsCg5y--\r\n
```

3.1.8 Reboot server

Restart server.

Note: This requires administrator access (administrator authorization).

Method: GET

Syntax:

```
http://<servername>/reboot.cgi
```

Alias Syntax:

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

3.1.9 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>/systemlog.cgi
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
<system log information>
```

3.1.10 System date and time

Get or set the system date and time.

Method: GET/POST

Syntax:

```
http://<servername>/time.cgi?<parameter>=<value>
```

Alias Syntax:

```
http://<servername>/date.cgi?<parameter>=<value>
```

with the following parameter and values

<parameter>=<value>	Values	Description
action=<string>	get or set	Specifies what to do. get = get the current date and time. set = set the current date and/or time.

3.1.10.1 Get system date and time

Syntax:

```
http://<servername>/time.cgi?action=get
```

Alias Syntax:

```
http://<servername>/date.cgi?action=get
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
<month> <day>, <year> <hour>:<minute>:<second>\r\n
```

Example:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Apr 03, 2003 15:16:04\r\n
```

3.1.10.2 Set system date and time

Syntax:

```
http://<servername>/time.cgi?action=set[&<parameter>=<value>...]
```

Alias Syntax:

```
http://<servername>/date.cgi?action=set[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
year=<int>	2000 - 2031	Current year.
month=<int>	1 - 12	Current month.
day=<int>	1 - 31	Current day.
hour=<int>	0 - 23	Current hour.
minute=<int>	0 - 59	Current minute.
second=<int>	0 - 59	Current second.
timezone=<string>	GMT	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. 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. Currently only GMT is considered valid input. The rest of the time zones are subject to future expansion.

The set action produces one of the following server responses:

Return: A successful set.

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

Example: Set the date.

```
http://myserver/time.cgi?action=set&year=2005&month=4&day=3
```

Alias:

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

Response:

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

3.1.11 OSD string

Set OSD string

Method: GET/POST

Syntax:

```
http://<servername>/setosd.cgi?<parameter>=<value>
```

with the following parameter and values

<parameter>=<value>	Values	Description
Commit=<string>	yes,no	yes: write to flash no: do not write to flash
DateEnabled=<string>	yes,no	Add system date to osd string
ClockEnabled=<string>	yes,no	Add system clock to osd string
TextEnabled=<string>	yes,no	Add extra osd string to osd string
String=<string>	string	extra osd string
OverlayImageEnabled=<string>	yes,no	Add overlay image
OverlayImageX=<string>	0~<max width of resolution>	Modify overlay image X coordinate
OverlayImageY=<string>	0~<max height of resolution>	Modify overlay image Y coordinate
UTF8OSDEnabled=<string>	yes,no	Enable/Disable? Simplified Chinese UTF-8

		OSD string
UTF8OSDStr=<string>	string	Simplified Chinese UTF-8 OSD string

3.1.12 Upload User Defined OSD File

Upload the user defined OSD file.

Note: This requires administrator access (administrator authorization).

Method: POST

Syntax:

```
http://<servername>/userdeftext.cgi
```

The file content is provided in the HTTP body according to the format given in RFC 1867. The body is created automatically by the browser if using HTML form with input type "file".

Example:

```
POST /userdeftext.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="text.osd"; filename="text.osd"\r\n
Content-Type: application/octet-stream\r\n
\r\n
<OSD file content>
\r\n
--AsCg5y--\r\n
```

3.1.13 Upload Overlay Image File

Upload the overlay image file.

Note: This requires administrator access (administrator authorization). Note: Only support 16 bit RGB 1555 bitmap file.

Method: POST

Syntax:

```
http://<servername>/overlayimage.cgi
```

The file content is provided in the HTTP body according to the format given in RFC 1867. The body is created automatically by the browser if using HTML form with input type "file".

Example:

```
POST /overlayimage.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="test.bmp"; filename="test.bmp"\r\n
Content-Type: application/octet-stream\r\n
\r\n
<OSD file content>
\r\n
--AsCg5y--\r\n
```

3.1.14 Image default

Setup default of all image setting in picture setting.

Method: GET/POST

Syntax:

```
http://<servername>/image_default.cgi
```

3.1.15 Connection Log

Get Connection Log

Method: GET

Syntax:

```
http://<servername>/connectionlog.cgi
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
<connection log information>
```

3.2 Image and Video

3.2.1 Image size

Get the actual image size of default image settings, or with given parameters.

Method: GET/POST

Syntax:

```
http://<servername>/resolution.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

Alias Syntax:

```
http://<servername>/imagesize.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
resolution=<string>	CIF...	Returns image size of the camera.

Example: Request image size of default settings.

```
http://myserver/resolution.cgi
```

Alias:

```
http://myserver/imagesize.cgi
```

Example: Returned data after a successful request.

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
image width = 176\n
image height = 144\n
```

Example: Request image size with supplied parameters for camera 1.

```
http://myserver/resolution.cgi?resolution=QCIF
```

Alias:

```
http://myserver/imagesize.cgi?resolution=QCIF
```

Example: Returned data after a successful request.

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
image width = 160\n
image height = 120\n
```

3.2.2 Video status

Check the status of one or more video sources.

Method: GET

Syntax:

```
http://<servername>/videostatus.cgi?<parameter>=<value>
```

Example: Request video status.

```
http://myserver/videostatus.cgi
```

Example: Returned data after a successful request.

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Video 1 = video
```

3.2.4 JPEG/MJPG

The requests specified in the JPEG/MJPG section are supported by those video products that use JPEG and MJPG encoding.

3.2.4.1 JPEG image request

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

Method: GET

Syntax:

```
http://<servername>/jpg[/<camera>]/image.jpg
```

with the following parameter

Parameter	Values	Description
resolution=<string>	1280x1024, 1024x768, 4CIF, D1, VGA, 640x480, QVGA, CIF, 320x240, QQVGA, QCIF, 160x120	Specify the resolution of the returned image.
profilename=<string>		Specify the profile name.

Example: Request JPEG image with default resolution and compression.

```
http://myserver/jpg/image.jpg
```

3.2.4.2 JPEG image (snapshot) CGI request

Request a JPEG image (snapshot) with specified properties.

Method: GET

Syntax:

```
http://<servername>/image.cgi[?<parameter>=<value>[&<parameter>=<value>...]]
```

with the following parameters and values

<parameter>=<value>	Values	Description
resolution=<string>	1280x1024, 1024x768, 4CIF, D1, VGA, 640x480, QVGA, CIF, 320x240, QQVGA, QCIF, 160x120	Specify the resolution of the returned image.
profilename=<string>		Specify the profile name.

Example: Request a JPEG image

```
http://myserver/image.cgi
```

3.2.4.3 JPEG image response

When a JPEG image is requested, the server returns either the specified JPEG image file or an error.

Return:

```
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
```

Example: Requested JPEG image.

```
HTTP/1.0 200 OK\r\n
Content-Type: image/jpeg\r\n
Content-Length: 15656\r\n
\r\n
<JPEG image data>\r\n
```

3.2.4.4 MJPG video request

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

Method: GET

Syntax: Request Multipart JPEG image.

```
http://<servername>/mjpg[/<camera>]/video.mjpg
```

with the following parameter

Parameter	Values	Description

resolution=<string>	1280x1024, 1024x768, 4CIF, D1, VGA, 640x480, QVGA, CIF, 320x240, QQVGA, QCIF, 160x120	Specify the resolution of the returned image.
profilename=<string>		Specify the profile name.

Example: Request JPEG image stream with default resolution and compression.

```
http://myserver/video.mjpg
```

3.2.4.5 MJPG video CGI request

Request a Multipart-JPEG image stream (video) with specified properties.

Method: GET

Syntax:

```
http://<servername>/mjpg/video.cgi
[?<parameter>=<value>[&<parameter>=<value>...]]
```

with the following parameters and values

<parameter>=<value>	Values	Description
resolution=<string>	1280x1024, 1024x768, 4CIF, D1, VGA, 640x480, QVGA, CIF, 320x240, QQVGA, QCIF, 160x120	Specify the resolution of the returned image.
profilename=<string>		Specify the profile name.

Example: Request a Multipart JPEG image stream

```
http://myserver/mjpg/video.cgi
```

3.2.4.6 MJPG video response

When MJPG video is requested, the server returns a continuous flow of JPEG files. The content type is "multipart/x-mixed-replace" and each image ends with a boundary string <boundary>. The returned image and HTTP data is equal to the request for a single JPEG image.

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=<boundary>\r\n
\r\n
--<boundary>\r\n
<image>
where the proposed <boundary> is
myboundary
and the returned <image> field is
Content-Type: image/jpeg\r\n
Content-Length: <image size>\r\n
\r\n
<JPEG image data>\r\n
--<boundary>\r\n
<image>
```

Example: Requested Multipart JPEG image.

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=myboundary\r\n
\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
Content-Length: 15656\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
Content-Length: 14978\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
Content-Length: 15136\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
 .
 .
 .
```

3.3 PTZ

The requests specified in the PTZ section are supported by those video products that have support for Pan/Tilt/Zoom? devices. Note! Installing a PTZ driver is done in two main steps:

1. Associate the driver with a serial port
2. Connect cameras to the serial port

PTZ administration can accomplish step 1 if the driver is already present in the product.

Open serial port accomplishes step 2.

3.3.1 PTZ control

To control the Pan, Tilt and Zoom behavior of a PTZ unit, the following PTZ control URL is used. This URL has view access rights. Important:

Some PTZ units automatically reduce pan and tilt movements as the zoom factor increases. Therefore, the actual movement may be less than what is requested of these units.

The PTZ control is device-dependent. For information about supported parameters and actual parameter values, please check the specification of the PTZ driver you intend to use. The following table is only an overview.

Note: The URL must follow the standard way of writing a URL, (RFC 2396: Uniform Resource Identifiers (URI) Generic Syntax); that is, 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 will have to be replaced with %20, My%20camera.

Method: GET/POST

Syntax:

```
http://<servername>/com/ptz.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
center=<int>,<int>	<x>,<y>	Absolute: Used to send the coordinates for the point in

		the image where the user clicked. This information is then used by the server to calculate the pan/tilt move required to (approximately) center the clicked point. Relative: Used to send the coordinates for the point in the image where the user clicked. This information is then used by the server to calculate the direction and number of degrees to move. The number of degrees increases with the distance from the center of the image to the point clicked. Digital: Used to send the coordinates for the point in the image where the user clicked. This information is then used by the server to center the clicked point.
imagewidth=<int>	1, ...	Required in conjunction with center if the image width displayed is different from the default size of the image, which is product-specific.
imageheight=<int>	1, ...	Needed in conjunction with center if the image height is different from the default size, which is product-specific.
move=<string>	home,up,down,left,right,upleft,upright,downleft,downright	Absolute: Moves the device 5 degrees in the specified direction. Relative: Moves the device approx. 50-90 degrees2 in the specified direction. Digital: Moves the image 25% of the image field width in the specified direction. Note: home is only valid if any home position has been previously set with "home=yes".
pan=<float>	-180.0 - 180.0	Absolute: Pans the device relative to the (0,0) position. Relative: n/a Digital: Pans the device relative to the (0,0) position.
tilt	-180.0 - 180.0	Absolute: Tilts the device relative to the (0,0) position. Relative: n/a Digital: Tilts the device relative to the (0,0) position.
rpan=<float>	-360.0 - 360.0	Absolute: Pans the device n degrees relative to the current position. Relative: Pans the device approx. n degrees relative to the current position. Digital: Pans the device n degrees relative to the current position.
rtilt=<float>	-360.0 - 360.0	Absolute: Tilts the device n degrees relative to the current position. Relative: Tilts the device approx. n degrees relative to the current position. Digital: Tilts the device n degrees relative to the current position.
rzoom=<int>	-9999 - 9999	Absolute: Zooms the device n steps relative to the current position. Positive values mean zoom in, negative values mean zoom out. Relative: Zooms the device approx. n steps relative to the current position. Positive values mean zoom in, negative values mean zoom out. Digital: Zooms the device n steps relative to the current

		position. Positive values mean zoom in, negative values mean zoom out.
rfocus=<int>	-9999 - 9999	Absolute: Move Focus n steps relative to the current position. Positive values mean focus far, negative values mean focus near. Relative: Move Focus approx. n steps relative to the current position. Positive values mean focus far, negative values mean focus near. Digital: n/a
riris=<int>	-9999 - 9999	Absolute: Move IRIS n steps relative to the current position.
autofocus=<string>	on, off	Autofocus On/Off?.
autoiris=<string>	on, off	Autoiris On/Off?.
continuouspantiltmove=<int>,<int>	-100 - 100,-100 - 100	Continuous pan/tilt motion. Positive values mean right (pan) and up (tilt), negative values mean left (pan) and down (tilt). "0,0" means stop. Values as <pan speed>,<tilt speed>
continuouszoommove=<int>	-100 - 100	Continuous zoom motion. Positive values mean zoom in and negative values mean zoom out. "0" means stop.
continuousfocusmove=<int>	-100 - 100	Continuous focus motion. Positive values mean focus near and negative values mean focus far. "0" means stop. Digital: n/a
continuousirismove=<int>	-100 - 100	Continuous iris motion. Positive values mean iris open and negative values mean iris close. "0" means stop. Digital: n/a
gotoserverpresetname=<string>	<preset name>	Move to the position associated with the <preset name>.
gotoserverpresetno=<int>	1, ...	Move to the position associated with the specified preset position number.
speed=<int>	1 - 100	Sets the head speed of the device that is connected to the specified camera. Digital: n/a
imagerotation=<int>	0, 90, 180, 270	If PTZ command refers to an image stream that is rotated differently than the current image setup, then the image stream rotation must be added to each command with this parameter to allow the server to compensate.
autopan=<string>	on, off	Autopan On/Off?.
tour=<string>	on, off	tour On/Off?.
tournum=<string>		tour index. Must used with tour.
vertical_reverse=<string>	1, 0	camera image vertical reverse 1/0.
horizontal_reverse=<string>	1, 0	camera image horizontal reverse 1/0.
rawdata=<string>	<string>	Raw data send to 485 bus. For example: ff,id,00,08,00,00,checksum
interface=<int>	0	interface 0 (RS485) or 1 (RS232 / Motor Lens)

3.3.2 PTZ configuration and OSD control

Configure PTZ preset positions and On Screen Display (OSD) control.

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

Method: GET/POST

Syntax:

```
http://<servername>/com/configptz.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

Alias Syntax:

```
http://<servername>/com/ptzconfig.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
setserverpresetname=<string>	<preset name>	Associates the current position to <preset name> as a preset position in the server.
setserverpresetno=<int>	1, ...	Saves the current position as a preset position number in the server.
settourpreset=<int>:<int>_<int>,<int>_<int>,...	<tour index>:<preset number>_<waiting time>,<preset number>_<waiting time>,...	Add presets to tour.
osdmenu=<string>	open,close,up,down,left,right,select,back	Commands to control the OSD menu in the camera. Note that the support of the different commands, and the behavior of the commands, is driver dependant.
removeserverpresetname=<string>	<preset name>	Removes the specified preset position associated with <preset name>.
removeserverpresetno=<int>	1, ...	Removes the specified preset position.

3.4 Motion Detection

To be able to define Motion Detection parameters, the video product must have built-in Motion Detection. A motion detection window is defined by several parameters. The motion detection parameters reside within a dynamic parameter group. Accordingly it is possible to add, remove, list and update the motion detection parameters with param.cgi, The dynamic motion detection parameter groups are divided into sub groups of the main motion parameter group, i.e. Motion.M<group number>.<parameter name>. group number is a unique number which is stated when a new dynamic parameter group is created, i.e. Motion.M3.

3.4.1 Add a Motion Detection window

When adding a Motion Detection window, the template file motion is used. The group number should be excluded when adding a new Motion Detection window with specified values since the group number will be defined when the new dynamic group is created.

Example: Add a new Motion Detection window with default values.

```
http://myserver/config.cgi?action=add&group=Motion&template=motion
```

Alias:

```
http://myserver/param.cgi?action=add&group=Motion&template=motion
```

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

```
http://myserver/config.cgi?
action=add&group=Motion&template=motion&Motion.M.Name=Entrance&Motion.M.Top=500&
```

```
    Motion.M.Bottom=7000&Motion.M.Left=5000&Motion.M.Right=8500
```

Alias:

```
http://myserver/param.cgi?
action=add&group=Motion&template=motion&Motion.M.Name=Entrance&Motion.M.Top=500&
    Motion.M.Bottom=7000&Motion.M.Left=5000&Motion.M.Right=8500
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
M<group number> OK\r\n
```

3.4.2 Remove a Motion Detection window

Example: Remove Motion Detection window defined within Motion.M3 and Motion.M5.

```
http://myserver/config.cgi?action=remove&group=Motion.M3,Motion.M5
```

Alias:

```
http://myserver/param.cgi?action=remove&group=Motion.M3,Motion.M5
```

3.4.3 Update the Motion Detection parameters

Example: Update the parameters for an existing Motion Detection window.

```
http://myserver/config.cgi?action=update&Motion.M1.Top=1500
&Motion.M1.Bottom=8000
```

Alias:

```
http://myserver/param.cgi?action=update&Motion.M1.Top=1500
&Motion.M1.Bottom=8000
```

3.4.4 List the Motion detection parameters

Example: List the Motion.M1 and Motion.M2 parameters.

```
http://myserver/config.cgi?action=list&group=Motion.M1,Motion.M2
```

Alias:

```
http://myserver/param.cgi?action=list&group=Motion.M1,Motion.M2
```

Example: List all Motion Detection windows.

```
http://myserver/config.cgi?action=list&group=Motion
```

Alias:

```
http://myserver/param.cgi?action=list&group=Motion
```

3.4.5 Get the Motion Detection level

It is possible to get the current Motion Detection levels from certain Motion Detection windows or from all MD windows, except from those windows that are defined to be Motion Detection exclude windows. It is also possible to define a Motion Detection window configuration and get the related

motion levels in return. The URL stated below is used.

Method: GET/POST

Syntax:

```
http://<servername>/motion/motion.cgi[?<parameter>=<value>...]
```

Alias:

```
http://<servername>/motion/motiondata.cgi[?<parameter>=<value>...]
```

with the following parameter and value

<parameter>=<value>	Values	Description
group=<int>[,<int>, ...]	<group number>[,<group number>, ...]	Specify the Motion Detection windows that are of interest. Excluding the group parameter will return all Motion Detection level information from all Motion Detection windows. Exclude windows are ignored.

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=<boundary>\r\n
\r\n
--<boundary>\r\n
<motion levels>
where the proposed boundary <boundary> is ipcammdb
and the <motion levels> part is
Content-Type: text/plain\r\n
\r\n
<motion level for window with lowest group number>
--<boundary>\r\n
and <motion level for window with group number n>" is
group=<group number n>;level=<motion level for n>;threshold=
<threshold level for n>;\r\n[ <motion level for window n+1> ]
```

Example: Get Motion Detection levels related to Motion Detection windows defined within Motion.M0 and Motion.M1.

```
http://myserver/motion/motion.cgi?group=0,1
```

Alias:

```
http://myserver/motion/motiondata.cgi?group=0,1
```

Return: The example above returns the following.

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=ipcammdb\r\n\r\n
--ipcammdb\r\n
Content-Type:text/plain\r\n\r\n
group=0;level=28;threshold=45;\r\n
group=1;level=43;threshold=25;\r\n
--ipcammdb\r\n
Content-Type:text/plain\r\n\r\n
group=0;level=54;threshold=45;\r\n
group=1;level=38;threshold=25;\r\n
--ipcammdb\r\n
```

```
Content-Type:text/plain\r\n\r\n
group=0;level=49;threshold=45;\r\n
group=1;level=19;threshold=25;\r\n
--ipcammdb\r\n
 .
 .
 .
```

1.If no Motion Detection windows are defined or only exclude windows are defined, HTTP/1.0 204 No Content is returned.
2.If any errors are found in the CGI request, HTTP/1.0 400 Bad Request is returned.
3.If too many clients try to get motion data, HTTP/1.0 503 Service Unavailable is returned.

3.5 I/O

The requests specified in the I/O section are supported by those products that have Input/Output? connectors.

3.5.1 I/O control

Method: GET

Syntax:

```
http://<servername>/io/port.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
check=<int>[,<int>, ...]	<id1>[,<id2>, ...]	Returns the status (1 or 0) of one or more inputs numbered id1 ,id2, ...
checkactive=<int>[, <int>, ...]	<id1>[,<id2>, ...]	Returns the status (active or inactive) of one or more inputs numbered id1,id2, ...
checkdirection	id1, [id2,..]	Returns the direction (input or output) of one or more ports numbered id1, id2, …
monitor=<int>[,<int>, ...]	<id1>[,<id2>, ...]	Returns a multipart stream of "check" inputs (see return description below).
action	[id]:<a>[,<wait><a>…]	Valid only for output ports. <id> = Port number, output1 is default <a> = action character: / or \ /=active, \=inactive <wait> = delay in milliseconds

3.5.2 Input

Method: GET

Syntax:

```
http://<servername>/io/input.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
check=(<int>|<string>) [,<int>, ...]	(<id1>|pir) [,<id2>, ...]	Returns the status (1 or 0) of inputs numbered id1 ,id2, ..., or PIR device
checkactive=(<int>| <string>)[,<int>, ...]	(<id1>|pir) [,<id2>, ...]	Returns the status (active or inactive) of inputs numbered id1,id2, ..., or PIR device

monitor=(<int>|<string>) [,<int>, ...]	(<id1>|pir) [,<id2>, ...]	Returns a multipart stream of "check" inputs or PIR device (see return description below).

Return: "monitor", i.e., multipart "check" parameter

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=<boundary>\r\n
\r\n
--<boundary>\r\n
<monitor data>
where the proposed boundary <boundary> is
ioboundary
and the <monitor data> part is
Content-Type: text/plain\r\n
\r\n
<check data>
--<boundary>\r\n
and <check data> is
IO<n>:<char>\r\n
and <n> is the I/O port number and <char> is / or H when the port is active and \ or L when the
port is inactive.
Note: The output can contain extra blank lines, i.e., extra \r\n within the sections.
```

Example: Monitor data on input ports 1, 2, 3, and 4.

```
http://myserver/io/input.cgi?monitor=1,2,3,4
```

Example: Return monitor data on input port 1.

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace; boundary=ioboundary\r\n
\r\n
\r\n
\r\n
\r\n
--ioboundary\r\n
Content-Type: text/plain\r\n
\r\n
IO0:/\n
\r\n
\r\n
--ioboundary\r\n
Content-Type: text/plain\r\n
\r\n
IO0:H\n
\r\n
--ioboundary\r\n
Content-Type: text/plain\r\n
\r\n
\r\n
IO0:\\n
\r\n
\r\n
--ioboundary\r\n
Content-Type: text/plain\r\n
\r\n
\r\n
\r\n
```

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

3.5.3 Output

Method: GET

Syntax:

```
http://<servername>/io/output.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameters and values

<parameter>=<value>	Values	Description
check=<int>[,<int>, ...]	<id1>[,<id2>, ...]	Returns the status (1 or 0) of one or more outputs numbered id1 ,id2, ...
checkactive=<int>[,<int>, ...]	<id1>[,<id2>, ...]	Returns the status (active or inactive) of one or more outputs numbered id1 ,id2, ...
monitor=<int>[,<int>, ...]	<id1>[,<id2>, ...]	Returns a multipart stream of "check" outputs (see return description below).
action=<string>	[<id>1]:<a>[<wait> <a> ...]	Sets the output relay <id> active or inactive and waits <wait> milliseconds. Note that only one output relay can be activated/deactivated per request. <id> = Output number. If omitted, output 1 is selected. <a> = Action character: / or \ / = active, \ = inactive. <wait> = Delay in milliseconds.

Example: Set output 1 active.

```
http://myserver/io/output.cgi?action=1:/
```

Example: Set two 300 ms pulses with 500 ms delay between the pulses on output 1.

```
http://myserver/io/output.cgi?action=1:/300\500/300\
```

Example: Wait 1 second before setting output 1 active.

```
http://myserver/io/output.cgi?action=1:1000/
```

3.6 IP filter

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

3.6.1 IP filter administration

Allow or deny the listed IP addresses to access the device.

Method: GET

Syntax:

```
http://<servername>/ipfilter.cgi?<parameter>=<value>[&<parameter>=<value>... ]
```

with the following parameters and values

<parameter>=<value>	Values	Description
action=<string>	add, remove, removeall, update or 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. ipaddress=<string>[%20<string>...] <IP addresses> The addresses allowed passing through the filter. A space separated list of IP addresses and network addresses in the CIDR notation (IP address/netmask bits). Note: If accessing the device via a proxy server, the proxy server's IP address must be added to the list of allowed addresses.
enable=<string>	yes,no	Enable/disable the IP address filtering function.
policy=<string>	allow,deny	Allow or deny access for the addresses in the list. If omitting this parameter the policy will be allow by default. Note: The policy used will apply for all the addresses in the list‖

Example: Add a list of IP addresses and enable the IP address filtering function. Verification that the device is still accessible will automatically be done.

```
http://myserver/ipfilter.cgi?action=add
&ipaddress=10.13.10.12%2010.13.17.0/24&enable=yes
```

Example: List settings for the IP address filtering function.

```
http://myserver/ipfilter.cgi?action=list
```

Example: Remove an entry in the list of addresses. Verification will automatically be done if the IP filter function is enabled.

```
http://myserver/ipfilter.cgi?action=remove&ipaddress=10.13.10.12
```

Example: Add 10.13.10.12 to the list of addresses which will be allowed access to the device.

```
http://myserver/ipfilter.cgi?action=add&ipaddress=10.13.10.12
&policy=allow&enable=yes
```

Example: Add 10.13.10.12 to the list of addresses which will be denied access to the device.

```
http://myserver/ipfilter.cgi?action=add&ipaddress=10.13.10.12
&policy=deny&enable=yes
```

Example: Remove all IP addresses and automatically disable the IP address filtering function

```
http://myserver/ipfilter.cgi?action=removeall
```

3.6.2 Server responses

Return: A successful add, remove, removeall or update.

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

Return: A list.

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
Accept addresses: <IP addresses>\r\n
Enabled: <yes/no>\r\n
```

3.7 Audio

The requests specified in the Audio section are supported by products that have audio capability.

3.7.1 Audio MIME types

Supported MIME types for audio

audio/raw	16bit/s PCM
audio/basic	which is G.711 ulaw 64kbit/s
audio/32KADPCM	which is G.726 32kbit/s
audio/G723	which is G.726 24kbit/s

3.7.2 Audio data request

Request an audio stream.

Method: GET

Syntax:

```
http://<servername>/audio/fromserver.cgi[?<parameter>=<value>]
```

Alias:

```
http://<servername>/audio/receive.cgi[?<parameter>=<value>]
```

with the following parameters and values

<parameter>=<value>	Values	Description
httptype=<string>	singlepart,multipart	Choose streaming method. Default value is defined by the parameter Audiod.HttpMessageType

Example: Request a singlepart audio stream

```
http://myserver/audio/fromserver.cgi?httptype=singlepart
```

Alias:

```
http://myserver/audio/receive.cgi?httptype=singlepart
```

3.7.3 Singlepart audio data response

When an audio stream is requested/transmitted, the server returns/receives a continuous flow of audio packets. The content type is only set at the beginning of the connection. When the connection is up and running the audio packets will come right after another without any extra information between the packets. The message body contains a block of binary data. Each block

of coded audio data is 240 byte.

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: <audio MIME>\r\n
\r\n
<Audio data>
```

Example: Singlepart Audio data encoded with G.711 ulaw.

```
HTTP/1.0 200 OK\r\n
Content-Type: audio/basic\r\n
\r\n
<Audio data>
<Audio data>
<Audio data>
.
.
.
```

3.7.4 Multipart audio data response

When an audio stream is requested/transmitted, the server returns/receives a continuous flow of audio packets. The content type is "mutipart/x-mixed-replace" and each audio packet ends with a boundary string. The message body contains a block of binary data. Each block of coded audio data is 240 byte.

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace; boundary=--<boundary>\r\n
\r\n
--<boundary>\r\n
<audio>
where the proposed <boundary> is:
myboundary
and the <audio> field is
Content-Type: <audio MIME>\r\n
\r\n
<Audio data>\r\n
--<boundary>\r\n
<audio>
```

Example: Multipart Audio data encoded with G.726 32kbit/s (G.721).

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=myboundary\r\n
\r\n
--myboundary\r\n
Content-Type: audio/32KADPCM\r\n
\r\n
<Audio data>\r\n
--myboundary\r\n
Content-Type: audio/32KADPCM\r\n
\r\n
<Audio data>\r\n
--myboundary\r\n
Content-Type: audio/32KADPCM\r\n
<Audio data>\r\n
```

```
--myboundary\r\n
Content-Type: audio/32KADPCM\r\n
\r\n
<Audio data>\r\n
--myboundary\r\n
.
.
.
```

3.7.5 Audio data transmit

Transmit a Singlepart/Multipart? Audio data stream. Each block of coded audio data is 240 byte.

Method: POST

Syntax:

```
http://<servername>/audio/toserver.cgi
```

Alias:

```
http://<servername>/audio/transmit.cgi
```

There are no valid parameters and values.

Example 1: Singlepart audio data transmit with G.711 ulaw (authorization omitted)

```
POST /audio/transmit.cgi HTTP/1.0\r\n
Content-Type: audio/basic\r\n
Content-Length: 9999999\r\n
Connection: Keep-Alive\r\n
Cache-Control: no-cache\r\n
\r\n
<Audio data>
<Audio data>
<Audio data>
.
.
.
```

Example 2: Multipart audio data transmit with G.711 ulaw (authorization omitted)

```
POST /audio/transmit.cgi HTTP/1.0\r\n
Content-Type: multipart/x-mixed-replace; boundary=--myboundary\r\n
Content-Length: 9999999\r\n
Connection: Keep-Alive\r\n
Cache-Control: no-cache\r\n
\r\n
--myboundary\r\n
Content-Type: audio/basic\r\n
\r\n
<Audio data>\r\n
--myboundary\r\n
Content-Type: audio/basic\r\n
\r\n
<Audio data>
--myboundary\r\n
Content-Type: audio/basic\r\n
\r\n
<Audio data>
```

```
--myboundary\r\n
Content-Type: audio/basic\r\n
\r\n
.
.
.
```

3.8 Storage

The requests specified in the storage section are supported by products that have micro SD socket.

3.8.1 mount storage

mount / umount the storage

Method: GET/POST

Syntax:

```
http://<servername>/mount.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
diskid=<string>	SD_DISK	specified the storage device
action=<string>	mount, umount	specified the action to mount/umount storage device

3.8.2 format storage

format the storage

Syntax:

```
http://<servername>/format.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
diskid=<string>	SD_DISK	specified the storage device

3.8.3 list storage status

list the storage status

Syntax:

```
http://<servername>/list.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
diskid=<string>	SD_DISK, all	specified the storage device

return value

```
<?xml version="1.0" ?>
  <root>
  <disks numberofdisks="1">
  <disk diskid="SD_DISK" totalsize="124048" freesize="124032" cleanuplevel="95" cleanupmaxage="1"
cleanuppolicy="none"
   locked="no" full="no" readonly="no" status="OK" />
  </disks>
  </root>
```

3.8.4 list recording files

list the recording files

Syntax:

```
http://<servername>/list.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
recordingid=<string>	all	specified the recording file

return value

```
    <?xml version="1.0" ?>
      <root>
      <recordings>
      <recording recordingid="20090101_010101.avi" diskid="SD_DISK" starttime="20090101_010101"
date="20090101" recordingtype="scheduled">
      </recording>
      <recording recordingid="20090101_010201.avi" diskid="SD_DISK" starttime="20090101_010201"
date="20090101" recordingtype="scheduled">
      </recording>
      </recordings>
```

3.8.5 play recording file

play the recording file

Syntax:

```
http://<servername>/play.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
recordingid=<string>	string	specified the recording file

3.8.6 remove recording file

remove the recording file

Syntax:

```
http://<servername>/remove.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
recordingid=<string>	string	specified the recording file

3.8.7 list continuous recording files

list the continuous recording files

Syntax:

```
http://<servername>/listcontinuous.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
recordingid=<string>	all	specified the recording file

return value

```
    <?xml version="1.0" ?>
      <root>
      <recordings>
      <recording recordingid="20090101_010101.avi" diskid="SD_DISK" starttime="20090101_010101"
date="20090101" recordingtype="continuous">
      </recording>
      <recording recordingid="20090101_010201.avi" diskid="SD_DISK" starttime="20090101_010201"
date="20090101" recordingtype="continuous">
      </recording>
      </recordings>
```

3.8.8 play continuous recording file

play the continuous recording file

Syntax:

```
http://<servername>/playcontinuous.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
recordingid=<string>	string	specified the recording file

3.8.9 remove continuous recording file

remove the continuous recording file

Syntax:

```
http://<servername>/removecontinuous.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
recordingid=<string>	string	specified the recording file

3.9 SAMBA Server

3.9.1 Set SAMBA Sever

Set SAMBA Server

Method: GET/POST

Syntax:

```
http://<servername>/config.cgi?action=update&<parameter>=<value>
```

with the following parameter and values

<parameter>=<value>	Values	Description
Storage.S1.Address=<string>	string	SAMBA Server hostname or IP Address
Storage.S1.Path=<string>	string	The mounted path
Storage.S1.Login=<string>	string	Username of SAMBA Server
Storage.S1.Password=<string>	string	Password of SAMBA Server

3.9.2 mount storage

mount / umount the SAMBA Server

Method: GET/POST

Syntax:

```
http://<servername>/mount.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
diskid=<string>	SMB_DISK	specified the storage device
action=<string>	mount, umount	specified the action to mount/umount SAMBA Server

3.9.3 list storage status

list the storage status

Syntax:

```
http://<servername>/list.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
diskid=<string>	NetworkShare	specified the storage device

return value

```
<?xml version="1.0" ?>
  <root>
  <disks numberofdisks="1">
  <disk diskid="SMB_DISK" totalsize="124048" freesize="124032" cleanuplevel="95" cleanupmaxage="1"
cleanuppolicy="none"
   locked="no" full="no" readonly="no" status="OK" />
  </disks>
  </root>
```

3.9.4 list continuous recording files

list the continuous recording files

Syntax:

```
http://<servername>/listcontinuous.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
recordingid=<string>	all	specified the recording file

return value

```
    <?xml version="1.0" ?>
      <root>
      <recordings>
      <recording recordingid="20090101_010101.avi" diskid="SMB_DISK" starttime="20090101_010101"
date="20090101" recordingtype="continuous">
      </recording>
      <recording recordingid="20090101_010201.avi" diskid="SMB_DISK" starttime="20090101_010201"
date="20090101" recordingtype="continuous">
      </recording>
      </recordings>
```

3.9.5 play continuous recording file

play the continuous recording file

Syntax:

```
http://<servername>/playcontinuous.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
recordingid=<string>	string	specified the recording file

3.9.6 remove continuous recording file

remove the continuous recording file

Syntax:

```
http://<servername>/removecontinuous.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
recordingid=<string>	string	specified the recording file

3.10 Continuous Recording

3.10.1 Set Continuous Recording

Set Continuous Recording

Method: GET/POST

Syntax:

```
http://<servername>/config.cgi?action=update&<parameter>=<value>
```

with the following parameter and values

<parameter>=<value>	Values	Description
ContinuousRecording.Enabled=<string>	yes,no	Enabled/Disable? continuous recording
ContinuousRecording.C1.ProfileNo=<integer>	1~max no. of profile	The profile number
ContinuousRecording.C1.RecordPath=<string>	/mnt/sd/<path>, /mnt/smb/<path>	Record path of SAMBA server or SD card

3.11 Fisheye Control

3.11.1 Fisheye Control

Fisheye control

Method: GET/POST

Syntax:

```
http://<servername>/fisheyectrl.cgi?<parameter>=<value>
```

with the following parameter and values

<parameter>=<value>	Values	Description
Window=<int>	0~3	Assign window number

Position=<string>	Ceil,Wall,Desk,Wide_Panorama	Refer parameter Fisheye.SupportPosition
DisplayMode=<string>	1O4R...	Refer parameter Fisheye.SupportDisplayMode
Move=<string>	home,up,down,left,right, upleft,upright,downleft,downright tele,wide	Need to assign window number
gotopresetname=<string>	preset name	Goto preset by name.
gotopresetno=<int>	preset number	Goto preset by number.
setpresetname=<string>	preset name	Set preset by name.
setpresetno=<int>	preset number	Set preset number.
removepresetname=<string>	preset name	Remove preset by name.
removepresetno=<int>	preset number	Remove preset by number.
Speed=<int>	1-100	Set the speed of the move function.

3.12 Force Trigger

3.12.1 Force Trigger

Force trigger

Method: GET/POST

Syntax:

```
http://<servername>/trigger.cgi?<parameter>=<value>
```

with the following parameter and values

<parameter>=<value>	Values	Description
Event=E<int>	0 ~ max event number	Event index
Prefix=<string>	length: 48 characters, Character: 0-9, a-z, A-Z, _, -	Record File Prefix
TriggerTime=<string>	time	Event trigger time.
TriggerServer=<string>	S3	Used for S3 update.
Duration=<int>	120	Duration for S3 update file.

3.13 Apple Push Notification Service (APNS)

Force trigger

Method: GET/POST

Syntax:

```
http://<apns_server_path>?<parameter>=<value>
```

apns_server_path: for nuuolink

```
http://sat-event.nuuolink.com/c2dm/push.php
```

with the following parameter and values

<parameter>=<value>	Values	Description	
Parameter	Note	Format	Example
dipsid	DIPS ID	[0 - 9]	000024097
deviceuid	SAT UID*	[A - Z][0 - 9][-]	A01CC-000EAEA24097
devicemac	MAC Address	[A - Z][0 - 9]	000EAEA24097
dpath	Recording File Download Path	URL Sub-path	/eventdownload.cgi?eventid=20120417_034037.avi
lpath	Recording File List Path	URL Sub-path	/eventlist.cgi?eventid=all
eventid	Recording File ID (or File Name)	[A - Z,a - z][0 - 9][-][_]	20120417_034037.avi
size	Recording File Size	[0 - 9][A - Z]	878KB
eventtype	Recording Type	[A - Z,a - z][0 - 9]	motion0Occur ~ motion9Occur, schedule, DI Change Enabled, DI Change Disabled, PIRStatus Change Disabled, PIRStatus Change Enabled
starttime	Event Start Time	[0 - 9], hhmmss	184023
date	Event Date	[0 - 9], yyyyMMdd	20120425

3.14 Audio Detection

3.14.1 Get Audio Alarm Level

Get Audio Alram Level

Method: GET/POST

Syntax:

```
http://<servername>/config.cgi?action=list&group=AudioSource.A0.AlarmLevel
```

Alias:

```
http://<servername>/param.cgi?action=list&group=AudioSource.A0.AlarmLevel
```

3.14.2 Set Audio Alarm Level

Set Audio Alram Level

Method: GET/POST

Syntax:

```
http://<servername>/config.cgi?action=update&<parameter>=<value>
```

Alias:

```
http://<servername>/param.cgi?action=update&<parameter>=<value>
```

with the following parameter and values

<parameter>=<value>	Values	Description
AudioSource.A0.AlarmLevel=<integer>	0~100	The audio alarm level

3.14.3 Get The Audio Detection Level

Get the current audio detection levels

Method: GET/POST

Syntax:

```
http://<servername>/audiodetection.cgi
```

Alias:

```
http://<servername>/audiodetectiondata.cgi
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=ipcamaddb\r\n\r\n
--ipcamaddb \r\n
Content-Type: text/plain \r\n\r\n
level=2;threshold=50; \r\n
--ipcamaddb \r\n
Content-Type: text/plain \r\n\r\n
level=38;threshold=50; \r\n
```

3.15 Face Detection

3.15.1 Add a Face Detection Window

When adding a Face Detection window, the template file FaceDetection is used. The group number should be excluded when adding a new Face Detection window with specified values since the group number will be defined when the new dynamic group is created.

Example: Add a new Face Detection window with default values.

```
http://myserver/config.cgi?action=add&group=FaceDetection&template=FaceDetection
```

Alias:

```
http://myserver/param.cgi?action=add&group=FaceDetection&template=FaceDetection
```

Example: Add a new Face Detection window with specified values.

```
http://myserver/config.cgi?
action=add&group=FaceDetection&template=FaceDetection&FaceDetection.F.Name=Win&FaceDetection.F.Top
=500&
    FaceDetection.F.Bottom=7000&FaceDetection.F.Left=5000&FaceDetection.F.Right=8500
```

Alias:

```
http://myserver/param.cgi?
action=add&group=FaceDetection&template=FaceDetection&FaceDetection.F.Name=Win&FaceDetection.F.Top
=500&
    FaceDetection.F.Bottom=7000&FaceDetection.F.Left=5000&FaceDetection.F.Right=8500
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
F<group number> OK\r\n
```

3.15.2 Remove a Face Detection window

Example: Remove Face Detection window defined within FaceDetection.F3 and FaceDetection.F5.

```
http://myserver/config.cgi?action=remove&group=FaceDetection.F3,FaceDetection.F5
```

Alias:

```
http://myserver/param.cgi?action=remove&group=FaceDetection.F3,FaceDetection.F5
```

3.15.3 Update the Face Detection Window parameters

Example: Update the parameters for an existing Face Detection window.

```
http://myserver/config.cgi?action=update&FaceDetection.F1.Top=1500
&FaceDetection.F1.Bottom=8000
```

Alias:

```
http://myserver/param.cgi?action=update&FaceDetection.F1.Top=1500
&FaceDetection.F1.Bottom=8000
```

3.15.4 List the Face detection parameters

Example: List the FaceDetection.F1 and FaceDetection.F2 parameters.

```
http://myserver/config.cgi?action=list&group=FaceDetection.F1,FaceDetection.F2
```

Alias:

```
http://myserver/param.cgi?action=list&group=FaceDetection.F1,FaceDetection.F2
```

Example: List all Face Detection windows.

```
http://myserver/config.cgi?action=list&group=FaceDetection
```

Alias:

```
http://myserver/param.cgi?action=list&group=FaceDetection
```

3.15.5 Get Face Detection Level

Get face detection data

Method: GET/POST

Syntax:

```
http://<servername>/facedetection.cgi
```

Return Syntax:

```
id=0;start=(50x50);end=(100x300);
```

with the following parameter and values

<parameter>=<value>	Values	Description
id	0~10	The face id
start	(PosXxPosY)	The face start position
end	(PosXxPosY)	The face end position

Return Example:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=ipcamaddb\r\n\r\n
--ipcamaddb \r\n
Content-Type: text/plain \r\n\r\n
id=0;start=(50x50);end=(100x300); \r\n
--ipcamaddb \r\n
Content-Type: text/plain \r\n\r\n
id=2;start=(50x50);end=(100x300); \r\n
```

3.16 Camera Tampering

3.16.1 Get minimum tampering period

Get minimum tampering period

Method: GET/POST

Syntax:

```
http://<servername>/config.cgi?action=list&group=CameraTampering.T0.MinDuration
```

Alias:

```
http://<servername>/param.cgi?action=list&group=CameraTampering.T0.MinDuration
```

3.16.2 Set minimum tampering period

Set minimum tampering period

Method: GET/POST

Syntax:

```
http://<servername>/config.cgi?action=update&<parameter>=<value>
```

Alias:

```
http://<servername>/param.cgi?action=update&<parameter>=<value>
```

with the following parameter and values

<parameter>=<value>	Values	Description
CameraTampering.T0.MinDuration=<integer>	1~60	The minimum duration

3.16.3 Get Camera Tampering Status

Get Camera Tampering Status

Method: GET/POST

Syntax:

```
http://<servername>/cameratampering.cgi
```

Return Syntax:

```
CameraTampering=active
```

with the following parameter and values

<parameter>=<value>	Values	Description
CameraTampering	active/inactive	Status of Camera Tampering

Return Example:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=ipcamaddb\r\n\r\n
--ipcamaddb \r\n
Content-Type: text/plain \r\n\r\n
CameraTampering=inactive \r\n
--ipcamaddb \r\n
Content-Type: text/plain \r\n\r\n
CameraTampering=active \r\n
```

3.17 Object Detection

3.17.1 Add a Object Detection Window

When adding a Object Detection window, the template file ObjectDetection is used. The group number should be excluded when adding a new Object Detection window with specified values since the group number will be defined when the new dynamic group is created.

Example: Add a new Object Detection window with default values.

```
http://myserver/config.cgi?action=add&group=ObjectDetection&template=ObjectDetection
```

Alias:

```
http://myserver/param.cgi?action=add&group=ObjectDetection&template=ObjectDetection
```

Example: Add a new Object Detection window with specified values.

```
http://myserver/config.cgi?
action=add&group=ObjectDetection&template=ObjectDetection&ObjectDetection.0.Name=Win&ObjectDetecti
on.0.Top=500&
    ObjectDetection.0.Bottom=7000&ObjectDetection.0.Left=5000&ObjectDetection.0.Right=8500
```

Alias:

```
http://myserver/param.cgi?
action=add&group=ObjectDetection&template=ObjectDetection&ObjectDetection.0.Name=Win&ObjectDetecti
on.0.Top=500&
    ObjectDetection.0.Bottom=7000&ObjectDetection.0.Left=5000&ObjectDetection.0.Right=8500
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
0<group number> OK\r\n
```

3.17.2 Remove a Object Detection window

Example: Remove Object Detection window defined within ObjectDetection.03 and ObjectDetection.05.

```
http://myserver/config.cgi?action=remove&group=ObjectDetection.03,ObjectDetection.05
```

Alias:

```
http://myserver/param.cgi?action=remove&group=ObjectDetection.03,ObjectDetection.05
```

3.17.3 Update the Object Detection Window parameters

Example: Update the parameters for an existing Object Detection window.

```
http://myserver/config.cgi?action=update&ObjectDetection.O1.Top=1500
&ObjectDetection.O1.Bottom=8000
```

Alias:

```
http://myserver/param.cgi?action=update&ObjectDetection.O1.Top=1500
&ObjectDetection.O1.Bottom=8000
```

3.17.4 List the Object detection parameters

Example: List the ObjectDetection.O1 and ObjectDetection.O2 parameters.

```
http://myserver/config.cgi?action=list&group=ObjectDetection.O1,ObjectDetection.O2
```

Alias:

```
http://myserver/param.cgi?action=list&group=ObjectDetection.O1,ObjectDetection.O2
```

Example: List all Object Detection windows.

```
http://myserver/config.cgi?action=list&group=ObjectDetection
```

Alias:

```
http://myserver/param.cgi?action=list&group=ObjectDetection
```

3.17.5 Get Object Detection Level

Get object detection data

Method: GET/POST

Syntax:

```
http://<servername>/objdetection.cgi
```

Return Syntax:

```
id=0;status=active;
```

with the following parameter and values

<parameter>=<value>	Values	Description
id	0~Properties.DynamicParam.MaxObjectDetection	The object detection window id
status	active/inactive	Status of Object Detection

Return Example:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=ipcamobjdb\r\n\r\n
--ipcamobjdb \r\n
Content-Type: text/plain \r\n\r\n
id=0;status=active;\r\n
--ipcamobjdb \r\n
Content-Type: text/plain \r\n\r\n
id=0;start=inactive;\r\n
```

3.18 Cross Line Detection

3.18.1 Add a line of Cross Line Detection

When adding a line of Cross Line Detection, the template file CrossLine is used. The group number should be excluded when adding a new line with specified values since the group number will be defined when the new dynamic group is created.

Example: Add a new line of Cross Line Detection with default values.

```
http://myserver/config.cgi?action=add&group=CrossLine&template=CrossLine
```

Alias:

```
http://myserver/param.cgi?action=add&group=CrossLine&template=CrossLine
```

Example: Add a new line of Cross Line Detection with specified values.

```
http://myserver/config.cgi?
action=add&group=CrossLine&template=CrossLine&CrossLine.C.Name=Line&CrossLine.C.StartX=1500&

CrossLine.C.StartY=100&CrossLine.C.EndX=1500&CrossLine.C.EndY=2500&CrossLine.C.Direction=1&CrossLi
ne.C.SlowSpeed=1&

CrossLine.C.FastSpeed=90&CrossLine.C.MinInspectionSize=1x1&CrossLine.C.MaxInspectionSize=1920x1080
```

Alias:

```
http://myserver/param.cgi?
action=add&group=CrossLine&template=CrossLine&CrossLine.C.Name=Line&CrossLine.C.StartX=1500&

CrossLine.C.StartY=100&CrossLine.C.EndX=1500&CrossLine.C.EndY=2500&CrossLine.C.Direction=1&CrossLi
ne.C.SlowSpeed=1&

CrossLine.C.FastSpeed=90&CrossLine.C.MinInspectionSize=1x1&CrossLine.C.MaxInspectionSize=1920x1080
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
C<group number> OK\r\n
```

3.18.2 Remove a line of Cross Line Detection

Example: Remove line of Cross Line Detection defined within CrossLine.C3 and CrossLine.C5.

```
http://myserver/config.cgi?action=remove&group=CrossLine.C3,CrossLine.C5
```

Alias:

```
http://myserver/param.cgi?action=remove&group=CrossLine.C3,CrossLine.C5
```

3.18.3 Update the line of Cross Line Detection parameters

Example: Update the parameters for an existing line of Cross Line Detection.

```
http://myserver/config.cgi?action=update&CrossLine.C1.StartX=1500
&CrossLine.C1.EndX=8000
```

Alias:

```
http://myserver/param.cgi?action=update&CrossLine.C1.StartX=1500
&CrossLine.C1.EndX=8000
```

3.18.4 List the Cross Line Detection parameters

Example: List the CrossLine.C1 and CrossLine.C2 parameters.

```
http://myserver/config.cgi?action=list&group=CrossLine.C1,CrossLine.C2
```

Alias:

```
http://myserver/param.cgi?action=list&group=CrossLine.C1,CrossLine.C2
```

Example: List all lines of Cross Line Detection.

```
http://myserver/config.cgi?action=list&group=CrossLine
```

Alias:

```
http://myserver/param.cgi?action=list&group=CrossLine
```

3.18.5 Get Cross Line Detection Level

Get cross line detection data

Method: GET/POST

Syntax:

```
http://<servername>/crosslinedetection.cgi
```

Return Syntax:

```
id=0;status=active;
```

with the following parameter and values

<parameter>=<value>	Values	Description
id	0~Properties.DynamicParam.MaxCrossLine	The id of line of Cross Line Detection
status	active/inactive	Status of Cross Line Detection

Return Example:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=ipcamcldb\r\n\r\n
--ipcamcldb \r\n
Content-Type: text/plain \r\n\r\n
id=0;status=active;\r\n
id=1;start=inactive;\r\n
--ipcamcldb \r\n
Content-Type: text/plain \r\n\r\n
id=0;status=inactive;\r\n
id=1;start=active;\r\n
```

3.19 VMD

3.19.1 Add a VMD Window

When adding a VMD window, the template file VMDWindow.Detect or VMDWindow.NonDetect is

used. The template file VMDWindow.Detect is used to add a detection window, and VMDWindow.NonDetect is used to add a non-detection window. The group number should be excluded when adding a new Object Detection window with specified values since the group number will be defined when the new dynamic group is created.

Example: Add a new detection VMD window with default values.

```
http://myserver/config.cgi?action=add&group=VMDWindow.Detect&template=VMDWindow.Detect
```

Alias:

```
http://myserver/param.cgi?action=add&group=VMDWindow.Detect&template=VMDWindow.Detect
```

Example: Add a new non-detection VMD window with default values.

```
http://myserver/config.cgi?action=add&group=VMDWindow.NonDetect&template=VMDWindow.NonDetect
```

Alias:

```
http://myserver/param.cgi?action=add&group=VMDWindow.NonDetect&template=VMDWindow.NonDetect
```

Example: Add a new detection VMD window with specified values.

```
http://myserver/config.cgi?
action=add&group=VMDWindow.Detect&template=VMDWindow.Detect&VMDWindow.Detect.V.Name=Win&VMDWindow.
Detect.V.Left=0&
VMDWindow.Detect.V.Right=9998&VMDWindow.Detect.V.Top=0&VMDWindow.Detect.V.Bottom=9998
```

Alias:

```
http://myserver/param.cgi?
action=add&group=VMDWindow.Detect&template=VMDWindow.Detect&VMDWindow.Detect.V.Name=Win&VMDWindow.
Detect.V.Left=0&
VMDWindow.Detect.V.Right=9998&VMDWindow.Detect.V.Top=0&VMDWindow.Detect.V.Bottom=9998
```

Example: Add a new non-detection VMD window with specified values.

```
http://myserver/config.cgi?
action=add&group=VMDWindow.NonDetect&template=VMDWindow.NonDetect&VMDWindow.NonDetect.V.Name=Win&V
MDWindow.NonDetect.V.Left=0&
VMDWindow.NonDetect.V.Right=9998&VMDWindow.NonDetect.V.Top=0&VMDWindow.NonDetect.V.Bottom=9998
```

Alias:

```
http://myserver/param.cgi?
action=add&group=VMDWindow.NonDetect&template=VMDWindow.NonDetect&VMDWindow.NonDetect.V.Name=Win&V
MDWindow.NonDetect.V.Left=0&
VMDWindow.NonDetect.V.Right=9998&VMDWindow.NonDetect.V.Top=0&VMDWindow.NonDetect.V.Bottom=9998
```

Return:

```
HTTP/1.0 200 OK\r\n
Content-Type: text/plain\r\n
\r\n
V<group number> OK\r\n
```

3.19.2 Remove a VMD window

Example: Remove VMD window defined within VMDWindow.Detect.V3 and VMDWindow.NonDetect.V5.

```
http://myserver/config.cgi?action=remove&group=VMDWindow.Detect.V3,VMDWindow.NonDetect.V5
```

Alias:

```
http://myserver/param.cgi?action=remove&group=VMDWindow.Detect.V3,VMDWindow.NonDetect.V5
```

3.19.3 Update the VMD Window parameters

Example: Update the parameters for an existing VMD window.

```
http://myserver/config.cgi?action=update&VMDWindow.Detect.V1.Top=1500
&VMDWindow.NonDetect.V1.Bottom=8000
```

Alias:

```
http://myserver/param.cgi?action=update&VMDWindow.Detect.V1.Top=1500
&VMDWindow.NonDetect.V1.Bottom=8000
```

3.19.4 List the VMD parameters

Example: List the VMDWindow.Detect.V1 and VMDWindow.NonDetect.V1 parameters.

```
http://myserver/config.cgi?action=list&group=VMDWindow.Detect.V1,VMDWindow.NonDetect.V1
```

Alias:

```
http://myserver/param.cgi?action=list&group=VMDWindow.Detect.V1,VMDWindow.NonDetect.V1
```

Example: List all VMD windows.

```
http://myserver/config.cgi?action=list&group=VMDWindow.Detect,VMDWindow.NonDetect
```

Alias:

```
http://myserver/param.cgi?action=list&group=VMDWindow.Detect,VMDWindow.NonDetect
```

3.19.5 Get VMD Level

Get VMD Window data

Method: GET/POST

Syntax:

```
http://<servername>/vmd.cgi
```

Return Syntax:

```
type=Dynamic;def =-;id=101;start=(100,10);end=(300,400);
type=Static;def =Desertion;id=11;start=(0,155);end=(100,255);
```

with the following parameter and values

<parameter>=<value>	Values	Description
type	Dynamic/Static?	The type of detection window
def	Desertion/Theft/?-	The defition of static window.
id	Integer	The detection window id
start	(PosX,PosY)	Start position of detection window
end	(PosX,PosY)	End position of detection window

Return Example:

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=ipcamvmddb\r\n\r\n
--ipcamvmddb \r\n
Content-Type: text/plain \r\n\r\n
type=Dynamic;def =-;id=101;start=(100,10);end=(300,400);\r\n
type=Static;def =Desertion;id=11;start=(0,155);end=(100,255);\r\n
--ipcamvmddb \r\n
Content-Type: text/plain \r\n\r\n
type=Dynamic;def =-;id=1050;start=(188,32);end=(777,211);\r\n
type=Static;def =Theft;id=2     1;start=(5,200);end=(17,400);\r\n
```

3.20 Snapshot CGI request

Request a snapshot with specified properties.

Method: GET

Syntax:

```
http://<servername>/snapshot.cgi[?<parameter>=<value>[&<parameter>=<value>...]]
```

with the following parameters and values

<parameter>=<value>	Values	Description
Resolution=<string>		Specify the resolution of the returned image.
Format=<string>	jpeg,h264	Specify the image format.
Recordfile=<string>	avi,jpg	Specify the file format.
UploadAddr		IP address of FTP server.
Username		FTP user name.
Password		FTP password.
Port		FTP port.
UploadPath		Directory where uploaded files go.
Passive		Use passive FTP.

Example: Request a image

```
http://myserver/snapshot.cgi?Resolution=1920x1080
```

3.21 CHT CGI request

CHT control.

Method: GET

Syntax:

```
http://<servername>/cht.cgi[?<parameter>=<value>[&<parameter>=<value>...]]
```

with the following parameters and values

<parameter>=<value>	Values	Description
OperateMode	none,event,fulltime	S3 upload type.
Host		S3 host name, "s3.hicloud.net.tw"
HostIpAddr?		S3 host ip address or domain name
AccessKey		S3 access key.
SecretKey		S3 secret key

Bucket		S3 cloud bucket.
ServiceNumber		S3 cloud folder - service number.
CameraNumber		S3 cloud folder - camera number
RecStartTime		Recording start time index (0 - 29)
LifeTime		S3 cloud file - LifeTime/ExpireDay

3.22 Autoupdate CGI request

Auto firmware upgrade control.

Method: GET

Syntax:

```
http://<servername>/autoupdate.cgi[?<parameter>=<value>[&<parameter>=<value>...]]
```

with the following parameters and values

<parameter>=<value>	Values	Description
fw_url	URL	Firmware path, for example : fw_url=http://168.168.9.89/AMTK-2014Dec8-6.S.0.15620-ipcam.pck

3.23 iSCSI

3.23.1 Set iSCSI

Set iSCSI

Method: GET/POST

Syntax:

```
http://<servername>/config.cgi?action=update&<parameter>=<value>
```

with the following parameter and values

<parameter>=<value>	Values	Description
Storage.S2.Address=<string>	string	iSCSI IP Address
Storage.S2.Login=<string>	string	Username of iSCSI
Storage.S2.Password=<string>	string	Password of iSCSI
Storage.S2.CleanupMaxAge=<int>	1~999	Remove recordings older than CleanupMaxAge days
Storage.S2.CleanupLevel=<int>	1~100	Remove oldest recordings when disk is CleanupLevel% full
Storage.S2.Locked=<string>	yes, no	Lock diskI

3.23.2 Connect to iSCSI

Connect to iSCSI to get Targets

Method: GET/POST

Syntax:

```
http://<servername>/iscsi.cgi?action=discover&<parameter>=<value>
```

with the following parameter and values

<parameter>=<value>	Values	Description
ip=<string>	string	iSCSI IP Address

return value

```
<?xml version="1.0"?>
<targets>
<target>iqn.2006-03.com.kernsafe:user-pc.SuperSAN0</target>
<target>iqn.2006-03.com.kernsafe:user-pc.ImageDisk2</target>
</targets>
```

3.23.3 Login to Target of iSCSI

Login to Target of iSCSI

Method: GET/POST

Syntax:

```
http://<servername>/iscsi.cgi?action=login&<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
ip=<string>	string	iSCSI IP Address
target=<string>	string	Target of iSCSI

return value

```
<?xml version="1.0"?>
<loginstatus>
<status>OK</status>
<storagestatus>Online</storagestatus>
</loginstatus>
```

3.23.4 Logout to Target of iSCSI

Logout to Target of iSCSI

Method: GET/POST

Syntax:

```
http://<servername>/iscsi.cgi?action=logout
```

return value

```
<?xml version="1.0"?>
<logoutstatus>
<status>OK</status>
<storagestatus>NoConnect</storagestatus>
</logoutstatus>
```

3.23.5 Get disk information of target

Get disk information of target

Method: GET/POST

Syntax:

```
http://<servername>/iscsi.cgi?action=getdisk
```

return value

```
<?xml version="1.0"?>
```

```
<disks>
<mountstatus>1</mountstatus>
<storagestatus>Mount</storagestatus>
<disk>
<name>/dev/sda</name>
<lun>LUN0</lun>
<size>3221225472</size>
<partitions>
<partition>
<name>/dev/sda1</name>
<totalsize>3218809856</totalsize>
</partition>
</partitions>
</disk>
<disk>
<name>/dev/sdb</name>
<lun>LUN1</lun>
<size>4294967296</size>
<partitions>
<partition>
<name>/dev/sdb1</name>
<mountpath>/mnt/iscsi</mountpath>
<totalsize>4226257408</totalsize>
<freesize>4011553280</freesize>
</partition>
</partitions>
</disk>
</disks>
```

3.23.6 Mount to partition of LUN

Mount to partition of LUN

Method: GET/POST

Syntax:

```
http://<servername>/iscsi.cgi?action=mount&<parameter>=<value>
```

with the following parameter and values

<parameter>=<value>	Values	Description
dev=<string>	string	Partition of LUN

return value

```
<?xml version="1.0"?>
<mountstatus>
<status>OK</status>
<storagestatus>Mount</storagestatus>
</mountstatus>
```

3.23.7 Umount from partition of LUN

Umount from partition of LUN

Method: GET/POST

Syntax:

```
http://<servername>/iscsi.cgi?action=unmount
```

return value

```
<?xml version="1.0"?>
<unmountstatus>
<status>OK</status>
<storagestatus>Online</storagestatus>
</unmountstatus>
```

3.23.8 list continuous recording files

list continuous recording files

Method: GET/POST

Syntax:

```
http://<servername>/listcontinuous.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
recordingid=<string>	all	specified the recording file
path=<string>	/mnt/iscsi	specified the path recording file

return value

```
<?xml version="1.0"?>
<root>
<recordings>
<recording timetag="1421133617" duration="301" tag="/mnt/iscsi/000EAEA26253/1421133617638.avi"
date="20150113" starttime="072017" recordingtype="continuous" size="50519 KB" diskid="iSCSI"
recordingid="20150113_072017.avi"/>
</recordings>
</root>
```

3.23.9 play continuous recording file

play continuous recording file

Method: GET/POST

Syntax:

```
http://<servername>/playcontinuous.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description
recordingid=<string>	all	specified the recording file
path=<string>	/mnt/iscsi	specified the path recording file

3.23.10 remove continuous recording file

remove continuous recording file

Method: GET/POST

Syntax:

```
http://<servername>/removecontinuous.cgi?<parameter>=<value>[&<parameter>=<value>...]
```

with the following parameter and values

<parameter>=<value>	Values	Description

recordingid=<string>	all	specified the recording file
path=<string>	/mnt/iscsi	specified the path recording file

3.24 IEEE 802.1X

3.24.1 Set IEEE 802.1X

Set IEEE 802.1X

Method: GET/POST

Syntax:

```
http://<servername>/config.cgi?action=update&<parameter>=<value>
```

with the following parameter and values

<parameter>=<value>	Values	Description
Network.8021X.Enabled=<string>	yes,no	Enable/Disable? 802.1X
Network.8021X.EAPType=<string>	TLS/LEAP	802.1X type
Network.8021X.EAPOLVer=<int>	1,2	802.1X version
Network.8021X.Username=<string>	string	Username of 802.1X
Network.8021X.Password=<string>	string	Password of 802.1X

3.24.2 Upload CA certificate

Upload CA certificate when Network.8021X.EAPType is TLS

Method: POST

Syntax:

```
http://<servername>/8021xconfig.cgi?action=upload&type=cacert
```

The file content is provided in the HTTP body according to the format given in RFC 1867. The body is created automatically by the browser if using HTML form with input type "file".

Example:

```
POST /asp/8021xconfig.cgi?action=upload&type=cacert HTTP/1.1\r\n
Accept: text/html, application/xhtml+xml, */*\r\n
Content-Type: multipart/form-data; boundary=---------------------------7df66351441260\r\n
Content-Length: 2011\r\n
\r\n
-----------------------------7df66351441260\r\n
Content-Disposition: form-data; name="file"; filename="ca.crt"\r\n
Content-Type: application/x-x509-ca-cert\r\n
\r\n
<CA file content>
\r\n
-----------------------------7df66351441260\r\n
```

3.24.3 Upload public client certificate

Upload public client certificate

Method: POST

Syntax:

```
http://<servername>/8021xconfig.cgi?action=upload&type=clientcert
```

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

body is created automatically by the browser if using HTML form with input type "file".

Example:

```
POST /asp/8021xconfig.cgi?action=upload&type=clientcert HTTP/1.1\r\n
Accept: text/html, application/xhtml+xml, */*\r\n
Content-Type: multipart/form-data; boundary=--------------------------7df1751a1441260\r\n
Content-Length: 5676\r\n
\r\n
----------------------------7df1751a1441260\r\n
Content-Disposition: form-data; name="file"; filename="client.crt"\r\n
Content-Type: application/x-x509-ca-cert\r\n
\r\n
<public client certificate file content>
\r\n
----------------------------7df1751a1441260\r\n
```

3.24.4 Upload private client key

Upload private client key

Method: POST

Syntax:

```
http://<servername>/8021xconfig.cgi?action=upload&type=clientkey
```

The file content is provided in the HTTP body according to the format given in RFC 1867. The body is created automatically by the browser if using HTML form with input type "file".

Example:

```
POST /asp/8021xconfig.cgi?action=upload&type=clientkey HTTP/1.1\r\n
Accept: text/html, application/xhtml+xml, */*\r\n
Content-Type: multipart/form-data; boundary=--------------------------7df3bf351441260\r\n
Content-Length: 2034\r\n
\r\n
----------------------------7df3bf351441260\r\n
Content-Disposition: form-data; name="file"; filename="client.key"\r\n
Content-Type: text/plain\r\n
\r\n
<private client key file content>
\r\n
----------------------------7df3bf351441260\r\n
```

Appendix: Alias summary

Group	Formal name	Alias name	Remark
General	config.cgi	param.cgi	
General	usrgrp.cgi	pwdgrp.cgi	
General	reboot.cgi	restart.cgi	
General	time.cgi	date.cgi	
Image & Video	resolution.cgi	imagesize,cgi	
PTZ	configptz.cgi	ptzconfig.cgi	
Motion	motion.cgi	motiondata.cgi	
Audio	fromserver.cgi	receive.cgi	
Audio	toserver.cgi	transmit.cgi	

Audio	audiodetection.cgi	audiodetectiondata.cgi	