IP Camera API GUIDE

VN-H37/137/237/237VP
VN-H57/157WP/257/257VP

This document provides information of protocol and API of JVC new IP cameras, VN-H series.
Specifications subject to change without notice.

2012.06.29. (V4.00)

Updates

Version	Date	Updates
1.00	2012/1/31	New release
1.01a	2012/02/23	Corrections of typographical error Page 5, section 2.1: change boundary to server_push Page 6, section 2.2: change boundary to server_push Page 14, Getting Enhance of Scene File: change 11 internal levels to 14 internal levels. Page 15, Setting Enhance of Scene File: change 11 internal levels to 14 internal levels.
		Page 17, Getting / Setting Shutter Speed of a Scene File:“auto” is added.
		Page 22, Setting Frame Rate: 15, 10, and 7.5 is added.
1.01b	2012/03/06	Page 6, section 2.3. Response is added. Page 7, section 2.4. Restrictions is added. Page 7, section 2.5. JPEG File Format Sent Out by the camera is added. Page 10, section 3.3. Response is added. Page 10, section 3.4. Restrictions is added. Page 10, section 3.5. H.264 Stream Format Send Out by the camera is added.
		Page 81, Default User Name is changed from PSIATest to psia. Default Password is changed from PSIATest to jvc.
2.00	2012/04/16	Page 6, section 2.2, Setting Frame Rate: 30, 15, 10, and 7.5 is added.
		Page 12, section 4, “JVC Protocol :MPEG-4 Streaming” is added.
		Page 19, section 8, “Getting Preset Data of Scene File” is added.
		Page 21, section 8, “Enhance”: Explanation of parameter is added.
		Page 22, section 8, Getting and Setting 3DDNR is added.
		Page 25, section 8, Getting and Setting ALC priority is added.
		Page 26, section 8, Corrections of typographical error: Easy is removed.
		Page 26, section 8, Getting and Setting Easy Day and Night is added.
		Page 27, section 8, Getting and Setting CLVI is added.
		Page 28, section 9, Setting Compression Format : mpeg4 is added.
		Page 29, section 9, Setting Resolution : 320x180 is removed.
		Page 29, section 9, Getting and Setting Rate Control Setting : MPEG-4 is added. Page 29, section 9, Setting Rate Control: Explanation of parameter is added.
		Page 29, section 9, Getting and Setting bitrate : MPEG-4 is added.
		Page 30, section 9, Getting and Setting I-Frame Interval : MPEG-4 is added.
		Page 31, section 9, Getting and Setting Monitor Out is added.
		Page 59, section 16, JVC API for Tampering Detect is added.
		Page 69, section 18, Getting and Setting Status of PSIA Protocol is added.
		Page 69, section 18, Getting and Setting Status of ONVIF Protocol is addded.
		Page 84, section 26, Getting and Setting Port Number of RTSP Server is added.
		Continue on next page

Version	Date	Updates
2.01	2012/05/08	Page 5, “JVC API for Audio” is added. Page 5, “Getting Audio from the Camera via HTTP” is added. Page 5, “Sending Audio to the Camera” is added. Page 33, section 10, JVC API for Audio is added. Page 36, section 11, Getting and Setting Alarm Action: “audioplay” and “pinout” are added. Page 41, section 11, Getting and Setting Alarm Trigger: “m1”, “b1”, “m2”, “b2”, “audio_detect1”, “audio_detect2”, “tampering_detect”, “ncbwe” and “ncbws” are added. Page 87, section 28, “Getting Audio from the Camera via HTTP” is added. Page 90, section 29, “Sending Audio to the Camera” is added. Page 92, section 31, List of ActiveX: “Audio Monitor” and “Audio Sending Client” are added. Page 93, section 31, Properties of ActiveX: Explanation of default Folder Name is added. Page 94, section 31, Properties of ActiveX: Audio Monitor / Audio Sending Client is added. Page 95, section 31, Method of ActiveX Control: Audio Monitor / Audio Sending Client is added. Page 96, section 31, How to use ActiveX control by HTML: Audio Monitor and Audio Sending Client are added. Page 97, section 31, HTML Sample: Audio Monitor and Audio Sender are added.
3.00	2012/05/21	Page 5, 13. JVC API for SD Card Record is added. Page 5, 31. Getting SD Card data from the Camera via RTSP/RTP is added. Page 5, 32. Exporting H.264 data from SD Card to the PC is added. Page 35, section 11, Explanation of SD Card recording is added. Page 51, section 12, Getting and Setting Parameters of Pre/Post Recording for FTP : Explanation of Encoder No. is added. Page 53, 13. JVC API for SD Card Record is added. Page 96, 31. Getting SD Card data from the Camera via RTSP/RTP is added. Page 97, 32. Exporting H.264 data from SD Card to the PC
4.00	2012/06/29	Page 30, section 9, Corrections of typographical error change from “channel is saved” to “channel is availed” Page 30, section 9, Example of Setting Compression Format is added. Page 35, section 11, event No.10 is added. Page 60, section 14, “Moving Specified Position to Center” is added.

Preface

This document is for VMS to support JVC new cameras, VN-H37/137/237.
If VMS supports only streaming, i.e. VMS does not have camera setting pages, the chapter "Streaming Protocol" provides how to get stream from a camera.
If VMS have setting page of the camera, focusing on necessary functionalities is recommended. Typical necessary functionalities are Image settings and Encode settings. Supporting all functionalities of camera will not pay. For example, if VMS does not get multiple streams from a camera, Encode settings can be simple because setting multiple resolution/encode to camera is not required.

Content

1. Streaming Protocol

- Both JVC protocol and standard RTSP/RTP are supported.
- JPEG, H.264 baseline profile, and H.264 high profile are supported. MPEG-4 will be supported in future.
- Maximum resolution is 1920x1080.
- VN-H series can send 3 different resolution streams of JPEG simultaneously.
- VN-H series can send 3 different resolution streams of H.264 simultaneously.
- Sending JPEG stream and H.264 stream simultaneously is supported.

2. JVC Protocol: JPEG Streaming

2.1. Basic Procedures

1) The client establishes a TCP connection to port number 80.

2) The client sends out API.

Example to get JPEG stream encoded by first channel of the camera

GET /api/video?encode=jpeg(1)&framerate=5&server_push=on HTTP/1.1<CRLF>

Host: 192.168.0.2<CRLF><CRLF>

Note <CRLF> denotes the line feed code (0x0D, 0x0A).

3) The camera returns HTTP response and JPEG stream.

JPEG files in boundary structure will be sent out continuously after HTTP response. Each Content-Length is the size of each JPEG data. Using the size, reading the whole data of each JPEG is possible. HTTP Response and JPEG data sent out by the camera are as follows.

HTTP Response
--foo<CRLF> Content-Type: image/jpeg<CRLF> Content-Length: 31614<CRLF><CRLF>
JPEG (No. 1) <CRLF>
--foo<CRLF> Content-Type: image/jpeg<CRLF> Content-Length: 32756<CRLF><CRLF>
JPEG (No. 2) <CRLF>
,,,

4) When the client wants to stop current JPEG transmission, the client disconnects TCP80.
The camera does not accept further API via current TCP that is used for JPEG transmission. To change parameter, disconnect current TCP to stop the JPEG transmission, connect new TCP, and send API with new parameter.

2.2. API Format

Structure

GET	space	API	space	HTTP/1.1	0x0D 0x0A
Host:	space	IP Address of Camera	0x0D 0x0A 0x0D 0x0A		

Unlike APIs for getting/setting parameters, Accept line is not required. Basic authentication is also not necessary.

Example

GET /api/video?encode=jpeg(1)&framerate=5&server_push=on HTTP/1.1<CRLF>

Host: 192.168.0.2<CRLF><CRLF>

Parameter value is indicated using =. Do not insert space before and after =.
Example framerate=1

Parameters are segmented using &. Do not insert space before and after &.
Example encode=jpeg&framerate=5
There is no need to specify all parameters. Default values will be used for parameters that are not specified.

Parameter Description

encode For specifying compression format with channel number. For example, specify as encode=jpeg(1) to get JPEG encoded by channel 1. To know compression format of each channel, open Encoder setting page by IE described in INSTRUCTIONS manual, or issue "encode" API described in later chapter of this document.
framerate For specifying the frame rate. For example, specify as framerate=5 to get at 5 fps. Specify as framerate=-5 to get at 1/5 fps, or in other words, 1 frame in 5 seconds. Selection range for JPEG is as follows.
30, 15, 10, 7.5, 5, 3, 2, 1, 0, -2, -3, -5, -10, -15, -20, -30, -60
When the parameter is specified as framerate=0, the camera sends 1 frame of JPEG data, and disconnect the TCP connection.
server_push For specifying streaming structure. For example, specify as server_push=on to get Server Push

structured JPEG. When framerate=0 is specified, Server Push is disabled even if server_push=on is specified.

2.3. Response

When API with server_push=on is successfully received.

The camera will return 200 OK. The x-vnh37_response line indicates actual parameter.

Example of VN-H137

HTTP/1.1 200 OK<CRLF>

Content-Type: multipart/x-mixed-replace;boundary=foo<CRLF>

Date: Tue, 06 Mar 2012 13:32:57 GMT<CRLF>

Server: JVC VN-H137 Network Camera<CRLF>

x-vnh37_response: encode=jpeg&framerate=5.0&framesize=1920x1080&server_push=on&ptz_info=off<CRLF>

<CRLF>

When API without server_push option is successfully received.

The camera will return 200 OK. The x-vnh37_response line indicates actual parameter.

Example of VN-H137

HTTP/1.1 200 OK<CRLF>

Connection: Keep-Alive<CRLF>

Content-Type: image/jpeg<CRLF>

Date: Tue, 06 Mar 2012 14:06:07 GMT<CRLF>

Server: JVC VN-H137 Network Camera<CRLF>

x-vnh37_response: encode=jpeg&framerate=5.0&framesize=1920x1080&server_push=off&ptz_info=off<CRLF>

<CRLF>

2.4. Restrictions

Access restriction

The camera has access restriction feature that enables to deny access from a specific IP address. If JPEG is requested from the IP address of access restriction, the camera disconnects the TCP connection after API is sent.

Restriction by maximum bitrate of the camera.

The maximum bitrate of the camera is about 20 Mbps.

Number of clients

The maximum number of clients that can get JPEG stream depends on encode settings and requests from client.

Refer the instruction manual for detailed infomation.

2.5. JPEG File Format Sent Out by the camera

JPEG file from the camera is JFIF compliant and consist of the following.

FFD8	Start Code
FFE0	Application Segment
FFFE	Comment Segment 1
FFFE	Comment Segment 2 (reserved)
FFC4	DHT Huffman Table
FFDB	DQT Quantization Table
FFDD	DRI Restart Interval
FFC0	SOF Frame Information
FFDA	Data Start Segment
FFD9	End Code

The following information is stored in the comment segment 1. Each item has a fixed length.

Item	Size	Example	Note
Version Information	9	JVC V1.0	Indicates the version of information stored in the comment segment.
File Size	18	size = 123456	Indicates JPEG size in bytes.
Width	13	width = 1920	Width of JPEG.
Height	14	height = 1080	Height of JPEG.
Model Name	18	type = VN-H137	Name of model that created the JPEG.
(reserved)	12	reverse = 0	(reserved)
Time Stamp	70	timestamp = 20120306233412538UTC	Indicates the time when the JPEG is created. This is made up of the year/month/day, hour/minute/second, millisecond and timezone code.
(reserved)	13	alarm = 00000000	(reserved)
Camera ID	50	camera = input01	Stores camera information set at VN-X35/235.
Motion Detect Setting	11	motion = 1	Specified as 1 when the motion detect is ON.
Motion Detect Result	7	md = 1	Specified as 1 if motion is detected at the time when JPEG is created.
Tampering Detect Result	14	tampering = 0	Specified as 1 if tampering is detected at the time when JPEG is created.
Pan position	16	digipan = 123	Indicates pan position in pixels from 0 to 1278.
Tilt position	17	digitilt = 123	Indicates tilt position in pixels from 0 to 958.
Zoom position	17	digizoom = 1.23	Indicates zoom value from 0.25 to 8.00.
Preset Posision Number	15	position = 19	Indicates preset position number after moving to preset position. In other cases, position = NA.

Item names and values, excluding the version information that does not include =, are stored in the following format.

name	space	=	space	value	(stuffed with 0x00)

fixed length for each item

Example: When width=640, the 13-byte area will be written as follows.

w	i	d	t	h		=		6	4	0	0x00	0x00

3. JVC Protocol: H.264 Streaming

3.1. Basic Procedures

1) The client establishes a TCP connection to port number 80.

2) The client sends out API.

Example to get H.264 high profile stream encoded by first channel of the camera

GET /api/video?encode=h264(1) HTTP/1.1<CRLF>

Host: 192.168.0.2<CRLF><CRLF>

Note <CRLF> denotes the line feed code (0x0D, 0x0A).

3) The camera returns HTTP response and H.264 stream.

HTTP Response and H.264 stream sent out by the camera are as follows.

HTTP Response
I Picture of H.264 (First Frame)
P Picture of H.264 (Second Frame)
,,,

4) When the client wants to stop current H.264 transmission, the client disconnects TCP80.

The camera does not accept further API via current TCP that is used for H.264 transmission. To change parameter, disconnect current TCP to stop the H.264 transmission, connect new TCP, and send API with new parameter.

3.2. API Format

Structure

GET	space	API	space	HTTP/1.1	0x0D 0x0A

Host:	space	IP Address of Camera	0x0D 0x0A 0x0D 0x0A

Unlike APIs for getting/setting parameters, Accept line is not required. Basic authentication is also not necessary.

Example

GET /api/video?encode=h264(1) HTTP/1.1<CRLF>

Host: 192.168.0.2<CRLF><CRLF>

Parameter value is indicated using =. Do not insert space before and after =.

Example encode=h264(1)

Parameter Description

encode For specifying compression format. For example, specify as encode=h264(1) to get H.264 encoded by channel 1. To know compression format of each channel, open Encoder setting page by IE described in INSTRUCTIONS manual, or issue "encode" API described in later chapter of this document.

3.3. Response

When API is successfully received.

The camera will return 200 OK. The x-vnh37_response line indicates actual parameter.

Example of VN-H137

HTTP/1.1 200 OK<CRLF>

Connection: Keep-Alive<CRLF>

Content-Type: video/mp4v-es<CRLF>

Date: Tue, 06 Mar 2012 15:10:55 GMT<CRLF>

Server: JVC VN-H137 Network Camera<CRLF>

x-vnh37_response: encode=h264&framesize=1920x1080<CRLF>

3.4. Restrictions

Access restriction

The camera has access restriction feature that enables to deny access from a specific IP address. If H.264 is requested from the IP address of access restrictions, the camera disconnects the TCP connection after API is send.

3.5. H.264 Stream Format Send Out by the camera

H.264 stream form the camera is sequence of I Picture and P Picture. Ratio of I Picture and P Picture depends on I-Frame interval setting. Encode page of Web has the setting.

Example of H.264 Stream

HTTP response
Sequence Parameter Set
Picture Parameter Set
User data
I Picture
User data
P Picture
~
User data
I Picture

There are Sequence Parameter Set, Picture Parameter Set, and User data before each I Picture and there is User data before each P Picture.

The following information is stored in the User data. Each item has a fixed length.

Item	Size	Example	Note
Start code	4	0x00000001	Start code of User data in H.264 stream.
NAL unit type	1	0x66	NAL unit type of User data in H.264 stream.
Payload type	1	0x05	Payload type of User data in H.264 stream.
User data size	1	0xf0	Size of User data in H.264 stream.
Reserved	16	0x0303030303030303030303030303030303	–
Model Name	18	type = VN-H137	Product Name
Time Stamp	70	timestamp = 20120306233412538UTC	This is made up of the year/month/day, hour/minute/second, millisecond and timezone code.
Camera ID	50	camera = input01	Camera ID that user can define
Motion Detect Result	7	md = 1	Specified as 1 if motion is detected at the time when data is created.
Tampering Detect Result	14	tampering = 0	Specified as 1 if tampering is detected at the time when data is created.

Pan position	16	digipan = 123	Indicates pan position in pixels from 0 to 1278.
Tilt position	17	digitilt = 123	Indicates tilt position in pixels from 0 to 958.
Zoom position	17	digizoom = 1.23	Indicates zoom value from 1.00 to 8.00.
Preset Posision Number	15	position = 19	Indicates preset position number after moving to preset position. In other cases, position = NA.

4. JVC Protocol: MPEG-4 Streaming

4.1. Basic Procedures

1) The client establishes a TCP connection to port number 80.

2) The client sends out API.

Example

GET /api/video?encode=mpeg4 HTTP/1.1<CRLF>

Host: 192.168.0.2<CRLF><CRLF>

Note <CRLF> denotes the line feed code (0x0D, 0x0A).

3) The camera returns HTTP response and MPEG-4 stream.

HTTP Response and MPEG-4 stream sent out by the camera are as follows.

HTTP Response
VOP of MPEG_4 (First Frame)
VOP of MPEG-4 (Second Frame)
,,,

4) When the client wants to stop current MPEG-4 transmission, the client disconnects TCP80.

The camera does not accept further API via current TCP that is used for H.264 transmission. To change parameter, disconnect current TCP to stop the MPEG-4 transmission, connect new TCP, and send API with new parameter.

4.2. API Format

Structure

GET	space	API	space	HTTP/1.1	0x0D 0x0A
Host:	space	IP Address of Camera	0x0D 0x0A 0x0D 0x0A		

Unlike APIs for getting/setting parameters, Accept line is not required. Basic authentication is also not necessary.

Example

GET /api/video?encode=mpeg4 HTTP/1.1<CRLF>

Host: 192.168.0.2<CRLF><CRLF>

Parameter value is indicated using =. Do not insert space before and after =.

Example encode=h264

Parameter Description

encode For specifying compression format.

4.3. Response

When API is successfully received.

The camera will return 200 OK. The x-vnh37_response line indicates actual parameter.

Example of VN-H137

HTTP/1.1 200 OK<CRLF>

Connection: Keep-Alive<CRLF>

Content-Type: video/mp4v-es<CRLF>

Date: Tue, 06 Mar 2012 15:10:55 GMT<CRLF>

Server: JVC VN-H137 Network Camera<CRLF>

x-vnh37_response: encode=mpeg4&framesize=640x480<CRLF>

4.4. Restrictions

Access restriction

The camera has access restriction feature that enables to deny access from a specific IP address. If MPEG-4 is requested from the IP address of access restrictions, the camera disconnects the TCP connection after API is send.

4.5. MPEG-4 Stream Format Send Out by the camera

MPEG-4 stream form the camera is MPEG-4 Part2 (ISO/IEC 14496-2) compliant, level3 of simple profile. Its is a sequence of I-VOPs, or I-VOPs and P-VOPs.

I-VOP: Inter frame compressed data

P-VOP: Inter frame compressed data with previous frame

Ratio of I-VOP and P-VOP depends on I-Frame interval setting. Encode page of Web has the setting.

First VOP can be I-VOP or P-VOP. If client want to decode from I-VOP, please skip P-VOP and wait first I-VOP.

Example of MPEG-4 Stream

HTTP response
P-VOP
P-VOP
P-VOP
VOL
I-VOP
P-VOP
~

There are VOL, Userdata1, GOV and Userdata2 before each I-VOP.

Data structure before I-VOP

Item	Note
VOL	VOL of MPEG-4 Video
Userdata1	Reserved
GOV	GOV of MPEG-4 Video
Userdata2	Userdata

Data structure of Userdata2

Item	Size	Example	Note
Start code	4	0x000001B2	Start code of User data in MPEG-4 stream.
Model Name	18	type = VN-H137	Product Name
Time Stamp	70	timestamp = 20120306233412538UTC	This is made up of the year/month/day, hour/minute/second, millisecond and timezone code.
Camera ID	50	camera = input01	Camera ID that user can define
Motion Detect Result	7	md = 1	Specified as 1 if motion is detected at the time when data is created.
Tampering Detect Result	14	tampering = 0	Specified as 1 if tampering is detected at the time when data is created.
Pan position	16	digipan = 123	Indicates pan position in pixels from 0 to 1278.
Tilt position	17	digitilt = 123	Indicates tilt position in pixels from 0 to 958.
Zoom position	17	digizoom = 1.23	Indicates zoom value from 1.00 to 8.00.
Preset Posision Number	15	position = 19	Indicates preset position number after moving to preset position. In other cases, position = NA.

5. RTSP/RTP

5.1. URI

RTSP of the camera is RFC2326 compliant.

Three encoders can be enabled in the camera at its maximum. Each encoder's URI for RTSP is:

Encoder Channel	URI of RTSP
1	rtsp://ipaddress/PSIA/Streaming/channels/0
2	rtsp://ipaddress/PSIA/Streaming/channels/1
3	rtsp://ipaddress/PSIA/Streaming/channels/2

To know compression format of each channel, open Encoder setting page by IE described in INSTRUCTIONS manual, or issue "encode" API described in later chapter of this document.

5.2. JPEG

- RFC

JPEG/RTP of the camera is RFC2435 compliant.

- Frame Rate of JPEG

In case of JPEG/RTP, the client can request frame rate to the camera.

Example to get 5fps JPEG: (This is valid when encode channel 1 is set to JPEG.)

rtsp://ipaddress/PSIA/Streaming/channels/0?maxFrameRate=5

If maxFrameRate is not specified, the camera tries to send JPEG at its maximum frame rate.

5.3. H.264

H.264/RTP of the camera is RFC3984 compliant.

6. API to Search Camera

The camera in LAN can be searched by broadcast/multicast packet that has search API.

Search Camera in LAN

Protocol Send broadcast/multicast packet with following text in UDP payload to destination port number 80. Source port number can be any value. Multicast address is 239.0.255.255.

system.id<CRLF>

Response The camera that received this packet sends unicast udp packet to the source port number of the search packet. UDP payload of response packet has model name, IP address, and subnet mask. The camera

waits 0-0.7 second before sending response to avoid too many responses are sent in short period from many cameras.

Response Example **system.id=VN-H37(192.168.0.2/24)&200 OK<CRLF>**

7. Using API that Requires Basic Authentication

Basic authentication is required for JVC API explained in Section 7 or later. This section provides general explanation of those APIs.

7.1. Procedure

1) The client establishes a TCP connection to port number 80.

2) The client sends API.

API has following structure.

GET	space	API Characters	space	HTTP/1.1	0x0D 0x0A
Accept:	space	text/plain (or text/html)	0x0D 0x0A		
Host:	space	IP Address of Camera	0x0D 0x0A		
Authorization: Basic	space	Encoded User Name and Password	0x0D 0x0A 0x0D 0x0A		

The following is an example of API for Getting subnet mask of the camera.

Example

GET /api/param?network.interface.subnetmask HTTP/1.1<CRLF>

Accept: text/plain<CRLF>

Host: 192.168.0.2<CRLF>

Authorization: Basic YWRtaW46anZj<CRLF><CRLF>

Specify the response format by Accept line. Plain text response is returned when this is specified as text/plain. HTML response is returned when text/html is specified. HTML response is returned when Accept is not specified. These APIs for getting/setting parameters are protected by basic authentication. Authorization line needs to include encoded username and password. There are 3 types of usernames, namely admin, operator and user. Available APIs are different for each username. Join the user name and the password using a colon, Base64 encode this character string and enter this in the Authorization line.

For example, when

User name admin

Password jvc

then the character string joining the user name and the password with a colon is:

admin:jvc

Base64 encoding of this string yields YWRtaW46anZj. Enter this in the Authorization line. Default password for each username is jvc.

3) The camera returns a response to the client. In the following example, current subnet mask is 255.0.0.0. In addition, 255.0.0.0 is followed by & and 200 OK, indicating that getting parameter is successful.

Example

HTTP/1.1 200 OK<CRLF>

Connection: close<CRLF>

Content-Length: 80<CRLF>

Content-type: text/plain<CRLF>

Date: Fri, 13 MAY 2011 07:33:12 GMT<CRLF>

Server: JVC VN-H37 API Server<CRLF>

network.interface.subnetmask=255.0.0.0&200 OK<CRLF>

4) The client disconnects TCP80 to end the use of API.

Note: APIs for getting/setting parameters are not restricted by the access restriction function.

7.2. Getting Parameter

Specify API in GET line according to the format below when getting a parameter from the camera.

/api/param?ParamA.ParamB.ParamC

It is possible to get multiple parameters at a time. Connect parameters with &. Do not insert space before and after &.

/api/param?ParamA.ParamB.ParamC&ParamA.ParamD.ParamE

The upper limit of this character string is 1024 bytes. The maximum number of parameters that can be acquired at a time is 15. Status settings, i.e. network.interface.status, network.dns.status, network.ntp.status, etc., can not be acquired at a time.

When acquisition is successfully completed, values will be shown in the body of HTTP response, followed by "&200 OK" message.

Example:

ParamA.ParamB.ParamC=Data&200 OK

When an error occurs, an error code will be returned instead of indicating a value in the body of HTTP response.
Example:

ParamA.ParamB.ParamC&401 Unauthorized

When multiple APIs for getting are performed at one time, a response will be returned for each setting.

ParamA.ParamB.ParamC&200 OK<CRLF>

ParamA.ParamB.ParamD&200 OK<CRLF>

7.3. Setting Parameter

Specify API in GET line according to the format below when setting a parameter for the camera.

/api/param?ParamA.ParamB.ParamC=Data

Parameter values are indicated using =. Do not insert space before and after =.

It is possible to perform multiple settings at a time. Connect parameters with &. Do not insert space before and after &.

/api/param?ParamA.ParamB.ParamC=Data&ParamA.ParamB.ParamD=Data

The upper limit of this character string is 1024 bytes. The maximum number of parameters that can be set at a time is 15. Status settings, i.e. network.interface.status, network.dns.status, network.ntp.status, etc., can not be acquired at a time.

Response will be in the following format.

ParamA.ParamB.ParamC&200 OK

An error code will be returned when setting is not properly performed. Example:

ParamA.ParamB.ParamC&401 Unauthorized

When multiple settings are performed at one time, a response will be returned for each setting.

ParamA.ParamB.ParamC&200 OK<CRLF>

ParamA.ParamB.ParamD&200 OK<CRLF>

8. JVC API for Camera

These APIs are related to camera settings. Same functions are shown on the Camera page of the WEB setting page. Refer to the instruction manual for details on the Camera page.

Getting Camera ID

Format **/api/param?camera.id**

Example of response **camera.id=VN-H37&200 OK**

Response example when setting field is left blank **camera.id=&200 OK**

Interpretation Acquire Camera ID comment. This comment is stored in comment segment of JPEG. The Camera ID is used as sender's display name of alarm mail. If you want to set sender's mail address, see "Setting Sender Mail Address".

Example of response **camera.id=Camera01&200 OK**

Sender **Camera01<somename@somecompany.com>**

Allowed users admin, operator, user

Setting Camera ID

Format **/api/param?camera.id=data**

Example **/api/param?camera.id=Camera01**

Example when setting as blank **/api/param?camera.id=%00**

Example of response **camera.id&202 Accepted(camera.status=save)**

Interpretation Change the camera ID stored in comment segment of JPEG. Maximum size is 40 bytes.

To use following characters, specify by hexadecimal number after %.

space & / < > # % " { } | \ ^ [ ] `

To set as blank, specify as %00(0x25, 0x30, 0x30).

To use space, specify as %20(0x25, 0x32, 0x30). If you want to set "Comment In JPEG" for example, specify as follows. **/api/param?camera.id=Comment%20In%20JPEG**

The Camera ID is used as sender's display name of alarm mail. If you want to set sender's mail address, see "Setting Sender Mail Address".

Example of setting **/api/param?camera.id=Camera01**

Sender **Camera01<somename@somecompany.com>**

The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Current Scene File Number

Format **/api/param?camera.scene.status**

Example of response **camera.scene.status=0&200 OK**

Interpretation Acquire current scene file number. A number from 0 to 7 is returned.

A scene file is a set of preset parameters below.

auto_exposure.reference, color, monitortype, pedestal, gamma, enhance, white_balance, brightness, white_balance, white_balance.r, white_balance.b, senseup_limit, brightness.highgain, true_daynight, blc, auto_exposure.priority, shutter

Allowed users admin, operator, user

Getting Preset Data of Scene File

Format /api/param?camera.scene(number).status

Example of getting scene file 0 /api/param?camera.scene(0).status

Example of response

camera.scene(0).status=General-55--0.45-30-auto-off-0-51-off-low-53-mid-autoM-2-combo-color-8-8-auto W-107-65-off-off-0-normal&200 OK

Interpretation Acquire preset data of specified scene file. The preset data is joined with hyphen as follows.

scenename-color-monitortype[*1]-gamma-shutter-brightness.highgain[*2]-auto_focus-iris-pedestal-autoblack[*1]-enhance_band[*1]-enhance-3ddnr-brightness-senseup_limit-auto_exposure.priority-true_daynight[*2]-avpk_color-avpk_bw[*3]-white_balance-white_balance_r-white_balance_b-blc-clvi-autoexposure.reference-atw_convergence[*1]

parameter that is marked with [*1] : The parameter is not used or data value is invalid.

parameter that is marked with [*2] : The parameter is used for VN-H37and VN-H237VP.

parameter that is marked with [*3] : The parameter is used for VN-H137and VN-H237.

Allowed users admin, operator, user

Loading/Saving/Initializing Scene File

Format /api/param?camera.scene(number).status=data

Example of loading scene file 0 /api/param?camera.scene(0).status=goto

Example of saving scene file 0 /api/param?camera.scene(0).status=save

Example of initializing scene file 0 /api/param?camera.scene(0).status=initialize

Example of response camera.scene(0).status&200 OK

Interpretation Load/save/initialize scene file setting. Specify from scene(0) to scene(7). Loading scene file changes current camera settings. Saving scene file saves setting s of specified scene file. Initializing scene file changes settings of specified scene file to default values.

Allowed users admin, operator

Getting Current Scene File Name

Format /api/param?camera.scene(number).name

Example of response camera.scene(0).name=general&200 OK

Interpretation Acquire current scene file name. Range of scene file number is between 0 to 7.

Scene file names are General, Indoor, Outdoor, CLVI, Traffic, DataSaving, Day, and Night.

Scene file name is read only.

Allowed users admin, operator, user

Getting Auto Exposure Reference of a Scene File

Format **/api/param?camera.scene(number).auto_exposure.reference**

Example of response **camera.scene(0).auto_exposure.reference=0&200 OK**

Interpretation Acquire auto exposure reference. A number from -5 to 5 is returned. When the number is bigger, image becomes brighter.

Allowed users admin, operator, user

Setting Auto Exposure Reference of a Scene File

Format **/api/param?camera.scene(number).auto_exposure.reference=data**

Example **/api/param?camera.scene(0).auto_exposure.reference=0**

Example of response **camera.scene(0).auto_exposure.reference&202**

Accepted(camera.scene.status=save)

Interpretation Change auto exposure reference. Specify a number from -5 to 5, or "+", "-". When the number is bigger, image becomes brighter. The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Color Level of a Scene File

Format **/api/param?camera.scene(number).image.color**

Example of response **camera.scene(0).image.color=50&200 OK**

Interpretation Acquire color level value. Range of color level is between 0 to 100. The value is mapped to 11 internal levels. The larger the value, the stronger will be the color.

Allowed users admin, operator, user

Setting Color Level of a Scene File

Format **/api/param?camera.scene(number).image.color=data**

Example of setting a value **/api/param?camera.scene(0).image.color=50**

Example of 1 step change **/api/param?camera.scene(0).image.color=+**

Example of response

camera.scene(0).image.color&202 Accepted(camera.scene.status=save)

Interpretation Change color level value. Specify 0 to 100, "+" or "-". The value is mapped to 11 internal levels. The larger the value, the stronger will be the color. It becomes stronger 1 step by specifying "+", softer 1 step by specifying "-". The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Enhance of a Scene File

Format **/api/param?camera.scene(number).image.enhance**

Example of response **camera.scene(0).image.enhance=50&200 OK**

Interpretation Acquire enhance setting. The enhance is equal to sharpness of image. Range of enhance is between 0 to 100, and it is mapped to 14 internal levels. The larger the value, the sharper will be the image.

Allowed users admin, operator, user

Setting Enhance of a Scene File

Format **/api/param?camera.scene(number).image.enhance=data**

Example of setting a value **/api/param?camera.scene(0).image.enhance=50**

Example of 1 step change **/api/param?camera.scene(0).image.enhance=+**

Example of response **camera.scene(0).image.enhance&202 Accepted(camera.scene.status=save)**

Interpretation Change enhance setting. The enhance is equal to sharpness of image. Specify 0 to 100, "+" or "-". The value is mapped to 14 internal levels. It becomes sharper 1 step by specifying "+", softer 1 step by specifying "-". The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting 3DDNR of a Scene File

Format **/api/param?camera.scene(number).image.3ddnr**

Example of response **camera.scene(0).image.3ddnr=mid&200 OK**

Interpretation Acquire 3DDNR (3 Dimension Digital Noise Reduction) setting. “off”, “low”, “mid” of “high”is returned.

Allowed users admin, operator, user

Setting 3DDNR of a Scene File

Format **/api/param?camera.scene(number).image.3ddnr=data**

Example of setting a value **/api/param?camera.scene(0).image.3ddnr=mid**

Example of response **camera.scene(0).image.3ddnr&202 Accepted(camera.scene(0).status=save)**

Interpretation Change 3DDNR setting. Specify “off”, “low”, “mid” of “high”. The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting White Balance of a Scene File

Format **/api/param?camera.scene(number).image.white_balance**

Example of response **camera.scene(0).image.white_balance=auto&200 OK**

Interpretation Acquire white balance setting. "autoW", "autoN", or "manual" is returned.

Allowed users admin, operator, user

Setting White Balance of a Scene File

Format /api/param?camera.scene(number).image.white_balance=data

Example /api/param?camera.scene(0).image.white_balance=auto

Example of response camera.scene(0).image.white_balance&202 Accepted(camera.scene.status=save)

Interpretation Change white balance setting. Specify "autoW", "autoN", or "manual". If "op_auto" is specified, one push auto white balance control is done, and setting becomes "manual". The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting R-Gain of White Balance of a Scene File

Format /api/param?camera.scene(number).image.white_balance.r

Example of response camera.scene(0).image.white_balance.r=s85&200 OK

Interpretation Acquire R-gain of white balance setting. s0 to s255 is returned. The s before number means "step". Default value is s85.

Allowed users admin, operator, user

Setting R-Gain of White Balance of a Scene File

Format /api/param?camera.scene(number).image.white_balance.r=data

Example /api/param?camera.scene(0).image.white_balance.r=s100

Example of response

camera.scene(0).image.white_balance.r&202 Accepted(camera.status=save)

Interpretation Change R-gain white balance setting. Specify s0 to s255. The s before number means "step". Default value is s85. The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting B-Gain of White Balance of a Scene File

Format /api/param?camera.scene(number).image.white_balance.b

Example of response camera.scene(0).image.white_balance.b=s219&200 OK

Interpretation Acquire B-gain of white balance setting. s0 to s255is returned. The s before number means "step". Default value is s219.

Allowed users admin, operator, user

Setting B-Gain of White Balance of a Scene File

Format **/api/param?camera.scene(number).image.white_balance.b=data**

Example **/api/param?camera.scene(0).image.white_balance.b=s100**

Example of response

camera.scene(0).image.white_balance.b&202 Accepted(camera.status=save)

Interpretation Change B-gain white balance setting. Specify s0 to s255. The s before number means "step". Default value is s219. The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting AGC of a Scene File

Format **/api/param?camera.scene(number).image.brightness**

Example of response **camera.scene(0).image.brightnesss=autoL&200 OK**

Interpretation Acquire AGC setting. "manual", "autoM" or "autoH" is returned.

Allowed users admin, operator, user

Setting AGC of a Scene File

Format **/api/param?camera.scene(number).image.brightness=data**

Example **/api/param?camera.scene(0).image.brightness=auto**

Example of response **camera.scene(0).image.brightness&202 Accepted(camera.scene.status=save)**

Interpretation Change AGC setting. Specify "manual", "autoM" or "autoH". The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

The AGC setting is limited by Day and Night setting. Change Day and Night first, then change AGC setting.

Allowed users admin, operator

Getting Limit of Sense Up of a Scene File

Format **/api/param?camera.scene(number).image.senseup_limit**

Example of response **camera.scene(0).image.senseup_limit=0&200 OK**

Interpretation Acquire limit of sense up. 0, 2, 4, 8, 16, 32 or 60 is returned. 0 means sense up is disabled. Other numbers mean frame number of sense up.

Allowed users admin, operator, user

Setting Limit of Sense Up of a Scene File

Format **/api/param?camera.scene(number).image.senseup_limit=data**

Example **/api/param?camera.scene(0).image.senseup_limit=4**

Example of response **camera.scene(0).image.senseup_limit&202 Accepted(camera.status=save)**

Interpretation Change limit of sense up. Specify 0, 2, 4, 8, 16, 32, 60, "+" or "-". It becomes bigger 1 step by specifying "+", smaller 1 step by specifying "-". The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting ALC priority of Scene File

Format /api/param?camera.scene(number).auto_exposure.priority

Example of response camera.scene(0).auto_exposure.priority=combo&200 OK

Interpretation Acquire ALC priority. ALC priority decides what is used first for auto exposure. “combo”, “motion” or “quality”is returned.

Allowed users admin, operator, user

Setting ALC priority of Scene File

Format /api/param?camera.scene(number).auto_exposure.priority=data

Example /api/param?camera.scene(0).auto_exposure.priority=combo

Example of response camera.scene(0).auto_exposure.priority&202

Accepted(camera.scene(0).status=save)

Interpretation Change ALC priority. ALC priority decides what is used first for auto exposure. “combo”, “motion” or “quality”is returned. In case of “combo”, selects the best combination automatically. In case of “motion”, assigns priority to AGC. In case of “quality”, assigns priority to the Sense Up function.

Allowed users admin, operator, user

Getting Shutter Speed of a Scene File

Format /api/param?camera.scene(number).shutter

Example of response camera.scene(0).shutter=60&200 OK

Interpretation Acquire shutter speed setting. “auto”, "auto100", "auto1000", 30, 50, 60, 100, 250, 500, 1000, 2000, 4000, 10000 or "flickerless" is returned. For example, 60 means shutter speed 1/60. In case of “auto”, the shutter speed is adjusted from 1/30 to 1/10000. In case of "auto100", the shutter speed is adjusted from 1/30 to 1/100. In case of "auto1000", the shutter speed is adjusted from 1/30 to 1/1000. In case of "flickerless", the shutter speed that avoids flicker is selected automatically.

Allowed users admin, operator, user

Setting Shutter Speed of a Scene File

Format /api/param?camera.scene(number).shutter=data

Example of setting a value /api/param?camera.scene(0).shutter=60

Example of 1 step change /api/param?camera.scene(0).shutter=+

Example of response **camera.scene(0).shutter&202 Accepted(camera.scene.status=save)**

Interpretation Change shutter speed setting. Specify "auto100", "auto1000", 30, 50, 60, 100, 250, 500, 1000, 2000, 4000, 10000 or "flickerless", "+" or "-". To set 1/60 for example, specify 60. It becomes shorter 1 step by specifying "+", longer 1 step by specifying "-". The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Day and Night Setting of a Scene File (for VN-H37 and VN-H237VP)

Format **/api/param?camera.scene(number).image.true_daynight**

Example of response **camera.scene(0).image.true_daynight=off&200 OK**

Interpretation Acquire Day and Night setting. "color", "bw", "autoL", "autoM", or "autoH" is returned.

Allowed users admin, operator, user

Setting Day and Night Setting of a Scene File (for VN-H37 and VN-H237VP)

Format **/api/param?camera.scene(number).image.true_daynight=data**

Example **/api/param?camera.scene(0).image.true_daynight=on**

Example of response

camera.scene(0).image.true_daynight&202 Accepted(camera.scene.status=save)

Interpretation Change Day and Night setting. Specify "color", "bw", "autoL", "autoM", or "autoH". The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

The AGC setting is limited by Day and Night setting. Change Day and Night first, then change AGC setting.

Allowed users admin, operator

Getting Easy Day and Night Setting of a Scene File (for VN-H137 and VN-H237)

Format **/api/param?camera.scene(number).image.brightness.highgain**

Example of response **camera.scene(0).image.brightness.highgain=color&200 OK**

Interpretation Acquire Easy Day and Night setting. "color", "bw" or "auto" is returned.

Allowed users admin, operator, user

Setting Easy Day and Night Setting of a Scene File (for VN-H137 and VN-H237)

Format **/api/param?camera.scene(number).image.brightness.highgain=data**

Example **/api/param?camera.scene(0).image.brightness.highgain=color**

Example of response

camera.scene(0).image.brightness.highgain&202 Accepted(camera.scene.status=save)

Interpretation Change Day and Night setting. Specify "color", "bw" or "auto". The change of scene file 0 is saved

by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.
The AGC setting is limited by Day and Night setting. Change Day and Night first, then change AGC setting.
Allowed users admin, operator

Getting Back Light Compensation of a Scene File

Format /api/param?camera.scene(number).image.blc

Example of response camera.scene(0).image.blc=off&200 OK

Interpretation Acquire Back Light Compensation setting. "off", "a", "b", "c" or "d" is returned. Refer the instruction manual for detailed information of "a", "b", "c" and "d".

Allowed users admin, operator, user

Setting Back Light Compensation of a Scene File

Format /api/param?camera.scene(number).image.blc=data

Format of setting ON /api/param?camera.scene(0).image.blc=a

Example of response camera.scene(0).image.blc&202 Accepted(camera.scene.status=save)

Interpretation Change Back Light Compensation setting. Specify "off", "a", "b", "c" or "d". Refer the instruction manual for detailed information of "a", "b", "c" and "d". The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting CLVI of a Scene File

Format /api/param?camera.scene(number).image.clvi

Example of response camera.scene(0).image.clvi=off&200 OK

Interpretation Acquire CLVI (Clear Logic Video Intelligence) setting. "on" or "off" is returned.

Allowed users admin, operator, user

Setting CLVI of a Scene File

Format /api/param?camera.scene(number).image.clvi=data

Format of setting ON /api/param?camera.scene(0).image.clvi=on

Example of response camera.scene(0).image.clvi&202 Accepted(camera.scene.status=save)

Interpretation Change CLVI (Clear Logic Video Intelligence) setting. Specify "on" or "off". The change of scene file 0 is saved by the API, camera.scene(0).status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

9. JVC API for Encode

These APIs are related to camera settings. Same functions are shown on the Encode page of the WEB setting page. Refer to the instruction manual for details on the Encode page.

Though multiple encode is available, there are limitations to set multiple encode channels. If VMS does not get multiple streams from a camera, setting only first channel is recommended that can simplify such limitations. Refer the Encode page of the camera to see those limitations.

Getting Compression Format

Format **/api/param?encode(number).type**

Example of response **encode(1).type=jpeg&200 OK**

Interpretation Acquire compression format of the encode channel. Encode channel is from encode(1) to encode(3).

Allowed users admin, operator, user

Setting Compression Format

Format **/api/param?encode(number).type=data**

Example **/api/param?encode(1).type=h264high**

Example of response **encode(1).type&202 Accepted(encode.status=save)**

Interpretation Change compression format of the encode channel. Set "jpeg", "h264high", "h264baseline", “mpeg4”or "off". The change of the first encode channel is availed by the API, encode(1).status=save.

Example When Changing Compression Format from H.264 to JPEG,

it is necessary that rate control setting is specified to “afs” or “vfs”.

/api/param?encode(1).type=jpeg&encode(1).cbr_mode=afs

/api/param?encode(1).status=save

Example When Changing Compression Format from JPEG to H.264,

it is necessary that rate control setting is specified to “cbr” or “vbr”.

/api/param?encode(1).type=h264high&encode(1).cbr_mode=cbr

/api/param?encode(1).status=save

Caution: In case of multiple resolution, 3 channels are available at the maximum. In case of multiple encoding, 2 channels are available at the maximum, i.e. 3rd channel is not available.

Allowed users admin, operator

Getting Resolution (Frame Size)

Format **/api/param?encode(number).framesize**

Example of response **encode(1).framesize=1920x1080&200 OK**

Interpretation Acquire resolution of the encode channel. Encode channel is from encode(1) to encode(3).

Allowed users admin, operator, user

Setting Resolution (Frame Size)

Format /api/param?encode(number).framesize=data

Example /api/param?encode(1).framesize=1920x1080

Example of response encode(1).type&202 Accepted(encode.status=save)

Interpretation Change resolution of the encode channel. Set "1920x1080", "quadvga", "1280x720", "vga", "qvga" or "640x360". The change of the first encode channel is availed by the API, encode(1).

Maximum resolution of VN-H37/137/237/57/157/257 is 1920x1080.

Maximum resolution of VN-V17/217 is 1280x720.

Caution: All channels need to have same aspect ratio, 16:9 or 4:3.

Allowed users admin, operator

Getting Rate Control Setting

Format /api/param?encode(number).cbr_mode

Example of response encode(1).cbr_mode=afs&200 OK

Interpretation Acquire the rate control setting.

When compression format is JPEG, "vfs" or "afs" is returned. Quantization table is fixed in the case of vfs (VariableFileSize). In the case of afs (AverageFileSize), bit rates are controlled such that the average size of multiple files remains constant.

When compression format is H.264 or MPEG-4, "cbr" or "vbr" is returned. Bitrate is controlled to be constant in the case of cbr (Constant Bitrate). In the case of vbr (Variable Bitrate), bitrate can be larger by input image.

Allowed users admin, operator, user

Setting Rate Control

Format /api/param?encode(number).cbr_mode=data

Example /api/param?encode(1).cbr_mode=vfs

Example of response encode(1).cbr_mode&202 Accepted(encode.status=save)

Interpretation Change rate control. When compression format is JPEG, set "vfs" or "afs". This parameter is not included file size/quality information. Parameters of JPEG File Size can be set by other APIs, encode(number).quality.

When compression format is H.264 or MPEG-4, set "cbr" or "vbr". The change of the first channel is availed by the API, encode(1).status=save.

Allowed users admin, operator

Getting H.264 or MPEG-4 bitrate

Format **/api/param?encode(number).bitrate**

Example of response **encode(1).bitrate=4000000&200 OK**

Interpretation Acquire the bitrate setting of H.264 or MPEG-4. This API is valid when compression format is h264high, h264baseline or mpeg4. If the response is 4000000 for example, the setting is 4Mbps.

Allowed users admin, operator, user

Setting H.264 or MPEG-4 bitrate

Format **/api/param?encode(number).bitrate=Data**

Example **/api/param?encode(1).bitrate=2000000**

Example of response **encode(1).bitrate&202 Accepted(encode.status=save)**

Interpretation Change the bitrate setting of H.264 or MPEG-4. This API is valid when compression format is h264high, h264baseline or mpeg-4. In case of H.264, specify from 64000 to 8000000. In case of MPEG-4, specify from 64000 to 3000000. The change of the first channel is availed by the API, encode(1).status=save.

Allowed users admin, operator

Getting JPEG File Size Setting

Format **/api/param?encode(number).quality**

Example of response **encode(1).quality=40k&200 OK**

Interpretation Acquire the file size setting of JPEG. This API is valid when compression format is jpeg. If the response is 40k for example, the setting is 40KB.

Allowed users admin, operator, user

Setting JPEG File Size

Format **/api/param?encode(number).quality=Data**

Example **/api/param?encode(1).quality=30k**

Example of response **encode(1).quality&202 Accepted(encode.status=save)**

Interpretation Change the file size setting of JPEG. This API is valid when compression format is jpeg. The unit of set values is in KB.

Allowed users admin, operator

Getting H.264 or MPEG-4 I-Frame Interval Setting

Format **/api/param?encode(number).iframeinterval**

Example of response **encode(1).iframeinterval=15&200 OK**

Interpretation Acquire I-Frame interval of H.264 or MPEG-4 encoding. This API is valid when compression format is h264high, h264baseline or mpeg4.

Allowed users admin, operator, user

Setting H.264 or MPEG-4 I-Frame Interval

Format /api/param?encode(number).iframeinterval=data

Example /api/param?encode(1).iframeinterval=15

Example of response encode(1).iframeinterval&202 Accepted(encode.status=save)

Interpretation Change I-Frame interval of H.264 or MPEG-4. This API is valid when compression format is h264high, h264baseline or mpeg4. In case of H.264, specify 5, 10, 15, 30, 60, 90, or 120. In case of MPEG-4, specify 15 or 30. The change of the first channel is availed by the API, encode(1).status=save.

Allowed users admin, operator

Getting Frame Rate Setting

Format /api/param?encode(number).framerate

Example of response encode(1).framerate=15&200 OK

Interpretation Acquire frame rate of the encoding.

Allowed users admin, operator, user

Setting Frame Rate

Format /api/param?encode(number).framerate=data

Example /api/param?encode(1).framerate=30

Example of response encode(1).framerate&202 Accepted(encode.status=save)

Interpretation Change frame rate of the encoding. In case of H.264, specify 30, 25, 15, 10, 7.5, 5, 3, 2, or 1. In case of JPEG, specify 30, 15, 10, 7.5, 5, 3, 2, or 1. When 30fps or 25fps is set, 3DDNR, motion detection and analog viedeo output are not available. The change of the first channel is availed by the API, encode(1).status=save.

Allowed users admin, operator

Getting Monitor Out Status

Format /api/param?video.output.status

Example of response video.output.status=on&200 OK

Interpretation Acquire monitor out status. “on” or “off” is returned.

Allowed users admin, operator, user

Setting Monitor Out Status

Format /api/param?video.output.status=data

Example /api/param?video.output.status=on

Example of response **video.output.status&200 OK**

Interpretation Change monitor out status. Specify on to enable the monitor out, off to disable the monitor out.

Allowed users admin, operator

10. JVC API for Audio (VN-H57/157WP/257/257VP)

These APIs are related to audio settings. Same functions are shown on the Audio page of the WEB setting page. Refer to the instruction manual for details on the Audio page.

Getting Audio Duplex Mode

Format **/api/param?audio.input(1).halfduplex**

Example of response **audio.input(1).halfduplex=on&200 OK**

Interpretation Acquire audio duplex mode. "on" or "off" is returned. When the setting is "on", audio from the camera is muted during a client is sending audio to the camera. By setting "on", howling/echo can be suppressed.

Allowed users admin, operator, user

Setting Audio Duplex Mode

Format **/api/param?audio.input(1).halfduplex=data**

Example **/api/param?audio.input(1).halfduplex=on**

Example of response **audio.input(1).halfduplex&200 OK**

Interpretation Change audio duplex mode. Specify "on" or "off". When the setting is "on", audio from the camera is muted during a client is sending audio to the camera. By setting "on", howling/echo can be suppressed.

Allowed users admin, operator

Getting Mike Gain

Format **/api/param?audio.input(1).gain**

Example of response **audio.input(1).gain=32&200 OK**

Interpretation Acquire mike gain. "0", "20", "26", "32" or "auto" is returned. "32" measn 32 dB.

Allowed users admin, operator, user

Setting Mike Gain

Format **/api/param?audio.input(1).gain=data**

Example **/api/param?audio.input(1).gain=32**

Example of response **audio.input(1).gain&200 OK**

Interpretation Change mike gain. Specify "0", "20", "26", "32" or "auto". "32" measn 32 dB.

Allowed users admin, operator

Getting Mike Power Supply setting

Format **/api/param?audio.input(1).powersupply.status**

Example of response **audio.input(1).powersupply.status=on&200 OK**

Interpretation Acquire mike power supply setting. "on" or "off" is returned.

Allowed users admin, operator, user

Setting Mike Power Supply

Format **/api/param?audio.input(1).powersupply.status=data**

Example **/api/param?audio.input(1).powersupply.status=32**

Example of response **audio.input(1).powersupply.status&200 OK**

Interpretation Change mike power supply setting. Specify "on" or "off".

Allowed users admin, operator

11. JVC API for Alarm

These APIs are related to alarm settings. Same functions are shown on the Alarm page of the WEB setting page. Refer to the instruction manual for details on the Alarm page.

Getting On/Off of Alarm Action

Format **/api/param?application.event(Number).status**

Example When Getting the on/off status of alarm action No. 1

/api/param?application.event(1).status

Example of response **application.event(1).status=on&200 OK**

Interpretation Acquire the on/off status of the alarm action for the specified alarm action number. 5 alarm actions, 1 periodic FTP assigned to No.6, 1 pre/post FTP assigned to No.7, 1 SD Card constant recording assigned to No.8, and 1 SD Card alarm recording assigned to No.10 are available, so alarm action number can be 1 to 8 and 10. Note that alarm numbers are different from the alarm input pin numbers. Either on or off is returned.

Allowed users admin, operator

Setting On/Off of Alarm Action, or Enabling Changes to Alarm Action

Format **/api/param?application.event(Number).status=data**

Example When setting alarm action No. 1 to off

/api/param?application.event(1).status=off

Example of response **application.event(1).status&200 OK**

Interpretation Set the alarm action of the specified alarm action number to on/off, or enable changes to the alarm

action. 5 alarm actions, 1 periodic FTP assigned to No.6, 1 pre/post FTP assigned to No.7, and 1 SD Card constant recording assigned to No.8, and SD Card alarm recording assigned to No.10 are available, so alarm action number can be 1 to 8 and 10. Note that alarm numbers are different from the alarm input pin numbers. Either on or off will be returned.
Specify "on", "off" or "restart". By "restart", changes to alarm action and alarm trigger are enabled. By "on" after "restart", the alarm action starts working with the changed settings. If "restart" is not set after changes to alarm action and alarm trigger, APIs to get settings of alarm action and alarm trigger return previous values.
Allowed users admin, operator

Getting Alarm Action

Format /api/param?application.event(Number).action

Example When Getting action of alarm action No. 1

/api/param?application.event(1).action

Example of Response

application.event(1).action=mailto/somebody@somecompany.com/none/Message&200 OK

Interpretation Acquire the alarm action of the specified alarm action number. 5 alarm actions, 1 periodic FTP assigned to No.6, 1 pre/post FTP assigned to No.7, 1 SD Card constant recording assigned to No.8, and 1 SD Card alarm recording assigned to No.10 are available, so alarm action number can be 1 to 8, and 10. Note that alarm numbers are different from the alarm input pin numbers. A separate API (/api/param?application.event(Number).status) is used to acquire the on/off status of the alarm action.
When no action is specified, response below is returned.

Example of Response application.event(1).action=&200 OK

When sending mail is specified, mailto, mail address, JPEG attaching and the character string to be sent will be returned. When spaces are included in the character string, the character string with spaces will be returned. Segments are indicated by /. If JPEG attaching is on, "object(Number)" is returned, and if JPEG attaching is off, "none" is returned.

Example of Response

application.event(1).action=mailto/somebody@somecompany.com/object(1)/Message&200 OK

When sending via TCP is specified, tcpto, IP address, port number and the character string to be sent will be returned. Segments are indicated by /. If JPEG attaching is on, "object(Number)" is returned, and if JPEG attaching is off, "none" is returned.

Example of Response application.event(1).action=tcpto/10.0.0.100/20000/object(1)/Message&200 OK

When sending via UDP is specified, udpto, IP address, port number and the character string to be sent will be

returned. Segments are indicated by /.

Example of Response application.event(1).action=udpto/10.0.0.100/20000/Message&200 OK

When switch scene file is specified, scene file number will be returned.

Example of Response when scene file number is 7

application.event(1).action=camera.image.scene(7).status/goto&200 OK

When preset position is specified, position number will be returned.

Example of Response when position number is 2

application.event(1).action=camera.position(2).status/goto&200 OK

[VN-H57/157WP/257/257VP Only] When audio file playback is specified, audio file number will be returned. A separate API (/api/param?application.audioplay) is used to get/set parameters of audio file playback.

Example of Response when audio file number is 2

application.event(1).action=audioplay/audiofile02/ch01&200 OK

[VN-H57/157WP/257/257VP Only] When alarm output is specified, pinout, distinction between make/break (m1 or b1) and output time (millisecond) will be returned. Segments are indicated by /.

Example of Response application.event(1).action=pinout/m1/1500&200 OK

Alarm action of event number 6 is periodic FTP. Response to the API has ftpto, FTP number, and the attached object number. Segments are indicated by /. The FTP number is fixed as ftp01 at all times. The object number is fixed as object(6). Parameters of FTP can be gotten by another API, application.ftp.

Example of Response application.event(6).action=ftpto/ftp01/object(6)&200 OK

Alarm action of event number 7 is "PrePostRecording + FTP". When "PrePostRecording + FTP" is enabled, recftp, FTP number, and the attached object number will be returned. Segments are indicated by /. The FTP number is fixed as ftp01 at all times. The object number is fixed as object(7). Parameters of FTP can be gotten by other APIs, application.ftp and application.object.

Example of Response application.event(7).action=recftp/ftp01/object(7)&200 OK

Alarm action of event number 8 is “SD Card constant recording”. When “SD Card constant recording” is enabled, rec, SD Card number, and the attached object number will be returned. Segments are indicated by /. The SD Card number is fixed as sd01 at all times. The object number is fixed as object(8). Parameters of SD Card recording can be gotten by other APIs, application.object.

Example of Response application.event(8).action=rec/sd01/object(8)&200 OK

Alarm action of event number 10 is “SD Card alarm recording”. When “SD Card alarm recording” is enabled, rec, SD Card number, and the attached object number will be returned. Segments are indicated by /. The SD Card number is fixed as sd01 at all times. The object number is fixed as object(10). Parameters of SD Card recording can be gotten by other APIs, application.object.

Example of Response application.event(10).action=rec/sd01/object(10)&200 OK

Allowed users admin, operator

Setting Alarm Action

Format /api/param?application.event(Number).action=Data

Example When setting action of Alarm No. 1

/api/param?application.event(1).action=mailto/somebody@somecompany.com/none/Message

Example of Response

application.event(1).action&202 Accepted(application.event(1).status=restart)

Interpretation Set the alarm action of the specified alarm number. 5 alarm actions, 1 periodic FTP assigned to No.6, 1 pre/post FTP assigned to No.7 are available, 1 SD Card constant recording assigned to No.8 and 1 SD Card alarm recording assigned to No.10 so alarm action number can be 1 to 8 and 10. Note that alarm numbers are different from the alarm input pin numbers. A separate API (/api/param?application.event(Number).status=off) is used to set the alarm action to off.

The action will be activated by setting the alarm trigger. The API for setting the alarm trigger is /api/param?application.event(Number).trigger.

The changes to settings of alarm action become valid by /api/param?application.event(Number).status=restart.

Specify mailto, mail address, JPEG attach and the character string to be sent when sending via mail. Segments are indicated by /. The maximum number of characters for the mail address is 95. To attach JPEG, specify object(Number). If none is specified instead of object(Number), JPEG is not attached to the mail. Number of the character string is from 1 to 127 bytes. To use following characters, specify by hexadecimal number after %.

space & / < > # % " { } | \ ^ [ ] `

For example, specify 3 characters %20 when inserting a space in the character string. For example, to send the character string "This is alarm.", specify as "This%20is%20alarm.". %09 and %0D are not available.

Setting Example

/api/param?application.event(1).action=mailto/somebody@somecompany.com/object(1)/Message%20ON

The character string "Alarm from VN-H37" will be stored in the title field of the mail.

Specify tcpto, IP address, port number, none or object(Number), and the character string to be sent when sending via TCP. Segments are indicated by /. The number of character string is from 1 to 127 bytes. To use following characters, specify by hexadecimal number after %.

space & / < > # % " { } | \ ^ [ ] `

For example, specify 3 characters %20 when inserting a space in the character string. For example, to send the character string "This is alarm.", specify as "This%20is%20alarm.". %09 and %0D are not available.

Setting Example /api/param?application.event(1).action=tcpto/10.0.0.100/20000/none/Message

To add JPEG, specify object(Number) instead of “none”.

Setting Example /api/param?application.event(1).action=tcpto/10.0.0.100/20000/object(1)/Message

Specify udpto, IP address, port number and the character string to be sent when sending via UDP. Segments are indicated by /. The number of character string is from 1 to 127 bytes. To use following characters, specify by hexadecimal number after %.

space & / < > # % " { } | \ ^ [ ] `

For example, specify 3 characters %20 when inserting a space in the character string. For example, to send the character string "This is alarm.", specify as "This%20is%20alarm.". %09 and %0D are not available.

Setting Example /api/param?application.event(1).action=udpto/10.0.0.100/20000/Message

Specify scene file number when switch scene file is specified.

Setting Example /api/param?application.event(1).action=camera.image.scene(7).status/goto

Specify preset position number when preset position is specified.

Setting Example /api/param?application.event(1).action=camera.position(2).status/goto

[VN-H57/157WP/257/257VP Only] Specify audio file number when audio file playback is specified. The audio file number can be from 01 to 05.

Setting Example /api/param?application.event(1).action=audioplay/audiofile02/ch01

[VN-H57/157WP/257/257VP Only] Specify pinout, distinction between make/break (m1 or b1) and the time (millisecond) when alarm output is specified. Segments are indicated by /. The time is 0 or from 100 to 5000. When the time is 0, alarm output does not come back to previous state.

Setting Example /api/param?application.event(1).action=pinout/m1/1500

Alarm action of event number 6 is periodic FTP. Other Event number can not be set to periodic FTP. Parameters of FTP can be set by another API, application.ftp.

Setting Example /api/param?application.event(6).action=ftpto/ftp01/object(6)

Alarm action of event number 7 is "PrePostRecording + FTP". Specify recftp, FTP number and the object for PrePostRecording+FTP. The FTP number is fixed as ftp01 at all times. The object is fixed as object(7). Parameters of FTP can be set by other APIs, application.ftp and application.object. Ensure to set the FTP server (/api/param?application.ftp.host, /api/param?application.object.framerate etc.) before setting PrePostRecording+FTP.

Setting Example /api/param?application.event(7).action=recftp/ftp01/object(7)

Alarm action of event number 8 is “SD Card constant recording”. Specify rec, SD Card number and the object for SD Card constant recording. The SD Card number is fixed as sd01 at all times. The object is fixed as object(8). Parameters of SD Card recording can be set by other APIs, application.object.

Setting Example /api/param?application.event(8).action=rec/sd01/object(8)

Alarm action of event number 10 is “SD Card alarm recording”. Specify rec, SD Card number and the object for SD Card alarm recording. The SD Card number is fixed as sd01 at all times. The object is fixed as object(10). Parameters of SD Card recording can be set by other APIs, application.object.

Setting Example /api/param?application.event(10).action=rec/sd01/object(10)

Allowed users admin, operator

Getting Alarm Filter Setting

Format /api/param?application.event(Number).filter(WeekOfDay).status

Example When Getting Setting of Sunday filter of Alarm No. 1

/api/param?application.event(1).filter(sunday).status

Example of Response application.event(1).filter(sunday).status=off&200 OK

Interpretation Acquire filter setting of the alarm action for the specified alarm number. Up to 5 alarm actions can be specified, periodic FTP is assigned to event No.6, and pre/post FTP assigned to No.7. Therefore the number of event(number) can be set between the range of 1 to 7. Note that alarm numbers are different from the alarm input pin numbers.

Specify sunday, monday, tuesday, wednesday, thursday, friday or saturday for WeekOfDay.

When the filter is enabled, on will be returned. When the filter is disabled, off will be returned.

Allowed users admin, operator

Setting Alarm Filter

Format /api/param?application.event(Number).filter(WeekOfDay).status=data

Example When setting Sunday filter of Alarm No. 1

/api/param?application.event(1).filter(sunday).status=on

Example of Response

application.event(1).filter(sunday).status&202 Accepted(application.event(1).status=restart)

Interpretation Set filter setting of the alarm action for the specified alarm number. Up to 5 alarm actions can be specified, periodic FTP is assigned to event No.6, and pre/post FTP assigned to No.7. Therefore the number of event(number) can be set between the range of 1 to 7. Note that alarm numbers are different from the alarm input pin numbers.

Specify sunday, monday, tuesday, wednesday, thursday, friday or saturday for WeekOfDay.

Specify on to enable the filter, off to disable the filter.

The changes to filter of alarm action is saved by /api/param?application.event(Number).status=restart.

Allowed users admin, operator

Getting Alarm Filter Time

Format /api/param?application.event(Number).filter(WeekOfDay).time

Example When Getting Time of Sunday filter of Alarm No. 1

/api/param?application.event(1).filter(sunday).time

Example of Response application.event(1).filter(sunday).time=000000-240000&200 OK

Interpretation Acquire filter time of the alarm action for the specified alarm number. Up to 5 alarm actions can be specified, periodic FTP is assigned to event No.6, and pre/post FTP assigned to No.7. Therefore the number of event(number) can be set between the range of 1 to 7. Note that alarm numbers are different from the alarm input pin numbers.

Specify sunday, monday, tuesday, wednesday, thursday, friday or saturday for WeekOfDay. Start time and end time is returned in the format like hhmmss-hhmmss. Start time can be from 000000 to 235959. End time can be from 000001 to 240000.

Allowed users admin, operator

Setting Alarm Filter Time

Format /api/param?application.event(Number).filter(WeekOfDay).time=data

Example When setting Sunday filter time of Alarm No. 1

/api/param?application.event(1).filter(sunday).time=010200-040500

Example of Response

application.event(1).filter(sunday).time&202 Accepted(application.event(1).status=restart)

Interpretation Set filter time of the alarm action for the specified alarm number. Up to 5 alarm actions can be specified, periodic FTP is assigned to event No.6, and pre/post FTP assigned to No.7. Therefore the number of event(number) can be set between the range of 1 to 7. Note that alarm numbers are different from the alarm input

pin numbers.

Specify sunday, monday, tuesday, wednesday, thursday, friday or saturday for WeekOfDay.

Specify start time and end time in the format like hhmmss-hhmmss. Start time can be from 000000 to 235959. End time can be from 000001 to 240000. Start time must be earlier than end time.

The changes to filter of alarm action is saved by /api/param?application.event(Number).status=restart.

Allowed users admin, operator

Getting Alarm Filter Type

Format **/api/param?application.event(Number).filter(WeekOfDay).type**

Example When Getting Type of Sunday filter of Alarm No. 1

/api/param?application.event(1).filter(sunday).type

Example of Response **application.event(1).filter(sunday).type=mask&200 OK**

Interpretation Acquire filter type of the alarm action for the specified alarm number. Up to 5 alarm actions can be specified, periodic FTP is assigned to event No.6, and pre/post FTP assigned to No.7. Therefore the number of event(number) can be set between the range of 1 to 7. Note that alarm numbers are different from the alarm input pin numbers.

Specify sunday, monday, tuesday, wednesday, thursday, friday or saturday for WeekOfDay. "mask" or "unmask" is returned. When the setting is mask, alarm action is disabled during the filter time. When the setting is unmask, alarm action is enabled during the filter time.

Allowed users admin, operator

Setting Alarm Filter Type

Format **/api/param?application.event(Number).filter(WeekOfDay).type=data**

Example When setting Sunday filter type of Alarm No. 1 to be unmask

/api/param?application.event(1).filter(sunday).type=unmask

Example of Response

application.event(1).filter(sunday).type&202 Accepted(application.event(1).status=restart)

Interpretation Set filter type of the alarm action for the specified alarm number. Up to 5 alarm actions can be specified, periodic FTP is assigned to event No.6, and pre/post FTP assigned to No.7. Therefore the number of event(number) can be set between the range of 1 to 7. Note that alarm numbers are different from the alarm input pin numbers.

Specify sunday, monday, tuesday, wednesday, thursday, friday or saturday for WeekOfDay.

Specify mask or unmask. When the setting is mask, alarm action is disabled during the filter time. When the setting is unmask, alarm action is enabled during the filter time.

The changes to filter of alarm action is saved by /api/param?application.event(Number).status=restart.

Allowed users admin, operator

Getting Alarm Trigger

Format **/api/param?application.event(Number).trigger**

Example When Getting Trigger of Alarm No. 1

/api/param?application.event(1).trigger

Example of Response **application.event(1).trigger=m1&200 OK**

Interpretation Acquire Trigger of the alarm action for the specified alarm number. Up to 5 alarm actions can be specified, periodic FTP is assigned to event No.6, pre/post FTP assigned to No.7, SD Card constant recording to event No.8 and SD Card alarm recording to No.10. Therefore the number of event(number) can be set between the range of 1 to 8 and 10. Note that alarm numbers are different from the alarm input pin numbers.

When only 1 Trigger is set:

m1 will be returned in the case of make for alarm input 1. **[VN-H57/157WP/257/257VP Only]**

b1 will be returned in the case of break for alarm input 1. **[VN-H57/157WP/257/257VP Only]**

m2 will be returned in the case of make for alarm input 2. **[VN-H57/157WP/257/257VP Only]**

b2 will be returned in the case of break for alarm input 2. **[VN-H57/157WP/257/257VP Only]**

camera.position(num).status will be returned for preset position. "num" is from 0 to 19.

v1 will be returned for motion detection of video.

audio_detect1 will be returned for audio detection1. **[VN-H57/157WP/257/257VP Only]**

audio_detect2 will be returned for audio detection2. **[VN-H57/157WP/257/257VP Only]**

tampering_detect will be returned for tampering detect of video.

ncbwe will be returned for IR filter ON.

ncbws will be returned for IR filter OFF.

i(second) will be returned for periodic FTP trigger.

time/hhmmss will be returned for time trigger.

When a combination of 2 Triggers are set, responses such as m1(10)b2 will be returned. The example indicates that trigger will be activated when break is invoked at alarm input 2 within 10 seconds after make is invoked at alarm input 1.

Example of Response **application.event(1).trigger=m1(100)b2&200 OK**

Allowed users admin, operator

Setting Alarm Trigger

Format **/api/param?application.event(Number).trigger=data**

Example When setting Trigger of Alarm No. 1

/api/param?application.event(1).trigger=m1

Example of Response

application.event(1).trigger&202 Accepted(application.event(1).status=restart)

Interpretation Set Trigger of the alarm action for the specified alarm number. Up to 5 alarm actions can be specified, and periodic FTP is assigned to event No.6, pre/post FTP assigned to No.7, SD Card constant recording is assigned to event No.8 and SD Card alarm recording is assigned to event No.10. Therefore the number of event(number) can be set between the range of 1 to 8 and 10. Note that alarm numbers are different from the alarm input pin numbers.

The changes to settings of alarm action become valid by /api/param?application.event(Number).status=restart.

When setting only 1 Trigger:

Specify m1 in the case of Make for alarm input 1. **[VN-H57/157WP/257/257VP Only]**

Specify b1 in the case of Break for alarm input 1. **[VN-H57/157WP/257/257VP Only]**

Specify m2 in the case of Make for alarm input 2. **[VN-H57/157WP/257/257VP Only]**

Specify b2 in the case of Break for alarm input 2. **[VN-H57/157WP/257/257VP Only]**

Specify camera.position(num).status for preset position. "num" is from 0 to 19.

Specify v1 for motion detection of video.

Specify audio_detect1 for audio detection 1. **[VN-H57/157WP/257/257VP Only]**

Specify audio_detect2 for audio detection 2. **[VN-H57/157WP/257/257VP Only]**

Specify tampering_detect for tampering detect of video.

Specify ncbwe for IR Filter ON.

Specify ncbws for IR Filter OFF.

Specify i(second) for periodic FTP trigger.

Specify time/hhmmss for time trigger.

Setting Example /api/param?application.event(1).trigger=v1

Interval can be set to periodic ftp assigned to event(6). Set "i1500" for interval 1500 seconds.

Setting Example /api/param?application.event(6).trigger=i1500

When setting Trigger upon combining 2 alarm inputs, specify as m1(50)b2. The example above indicates that trigger will be activated when break is invoked at alarm input 2 within 50 seconds after make is invoked at alarm input 1. Additionally, combination is only allowed for alarm inputs and not motion detect nor IR Filter. And same alarm can not be combined. For example, m1(50)m1 is not available.

Setting Example /api/param?application.event(1).trigger=m1(100)b2

Allowed users admin, operator

12. JVC API for Alarm Environment

The APIs below are related to alarm environment setting. These are equivalent to the features on the Alarm Environment page of the WEB setting page. Refer to the instruction manual for details on the Alarm Environment page.

Getting SMTP Server Address Setting

Format **/api/param?application.smtp.host**

Example of Response **application.smtp.host=192.168.0.200&200 OK**

Response example when setting field is left blank **application.smtp.host=&200 OK**

Interpretation Acquire the address setting of the SMTP server.

Allowed users admin, operator, user

Setting SMTP Server Address

Format **/api/param?application.smtp.host=data**

Example **/api/param?application.smtp.host=192.168.0.200**

Example of Response **application.smtp.host&200 OK**

Interpretation Change the address setting of the SMTP server. Specify the IP address or FQDN. The maximum FQDN size is 63 bytes. Specify as 0.0.0.0 when the SMTP server is not set. It is also possible to leave the setting field blank as follows. **/api/param?application.smtp.host=%00**

Allowed users admin, operator

Getting SMTP Server Port Number Setting

Format **/api/param?application.smtp.port**

Example of Response **application.smtp.port=25&200 OK**

Interpretation Acquire the port number setting of the SMTP server.

Allowed users admin, operator, user

Setting SMTP Server Port Number

Format **/api/param?application.smtp.port=data**

Example **/api/param?application.smtp.port=25**

Example of Response **application.smtp.port&200 OK**

Interpretation Change the port number setting of the SMTP server.

Allowed users admin, operator

Getting Sender Mail Address Setting

Format **/api/param?application.smtp.mailfrom**

Example of Response **application.smtp.mailfrom=somebody@somecompany.com&200 OK**

Interpretation Acquire sender mail address setting. POP user name is used as local part of sender mail address when sender mail address setting is blank. When POP user name is also blank, the local-part is set to "vn_h37@hostname". When the hostname is also blank, SMTP server decide sender mail address.

Allowed users admin, operator, user

Setting Sender Mail Address

Format **/api/param?application.smtp.mailfrom=data**

Example **/api/param?application.smtp.mailfrom=somebody@somecompany.com**

Example of Response **application.smtp.mailfrom&200 OK**

Interpretation Change sender mail address setting. Maximum text number of sender mail address is 96. Alphanumeric and followings are available.

! # $ % & ' * + - / = ? ^ _ ` { } | ~

POP user name is used as local part of sender mail address when sender mail address setting is blank. When POP user name is also blank, the local-part is set to "vn_h37@hostname". When the hostname is also blank, SMTP server decide sender mail address.

Allowed users admin, operator

Getting "POP before SMTP" Setting

Format **/api/param?application.smtp.type**

Example of Response **application.smtp.type=pbs&200 OK**

Interpretation Acquire the "POP before SMTP" setting. "simple" is returned when this is set to off. "pbs" is returned when this is set to on.

Allowed users admin, operator, user

Setting "POP before SMTP"

Format **/api/param?application.smtp.type=data**

Example **/api/param?application.smtp.type=pbs**

Example of Response **application.event.smtp.type&200 OK**

Interpretation Change the "POP before SMTP" setting. Specify as "simple" when setting to off and "pbs" when setting to on.

Allowed users admin, operator

Getting POP Server Address Setting

Format **/api/param?application.pop.host**

Example of Response **application.pop.host=192.168.0.200&200 OK**

Response example when setting field is left blank **application.pop.host=&200 OK**

Interpretation Acquire the address setting of the POP server.

Allowed users admin, operator, user

Setting POP Server Address

Format **/api/param?application.pop.host=data**

Example **/api/param?application.pop.host=192.168.0.200**

Example of Response **application.pop.host&200 OK**

Interpretation Change the address setting of the POP server. Specify the IP address or FQDN. The maximum FQDN size is 63 bytes. Specify as 0.0.0.0 when the POP server is not set. It is also possible to leave the setting field blank as follows. **/api/param?application.pop.host=%00**

Allowed users admin, operator

Getting POP Server Port Number Setting

Format **/api/param?application.pop.port**

Example of Response **application.pop.port=110&200 OK**

Interpretation Acquire the port number setting of the POP server.

Allowed users admin, operator, user

Setting POP Server Port Number

Format **/api/param?application.pop.port=data**

Example **/api/param?application.pop.port=110**

Example of Response **application.pop.port&200 OK**

Interpretation Change the port number setting of the POP server.

Allowed users admin, operator

Getting POP Server User Name Setting

Format **/api/param?application.pop.user**

Example of Response **application.pop.user=somename&200 OK**

Response example when setting field is left blank **application.pop.user=&200 OK**

Interpretation Acquire the user name setting of the POP server. The user name is used as local part of sender mail address when sender mail address setting is blank. When the user name is blank, the local-part is set to "vn_h37".

Example of Response **application.pop.user=somename&200 OK**

Example of Mail Address **somename@somecompany.com**

Allowed users admin, operator, user

Setting POP Server User Name

Format **/api/param?application.pop.user=data**

Example **/api/param?application.pop.user=somename**

Example of Response application.pop.user&200 OK

Interpretation Change the user name setting of the POP server. The maximum user name size is 64 bytes. Set as follows when this is to be left blank.

/api/param?application.pop.user=%00

The user name is used as local part of sender mail address when sender mail address setting is blank. When the user name is blank, the local-part is set to "vn_h37". When POP before SMTP is disabled, it is not necessary to set POP server settings other than POP user name setting.

Example of setting /api/param?application.pop.user=somename

Example of Mail Address somename@somecompany.com

Following characters must not be used in user name.

space ( ) < > [ ] : ; ¥ ,(comma)

Allowed users admin, operator

Setting POP Server Password

Format /api/param?application.pop.password=data

Example /api/param?application.pop.password=someword

Example of Response application.pop.password&200 OK

Interpretation Change the password setting of the POP server. The maximum password size is 32 bytes. Set as follows when this is to be left blank. **/api/param?application.pop.password=%00**

Allowed users admin, operator

(Note: There is no API for reading passwords.)

Getting FTP Server Address Setting

Format /api/param?application.ftp.host

Example of Response application.ftp.host=192.168.0.200&200 OK

Response example when setting field is left blank application.ftp.host=&200 OK

Interpretation Acquire the FTP server address setting used for FTP transmission via alarm.

Allowed users admin, operator, user

Setting FTP Server Address

Format /api/param?application.ftp.host=data

Example /api/param?application.ftp.host=10.0.0.200

Example of Response application.ftp.host&200 OK

Interpretation Change the FTP server address setting used for FTP transmission via alarm. Specify the IP address or FQDN. The maximum FQDN size is 63 bytes. Specify as 0.0.0.0 when the FTP server is not set. It is also possible to leave the setting field blank as follows. **/api/param?application.ftp.path=%00**

Allowed users admin, operator

Getting FTP Server Path Setting

Format /api/param?application.ftp.path

Example of Response application.ftp.path=subdir1&200 OK

Response example when setting field is left blank application.ftp.path=&200 OK

Interpretation Acquire the FTP server directory setting used for FTP transmission via alarm.

Allowed users admin, operator, user

Setting FTP Server Path

Format /api/param?application.ftp.path=data

Example /api/param?application.ftp.path=subdir1

Example of Response application.ftp.path&200 OK

Interpretation Change the FTP server directory setting used for FTP transmission. It is possible to set FTP transmission to a directory under the FTP server home directory by specifying that directory name. Use %2F to segment the directory. ("2F" is ASCII code of "/".) The maximum directory name size is 63 bytes.

Example /api/param?application.ftp.path=subdir1%2Fsubdir2

By leaving the setting blank as follows, FTP transmission will be set to the FTP server home directory.

/api/param?application.ftp.path=%00

Allowed users admin, operator

Getting FTP Server User Name Setting

Format /api/param?application.ftp.user

Example of Response application.ftp.user=somename&200 OK

Response example when setting field is left blank application.ftp.user=&200 OK

Interpretation Acquire the FTP server user name setting used for FTP transmission via alarm.

Allowed users admin, operator

Setting FTP Server User Name

Format /api/param?application.ftp.user=data

Example /api/param?application.ftp.user=somename

Example of Response application.ftp.user&200 OK

Interpretation Change the FTP server user name setting used for FTP transmission via alarm. The maximum user name size is 32 bytes. Set as follows when this setting is to be left blank.

/api/param?application.ftp.user=%00

Allowed users admin, operator

Setting FTP Server Password

Format **/api/param?application.ftp.password=data**

Example **/api/param?application.ftp.password=someword**

Example of Response **application.ftp.password&200 OK**

Interpretation Change the FTP server password setting used for FTP transmission via alarm. The maximum password size is 32 bytes. Set as follows when this setting is to be left blank.

/api/param?application.ftp.password=%00

Allowed users admin, operator

(There is no API for Getting passwords.)

Getting File Naming of Periodic FTP

Format **/api/param?application.ftp.naming**

Example of Response **application.ftp.naming=default&200 OK**

Interpretation Acquire file naming of periodic FTP. "default", "type1" or "type2" is returned. When default is set, the file name is as YYYYMMDDHHMMSS-NNN-2.jpg.

Example 20060207201315-001-2.jpg

When type1 is set, the file name is as ***YYYMMDDHHMMSSNNN.jpg. "***" can be gotten by another API, /api/param?application.ftp.naming_option.

File Name Example Camera_20060207201315001.jpg

When type2 is set, the file name is as ***.jpg. "***" can be gotten by another API, /api/param?application.ftp.naming_option.

File Name Example Camera.jpg

Allowed users admin, operator

Setting File Naming of Periodic FTP

Format **/api/param?application.ftp.naming=data**

Example **/api/param?application.ftp.naming=type1**

Example of Response **application.ftp.naming&200 OK**

Interpretation Change file naming of periodic FTP. Specify "default", "type1" or "type2". When default is set, the file name is as YYYYMMDDHHMMSS-NNN-2.jpg.

Example 20060207201315-001-2.jpg

When type1 is set, the file name is as ***YYYYMMDDHHMMSSNNN.jpg. "***" can be set by another API, /api/param?application.ftp.naming_option.

File Name Example Camera_20060207201315001.jpg

When type2 is set, the file name is as ***.jpg. "***" can be set by another API,

/api/param?application.ftp.naming_option.

File Name Example Camera.jpg

Allowed users admin, operator

Getting User Define Name of File Naming

Format /api/param?application.ftp.naming_option

Example of Response application.ftp.naming_option=abc&200 OK

Interpretation Acquire user define name for file naming of periodic FTP. The maximum size is 16 bytes. When /api/param?application.ftp.naming_option is set to "type1", the file name is as ***YYYMMDDHHMMSSNNN.jpg, and "***" can be gotten by this API.

File Name Example Camera_20060207201315001.jpg

When /api/param?application.ftp.naming_option is set to "type2", the file name is as ***.jpg and "***" can be gotten by this API.

File Name Example Camera.jpg

Allowed users admin, operator

Setting User Define Name of File Naming

Format /api/param?application.ftp.naming_option=data

Example of Response application.ftp.naming_option&200 OK

Interpretation Change user define name for file naming of periodic FTP. The maximum size is 16 bytes. When /api/param?application.ftp.naming_option is set to "type1", the file name is as ***YYYMMDDHHMMSSNNN.jpg, and "***" can be set by this API.

File Name Example Camera_20060207201315001.jpg

When /api/param?application.ftp.naming_option is set to "type2", the file name is as ***.jpg and "***" can be set by this API.

File Name Example Camera.jpg

Allowed users admin, operator

Getting Parameters of Pre/Post Recording for FTP

Format

To get Frame Rate /api/param?application.object(7).framerate

To get Pre Duration /api/param?application.object(7).prerec

To get Post Duration /api/param?application.object(7).postrec

To get Encoder No. /api/param?application.object(7).source

Example of Response

For Frame Rate application.object(7).framerate=10&200 OK

For Pre Duration **/api/param?application.object(7).prerec=2&200 OK**

For Post Duration **/api/param?application.object(7).postrec=2&200 OK**

For Encoder No. **/api/param?application.object(7).source=encode(1)&200 OK**

Interpretation Acquire parameters for PrePost.

Allowed users admin, operator, user

Setting Parameters of Pre/Post Recording for FTP

Format

To set Frame Rate **/api/param?application.object(7).framerate=5**

To set Pre Duration **/api/param?application.object(7).prerec=3**

To set Post Duration **/api/param?application.object(7).postrec=3**

To set Encoder No. **/api/param?application.object(7).source=encode(1)**

Example of Response

For Frame Rate **application.object(7).framerate&200 OK**

For Pre Duration **/api/param?application.object(7).prerec&200 OK**

For Post Duration **/api/param?application.object(7).postrec&200 OK**

For Encoder No. **/api/param?application.object(7).source&200 OK**

Interpretation Change parameters for PrePost.

Specify 30, 15, 10, 7.5, 6, 5, 3, 2, or 1 for frame rate. Maximum Pre/Post duration is 60 seconds. Setting zero to Pre and Post duration is invalid. Specify encode(1), encode(2), or encode(3) for encoder No. Pre/Post Recording for FTP is valid when encode type is set to JPEG.

Allowed users admin, operator

13. JVC API for SD Card Record

The APIs below are related to SD Card Recording. These are equivalent to the features on the SD Card Record page of the WEB setting page. Refer to the instruction manual for details on the SD Card Record page.

Getting SD Card Status

Format **/api/param?storage.disk(1).status**

Example of Response **storage.disk(1).status=on&200 OK**

Interpretation Acquire SD Card status. “on”, “empty”, “read_only”, “off”, “off_read_only”, or “off_empty” will be returned.

Return value	Use / Disable	Status
off_empty	Disable	No SD card
off_read_only	Disable	LOCK switch is enabled
off	Disable	LOCK switch is disabled
empty	Use	No SD card

read_only	**Use**	**LOCK switch is enabled**
on	**Use**	**LOCK switch is disabled**

Allowed users admin, operator, user

Setting SD Card to Use/Disable

Format /api/param?storage.disk(1).status=data

Example of Response storage.disk(1).status&200 OK

Interpretation Change the Use/Disable status of SD Card. Specify “on” or “off”. In case of “on”, SD Card can be use. In case of “off”, SD Card is disabled.

Allowed users admin, operator, user

Getting Status of SD Card formatting

Format /api/param?storage.disk(1).initialize

Example of Response storage.disk(1).initialize=on&200 OK

Interpretation Acquire status of SD Card formatting. “on”, “off”, or “not_initialized” will be returned. In case of “on”, SD Card formatting is in progress. In case of “off”SD Card is formatted. In case of “not_initialized”, SD Card is unformatted.

Allowed users admin, operator

Formatting SD Card

Format /api/param?storage.disk(1).initialize=data

/api/param?storage.disk(1).initialize=start

Example of Response storage.disk(1).initialize&200 OK

Interpretation Specify as start to format the SD. When this API is issued, the camera reboots in about 1 minute.

Allowed users admin, operator

Getting SD Card Constant Recording On/Off Status

Format /api/param?application.event(8).status

Example of Response application.event(8).status=on&200 OK

Interpretation Acquire the on/off status of SD Card Constant Recording. “on” or “off”will be returned.

Allowed users admin, operator, user

Setting SD Card Constant Recording On/Off

Format /api/param?application.event(8).status=data

Example of Response application.event(8).status&200 OK

Interpretation Change the on/off status of SD card Constant Recording. Specify "on" or "off" to change the status.

Allowed users admin, operator

Getting SD Card Capacity

Format /api/param?storage.disk(1).size

Example of Response storage.disk(1).size=30543M&200 OK

Interpretation Acquire the capacity of SD card in megabytes.

Allowed users admin, operator

Getting SD Card Recording Status

Format /api/param?storage.disk(1).rec

Example of Response storage.disk(1).rec=on&200 OK

Interpretation Acquire the status of SD card recording. “on” or “off”will be returned. In case of “on”, SD card recording is in progress. In case of “off”, SD card recording is stopping.

Allowed users admin, operator, user

Getting Encoder No. for SD Card Recording

Format /api/param?application.object(8).source

Example of Response application.object(8).source=encode(1)&200 OK

Interpretation Acquire the encoder No. for SD card recording. “encode(1)”, “encode(2)”, or “encode(3)”will be returned.

Allowed users admin, operator, user

Setting Encoder No. for SD Card Recording

Format /api/param?application.object(8).source=data

Example /api/param?application.object(8).source=encode(1)

Example of Response application.object(8).source&200 OK

Interpretation Change the encoder No. for SD card recording. Specify “encode(1)”, “encode(2)”, or “encode(3)”for encoder No. This parameter is valid when encode type is set to h264high or h264baseline. Depending on Recording Quality setting, specify encoder as below,

To set recording quality to High

/api/param?encode(3).type=h264high

/api/param?encode(3).framesize=1920x1080

/api/param?encode(3).framerate=5

/api/param?encode(3).bitrate=1000000

/api/param?encode(3).cbr_mode=cbr

/api/param?encode(3).iframeinterval=5

To set recording quality to Mid

/api/param?encode(3).type=h264high

/api/param?encode(3).framesize=1280x720

/api/param?encode(3).framerate=5

/api/param?encode(3).bitrate=768000

/api/param?encode(3).cbr_mode=cbr

/api/param?encode(3).iframeinterval=5

To set recording quality to Low

/api/param?encode(3).type=h264high

/api/param?encode(3).framesize=640x360

/api/param?encode(3).framerate=5

/api/param?encode(3).bitrate=128000

/api/param?encode(3).cbr_mode=cbr

/api/param?encode(3).iframeinterval=5

Allowed users admin, operator

14. JVC API for Digital PTZ

The APIs below are related to digital PTZ control. These are equivalent to the features on the PTZ page of the WEB setting page. Refer to the instruction manual for details on the PTZ page.

Basic authentication is required for JVC API explained in Section 7 or later. This section provides general explanation of those APIs. The API is valid when the resolution is 640x360 or 640x480.

(1) Settings for PTZ Control

Getting Auto Return Mode

Format /api/param?camera.motion.auto_return.mode

Example of response camera.motion.auto_return.mode=home&200 OK

Interpretation Acquire Auto Return mode. "home" or "auto_patrol(0)" will be returned.

Allowed users admin, operator, user

Setting Auto Return Mode

Format /api/param?camera.motion.auto_return.mode=data

Example of Response camera.motion.auto_return.mode&202 Accepted(camera.status=save)

Interpretation Change Auto Return mode. Specify "home" or "auto_patrol(0)". The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Timeout of Auto Return

Format **/api/param?camera.motion.auto_return.timeout**

Example of response **camera.motion.auto_return.timeout=60&200 OK**

Interpretation Acquire timeout of Auto Return in seconds.

Allowed users admin, operator, user

Setting Timeout of Auto Return

Format **/api/param?camera.motion.auto_return.timeout=data**

Example of Response **camera.motion.auto_return.timeout&202 Accepted(camera.status=save)**

Interpretation Change timeout of Auto Return in seconds. Specify 60, 120, 180, 300, 600, 1200, 1800 or 3600. The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Auto Return Status

Format **/api/param?camera.motion.auto_return.status**

Example of response **camera.motion.auto_return.status=on&200 OK**

Interpretation Acquire status of Auto Return. "on" or "off" will be returned.

Allowed users admin, operator, user

Setting Auto Return Status

Format **/api/param?camera.motion.auto_return.status=data**

Example of Response **camera.motion.auto_return.status&202 Accepted(camera.status=save)**

Interpretation Change status of Auto Return. Specify "on" or "off" to change the status. Specify "start" or "stop" for manual operation. "on" or "off" is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Speed of Going to Preset Position

Format **/api/param?camera.motion.position.speed**

Example of response **camera.motion.position.speed=100&200 OK**

Interpretation Acquire speed of going to preset position. Value from 0 to 100 is returned. 100 is fastest. The speed is 4 steps internally. The speed is applied also to preset position of auto patrol.

Allowed users admin, operator, user

Setting Speed of Going to Preset Position

Format **/api/param?camera.motion.position.speed=data**

Example to set horizontal **/api/param?camera.motion.position.speed=100**

Example of Response **camera.motion.position.speed&202 Accepted(camera.status=save)**

Interpretation Set speed of going to preset position. Specify from 0 to 100. 5 is horizontal. 100 is fastest. The speed is 4 steps internally. The speed is applied also to preset position of auto patrol. The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

(2) PTZ Control

Getting Pan Position

Format **/api/param?camera.motion.pan**

Example of response **camera.motion.pan=s100&200 OK**

Interpretation Acquire current pan position, left edge of current area, in pixels. Value from 0 to 1278 is returned. "s" is added before the value.

Allowed users admin, operator, user

Moving to Specified Pan Position

Format **/api/param?camera.motion.pan=data**

Example to move to absolute 100 pixels **/api/param?camera.motion.pan=s100**

Example to move to relative 45 pixels **/api/param?camera.motion.pan=+s45**

Example of Response **camera.motion.pan&200 OK**

Interpretation Move to specified pan position, left edge of target area, in pixels. To move to absolute position, specify from 0 to 1278 with "s". Moved position can be adjusted automatically to prevent showing invalid area.

Allowed users admin, operator

Pan Operation

Format **/api/param?camera.motion.pan.status=data**

Example to start pan **/api/param?camera.motion.pan.status=start**

Example of Response **camera.motion.pan.status&200 OK**

Interpretation Start or stop pan operation. Specify start or stop.

Allowed users admin, operator

Setting Direction of Pan Operation

Format **/api/param?camera.motion.pan.mode=data**

Example to set to left **/api/param?camera.motion.pan.mode=left**

Example of Response **camera.motion.pan.mode&200 OK**

Interpretation Set direction of pan operation. Specify left or right.

Allowed users admin, operator

Setting Speed of Pan Operation

Format **/api/param?camera.motion.pan.speed=data**

Example to set maximum speed **/api/param?camera.motion.pan.speed=100**

Example of Response **camera.motion.pan.speed&200 OK**

Interpretation Set speed of pan operation. Specify 0 to 100. The speed is 8 steps internally.

Allowed users admin, operator

Getting Pan Operation Status

Format **/api/param?camera.motion.pan.status**

Example of Response **camera.motion.pan.status=moving&200 OK**

Interpretation Acquire current pan status. "moving" or "stop" is returned.

Allowed users admin, operator, user

Getting Tilt Position

Format **/api/param?camera.motion.tilt**

Example of response **camera.motion.tilt=s45&200 OK**

Interpretation Acquire current tilt position, top edge of current area, in pixels. Value from 0 to 958 is returned. "s" is added before the value.

Allowed users admin, operator, user

Moving to Specified Tilt Position

Format **/api/param?camera.motion.tilt=data**

Example to move to absolute 100 pixels **/api/param?camera.motion.tilt=s100**

Example to move to relative 45 degrees **/api/param?camera.motion.tilt=+s45**

Example of Response **camera.motion.tilt&200 OK**

Interpretation Move to specified tilt position, top edge of target area, in pixels. To move to absolute position, specify from 0 to 958 with "s". Moved position can be adjusted automatically to prevent showing invalid area.

Allowed users admin, operator

Tilt Operation

Format **/api/param?camera.motion.tilt.status=data**

Example to start pan **/api/param?camera.motion.tilt.status=start**

Example of Response **camera.motion.tilt.status&200 OK**

Interpretation Start or stop tilt operation. Specify start or stop.

Allowed users admin, operator

Setting Direction of Tilt Operation

Format **/api/param?camera.motion.tilt.mode=data**

Example to set to up **/api/param?camera.motion.tilt.mode=up**

Example of Response **camera.motion.tilt.mode&200 OK**

Interpretation Set direction of tilt operation. Specify up or down.

Allowed users admin, operator

Setting Speed of Tilt Operation

Format **/api/param?camera.motion.tilt.speed=data**

Example to set maximum speed **/api/param?camera.motion.tilt.speed=100**

Example of Response **camera.motion.tilt.speed&200 OK**

Interpretation Set speed of tilt operation. Specify 0 to 100. The speed is 8 steps internally.

Allowed users admin, operator

Getting Tilt Operation Status

Format **/api/param?camera.motion.tilt.status**

Example of Response **camera.motion.tilt.status=moving&200 OK**

Interpretation Acquire current tilt status. "moving" or "stop" is returned.

Allowed users admin, operator, user

Getting Zoom Position

Format **/api/param?camera.motion.zoom**

Example of response **camera.motion.zoom=x2.00&200 OK**

Interpretation Acquire current zoom multiple. Value from 1.00 to 8.00 is returned with "x". The API is valid when the resolution is 640x360 or 640x480.

Allowed users admin, operator, user

Moving to Specified Zoom Position

Format **/api/param?camera.motion.zoom=data**

Example to move to absolute multiple, x2.0 **/api/param?camera.motion.zoom=x2.00**

Example to move to relative multiple, 1.5 Tele **/api/param?camera.motion.zoom=+x1.5**

Example to move to relative multiple, 1.5 Wide /api/param?camera.motion.zoom=-x1.5

Example of Response camera.motion.zoom&200 OK

Interpretation Move to specified zoom multiple. To move to absolute multiple, specify from 1.00 to 8.00 with "x". The API is valid when the resolution is 640x360 or 640x480.

Allowed users admin, operator

Zoom Operation

Format /api/param?camera.motion.zoom.status=data

Example to start zoom /api/param?camera.motion.zoom.status=start

Example of Response camera.motion.zoom.status&200 OK

Interpretation Start or stop zoom operation. Specify start or stop.

Allowed users admin, operator

Setting Direction of Zoom Operation

Format /api/param?camera.motion.zoom.mode=data

Example to set to Tele /api/param?camera.motion.zoom.mode=in

Example of Response camera.motion.zoom.mode&200 OK

Interpretation Set direction of zoom operation. Specify in or out.

Allowed users admin, operator

Setting Speed of Zoom Operation

Format /api/param?camera.motion.zoom.speed=data

Example to set maximum speed /api/param?camera.motion.zoom.speed=100

Example of Response camera.motion.zoom.speed&200 OK

Interpretation Set speed of zoom operation. Specify 0 to 100. The speed is 4 steps internally.

Allowed users admin, operator

Getting Zoom Operation Status

Format /api/param?camera.motion.zoom.status

Example of Response camera.motion.zoom.status=moving&200 OK

Interpretation Acquire current zoom status. "moving" or "stop" is returned.

Allowed users admin, operator, user

Moving Specified Position to Center

Format /api/param?camera.motion.clickoncenter=X-Y

Example of Response camera.motion.clickoncenter&200 OK

Example to move (958, 534) to center (pixel)

/api/param?camera.motion.clickoncenter=s958-s534

Example to move (958, 534) to center (percentage)

/api/param?camera.motion.clickoncenter=50.00-50.00

Interpretation Moving specified position to center of image. To move to X position, specify from s0 to s1918 or 0.00 to 100.00. To move to Y position, specify from s0 to s1078 or 0.00 to 100.00.

Allowed users admin, operator

(3) Preset Position

Getting Current Preset Position

Format /api/param?camera.status

Example of response camera.status=3&200 OK

Interpretation Acquire current preset position after moving to preset position. "none" is returned after moved from preset position.

Allowed users admin, operator, user

Getting Status of Specified Preset Position

Format /api/param?camera.position(number).status

Example of response camera.position(3).status=unregistered&200 OK

Interpretation Acquire current status of specified preset position. Specify from 0 to 19 as position number. "unregistered" or "registered" is returned.

Allowed users admin, operator, user

Register Current Position as Preset Position

Format /api/param?camera.position(number).status=save

Example of Response camera.position(3).status&200 OK

Interpretation Save current position as preset position. Specify from 0 to 19 as position number.

Allowed users admin, operator

Initialize Preset Position

Format /api/param?camera.position(number).status=initialize

Example of Response camera.position(3).status&200 OK

Interpretation Initialize specified preset position. Specify from 0 to 19 as position number. Position number 0 is

home position and it is registered with default settings when initilized. Other positions are unregistered by initializing.

Allowed users admin, operator

Moving to Preset Position

Format /api/param?camera.position(number).status=goto

Example of Response camera.position(3).status&200 OK

Interpretation Move to specified preset position. Specify from 0 to 19 as position number.

Allowed users admin, operator

Getting Title of Preset Position

Format /api/param?camera.position(number).comment

Example of response camera.position(3).comment=entrance&200 OK

Interpretation Acquire title of specified preset position. Specify from 0 to 19 as position number.

Allowed users admin, operator, user

Setting Title to Preset Position

Format /api/param?camera.position(number).comment=data

Example of Response camera.position(3).status&200 OK

Interpretation Set tilte to specified preset position. Specify from 0 to 19 as position number. Maximum characters is 32. To erase title, specify %00, i.e. 0x25 0x30 0x30 in binary data. Use %20 to set space.

Allowed users admin, operator

15. JVC API for Auto Patrol

The APIs below are related to Auto Patrol. These are equivalent to the features on the AutoPatrol page of the WEB setting page. Refer to the instruction manual for details on the AutoPatrol page page.

Start/Stop of Auto Patrol

Format /api/param?camera.motion.auto_patrol(0).status=data

Example to start auto patrol /api/param?camera.motion.auto_patrol(0).status=start

Example of Response camera.motion.auto_patrol(0).status&200 OK

Interpretation Start/stop a mode of auto patrol. Specify start or stop.

Allowed users admin, operator

Getting Status of Auto Patrol

Format /api/param?camera.motion.auto_patrol(0).status

Example of response **camera.motion.auto_patrol(0).status=moving&200 OK**

Interpretation Acquire current status of auto patrol. "moving" or "stop" is returned.

Allowed users admin, operator, user

Getting Preset Postion Number of Auto Patrol

Format **/api/param?camera.motion.auto_patrol(0).position(number)**

Example to get preset position number of patrol nuber 3

/api/param?camera.motion.auto_patrol(0).position(3)

Example of response **camera.motion.auto_patrol(0).position(3)=5&200 OK**

Interpretation Acquire preset position number of specified patrol number of auto patrol. Patrol number is from 0 to 19. Preset position number from 0 to 19 is returned.

Allowed users admin, operator, user

Setting Preset Postion Number of Auto Patrol

Format **/api/param?camera.motion.auto_patrol(0).position(number)=data**

Example of Response **camera.motion.auto_patrol(0).position(3)&202**

Accepted(camera.motion.auto_patrol.status=save)

Interpretation Set preset position number of specified patrol number of auto patrol. Patrol number is from 0 to 19. Specify preset position number from 0 to 19. The change is saved by the API, camera.motion.auto_patrol.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Duration of Auto Patrol

Format **/api/param?camera.motion.auto_patrol(0).position(number).duration**

Example to get duration of patrol nuber 3

/api/param?camera.motion.auto_patrol(0).position(3).duration

Example of response **camera.motion.auto_patrol(0).position(3).duration=30&200 OK**

Interpretation Acquire duration of specified patrol number of auto patrol. Patrol number is from 0 to 19. 0, 10, 20, 30, 45, 60, or 120 is returned. 0 means skip. 10 means 10 seconds.

Allowed users admin, operator, user

Setting Duarion of Auto Patrol

Format **/api/param?camera.motion.auto_patrol(0).position(number).duration=data**

Example of Response **camera.motion.auto_patrol(0).position(3).duration&202**

Accepted(camera.motion.auto_patrol.status=save)

Interpretation Set duration of specified patrol number of auto patrol. Patrol number is from 0 to 19. Specify 0, 10,

20, 30, 45, 60, or 120. 0 means skip. 10 means 10 seconds. The change is saved by the API, camera.motion.auto_patrol.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Speed of Auto Patrol

Format /api/param?camera.motion.auto_patrol(0).position(number).speed

Example to get speed of patrol nuber 3

/api/param?camera.motion.auto_patrol(0).position(3).speed

Example of response camera.motion.auto_patrol(0).position(3).speed=30.00&200 OK

Interpretation Acquire speed of specified patrol number of auto patrol. Patrol number is from 0 to 19.

Allowed users admin, operator, user

Setting Speed of Auto Patrol

Format /api/param?camera.motion.auto_patrol(0).position(number).speed=data

Example to set maximum speed from patrol nuber 3 to 4

/api/param?camera.motion.auto_patrol(0).position(3).speed=100.00

Example of Response camera.motion.auto_patrol(0).position(3).speed&202

Accepted(camera.motion.auto_patrol.status=save)

Interpretation Set speed of specified patrol number of auto patrol. Patrol number is from 0 to 19. Specify from 0.00 to 100.00. 0 means skip. The change is saved by the API, camera.motion.auto_patrol.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Saving Preset Position Number/Duarion of Auto Patrol

Format /api/param?camera.motion.auto_patrol(0).status=save

Example of Response camera.motion.auto_patrol(0).status&202

Accepted(camera.motion.auto_patrol.status=save)

Interpretation Save preset position number and duration of auto patrol.

Allowed users admin, operator

16. JVC API for Privacy Masking

The APIs below are related to privacy masking. These are equivalent to the features on the Privacy Masking page of the WEB setting page. Refer to the instruction manual for details on the Privacy Masking page.

Getting Privacy Masking On/Off Status

Format **/api/param?camera.private_mask.status**

Example of response **camera.private_mask.status=on&200 OK**

Interpretation Acquire the on/off status of privacy masking.

Allowed users admin, operator, user

Setting Privacy Masking to On/Off

Format **/api/param?camera.private_mask.status=data**

Example of Response **camera.private_mask.status&202 Accepted(camera.status=save)**

Interpretation Change the on/off status of privacy masking. The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Privacy Masking Color

Format **/api/param?camera.private_mask.color**

Example of response **camera.private_mask.color=ffffff&200 OK**

Interpretation Acquire the color of privacy masking. RGB values are returned as hexadecimal number. For exmaple, ffffff is white, ff0000 is red, 00ff00 is green, and 0000ff is blue.

Allowed users admin, operator, user

Setting Privacy Masking Color

Format **/api/param?camera.private_mask.color=data**

Example of Response **camera.private_mask.color&202 Accepted(camera.status=save)**

Interpretation Change the color of privacy masking. Specify RGB values by hexadecimal number. For exmaple, ffffff for white, ff0000 for red, 00ff00 for green, and 0000ff for blue.The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Privacy Masking Area

Format **/api/param?camera.private_mask.area**

Example of response **camera.private_mask.area=ffffff,,,f&200 OK**

Interpretation Acquire the area of privacy masking. 510 characters are returned as hexadecimal number that show bitmap. A bit for privacy masking is 32x32 pixels block, and 1920x1080 is divided to 60x34 blocks. For example, f means 8 blocks are masked.

Allowed users admin, operator, user

Setting Privacy Masking Color

Format **/api/param?camera.private_mask.area=data**

Example of Response **camera.private_mask.area&202 Accepted(camera.status=save)**

Interpretation Change the area of privacy masking. Specify bitmap by 510 characters of hexadecimal number. A bit for privacy masking is 32x32 pixels block, and 1920x1080 is divided to 60x34 blocks. For example, specify f to mask 8 blocks. The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

17. JVC API for Motion Detect

The APIs below are related to motion detection. These are equivalent to the features on the Motion Detection page of the WEB setting page. Refer to the instruction manual for details on the Motion Detection page.

Getting Motion Detect On/Off Status

Format **/api/param?camera.detection.status**

Example of response **camera.detection.status=on&200 OK**

Interpretation Acquire the on/off status of motion detect.

Allowed users admin, operator, user

Setting Motion Detect to On/Off

Format **/api/param?camera.detection.status=data**

Example of Response **camera.detection.status&202 Accepted(camera.status=save)**

Interpretation Change the on/off status of motion detect. The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Motion Detect Sensitivity

Format **/api/param?camera.detection.level**

Example of response **camera.detection.level=20&200 OK**

Interpretation Acquire the motion detect sensitivity. A value between 0 to 100 will be returned. The larger the value, the higher will be the sensitivity.

Allowed users admin, operator, user

Setting Motion Detect Sensitivity

Format **/api/param?camera.detection.level=data**

Example of response **camera.detection.level&202 Accepted(camera.status=save)**

Interpretation Change the motion detect sensitivity. Specify a value between 0 to 100. The larger the value, the higher will be the sensitivity. The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Motion Detect Mask

Format /api/param?camera.detection.area

Example of response camera.detection.area=00010203040506070809,,,&200 OK

Interpretation Acquire the mask of motion detect. 20 ASCII characters will be returned.

The screen is made up of 15x 9 = 135 blocks, and mask can be set to on/off for each block. This information can be represented in 135 bits = 17-byte hexadecimal. (Response is returned in ASCII character strings. Therefore, 34 characters will be returned.) The bit string will appear as follows when mask is set to off for the top left block only.

10000000 00000000 00000000 ,,,

Storage in bytes will begin from the LSB and represented in a hexadecimal value as shown below.

01 00 00 00 00 00 00 00 00 00,,,

The hexadecimal value denotes the 34 ASCII characters acquired via this API that are expressed in ASCII codes. For example, the following character string will be returned when only the top left and bottom right blocks are masked.

camera.detection.area=0100000000000000,,,0080

Allowed users admin, operator, user

Setting Motion Detect Mask

Format /api/param?camera.detection.area=data

Example /api/param?camera.detection.area=00010203040506070809,,,

Example of response camera.detection.area&202 Accepted(camera.status=save)

Interpretation Change the motion detect mask. Specify using a 34 ASCII character string. Refer to the item on "Getting Motion Mask" on the interpretation of this character string. To mask all blocks, specify all zeros in the ASCII character string. The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

18. JVC API for Tampering Detect

The APIs below are related to the Tampering detection. These are equivalent to the features on the Tampering Detection page of the WEB setting page. Refer to the instruction manual for details on the Tampering Detection

page.

Getting Tampering Detect On/Off Status

Format **/api/param?camera.detection(tampering).status**

Example of response **camera.detection(tampering).status=on&200 OK**

Interpretation Acquire the on/off status of tampering detect.

Allowed users admin, operator, user

Setting Tampering Detect to On/Off

Format **/api/param?camera.detection(tampering).status=data**

Example of Response **camera.detection(tampering).status&202 Accepted(camera.status=save)**

Interpretation Change the on/off status of tampering detect. The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Tampering Detect Sensitivity

Format **/api/param?camera.detection(tampering).level**

Example of response **camera.detection(tampering).level=20&200 OK**

Interpretation Acquire the tampering detect sensitivity. A value between 0 to 100 will be returned. The larger the value, the higher will be the sensitivity.

Allowed users admin, operator, user

Setting Tampering Detect Sensitivity

Format **/api/param?camera.detection(tampering).level=data**

Example of response **camera.detection(tampering).level&202 Accepted(camera.status=save)**

Interpretation Change the tampering detect sensitivity. Specify a value between 0 to 100. The larger the value, the higher will be the sensitivity. The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

Getting Tampering Detect time

Format **/api/param?camera.detection(tampering).temporal**

Example of response **camera.detection(tampering).level=5&200 OK**

Interpretation Acquire the tampering detect time. A value between 0 to 120 will be returned.

Allowed users admin, operator, user

Setting Tampering Detect Sensitivity

Format **/api/param?camera.detection(tampering).temporal=data**

Example of response **camera.detection(tampering).temporal&202 Accepted(camera.status=save)**

Interpretation Change the tampering detect time. Specify a value between 0 to 120. The change is saved by the API, camera.status=save. If the change is not saved, the setting is restored by reboot.

Allowed users admin, operator

19. JVC API for Network Basics

The APIs below are related to the basics of networks. These are equivalent to the features on the Basic page of the WEB setting page. Refer to the instruction manual for details on the Basic page.

Enabling Network Setting Changes

Format **/api/param?network.interface.status=restart**

Example of Response **network.interface.status&200 OK**

Interpretation Changes of following network parameters become valid by this API.

DHCP, IP Address, Subnet Mask, TTL, MTU, TOS, Negotiation, IPv6

Changes are not reflected in the actions until this API is used. APIs to get settings of those parameters return previous values until this API is used. When this API is issued, the camera reboots in about 1 minute.

Allowed user admin

Getting DHCP Setting

Format **/api/param?network.interface.dhcp.status**

Example of Response **network.interface.dhcp.status=off&200 OK**

Interpretation Acquire the current DHCP setting.

Allowed users admin, operator, user

Setting DHCP

Format **/api/param?network.interface.dhcp.status=data**

Example **/api/param?network.interface.dhcp.status=on**

Example of Response

network.interface.dhcp.status&202 Accepted(network.interface.status=restart)

Interpretation Change the DHCP setting. Specify "on" or "off". To validate the change, use "network.interface.status=restart" API that reboots the camera in about 1 minute.

Allowed user admin

Getting IP Address

Format **/api/param?network.interface.ip**

Example of Response **network.interface.ip=192.168.0.2&200 OK**

Interpretation Acquire the current IP address.

Allowed users admin, operator, user

Setting IP Address

Format **/api/param?network.interface.ip=data**

Example **/api/param?network.interface.ip=192.168.0.2**

Example of Response

network.interface.ip&202 Accepted(network.interface.status=restart)

Interpretation Change the IP address. To validate the change, use "network.interface.status=restart" API that reboots the camera in about 1 minute. Set appropriate combination of IP address, subnet mask and default gateway before "network.interface.status=restart".

Allowed user admin

Getting Subnet Mask

Format **/api/param?network.interface.subnetmask**

Example of Response **network.interface.subnetmask=255.255.255.0&200 OK**

Interpretation Acquire the current subnet mask.

Allowed users admin, operator, user

Setting Subnet Mask

Format **/api/param?network.interface.subnetmask=data**

Example **/api/param?network.interface.subnetmask=255.0.0.0**

Example of Response

network.interface.subnetmask&202 Accepted(network.interface.status=restart)

Interpretation Change the subnet mask. To validate the change, use "network.interface.status=restart" API that reboots the camera in about 1 minute. Set appropriate combination of IP address, subnet mask and default gateway before "network.interface.status=restart".

Allowed user admin

Getting Default Gateway

Format **/api/param?network.gateway(version)**

Example to get default gateway of IPv4 **/api/param?network.gateway(ipv4)**

Example of Response **network.gateway(ipv4)=192.168.0.254&200 OK**

Interpretation Acquire the current default gateway. Specify ipv4 or ipv6.

Allowed users admin, operator, user

Setting Default Gateway

Format /api/param?network.gateway(ipv4)=data

Example /api/param?network.gateway(ipv4)=192.168.0.254

Example of Response network.gateway&200 OK

Interpretation Change the default gateway. To set static default gateway, disable DHCP. Default gateway can not be changed when DHCP is enabled. Specify IP address in same segment with the camera. Specify 0.0.0.0 to delete default gateway setting. Default gateway of IPv6 can not be set.

Allowed user admin

Getting Host Name

Format /api/param?network.hostname

Example of Response network.hostname=localhost&200 OK

Interpretation Acquire the current host name.

Allowed users admin, operator, user

Setting Host Name

Format /api/param?network.hostname=data

Example /api/param?network.hostname=somename

Example of Response network.hostname&200 OK

Interpretation Change the host name. Characters that may be used for the host name are alphanumerics, hyphens (-) and period. Maximum size is 63 bytes.

Specify as %00 when the host name setting is to be left blank.

Example when leaving field blank /api/param?network.hostname=%00

Allowed user admin

Getting DNS Server On/Off Status

Format /api/param?network.dns.status

Example of Response network.dns.status=off&200 OK

Interpretation Acquire the on/off status of the DNS server. Either on or off will be returned.

Allowed users admin, operator, user

Setting DNS Server Status to On/Off, or Validate Changes

Format /api/param?network.dns.status=data

Example **/api/param?network.dns.status=on**

Example of Response **network.dns.status&200 OK**

Interpretation Change status of DNS server setting, or validate changes to DNS server settings. Specify "on", "off" or "restart". Changes of DNS server settings become valid by "restart".

Allowed users admin, operator

Getting DNS Server IP Address

Format **/api/param?network.dns.ip**

Example of Response **network.dns.ip=10.0.0.150&200 OK**

Interpretation Acquire IP address of DNS server.

Allowed users admin, operator, user

Setting DNS Server IP Address

Format **/api/param?network.dns.ip=data**

Example **/api/param?network.dns.ip=10.0.0.150**

Example of Response

network.dns.ip&202 Accepted(network.dns.status=restart)

Interpretation Change IP address of DNS server. To validate the change, use "network.dns.status=restart" API.

Allowed users admin, operator

Getting IPv6 status

Format **/api/param?network.interface.ipv6.status**

Example of Response **network.interface.ipv6.status=off&200 OK**

Interpretation Acquire IPv6 status. "on" or "off" is returned.

Allowed users admin, operator, user

Setting IPv6 status

Format **/api/param?network.interface.ipv6.status=data**

Example **/api/param?network.interface.ipv6.status=on**

Example of Response

network.interface.ipv6.status&202 Accepted(network.dns.status=restart)

Interpretation Change IPv6 status. To validate the change, use "network.interface.status=restart" API that reboots the camera in about 1 minute.

Allowed users admin, operator

Getting Link Local Address of IPv6

Format **/api/param?network.interface.ipv6.link_local(Number)**

Exampleto get first link local address **/api/param?network.interface.ipv6.link_local(1)**

Example of Response **network.interface.ipv6.link_local(1)=fe80::280:88ff:fe41:400c&200 OK**

Interpretation Acquire the link local address of IPv6. Specify from 1 to 8 for Number, and get addresses from 1 till vacant address is returned. There is no API for setting link local address of IPv6.

Allowed users admin, operator, user

Getting Global Address of IPv6

Format **/api/param?network.interface.ipv6.global(Number)**

Exampleto get first global address **/api/param?network.interface.ipv6.global(1)**

Example of Response when no global address is set network.interface.ipv6.global(1)=&200 OK

Interpretation Acquire the global address of IPv6. Specify from 1 to 8 for Number, and get addresses from 1 till vacant address is returned. There is no API for setting global address of IPv6.

Allowed users admin, operator, user

Getting MAC Address

Format **/api/param?network.interface.mac**

Example of Response **network.interface.mac=008088001AEF&200 OK**

Interpretation Acquire the MAC address. A 12-byte ASCII character string will be returned. There is no API for setting MAC address.

Allowed users admin, operator, user

Getting TOS Value of Stream

Format **/api/param?network.interface.dscp**

Example of Response **network.interface.dscp =56&200 OK**

Interpretation Acquire TOS that includes DHCP field.

Allowed users admin, operator, user

Setting TOS Value of Stream

Format **/api/param?network.interface.dscp =data**

Example **/api/param?network.interface.dscp =56**

Example of Response

network.interface.dscp&202 Accepted(network.interface.status=restart)

Interpretation Change TOS that includes DHCP field. The range of set value is between 0 to 255 though MSB 6 bits in the value is valid. To validate the change, use "network.interface.status=restart" API.

Allowed user admin

Getting Unicast TTL Value

Format **/api/param?network.interface.ttl.unicast**

Example of Response **network.interface.ttl.unicast=16&200 OK**

Interpretation Acquire TTL of unicast. 1-255 is returned.

Allowed users admin, operator, user

Setting Unicast TTL

Format **/api/param?network.interface.ttl.unicast=data**

Example **/api/param?network.interface.ttl.unicast=56**

Example of Response

network.interface.ttl.unicast&202 Accepted(network.interface.status=restart)

Interpretation Change TTL of unicast. The range of set value is between 1 to 255. To validate the change, use "network.interface.status=restart" API.

Allowed user admin

Getting Multicast TTL Value

Format **/api/param?network.interface.ttl.multicast**

Example of Response **network.interface.ttl.multicast=16&200 OK**

Interpretation Acquire TTL of multicast. 1-255 is returned.

Allowed users admin, operator, user

Setting Multicast TTL

Format **/api/param?network.interface.ttl.multicast=data**

Example **/api/param?network.interface.ttl.multicast=56**

Example of Response

network.interface.ttl.multicast&202 Accepted(network.interface.status=restart)

Interpretation Change TTL of multicast. The range of set value is between 1 to 255. To validate the change, use "network.interface.status=restart" API.

Allowed user admin

Getting MTU Value

Format **/api/param?network.interface.mtu**

Example of Response **network.interface.mtu=1420&200 OK**

Interpretation Acquire the MTU value.

Allowed users admin, operator, user

Setting MTU Value

Format **/api/param?network.interface.mtu=data**

Example **/api/param?network.interface.mtu=1500**

Example of Response

network.interface.mtu&202 Accepted(network.interface.status=restart)

Interpretation Change the MTU value. The range of set value is between 1280 to 1500. To validate the change, use "network.interface.status=restart" API.

Allowed user admin

Getting Network Negotiation Setting

Format **/api/param?network.interface.negotiation**

Example of Response **network.interface.negotiation=auto&200 OK**

Interpretation Acquire the network Negotiation setting. Either auto, 100full, 100half, 10full or 10half will be returned.

Allowed users admin, operator, user

Setting Network Negotiation

Format **/api/param?network.interface.negotiation=data**

Example **/api/param?network.interface.negotiation=auto**

Example of Response

network.interface.negotiation&202 Accepted(network.interface.status=restart)

Interpretation Change the network Negotiation setting. Specify auto, 100full, 100half, 10full or 10half. To validate the change, use "network.interface.status=restart" API.

Allowed user admin

20. JVC API for Protocol

The APIs below are related to protocol. These are equivalent to the features on the Protocol page of the WEB setting page. Refer to the instruction manual for details on the Protocol page.

Getting Port Number of HTTP

Format **/api/param?network.http.port**

Example of Response **network.http.port=80&200 OK**

Interpretation Acquire port number of HTTP server in the camera.

Allowed users admin, operator

Setting Port Number of HTTP

Format **/api/param?network.http.port=data**

Example of Response **network.http.port&202 Accepted(network.http(configuration).status=restart)**

Interpretation Change port number of HTTP server in the camera. Default value is 80. To validate the change, use "network.http(configuration).status=restart" or "network.http.status=restart" API.

Allowed users admin, operator

Getting Status of AMX Discovery Protocol

Format **/api/param?network.amx.beacon.status**

Example of Response **network.amx.beacon.status=on&200 OK**

Interpretation Acquire status of AMX Discovery Protocol in the camera. "on" or "off" is returned.

Allowed users admin, operator

Setting Status of AMX Discovery Protocol

Format **/api/param?network.amx.beacon.status=data**

Example **/api/param?network.amx.beacon.status=on**

Example of Response **network.amx.beacon.status&200 OK**

Interpretation Change status of AMX Discovery Protocol in the camera. Specify "on" for interoperability with AMX products.

Allowed users admin, operator

Getting Status of PSIA Protocol

Format **/api/param?network.psia.status**

Example of Response **network.psia.status=on&200 OK**

Interpretation Acquire status of PSIA Protocol in the camera. "on" or "off" is returned.

Allowed users admin, operator, user

Setting Status of PSIA Protocol

Format **/api/param?network.psia.status=data**

Example **/api/param?network.psia.status=on**

Example of Response **network.psia.status&200 OK**

Interpretation Change status of PSIA protocol in the camera. Specify "on" or “off”. When status of ONVIF protocol is set to “on”, status of PSIA protocol will not be “on”.

Allowed users admin, operator

Getting Status of ONVIF Protocol

Format **/api/param?network.onvif.status**

Example of Response **network.onvif.status=on&200 OK**

Interpretation Acquire status of ONVIF Protocol in the camera. "on" or "off" is returned.

Allowed users admin, operator, user

Setting Status of ONVIF Protocol

Format **/api/param?network.onvif.status=data**

Example **/api/param?network.onvif.status=on**

Example of Response **network.onvif.status&200 OK**

Interpretation Change status of ONVIF protocol in the camera. Specify "on" or “off”. When status of PSIA protocol is set to “on”, status of ONVIF protocol will not be “on”.

Allowed users admin, operator

21. JVC API for Multicast Streaming

The APIs below are related to manual streaming. These are equivalent to the features on the Streaming page of the WEB setting page. Refer to the instruction manual for details on the Streaming page.

Getting Status of Multicast Streaming

Format **/api/param?network.destination(num).status**

Example of Response **network.destination(1).status=off&200 OK**

Interpretation Acquire status of multicast streaming. “num” is encoder channel from 1 to 3. Either on or off will be returned.

Allowed users admin, operator

Setting Status of Multicast Streaming, or Save Changes

Format **/api/param?network.destination(num).status=data**

Example **/api/param?network.destination(1).status=start**

Example of Response **network.destination(1).status&200 OK**

Interpretation Start/stop multicast streaming of specified encode channel, or save changes to multicast streaming settings. “num” is encoder channel from 1 to 3. Specify "start", "stop" or "save". Changes of multicast streaming settings become valid by "save".

Multicast stream is RTP compliant.

If power becomes off during multicast streaming, the streaming starts automatically after power on.

Allowed users admin, operator

Getting Multicast Address

Format **/api/param?network.destination(num).host**

Example of Response **network.destination(1).host=225.0.1.1&200 OK**

Interpretation Acquire multicast address of specified encode channel. “num” is encoder channel from 1 to 3.

Allowed users admin, operator

Setting Multicast Address

Format **/api/param?network.destination(num).host=data**

Example **/api/param?network.destination(1).host=225.0.1.1**

Example of Response

network.destination(1).host&202 Accepted(network.destination(1).host=save)

Interpretation Change multicast address of specified encode channel. “num” is encoder channel from 1 to 3. Specify from 224.0.0.0 to 239.255.255.255. To validate the change, use "network.destination(num).host=save" API. After the save, start streaming by "network.destination(num).host=start" API.

Allowed user admin

Getting Multicast Port Number

Format **/api/param?network.destination(num).port**

Example of Response **network.destination(1).port=49152&200 OK**

Interpretation Acquire multicast port number of specified encode channel. “num” is encoder channel from 1 to 3.

Allowed users admin, operator

Setting Multicast Port Number

Format **/api/param?network.destination(num).port=data**

Example **/api/param?network.destination(1).port=49152**

Example of Response

network.destination(1).port&202 Accepted(network.destination(1).host=save)

Interpretation Change multicast port number of specified encode channel. “num” is encoder channel from 1 to 3. Specify from 2 to 65534. To validate the change, use "network.destination(num).host=save" API. After the save, start streaming by "network.destination(num).host=start" API.

Allowed user admin

Getting Frame Rate of JPEG Multicast

Format **/api/param?network.destination(num).framerate**

Example of Response **network.destination(1).framerate=10&200 OK**

Interpretation Acquire JPEG multicast frame rate of specified encode channel. “num” is encoder channel from 1 to 3. The API is valid when the encoder channel is set to JPEG.

Allowed users admin, operator

Setting Frame Rate of JPEG Multicast

Format /api/param?network.destination(num).framerate=data

Example /api/param?network.destination(1).framerate=30

Example of Response

network.destination(1).framerate&202 Accepted(network.destination(1).host=save)

Interpretation Change JPEG multicast frame rate of specified encode channel. “num” is encoder channel from 1 to 3. The API is valid when the encoder channel is set to JPEG. Specify 30, 15, 10, 7.5, 6, 5, 3, 2, 1, -2, -3, -5, -10, -15, -20, or -30. -5 means 1/5fps for example. To validate the change, use "network.destination(num).host=save" API. After the save, start streaming by "network.destination(num).host=start" API.

Allowed user admin

Getting Status of Audio Multicast Streaming

Format /api/param?network.destination(4).status

Example of Response network.destination(4).status=off&200 OK

Interpretation Acquire status of audio multicast streaming. Either on or off will be returned.

Allowed users admin, operator

Setting Status of Audio Multicast Streaming, or Save Changes

Format /api/param?network.destination(4).status=data

Example /api/param?network.destination(4).status=start

Example of Response network.destination(4).status&200 OK

Interpretation Start/stop audio multicast streaming, or save changes to multicast streaming settings. Specify "start", "stop" or "save". Changes of multicast streaming settings become valid by "save".

Multicast stream is RTP compliant. If power becomes off during multicast streaming, the streaming starts automatically after power on.

Allowed users admin, operator

Getting Audio Multicast Address

Format /api/param?network.destination(4).host

Example of Response network.destination(4).host=225.0.1.3&200 OK

Interpretation Acquire audio multicast address.

Allowed users admin, operator

Setting Audio Multicast Address

Format **/api/param?network.destination(4).host=data**

Example **/api/param?network.destination(4).host=225.0.1.3**

Example of Response

network.destination(4).host&202 Accepted(network.destination(4).host=save)

Interpretation Change audio multicast address. Specify from 224.0.0.0 to 239.255.255.255. To validate the change, use "network.destination(4).host=save" API. After the save, start streaming by "network.destination(4).host=start" API.

Allowed user admin

Getting Audio Multicast Port Number

Format **/api/param?network.destination(4).port**

Example of Response **network.destination(4).port=39152&200 OK**

Interpretation Acquire audio multicast port number.

Allowed users admin, operator

Setting Audio Multicast Port Number

Format **/api/param?network.destination(4).port=data**

Example **/api/param?network.destination(4).port=39152**

Example of Response

network.destination(4).port&202 Accepted(network.destination(4).host=save)

Interpretation Change audio multicast port number. Specify from 2 to 65534. To validate the change, use "network.destination(4).host=save" API. After the save, start streaming by "network.destination(4).host=start" API.

Allowed user admin

22. JVC API for Access Restrictions

The APIs below are related to access restrictions. These are equivalent to the features on the Access Restrictions page of the WEB setting page. Refer to the instruction manual for details on the Access Restrictions page.

Getting Deny/Allow Setting of Client Restrictions

Format **/api/param?network.access_control(stream_out).logic**

Example of Response **network.access_control(stream_out).logic=deny&200 OK**

Interpretation Acquire the deny/allow setting of client restrictions. Either deny or allow will be returned. These

restrictions are applied to getting video stream and bi-directional Audio.

Allowed users admin, operator

Setting Client Restriction to Deny/Allow

Format /api/param?network.access_control(stream_out).logic=data

Example /api/param?network.access_control(stream_out).logic=deny

Example of Response network.access_control(stream_out).logic&200 OK

Interpretation Change the deny/allow setting of client restrictions. Specify as deny or allow. These restrictions are applied to getting video stream and bidirectional Audio.

Allowed user admin

Getting IP Address Setting of Restricted Client

Format /api/param?network.access_control(stream_out).host(Number)

Example When Getting the first IP address

/api/param?network.access_control(stream_out).host(1)

Example of Response network.access_control(stream_out).host(1)=10.0.0.100&200 OK

Interpretation Acquire the IP address setting of the restricted client. Setting is possible up to 10 items. Specify a value between 1 to 10 for the number. The following will be returned if subnet mask was specified.

Example of Response 2

network.access_control(stream_out).host(1)=10.0.0.0/24&200 OK

The above example indicates that the range is between 10.0.0.0 to 10.0.0.255. There are also cases when FQDN instead of IP address is set.

Example of Response 3

network.access_control(stream_out).host(1)=somedivision.somecompany.com&200 OK

Allowed users admin, operator

Setting IP Address of Restricted Client

Format /api/param?network.access_control(stream_out).host(Number)=data

Example When setting the first IP address

/api/param?network.access_control(stream_out).host(1)=10.0.0.100

Example of Response network.access_control(stream_out).host(1)&200 OK

Interpretation Change the IP address setting of client restriction. Setting is possible up to 10 items. Specify a value between 1 to 10 for the number. A range of IP address may be specified if the subnet mask is also specified. For example, set as follows to specify a range between 10.0.0.0 to 10.0.0.255.

Example /api/param?network.access_control(stream_out).host(1)=10.0.0.0/24

It is also possible to set using FQDN instead of IP address. Set as follows if the setting is to be left blank.

Example **/api/param?network.access_control(stream_out).host(1)=%00**

Allowed user admin

23. JVC API for Time

The APIs below are related to time. These are equivalent to the features on the Time page of the WEB setting page. Refer to the instruction manual for details on the Time page.

Getting On/Off of SNTP Client

Format **/api/param?network.ntp.status**

Example of Response **network.ntp.status=off&200 OK**

Interpretation Acquire the on/off status of SNTP client. Either on or off will be returned.

Allowed users admin, operator, user

Setting On/Off of SNTP Client, or Validate Changes

Format **/api/param?network.ntp.status=data**

Example **/api/param?network.ntp.status=on**

Example of Response **network.ntp.status&200 OK**

Interpretation Change the on/off status of SNTP client, or validate changes to settings. Specify "on", "off" or "restart". as on or off. IP address of NTP server and access interval are validated by "restart".

Allowed users admin, operator

Getting NTP Server Address

Format **/api/param?network.ntp.host**

Example of Response **network.ntp.host=10.0.0.100&200 OK**

Interpretation Acquire IP address of NTP server. Either the IP address or FQDN will be returned.

Allowed users admin, operator, user

Setting NTP Server Address

Format **/api/param?network.ntp.host=data**

Example **/api/param?network.ntp.host=10.0.0.100**

Example of Response **network.ntp.host&202 Accepted(network.ntp.status=restart)**

Interpretation Change IP address of NTP server. Specify IP address or FQDN. To validate the change, use "network.ntp.status=restart " API.

Allowed users admin, operator

Getting Access Interval to NTP Server

Format **/api/param?network.ntp.interval**

Example of Response **network.ntp.interval=10&200 OK**

Interpretation Acquire the interval for accessing the NTP server. Unit can be gotten by "network.ntp.unit" API.

Allowed users admin, operator, user

Setting Access Interval to NTP Server

Format **/api/param?network.ntp.interval=data**

Example **/api/param?network.ntp.interval=60**

Example of Response

network.ntp.interval&202 Accepted(network.ntp.status=restart)

Interpretation Change the interval for accessing the NTP server. Unit can be set by "network.ntp.unit" API. Specify 1-60 when the unit is min/hour, 1-31 when the unit is day. To validate the change, use "network.ntp.status=restart" API.

Allowed users admin, operator

Getting Access Interval Unit of NTP

Format **/api/param?network.ntp.unit**

Example of Response **network.ntp.unit=hour&200 OK**

Interpretation Acquire the unit of interval for accessing the NTP server. "min", "hour" or "day" is returned.

Allowed users admin, operator, user

Setting Access Interval Unit of SNTP

Format **/api/param?network.ntp.unit=data**

Example **/api/param?network.ntp.unit=day**

Example of Response

network.ntp.unit&202 Accepted(network.ntp.status=restart)

Interpretation Change the unit of interval for accessing the NTP server. Specify "min", "hour" or "day". To validate the change, use "network.ntp.status=restart" API.

Allowed users admin, operator

Getting Time

Format **/api/param?system.date**

Example of Response **system.date=20050614171537&200 OK**

Interpretation Acquire the time from the built-in clock of the camera. Time is arranged in the order of year, month, day, hour, minute and second. Year is denoted in a 4-digit decimal number, and month, day, hour, minute and

second are denoted in 2-digit decimal numbers.

Allowed users admin, operator, user

Setting Time

Format /api/param?system.date=data

Example /api/param?system.date=20050614171537

Example of Response system.date&200 OK

Interpretation Change the time of the built-in clock in the camera. Specify in the order of year, month, day, hour, minute and second. Specify year in a 4-digit decimal number, and month, day, hour, minute and second in 2-digit decimal numbers.

Allowed user admin

Getting Timezone

Format /api/param?system.timezone

Example of Response system.timezone=Pacific&200 OK

Interpretation Acquire the timezone from the camera. Character strings in the following table will be returned.

Timezone Character String	Description
GMT-12	Timezone that is 12 hours earlier than the Greenwich Mean Time.
GMT-11	Timezone that is 11 hours earlier than the Greenwich Mean Time.
GMT-10	Timezone that is 10 hours earlier than the Greenwich Mean Time.
Hawaii	Same timezone as GMT-10
GMT-9:30	Timezone that is 9 hours and 30 minutes earlier than the Greenwich Mean Time.
GMT-9	Timezone that is 9 hours earlier than the Greenwich Mean Time.
Alaska	Same timezone as GMT-9
GMT-8	Timezone that is 8 hours earlier than the Greenwich Mean Time.
Pacific	(GMT-8:00) US/Pacific Time
GMT-7	Timezone that is 7 hours earlier than the Greenwich Mean Time.
Arizona	Same timezone as GMT-7
Mountain	Same timezone as GMT-7
GMT-6	Timezone that is 6 hour earlier than the Greenwich Mean Time.
Central	Same timezone as GMT-6
GMT-5	Timezone that is 5 hour earlier than the Greenwich Mean Time.
East-Indiana	Same timezone as GMT-5.
Eastern	Same timezone as GMT-5.
GMT-4	Timezone that is 4 hour earlier than the Greenwich Mean Time.
Atlantic	Same timezone as GMT-4.
GMT-3:30	Timezone that is 3 hours and 30 minutes earlier than the Greenwich Mean Time.
GMT-3	Timezone that is 3 hour earlier than the Greenwich Mean Time.
GMT-2	Timezone that is 2 hour earlier than the Greenwich Mean Time.
GMT-1	Timezone that is 1 hour earlier than the Greenwich Mean Time.
UTC	Greenwich Mean Time
London	Same timezone as UTC.
GMT+1	Timezone that is 1 hour later than the Greenwich Mean Time.
Berlin	Same timezone as GMT+1.
Rome	Same timezone as GMT+1.
Madrid	Same timezone as GMT+1.

Paris	Same timezone as GMT+1.
CET	Same timezone as GMT+1.
GMT+2	Timezone that is 2 hours later than the Greenwich Mean Time.
EET	Same timezone as GMT+2
GMT+3	Timezone that is 3 hours later than the Greenwich Mean Time.
GMT+3:30	Timezone that is 3 hours and 30 minutes later than the Greenwich Mean Time.
GMT+4	Timezone that is 4 hours later than the Greenwich Mean Time.
GMT+4:30	Timezone that is 4 hours and 30 minutes later than the Greenwich Mean Time.
GMT+5	Timezone that is 5 hours later than the Greenwich Mean Time.
GMT+5:30	Timezone that is 5 hours and 30 minutes later than the Greenwich Mean Time.
Calcutta	Same timezone as GMT+5:30
GMT+5:45	Timezone that is 5 hours and 45 minutes later than the Greenwich Mean Time.
GMT+6	Timezone that is 6 hours later than the Greenwich Mean Time.
GMT+6:30	Timezone that is 6 hours and 30 minutes later than the Greenwich Mean Time.
GMT+7	Timezone that is 7 hours later than the Greenwich Mean Time.
GMT+8	Timezone that is 8 hours later than the Greenwich Mean Time.
GMT+8:45	Timezone that is 8 hours and 45 minutes later than the Greenwich Mean Time.
GMT+9	Timezone that is 9 hours later than the Greenwich Mean Time.
GMT+9:30	Timezone that is 9 hours and 30 minutes later than the Greenwich Mean Time.
Japan	Same timezone as GMT+9.
GMT+10	Timezone that is 10 hours later than the Greenwich Mean Time.
GMT+10:30	Timezone that is 10 hours and 30 minutes later than the Greenwich Mean Time.
GMT+11	Timezone that is 11 hours later than the Greenwich Mean Time.
GMT+11:30	Timezone that is 11 hours and 30 minutes later than the Greenwich Mean Time.
GMT+12	Timezone that is 12 hours later than the Greenwich Mean Time.
GMT+12:45	Timezone that is 12 hours and 45 minutes later than the Greenwich Mean Time.

Allowed users admin, operator, user

Setting Timezone

Format /api/param?system.timezone=data

Example /api/param?system.timezone=Pacific

Example of Response system.timezone&202 Accepted(system.status=restart)

Interpretation Change the timezone of the camera. Refer to "Getting Timezone" on the character string to specify. To validate the change, use "system.status=restart" API.

Allowed user admin

24. JVC API for Password

The APIs below are related to passwords. These are equivalent to the features on the Password page of the WEB setting page. Refer to the instruction manual for details on the Password page.

Setting Password of admin

Format /api/param?system.password.admin(num)=data2

Example /api/param?system.password.admin(0)=someword

Example of Response system.password.admin(0)&200 OK

Interpretation Change the password of admin(0), admin(1), admin(2) or admin(3). Set a password between 4 to 16 characters.

There is no API for Getting passwords.

Allowed user admin

Setting Password of operator

Format **/api/param?system.password.operator(num)=data2**

Example /api/param?system.password.operator(0)=someword

Example of Response system.password. operator(0)&200 OK

Interpretation Change the password of operator(0), operator(1), operator(2) or operator(3). Set a password between 4 to 16 characters.

There is no API for Getting passwords.

Allowed user admin

Setting Password of user

Format **/api/param?system.password.user(num)=data2**

Example /api/param?system.password.user(0)=someword

Example of Response system.password. user(0)&200 OK

Interpretation Change the password of user(0), user(1), user(2) or user(3). Set a password between 4 to 16 characters.

There is no API for Getting passwords.

Allowed user admin

Delete Acount

Format

/api/param?system.password.admin(num).status=initialize

/api/param?system.password.operator(num).status=initialize

/api/param?system.password.user(num).status=initialize

Example /api/param?system.password.admin(1)=initialize

Example of Response system.password. admin(1)=unregistered&200 OK

Interpretation Delete specified account. admin(0), operator(0) and user(0) can not be deleted.

Allowed user admin

25. JVC API for Maintenance

The APIs below are related to maintenance. These are equivalent to the features on the Maintenance page of the

WEB setting page. Refer to the instruction manual for details on the Maintenance page.

Restart the Camera

Format **/api/param?system.status=restart**

Example of Response **system.status&200 OK**

Interpretation Restarts the camera.

Allowed users admin

Initialization

Format **/api/param?system.status=initialize**

Example of Response **system.status&200 OK**

Interpretation Restore all the camera settings to factory defaults. Upon doing so, all transmission services that are in progress will be terminated. Initializing takes a few minutes. Response is returned after initializing. Do not power off during initializing.

Allowed user admin

Firmware Update

Version upgrading is not possible using API. To do so, use the Version Upgrade feature on the Maintenance page of the WEB setting page.

26. JVC API for LED Setting

The APIs below are related to LED. These are equivalent to the features on the LED page of the WEB setting page. Refer to the instruction manual for details on the LED page.

Getting LED mode

Format **/api/param?camera.stealth**

Example of Response **camera.stealth=off&200 OK**

Interpretation Acquire LED setting. "on" or "off" is returned. If thie is "on", LED becomes off after restarting.

Allowed users admin, operator, user

Setting LED mode

Format **/api/param?camera.stealth=data**

Example **/api/param?camera.stealth=on**

Example of Response

camera.stealth&202 Accepted(camera.status=save)

Interpretation Change LED setting. Specify "on" or "off". If "on" is set, LED becomes off after restarting. To validate the change, use "camera.status=save" API.

Allowed users admin, operator

Getting LED blinking mode

Format /api/param?camera.identify

Example of Response camera.identify=off&200 OK

Interpretation Acquire LED blinking setting. "on" or "off" is returned. If thie is "on", LED is blinking.

Allowed users admin, operator, user

Setting LED blinking mode

Format /api/param?camera.identify=data

Example /api/param?camera.identify=on

Example of Response

camera.identify&202 Accepted(camera.status=save)

Interpretation Change LED blinking setting. Specify "on" or "off". If "on" is set, LED starts blinking. To validate the change, use "camera.status=save" API.

Allowed users admin, operator

27. JVC API for Getting Status

The APIs below are related to status acquisition. These are equivalent to the features on the Operation/Settings page of the WEB setting page. Refer to the instruction manual for details on the Operation/Settings page.

Getting Sending Status

Format /api/param?system.session

Response Return the total transmission bit rate, and status of each sending operation. Transmission is not carried out in the following examples.

system.session=&200 OK

system.session.total_bitrate=0k&200 OK

system.session.sending_count=0&200 OK

system.session.sending_max=20&200 OK

In the examples below, 1 JPEG stream of TCP is being sent.

system.session=&200 OK

system.session.total_bitrate=388k&200 OK

system.session.sending_count=2&200 OK

system.session.sending_max=5&200 OK

system.session.sending(01).bitrate=326k&200 OK

system.session.sending(01).to.ip=10.0.0.100&200 OK

system.session.sending(01).to.port=1536&200 OK

system.session.sending(01).to.protocol=tcp_passive&200 OK

system.session.sending(01).to.session=http&200 OK

system.session.sending(01).from.encode=jpeg&200 OK

system.session.sending(01).from.framerate=1&200 OK

system.session.sending(01).from.framesize=vga&200 OK

In case of H.264, system.session.sending(01).from.encode=h264baseline or h264high is returned. In case of multicast, system.session.sending(01).to.ip becomes multicast IP address.

Interpretation Acquire the sending status of the camera. Starting and stopping stream can be occurred in random order, so it can happen that sending(01) is vacant though sending(02) has information.

Allowed users admin, operator, user

Getting Log

Format /api/param?system.log

Response Return the following information. These information will be initialized upon turning off the power of the camera.

Number of seconds after startup, Alarm input, Motion detect, Error

Response examples

system.log=&200 OK

system alive time: 2142sec <----- No. of seconds after startup

Dec 19 14:35:32 vn-h37 user.info evman: Motion Detect <----- Motion detect

Dec 19 14:36:03 vn-h37 user.info evman: Alarm Detect (m1) <----- Alarm input 1ch (make)

Dec 19 14:36:04 vn-h37 user.info evman: Alarm Detect (b2) <----- Alarm input 2ch (break)

Dec 19 14:35:18 vn-h37 user.info evman: Motion Detect <----- Motion detect

Interpretation Acquire the the camera log. Maximum size is 10KB.

Allowed user admin

Getting Model Name

Format /api/param?system.model

Example of Response system.model=VN-H37&200 OK

Interpretation Acquire the model name.

Allowed users admin, operator, user

Getting Firmware Revisions

Format **/api/param?system.software.revision**

Example of Response **system.software.revision=1.00&200 OK**

Interpretation Acquire revisions of the firmware.

Allowed users admin, operator, user

Getting Software ID

Format **/api/param?system.software.programid**

Response Return software ID.

Response examples

system.software.programid=SPL0123&200 OK

Interpretation Acquire the software ID.

Allowed user admin

28. JVC API for Others

These are APIs of features not found on the WEB setting page.

Getting Alarm Input Status (VN-H57/157/257)

Format **/api/param?peripheral.input_pin.pin(Number).status**

Example of Response **peripheral.input_pin.pin(1).status=make&200 OK**

Interpretation Acquire the current alarm input status. Specify 1 or 2 to Number. Either make or break will be returned.

Allowed users admin, operator, user

Getting Mode of FTP Server

Format **/api/param?application.ftp.mode**

Example of Response **application.ftp.mode=active&200 OK**

Interpretation Acquire the mode of FTP server that is used by alarm action. Either active or passive is returned.

active mode: Standard mode of FTP server. Also called PORT mode. TCP connection for data is established from 20 port of FTP server to 10020 port of the camera.

passive mode: TCP connection for data is established from the camera to FTP server. Port number depends on FTP server.

Allowed users admin, operator, user

Setting Mode of FTP Server

Format **/api/param?application.ftp.mode=data**

Example /api/param?application.ftp.mode=active

Example of Response application.ftp.mode&200 OK

Interpretation Change the mode of FTP server that is used by alarm action. Set active or passive. Default is active.

active mode: Standard mode of FTP server. Also called PORT mode. TCP connection for data is established from 20 port of FTP server to 10020 port of the camera.

passive mode: TCP connection for data is established from the camera to FTP server. Port number depends on FTP server.

Allowed user admin, operator

Getting Control Port Number of FTP Server

Format **/api/param?application.ftp.port**

Example of Response **application.ftp.port=21&200 OK**

Interpretation Acquire port number for control of FTP server that is used by alarm action. Port number for data plus one is the port number for control.

Allowed users admin, operator, user

Setting Control Port Number of FTP Server

Format **/api/param?application.ftp.port=data**

Example **/api/param?application.ftp.port=21**

Example of Response application.ftp.port&200 OK

Interpretation Change port number for control of FTP server that is used by alarm action. Default is 21. Port number for data plus one is the port number for control.

Allowed user admin, operator

Getting Port Number of RTSP Server

Format **/api/param?network.rtsp.port**

Example of Response **network.rtsp.port=554&200 OK**

Interpretation Acquire port number for RTSP server.

Allowed users admin, operator, user

Setting Port Number of RTSP Server

Format **/api/param?network.rtsp.port=data**

Example **/api/param?network.rtsp.port=554**

Example of Response network.rtsp.port&202 Accepted(network.rtsp(configuration).status=restart)

Interpretation Change port number of RTSP server. Default is 554.

Allowed user admin, operator

29. Getting Audio from the Camera via HTTP (VN-H57/157WP/257/257VP)

29.1. Basic Procedures

1) The client establishes a TCP connection to port number 80.

2) The client sends out API.

Example

GET /api/audio?lowdelay=1 HTTP/1.1<CRLF>

Host: 192.168.0.2<CRLF><CRLF>

Note <CRLF> denotes the line feed code (0x0D, 0x0A).

3) The camera returns HTTP response.

Example

HTTP/1.1 200 OK<CRLF>

Connection: close<CRLF>

Content-type: audio/ulaw<CRLF>

Date: Tue, 02 Oct 2007 07:33:12 GMT<CRLF>

Server: JVC VN-H57 Network Camera<CRLF>

x-vnh57_response: encode=ulaw&lowdelay=1&assured=1<CRLF><CRLF>

4) The camera sends out audio data after returning HTTP response.

Audio data with 12 bytes header will be sent out continuously after HTTP response. HTTP Response and audio data sent out by the camera are as follows.

HTTP Response
header (12 bytes)
u-Law data (512 bytes)
header (12 bytes)
u-Law data (512 bytes)

,,,

Structure of 12 bytes header is as below. First 4 bytes is payload type for u-Law.

0x00000080
Volume of payload (512 for u-Law)
Time stamp in 8kHz

5) When the client wants to stop current audio transmission, the client disconnects TCP80.

The camera does not accept further API via current TCP that is used for audio transmission. To change parameter, dsconnect current TCP to stop the audio transmission, connect new TCP, and send API with new parameter.

29.2. API Format

Structure

GET	space	API	space	HTTP/1.1	0x0D 0x0A
Host:	space	IP Address of the camera	0x0D 0x0A 0x0D 0x0A		

Unlike APIs for getting/setting parameters, Accept line is not required. Basic authentication is also not necessary.

Example

GET /api/audio?assured=1&lowdelay=1 HTTP/1.1<CRLF>

Host: 192.168.0.2<CRLF><CRLF>

Parameter value is indicated using =. Do not insert space before and after =.

Example assured=1

Parameters are segmented using &. Do not insert space before and after &.

Example assured=1&lowdelay=0

There is no need to specify all parameters. Default values will be used for parameters that are not specified.

Parameter Description

assured Recent audio data is stored in internal buffer of the camera. Specify as assured=0 to request for the newest data in the buffer and assured=1 to request for the oldest data in the buffer. Specify as assured=0 to

shorten the audio delay time. To enable stable playback in a network where jitter occurs, it is recommended that this be specified as assured=1. Default value is 1.

lowdelay Specifying as lowdelay=1 disables the Nagle algorithm of TCP, and audio delay time will be shortened. When lowdelay=0 is specified, the Nagle algorithm is enabled and audio delay time will be prolonged. However, transmission overhead will be enhanced. Default value is 1.

29.3. Response

When API is successfully received

The camera will return 200 OK. There is no Content-length field in the HTTP response. The x-vnh57_response line indicates actual parameter.

Example

HTTP/1.1 200 OK<CRLF>

Connection: close<CRLF>

Content-type: audio/ulaw<CRLF>

Date: Tue, 02 Oct 2007 07:33:12 GMT<CRLF>

Server: JVC VN-H57 Network Camera<CRLF>

x-vnh57_response: encode=ulaw&lowdelay=1&assured=1<CRLF><CRLF>

29.4. Restrictions

Access restriction

The camera has access restriction feature that enables to deny access from a specific IP address. If audio is requested from the IP address of access restriction, the camera disconnects the TCP connection after API is sent.

Restriction by maximum bitrate

The maximum bitrate of the camera is about 20 Mbps.

Number of clients

The maximum number of audio stream is 2, 2 TCP streams or 1 TCP stream and 1 multicast stream. When 2 streams are sent from the camera, new request for audio is disconnected.

30. Sending Audio to the Camera (VN-H57/157WP/257/257VP)

This section describes APIs for audio sending from a client to the camera. Make use of the APIs explained in this section in the way as mentioned in Section 7.

30.1. Procedures

1) The client establishes a TCP connection to port number 80.

2) The client sends out API.

API has following structure.

GET	space	API Characters	space	HTTP/1.1	0x0D 0x0A
Accept:	space	text/plain (or text/html)	0x0D 0x0A		
Host:	space	IP Address of the camera	0x0D 0x0A		
Authorization: Basic	space	Encoded User Name and Password	0x0D 0x0A 0x0D 0x0A		

Refer to Section 5 on details of the Accept and Authorization lines.

The API characters are as follows.

/api/receive?from=network&from.ip=data1&from.protocol=tcp_passive&from.ip_translate=on&to=audio

Example

/api/receive?from=network&from.ip=10.0.0.100&from.protocol=tcp_passive&from.ip_translate=on&to=audio

Specify the client IP address for from.ip=. When from.ip_translate is set to off, the camera will standby to receive audio data from the IP address specified at from.ip. When from.ip_translate is set to on, the camera will ignore from.ip and standby to receive audio data from the source IP address of this API.

2) The camera returns a response.

HTTP/1.1 200 OK<CRLF>

Connection: Keep-Alive<CRLF>

Content-type: text/plain<CRLF>

Date: Fri, 13 MAY 2005 07:33:12 GMT<CRLF>

Server: VN-H57 Network Camera/1.0.0<CRLF>

x-vnh57_response:

from=network&from.ip=10.0.0.100&from.protocol=tcp_passive&from.ip_translate=on&to=audio<CRLF><CRLF>

200 OK<CRLF>

The client may disconnect the TCP80 at this point of time.

3) The client establishes a TCP connection to port number 49298.

4) The client continues to send 512 bytes of u-Law data with a 12-byte header.

0x00000080
Volume of payload (512 for u-Law)
Time stamp in 8kHz
u-Law data (512 bytes)

5) To end, disconnect TCP49298.

30.2. Restrictions

Restrictions on Number of Clients

Only 1 client is allowed to send audio data to the camera. the camera will return an error for this API and TCP will be disconnected when this function is currently in use by another client.

Timing of Data Sending

512 bytes, or in other words, 64 msec of audio data may be sent during each transmission. Send audio data at intervals as evenly as possible. A part of the data may be lost if audio data exceeding 2 seconds are sent to the camera at one time.

31. Getting SD Card data from the Camera via RTSP/RTP

RTSP of the camera is RFC2326 compliant.

31.1. URI

URI for RTSP is

rtsp://ipaddress/PSIA/Streaming/tracks

31.2. Playback control

For Playback control, the messages is used as below,

Control command	Method	Header	Example
Play	PLAY	Range	Range: clock=20120518T135717Z
Pause	PAUSE	-	
KeepAlive	GET_PARAMETER	-	

Specify start time by request header “Range”.

For keep-alive control, issue the GET_PARAMETER method in 3 seconds during receiving data.

Keep the message interval is longer than 200 milliseconds.

31.3. Example of message sequence

C->S DESCRIBE rtsp://192.168.0.20/PSIA/Streaming/tracks RTSP/1.0

CSeq: 1

S->C RTSP/1.0 200 OK

CSeq: 1

Content-Base: rtsp://192.168.0.20/PSIA/Streaming/tracks/

Content-Type: application/sdp

Content-Length: 267

v=0

o=- 401875008 1 IN IP4 0.0.0.0

s=Media Presentation

c=IN IP4 0.0.0.0

t=0 0

m=video 0 RTP/AVP 96

a=control:video

a=rtpmap:96 H264/90000

a=fmtp:96

packetization-mode=1;profile-level-id=640028;sprop-parameter-sets=Z2QAKKzSAeAIn5cBbgwMDIA

AAAMAgAAACkeEQjUA,aO48MAD=

C->S SETUP rtsp://192.168.0.20/PSIA/Streaming/tracks/video RTSP/1.0

CSeq: 2

Transport: RTP/AVP;unicast;client_port=6970-6971

S->C RTSP/1.0 200 OK

CSeq: 2

Session: 401875008;timeout=60

Transport: RTP/AVP;unicast;client_port=6970-6971;server_port=1486-1487

C->S GET_PARAMETER rtsp://192.168.0.20/PSIA/Streaming/tracks RTSP/1.0

CSeq: 3

Connection: Keep-Alive

Session: 401875008

S->C RTSP/1.0 200 OK

CSeq: 3

Session: 401875008

Status: pause

C->S PLAY rtsp://192.168.0.20/PSIA/Streaming/tracks RTSP/1.0

CSeq: 4

Range: clock=20120518T135717Z

Session: 401875008

S->C RTSP/1.0 200 OK

CSeq: 4

Session: 401875008

C->S GET_PARAMETER rtsp://192.168.0.20/PSIA/Streaming/tracks RTSP/1.0

CSeq: 5

Connection: Keep-Alive

Session: 401875008

S->C RTSP/1.0 200 OK

CSeq: 5

Session: 401875008

Status: play

32. Exporting H.264 data from SD Card to the PC

This section describes APIs for audio exporting H.264 data from SD card to the PC.

Getting Total Number of Files and File Size

Format

/api/copy?pseudo=on&from.date.start=YYYYMMDDhhmmss&from.date.end=YYYYMMDDhhmmss

Example of response

14<CRLF>

200 OK,(Completed)<CRLF>

<CRLF>

1f<CRLF>

200 OK,count=1&t_size=7731371<CRLF>

<CRLF>

0<CRLF>

<CRLF>

0<CRLF>

Interpretation Acquire total number of files and file size. Specify start time and end time, then CHUNKED HTTP response will be returned. The value of count is total number of files. The value of t_size is file size.

Allowed users admin, operator

Exporting SD Card Data as a File

Format

/api/copy?pseudo=off&from.date.start=YYYYMMDDhhmmss&from.date.end=YYYYMMDDhhmmss

Example of response

c<CRLF>

200 OK,(0)<CRLF>

<CRLF>

14<CRLF>

200 OK,(Completed)<CRLF>

<CRLF>

40<CRLF> Size of main header

Data structure of header1

Item	Size	Example	Note
Revision	24	revision=0x0100	Revision of file format
Total number of files	16	count=2	Total number of files
Total size of files	24	t_size=12052495	Total size of files

38<CRLF> Header size of file2

Data structure of header1

Item		Size	Example	Note
Header of file1	Size	16	size=7731251	Size of file1
	File name	40	name-ckst0000	Name of file1

38<CRLF> Header size of file2

Data structure of header2

Item		Size	Example	Note
Header of file2	Size	16	size=4321244	Size of file2
	File name	40	name-ckst0001	Name of file2

C800<CRLF> Data(1) size of file1

data(1) of file1 (50 kB)

C800<CRLF> Data(2) size of file1

data(2) of file1 (50 kB)

...

C800<CRLF> Data(1) size of file2

data(1) of file2 (50 kB)

C800<CRLF> Data(2) size of file2

data(2) of file2 (50 kB)

...

0<CRLF> End of file

Interpretation Specify start time and end time, then CHUNKED HTTP response and H.264 elementary stream data will be returned.

Allowed users admin, operator

33. List of Protocols and Port Numbers Used

The camera uses the following protocols and port numbers.

Protocol / Port Number	**Use**
TCP 20, 21	FTP
TCP 25	SMTP (Mail by Alarm Action)
TCP 80	WEB setting page, API for Getting status and changing settings, video/audio streaming by JVC protocol
UDP 80	Search for the camera
TCP 110	POP (Mail by Alarm Action)
UDP 123	SNTP
TCP 554	RTSP
UDP 9131	AMX Device Discovery Protocol
TCP 10020, 10021, 10023	reserved for internal use
TCP 32040	Alarm server
TCP 49298	Audio sending from a client to the camera
TCP User Setting	Alarm on TCP
UDP User Setting	Alarm on UDP
UDP User Setting	Multicast Streaming

34. Customizing Built-in Viewer

The built-in viewer of the camera consists of five ActiveX controls. These ActiveX controls are available for customized viewer.

34.1. List of ActiveX

- JPEG/H.264 Viewer It can show JPEG and H.264 video, and save still image.
- PTZ Control Client It can control digital ptz.
- Audio Monitor It can playback audio.

- Audio Sending Client It can send audio from PC to the camera.

How to download ActiveX controls:

i) Please input URL below in Internet Explorer's url form.

http://(IP Address)/ IntegratedViewer.cab

Ex.) When IP address of the camera is "192.168.0.2":

http://192.168.0.2/IntegratedViewer.cab

ii) Download dialog box is showed. Please click save button and copy to some folder in the PC.

34.2. Properties of ActiveX

JPEG /H.264 Viewer

Property	Meaning
IP	IP Address of the camera: Required when RcvMode is unicast. Default: 192.168.0.2
HttpPort	Port Number of the camera: Required when RcvMode is unicast. (1 - 65535) Default: 80
MultiIP	IP Address of multicast: Required when RcvMode is multicast. Default: 225.0.1.1
MultiPort	Port Number of multicast: Required when RcvMode is multicast. (1 - 65535) Default: 49152
RcvMode	Desired stream (0: unicast, 1: multicast)
FrameRate *JPEG only	Frame Rate of JEPG To specify a frame rate lower than 1fps, use "-". For example, specify -5 for 1/5 fps. (15, 10, 7.5, 6, 5, 3, 2, 1, -2, -3, -5, -10, -15, -20, -30, -60) Default: 5
DispWidth	Width of Display When the size is different from original frame size, the image is scaled. Default: 640
DispHeight	Height of Display When the size is different from original frame size, the image is scaled. Default: 360
DispTitle	Display of Camera ID (0: hide, 1: display) Default: 0
DispMotion *JPEG only	Display of Motion Detection (0: hide, 1: display) Default: 0
DispPosTitle	Display of Position Title (0: hide, 1: display) Default: 0
DispTimeCode	Display of Time Code (0: hide, 1: display) Default: 0
TimeFormat	Format of Time Code (0: YYYY/MM/DD HH:MM:SS.mm 1: YYYY/MM/DD HH:MM:SS 2: DD/MM/YYYY HH:MM:SS 3: MM/DD/YYYY HH:MM:SS 4: MM/DD HH:MM:SS 5: HH:MM:SS 6: HH:MM) *Y: Year M: Month D: Day H: Hour M: Minute S: Second m: milli second Default: 1
FolderName *JPEG only	Folder Name of saving still images. This folder is created in WindowsXP : MyDocuments

	WindowsVista : Documents Windows7 : Documents Default: In case of VN-H37: VN-H37 In case of VN-H137: VN-H137 In case of VN-H237: VN-H237 In case of VN-H237VP: VN-H237VP In case of VN-H57: VN-H57 In case of VN-H157WP: VN-H157 In case of VN-H257: VN-H257 In case of VN-H257VP: VN-H257VP
OpPassword	Operator Password of the camera

PTZ Control Client

Property	Meaning
IP	IP address of the camera Default: 192.168.0.2
HttpPort	Port number of the camera (1 - 65535) Default: 80
DispLang	Language of error messages (0: Japanese, 1: English) Default: 0
OpPassword	Operator password of the camera
PanTiltSpeed	Speed of manual pan/tilt control (1 – 8) Default: 4
FocusZoomSpeed	Speed of manual zoom control (1 – 4) Default: 2
BlackAndWhiteMode	Easy Day and Night (0: Auto, 3: Color, 4: Black and White)
WhiteBalance	White Balance (0: ATW, 2: AWC)
BLC	Back Light Compensation (0: Off, 1: Area1, 2: Area2, 3: Area3, 4: Area4)
AutoFunctionStatus	Status of current auto function (0: stop, 1: auto patrol is working)
PositionTitle(n)	Getting the position title of registered preset position n: Position Number (0 – 19)
FocusAssistMode	Focus Assist Mode (0: stopped, 1: working)

Audio Monitor/Audio Sending Client [VN-H57/VN-H157WP/VN-H257/VN-H257VP only]

Property	Meaning
IP	[Audio Monitor] IP address of the camera in case of unicast receiving IP multicast address in case of multicast receiving [Audio Sending Client] IP address of the camera Default: 192.168.0.2
Port *Audio Monitor only	Port number of the camera in case of unicast receiving Port number of multicast in case of multicast receiving (1 – 65535) Default: 80
ApiPort *Audio Sending Client only	HTTP port number of the camera (1 – 65535) Default: 80
SoundPort *Audio Sending Client only	Destination port number of audio stream from PC to the camera (1 – 65535) Default: 49298
Result *Audio Sending Client only	Result of starting audio atream to the camera by “Play()” method. (0: failed, 1: success)
Password	Operator password of the camera

34.3. Method of ActiveX Control

JPEG /H.264 Viewer

Method	Meaning
Play()	Start playback
Stop()	Stop playback
Capture() *JPEG only	Save still image of JPEG (Saved folder is specified by “Folder Name” of property)

PTZ Control Client

Method	Meaning
Initialize()	Initialize PTZ Control Client *It must be called before using ptz control
Destroy()	Finalizing PTZ Control Client *It must be called when the application using ActiveX control is closed.
ManualCtrl(n)	Start Pan/Tilt according to specified direction **Direction** **Number of “n”** upper-left up upper-right 7 8 9 left right 4 5 6 under-left down under-right 1 2 3
ZoomCtrl(n)	Start Zoom-In/Zoom-Out (n = 0: Zoom-In, n = 1: Zoom-Out)
Stop()	Stop Pan/Tilt/Zoom
SetAutoFunction(n)	Control Auto Patrol (n = 0: stop auto patrol, n = 2: start auto patrol)
OnePushAWC()	Issue one push AWC
SetPosition(n, str)	Register current position as preset position n: Position Number (0 – 19) str: Position Title (0 - 32 characters)
DeletePosition(n)	Unregister specified preset position n: Position Number (1 – 19) *Cannot unregister Home Position
MovePosition(n)	Move to specified preset position n: Position Number (0 – 19)

Audio Monitor/Audio Sending Client

Method	Meaning
Play()	[Audio Monitor] Start playback [Audio Sending Client] Start audio stream *Result of starting audio stream is stored in “Result” of property
Stop()	[Audio Monitor] Stop playback [Audio Sending Client] Stop audio stream
Destroy() *Audio Sending Client only	Finalize Audio Sending Client *It must be called when the application using ActiveX control is closed.

34.4. How to use ActiveX Control by HTML

If write the next code in <Body> of HTML source code, It comes to be able to use ActiveX in HTML.

JPEG/H.264 Viewer

```
<OBJECT ID="Viewer"
        WIDTH = 1920
        HEIGHT= 1080
        CLASSID="CLSID:C0795FC0-14E7-4A78-A928-88C3FBD2A1D0"
        CODEBASE="./IntegratedViewer.cab#version=1,0,1, 37"
</OBJECT>
```

PTZ Control Client

```
<OBJECT ID="PTZCtrl"
        WIDTH = 1
        HEIGHT= 1
        CLASSID="CLSID:06731D1A-BD3C-49B7-8433-77C730D27F06 "
        CODEBASE="./IntegratedViewer.cab #version=1,0,0,0"
</OBJECT>
```

Audio Monitor

```
<OBJECT ID="AudioMonitor"
        WIDTH =1
        HEIGHT= 1
        CLASSID="CLSID:EEF1E8CA-D887-4530-97F9-4C79ABCAE520"
        CODEBASE="./IntegratedViewer.cab #version=1,0,0,0"
</OBJECT>
```

Audio Sending Client

```
<OBJECT ID="AudioSender"
        WIDTH = 1
        HEIGHT= 1
        CLASSID="CLSID:CAA77F3F-FADA-48d6-A9F3-C4B1D74C0E77"
        CODEBASE="./IntegratedViewer.cab #version=1,0,0,0"
</OBJECT>
```

34.5. HTML Sample

Sample code for functions below:

- Playback of JPEG or H.264 (Protocol: HTTP(unicast), Display Size: 640x360)
- Play/Pause of Playback
- Capture of still picture
- Playback of audio
- Send audio stream to the camera
- Digital PTZ Control (Up, Down, Left, Right, Zoom-in, Zoom-out)

Sample code

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML4.0 Transitional//EN">
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html;charset=euc-jp">
<TITLE>Sample Client</TITLE>
</HEAD>

<SCRIPT LANGUAGE=JAVASCRIPT>
```

```
// ********** Viewer **********
function play_click(play_btn, ip, http_port)
{
if(play_btn.value == "Play"){
InitViewer(ip, http_port);
Viewer.Play();
play_btn.value = "Stop";
}
else{
Viewer.Stop();
play_btn.value = "Play";
}
}

function capture_click()
{
Viewer.Capture();
}

function InitViewer(ip, http_port)
{
Viewer.OpPassword = "jvc";
Viewer.IP = ip.value;
Viewer.HttpPort = http_port.value;
Viewer.DispWidth = 640;
Viewer.DispHeight = 360;
Viewer.FolderName = "VN-H37";
Viewer.RcvMode = 0;
}

// ********** Audio Monitor **********
function receive_click(rcv_btn, ip, http_port)
{
if(rcv_btn.value == "Receive"){
InitMonitor(ip, http_port);
AudioMonitor.Play();
rcv_btn.value = "Stop";
}
else{
AudioMonitor.Stop();
rcv_btn.value = "Receive";
}
}

function InitMonitor(ip, http_port)
{
AudioMonitor.Password = "jvc";
AudioMonitor.IP = ip.value;
AudioMonitor.Port = http_port.value;
}

// ********** Audio Sender **********
function send_click(send_btn, ip, http_port)
{
if(send_btn.value == "Send"){
InitSender(ip, http_port);
AudioSender.Play();
send_btn.value = "Stop";
}
else{
AudioSender.Stop();
send_btn.value = "Send";
}
}
```

```
function InitSender(ip, http_port)
{
AudioSender.Password = "jvc";
AudioSender.IP = ip.value;
AudioSender.Port = http_port.value;
}

// ********** PTZ Control **********
var f_init = 0;
function PTControl(num){
if (f_init == 0)
InitPTZCtrl();

PTZCtrl.ManualCtrl(num);
}

function ZoomControl(num){
if (f_init == 0)
InitPTZCtrl();

PTZCtrl.ZoomCtrl(num);
}

function InitPTZCtrl(){
PTZCtrl.OpPassword = "jvc";
PTZCtrl.IP = myForm.IP.value;
PTZCtrl.HttpPort = myForm.HTTP_PORT.value;
PTZCtrl.Initialize();
f_init = 1;
}

function mouse_up(){
PTZCtrl.Stop();
}

function close_window(){
AudioSender.Destroy();
PTZCtrl.Destroy();
}

</SCRIPT>

<BODY STYLE="font-size:12px;font-family:arial;color:#ffffff" bgcolor="#000000"
onunload="close_window()">

<!-- Viewer ActiveX -->
<OBJECT ID="Viewer"
   WIDTH = 640
   HEIGHT= 360
   CLASSID="CLSID:C0795FC0-14E7-4A78-A928-88C3FBD2A1D0"
   STYLE="border-style:solid;border:1px;border-color:#ffffff;">
</OBJECT>

<!-- Audio Monitor ActiveX -->
<OBJECT ID="AudioMonitor"
   WIDTH = 1
   HEIGHT= 1
   CLASSID="CLSID:EEF1E8CA-D887-4530-97F9-4C79ABCAE520">
</OBJECT>

<!-- Audio Sender ActiveX -->
<OBJECT ID="AudioSender"
   WIDTH = 1
   HEIGHT= 1
   CLASSID="CLSID:CAA77F3F-FADA-48d6-A9F3-C4B1D74C0E77">
```

```
</OBJECT>

<!-- PTZ Control ActiveX -->
<OBJECT ID="PTZCtrl"
   WIDTH = 1
   HEIGHT= 1
   CLASSID="CLSID:5506B06A-9FED-4dc0-99E1-9AEF2F2B0509">
</OBJECT>

<FORM NAME="myForm">
<table>
   <tr>
      <td>
         VN-H37 IP Address
         <INPUT TYPE="TEXT" NAME="IP" VALUE="192.168.0.2">
         HTTP Port
         <INPUT TYPE="TEXT" NAME="HTTP_PORT" VALUE="80">
      </td>
   </tr>
   <tr>
      <td>
         Viewer
         <INPUT TYPE="BUTTON" NAME="PLAY_BTN" style="width:70px"
                                    VALUE="Play"          onclick="play_click(PLAY_BTN,          IP,
HTTP_PORT)">
         <INPUT TYPE="BUTTON" NAME="CAPTURE_BTN" style="width:70px"
                                    VALUE="Capture"  onclick="capture_click()">
      </td>
   </tr>
</table>

<p STYLE="top:506px;left:21px;position:absolute" >PTZ Control</p>
<INPUT TYPE="BUTTON" VALUE="Up"
         STYLE="width:40px;top:530px;left:61px;position:absolute"
         onmousedown="PTControl(8)" onmouseup="mouse_up()" onmouseout="mouse_up()">
<INPUT TYPE="BUTTON" VALUE="Left"
         STYLE="width:40px;top:550px;left:41px;position:absolute"
         onmousedown="PTControl(4)" onmouseup="mouse_up()" onmouseout="mouse_up()">
<INPUT TYPE="BUTTON" VALUE="Right"
         STYLE="width:40px;top:550px;left:81px;position:absolute"
         onmousedown="PTControl(6)" onmouseup="mouse_up()" onmouseout="mouse_up()">
<INPUT TYPE="BUTTON" VALUE="Down"
         STYLE="width:40px;top:570px;left:61px;position:absolute"
         onmousedown="PTControl(2)" onmouseup="mouse_up()" onmouseout="mouse_up()">

<INPUT TYPE="BUTTON" NAME="TELE_BTN" VALUE="+"
         STYLE="width:40px;top:535px;left:134px;position:absolute"
         onmousedown="ZoomControl(0)" onmouseup="mouse_up()" onmouseout="mouse_up()">
<INPUT TYPE="BUTTON" NAME="WIDE_BTN" VALUE="-"
         STYLE="width:40px;top:565px;left:134px;position:absolute"
         onmousedown="ZoomControl(1)" onmouseup="mouse_up()" onmouseout="mouse_up()">

</FORM>
</BODY>
</HTML>
```

34.6 Notes

- Enable the JPEG/H.264 frame size that you want in “Basic Settings2” or “Encoding” page of the camera.

- Start Multicast stream on the camera Web page to use Multicast. The ActiveX control does not send request to the camera for starting Multicast stream.

- Set unique Multicast address and port number to each Multicast stream if multiple multicast streams are required in the system.
- Reload of ActiveX control is required to change Multicast property.

35. PSIA

- PSIA Account

  Default User Name: psia

  Default Password: jvc

- RTSP URI

 See Chapter 4.

36. FAQ

(1) Low Frame rate due to long delay of network

- Causes of Low Frame Rate

During transmission via TCP, the camera sends out the following data by receiving the Ack of TCP. When network delay is long, reception of Ack will be delayed and sending rate will drop. This therefore leads to a drop in the frame rate.

- Countermeasure

This problem can be avoided by receiving via multicast. Multicast uses UDP and Ack does not exist. As such, the sender will be able to continue sending without being affected by network delays.