NVE RTSP
Reference Manual

UDP
Technology

Contents

WHAT'S NEW

FW K641.11710
[Add] enable, vdelay, hdelay, stream, and session_id of VIDEO_INFO
[Add] deinterlace of VIDEO_FORMAT

1. Introduction

The Real-Time Streaming Protocol (RTSP) is an application-level protocol for control over the delivery of continuous media such as video and audio with real-time. However, it is impossible such as changing a setting of server, changing a setting of media data and so on, by standard method of RTSP. Therefore UDP provide the extended RTSP method.

1.1. Reference

The RTSP protocol is described in RFC 2326(Real Time Streaming Protocol).

RTSP protocol
Real Time Streaming Protocol - RFC 2326
SDP protocol
Session Description Protocol - RFC 2327
HTTP protocol
Hypertext Transfer Protocol -- HTTP/1.0

1.2. RTSP Syntax

RTSP URL
RTSP URL is defined as follow;

Request syntax

```
COMMAND URI RTSP/1.0<CRLF>
Headerfield1: val1<CRLF>
Headerfield2: val2<CRLF>
...
<CRLF>
```

Response syntax

```
RTSP/1.0 ResultCode ResultString<CRLF>
Headerfield3: val3<CRLF>
Headerfield4: val41496.
<CRLF>
...
<CRLF>
```

The following header fields are accepted by all commands. Other header fields are silently ignored (unless stated otherwise in the sections below).

Field	Description
Authorization	Authorization information from the client
CSeq	Request sequence number
Session	Session identifier (returned by server in SETUP response)
Content-Length	Length of content

The following header fields can be generated for all responses by the RTSP server:

Field	Description
CSeq	Response sequence number (matches the sequence number of the request).
Session	Session identifier.
WWW-Authenticate	Authentication from client requested.

1.3. RTSP Status Code

Code	reason	
100	Continue	all
200	OK	all
201	Created	RECORD
250	Low on Storage Space	RECORD
300	Multiple Choices	all
301	Moved Permanently	all
302	Moved Temporarily	all
303	See Other	all
305	Use Proxy	all
400	Bad Request	all
401	Unauthorized	all
402	Payment Required	all
403	Forbidden	all
404	Not Found	all
405	Method Not Allowed	all
406	Not Acceptable	all
407	Proxy Authentication Required	all
408	Request Timeout	all
410	Gone	all
411	Length Required	all
412	Precondition Failed DESCRIBE	SETUP
413	Request Entity Too Large	all
414	Request-URI Too Long	all
415	Unsupported Media Type	all
451	Invalid parameter	SETUP
452	Illegal Conference Identifier	SETUP
453	Not Enough Bandwidth	SETUP
454	Session Not Found	all

455	Method Not Valid In This State	all
456	Header Field Not Valid	all
457	Invalid Range	PLAY
458	Parameter Is Read-Only	SET_PARAMETER
459	Aggregate Operation Not Allowed	all
460	Only Aggregate Operation Allowed	all
461	Unsupported Transport	all
462	Destination Unreachable	all
500	Internal Server Error	all
501	Not Implemented	all
502	Bad Gateway	all
503	Service Unavailable	all
504	Gateway Timeout	all
505	RTSP Version Not Supported	all
551	Option not support	all

2. Standard Method

The OPTIONS, DESCRIBE, SETUP, PLAY, PAUSE and TEARDOWN methods are supported.

Method	Direction	OBJECT
OPTIONS	C->S	P, S
DESCRIBE	C->S	P, S
SETUP	C->S	S
PLAY	C->S	P, S
PAUSE	C->S	P, S
TEARDOWN	C->S	P, S

(P: presentation, S: stream)

OPTIONS

The OPTIONS method requests a list of available RTSP methods.

Example

Request

```
OPTIONS * RTSP/1.0
CSeq: 1
```

Response

```
RTSP/1.0 200 OK
CSeq: 1
Public: OPTIONS, DESCRIBE, PAUSE, PLAY, SETUP, TEARDOWN, EXT_CMD,
EXT_SUBSESSION_CMD
```

DESCRIBE

The DESCRIBE method retrieves the description of a presentation or media object identified (SDP description) by the request URL from a server. It can be specified the description format by using the *Accept* header and only *application/SDP* is supported as *Accept* header. (SDP is detailed at 'Session Description Protocol – RPC2327)

The DESCRIBE method generates the following additional header fields

Content-Type	Type of content This value is always is *application/SDP.*
Content-Length	Length of SDP description
Content-Base	If relative URLs are used in the SDP description, then this is the base URL.

Example

Request

```
DESCRIBE / RTSP/1.0
CSeq: 1
Accept: application/sdp
```

Response

```
RTSP/1.0 200 OK
CSeq: 1
Date: Fri, Jul 28 2006 03:08:04 GMT
Content-Base: rtsp://192.168.11.15/
Content-Type: application/sdp
Content-Length: 482

v=0
o=- 1154055959603573 1 IN IP4 192.168.11.15
s=Media Presentation
t=0 0
a=type:broadcast
a=control:*
a=range:npt=0-
m=video 0 RTP/AVP 96
c=IN IP4 0.0.0.0
a=rtpmap:96 MP4V-ES/90000
a=fmtp:96 profile-level-id=245;
config=000001B0F5000001B509000001000000012000C888BA9860FA62C087828307
a=control:track1
m=audio 0 RTP/AVP 97
c=IN IP4 0.0.0.0
a=rtpmap:97 L16/16000
a=control:track2
m=event 0 RTP/AVP 100
c=IN IP4 0.0.0.0
a=rtpmap:100 EVENT-NVE/1000
a=control:track3
```

SETUP

The SETUP method configures the delivery method for the data. The SETUP method requires and generates the following additional header field

Transport	Specifies how the data stream is transported. Supported variants: RTP/AVP;unicast;client_port=port1-port2 RTP/AVP;multicast;client_port=port1-port2 RTP/AVP/TCP;unicast

The response returns a session identifier that should be used with stream control methods to the server (PLAY, PAUSE, TEARDOWN). If the Session header includes a timeout parameter, then the session needs to be kept alive. This can be done by sending RTSP requests to the server containing the session identifier (e.g. OPTIONS) within the specified timeout time or through the use of RTCP. The RTSP server does not support reconfiguring of the transport parameters.

Example

Request example

```
SETUP /track1 RTSP/1.0
CSeq: 3
Transport: RTP/AVP;unicast;client_port=20000-20001
```

Response example

```
RTSP/1.0 200 OK
CSeq: 3
Transport: RTP/AVP;unicast;destination=192.168.2.100;client_port=20000-
20001;server_port=1518-1519
Session: 6004
```

PLAY

The PLAY method requests the server to start sending data. It is possible to call PLAY only when SETUP request have been acknowledged as successfully. The *Range* is always ‘begin now’ and ‘have no stop time’ because only live stream are used.

Example

Request example

```
PLAY / RTSP/1.0
CSeq: 4
Session: 6004
```

Response example

```
RTSP/1.0 200 OK
CSeq: 4
Date: Fri, Jul 28 2006 03:16:50 GMT
Range: npt=0.000-
Session: 6004
RTP-Info: url=rtsp://192.168.11.15//track1;seq=34560;rtptime=1278107052
```

PAUSE

The PAUSE method request caused the stream delivery to be halted temporarily.

Example

Request example

```
PAUSE / RTSP/1.0
CSeq: 5
Session: 6004
```

Response example

```
RTSP/1.0 200 OK
CSeq: 5
Session: 6004
```

TEARDOWN

The TEARDOWN method requests the server to stop the stream delivery.

Example

Request example

```
TEARDOWN / RTSP/1.0
CSeq: 6
Session: 6004
```

Response example

```
RTSP/1.0 200 OK
CSeq: 6
Session: 6004
```

3. Extended Method

RTSP standard method cannot configure system setting and media setting. So UDP provides extended RTSP method to solve this problem.

The extended method has EXT_CMD and EXT_SUBSESSION_CMD. EXT_CMD method is for configuring the system and EXT_SUBSESSION_CMD method is for configuring the setting of each channel.

System configuration and channel configuration can be set and get using GET/SET operation and each operation have means as following.

Operation	Description
GET	Retrieve the value of variable.
SET	Assign the value of variable.
GSET	Assign the requested value of variable for retrieving the specific value of variable.

GET

GET retrieves the value of variable.

Syntax

```
GET <keyword> [<variable>, <variable>]
```

Parameters

keyword

Keyword is a kind of function. It can only recognize alphanumeric character and "_". It ignores case-sensitive

variable[.i, .s, .b, .e [#]]

It is the name of variable to get.
.i : It means that type of variable is decimal number
.s : It means that type of variable is string
.b : It means that type of variable is binary encoded by base64. In this case, the length of binary data must be declared using [#].
.e : It means that a data is encrypted. (It is not implemented)
[#] : The length of data.

Response

If operation succeeds, it responds as following format.
GET <keyword> <name>=<value>, [<name>=<value>, …]

If operation fails, it responds as following format.
GET <keyword> FAILED

If only keyword is used without name, it will respond about all variables.

Remarks

If a data type returned is binary, it returned as following format.
GET <keyword> <name[n]>=<value>
'n' is a length of binary data.

SET

SET operation set the value of variable.

```
SET <keyword> <variable>=<value>, [<variable>=<value>, …]
```

Parameters

keyword

Keyword is a kind of function. It can only recognize alphanumeric character and "_". it ignores case-sensitive

variable[.i, .s, .b, .e [#]]

It is the name of variable to set.

.i : It means that type of variable is decimal number
.s : It means that type of variable is string
.b : It means that type of variable is binary encoded by base64. In this case, the length of binary data must be declared using [#].
.e : It means that a data is encrypted. (It is not implemented)
[#] : The length of data.

Value

It is the value of variable to set. This value is case sensitive.

Response

If it succeeds, the response is as follow format.
SET <keyword> OK
If it fail, the response is as follow format.
SET <keyword> FAILED

Remarks

Default data type is decimal.

If a type of variable is string, a value must be enclosed double quotation marks and it can not use double quotation mark in value.

SET ADD_USER user_id=”robot=12”

If data type is binary, it need to set binary size. 'n' is binary size as base64 coding.
SET <keyword> <name[n]>=<value>

If the value of variable is encrypted, put ".e" after variable. This function is not implemented yet.
SET <keyword> <name.e>=<encrypted value>

3.1. A method for a system configuration

EXT_CMD method is for configuring System.

Example

Request of EXT_CMD method

```
EXT_CMD / RTSP/1.0
CSeq: 5
CmdCount: 3
GET HARDWARE_INFO firmware_version
GET DI ch=0, on
SET DO ch=0, on=1
```

Response of EXT_CMD method

```
RTSP/1.0 200 OK
CSeq: 5
Date: Thu, Jun 01 2006 09:10:36 GMT
CmdCount: 3
GET HARDWARE_INFO firmware_version=”1.2.1”
GET DI ch=0, on=1
SET DO OK
```

There are keyword group in EXT_CMD method as below.

System information
System management
Device management
DDNS
User management
Event automation

3.1.1. System information

Keyword	name	Type	Operation
EXT_CMD_SPEC	client_version	i	GET/SET
	server_version	i	GET
SYSTEM_INFO			
HARDWARE_INFO	firmware_version	s	GET
	hw_revision	s	GET
	model_id	s	GET
	max_video_ch	i	GET
	max_audio_ch	i	GET
	max_video_out_ch	i	GET
	max_video_loopback_ch	i	GET
	max_audio_out_ch	i	GET
	max_di	i	GET
	max_do	i	GET
	max_serial_port	i	GET
	has_factory_default	i	GET
	has_watchdog	i	GET
	has_rtc	i	GET
	usn	b[#]	GET
	user_region	b[#]	GET/SET
	Information_string	s	GET

EXT_CMD_SPEC

client_version.i

[GET/SET] Extended method version that client uses. (ex. 1000)

server_version.i

[GET] Extended method version that server uses. (ex. 1000)

SYSTEM_INFO

It is not implemented yet

HARDWARE_INFO

firmware_version.s

[GET] Firmware version of server.

hw_revision.s

[GET] Hardware revision of server.

model_id.s

[GET] Unique number of the model.

max_video_ch.i

[GET] Maximum number of video input channel supported.

max_audio_ch.i

[GET] Maximum number of audio input channel supported.

max_video_out_ch.i

[GET] Maximum number of video output channel supported.

max_video_loopback_ch.i

[GET] Maximum number of video loop out channel supported.

max_audio_out_ch.i

[GET] Maximum number of audio output channel supported.

max_di.i

[GET] Maximum number of DI supported.

max_do.i

[GET] Maximum number of DO supported.

max_serial_port.i

[GET] Maximum number of serial port(RS-485/RS-232C) supported.

has_factory_default.i

[GET] This flag is for checking “factory default” function.

has_watchdog.i

[GET] This flag is for checking “watchdog” function.

has_rtc.i

[GET] This flag is for checking “real time clock” function..

usn.b[#]

[GET] Unique serial number.

user_region.b[#]

[GET/SET] Serial number that can be set by a user.

information_string.s

[GET] String that descript a hardware information.

3.1.2. System management

Keyword	name	Type	Operation
ACTIVATE	code	?	SET
NETWORK	mode	s	GET/SET
	ipaddr	s	GET/SET
	subnetmask	s	GET/SET
	gateway	s	GET/SET
	mac	s	GET/SET
	dns1	s	GET/SET
	dns2	s	GET/SET
	domain_name	s	GET/SET
	host_name	s	GET/SET
	pppoe_id	s	GET/SET
	pppoe_pswd	s	GET/SET
	time_mode	s	GET/SET
	ntp_server	s	GET/SET
	ntp_update	s	GET/SET
	http_port	i	GET/SET
	rtsp_port	i	GET/SET
	timeout	i	GET/SET
	packet_size	i	GET/SET
	padding	i	GET/SET
QOS	video_dscp	i	GET/SET
	audio_dscp	i	GET/SET
	event_dscp	i	GET/SET
	http_dscp	i	GET/SET
SYSTEM_MGR	system_name	s	GET/SET
	time	s	GET/SET
	timezone	s	GET/SET
	login_mode	s	GET/SET
	save_mode	s	GET/SET
	encrypt_mode	s	GET/SET
CMD_REBOOT	reboot	i	SET
	factory_default	i	SET
BRAND	brand_name	s	GET/SET
	product_full_name	s	GET/SET
	product_short_name	s	GET/SET
	product_number	s	GET/SET
	product_type	s	GET/SET
	web_url	s	GET/SET
DAYLIGHT_SAVING_TIME	enable	i	GET/SET
	offset	s	GET/SET

	start_day	i	GET/SET
	start_month	i	GET/SET
	start_time	s	GET/SET
	start_type_of_date	i	GET/SET
	stop_day	i	GET/SET
	stop_month	i	GET/SET
	stop_time	s	GET/SET
	stop_type_of_date	i	GET/SET

* ACTIVATE is not implemented.

ACTIVATE

It is not implemented yet.

ACTIVATE method is set activation code for protecting software and hardware from cracker .This function is optional and the default setting of activation function is disable.
For using this function, use "Console tool" to set configuration as enable and so on

NETWORK

mode.s

[GET/SET] Mode of IP. This variable can be one of static, dhcp, pppoe.

ipaddr.s

[GET/SET] IP address. This variable is available when mode is 'static'. After changing IP address, the server must restart.

subnetmask.s

[GET/SET] Submask. This variable is available when mode is 'static'.

gateway.s

[GET/SET] Gateway. This variable is available when mode is 'static'.

mac.s

[GET/SET] MAC address of system.

dns1.s

[GET/SET] First DNS address.

dns2.s

[GET/SET] Second DNS address.

domain_name.s

[GET/SET] The name of the domain to which the device belongs.

host_name.s

[GET/SET] The name of the device on the network, usually the same as the DNS name.

pppoe_id.s

[GET/SET] ID to connect by PPPoE. This variable is available when mode is 'pppoe'.

pppoe_pswd.s

[GET/SET] Password to connect by PPPoE. This variable is available when mode is 'pppoe'.

time_mode.s

[GET/SET] Mode to set server time. There are three types.

Mode	Remarks
ntp	Retrieving time from NTP server. It changes only system time.

rtc	Retrieving time from RTC.
ntp to rtc	Change system time and RTC after retrieving NTP server

ntp_server.s

[GET/SET] Specifies NTP server. Both URL and IP address are available. It is able to be set up to 3.
For example,
ntp_server=”url1 url2 url3”

ntp_update.s

[GET/SET] Update interval to update time from NTP server. Unit is second.

http_port.i

[GET/SET] Port number of HTTP server.

rtsp_port.i

[GET/SET] Port number of RTSP server.

timeout.i

[GET/SET] timeout.i is interval time to check the connection with the client. If the client does not send any command to the server in timeout, the server closes the socket for connecting with the client. If timeout.i is set to 0, the server does not check. This value can be 0 or more than 5.

packet_size.i

[GET/SET] It is a packet size of RTP. The range is 1448 to 8192. The first 12 bytes are being RTP header, actual media size is packet size -12 (bytes).

padding.i

[GET/SET] If padding is 1, the stream has padding data for UDA5 API. The padding data includes the time information and the sequence number.

QOS

In order to use QOS function, a router must support DiffServ model.

DSCP which DiffServer Model recommends is as follow
- Default PHB : 000000b
- Expedited Forwarding PHB : 101110b
- Assured Forwarding PHB

Drop Precedence	Class 1	Class 2	Class 3	Class 4
Low Drop	001010	010010	011010	100010
Med Drop	001100	010100	011100	100100
High Drop	001110	010110	011110	100110

video_dscp.i

[GET/SET] DSCP of the video packet

audio_dscp.i

[GET/SET] DSCP of the audio packet

event_dscp.i

[GET/SET] DSCP of the event packet

http_dscp.i

[GET/SET] DSCP of the http packet
It is not implemented.

SYSTEM_MGR

system_name.s

[GET/SET] Specific system name for distinguishing among the systeams.

time.s

[GET/SET] Current time. The time format is ‘YYYY:MM:DD HH:MM:SS’..

timezone.s

[GET/SET] Speicifies the timezone.
For example
Timezone.s=”GMT+5”

login_mode.s

[GET/SET] RTSP login mode. This variable can be one of the following values.

login_mode	Description
open	can control all of functions without login
login_only	can control all of functions with login
login_access	can control a function as depending on user's access level

save_mode.s

[GET/SET] Specifies the save time for changed configuration.

save_mode	Description
auto	The configuration is saved automatically whenever the configuration is changed.
manual	The configuration is saved when the client requests to save it.

encrpyt_mode.s

[GET/SET] The encryption mode of the password.
(It is not implemented)

CMD_REBOOT

reboot.i

[SET] Set 1 to reboot system.

factory_default.i

[SET] Set 1 to restore the factory default.
If you execute factory default, all environment variables except variables at below table are restored to default.

Keyword	Name
HARDWARE_INFO	firmware_version
	hw_revision
	model_id
	max_video_ch
	max
	max_audio_ch
	max_video_out_ch
	max_video_loopback_ch
	max_audio_out_ch
	max_di
	max_do
	max_serial_port
	has_factory_default
	has_watchdog
	has_rtc
	usn.b[#]
	user_region.b[#]
	information_string
ACTIVATE	code
NETWORK	mode
	ipaddr
	subnetmask
	gateway
	mac
	pppoe_id
	pppoe_pswd
	http_port
	rtsp_port
SYSTEM_MGR	time
DDNS	address
	user_id
	user_pw
	dns_name

BRAND

brand_name.s

[GET/SET] The brand of the product

product_full_name.s

[GET/SET] The full name of the product

product_short_name.s

[GET/SET] The short name of the product

product_number.s

[GET/SET] The product serial number

product_type.s

[GET/SET] The product type

web_url.s

[GET/SET] The URL to visit for support and information about the product

DAYLIGHT_SAVING_TIME

enable.i

[GET/SET] Enable/disable DST (Daylight Saving Time).

offset.s

[GET/SET] The amount of time the clock should be turned back/forward (hh:mm:ss), due to DST.

start_day.i

[GET/SET] The range is 1 to 31.

start_month.i

[GET/SET] The number of months since January in the range 0 to 11.

start_time.s

[GET/SET] Indicates the time (hh:mm:ss) when DST should be enabled. start_time = 02:00:00 means that DST should be enabled two hours after midnight.

stop_day.i

[GET/SET] The range is 1 to 31.

stop_month.i

[GET/SET] The number of months since January in the range 0 to 11.

stop_time.s

[GET/SET] Indicates the time (hh:mm:ss) when DST should be disabled. stop_time = 02:00:00 means that DST should be disabled two hours after midnight.

3.1.3. Device management

Keyword	name	Type	Operation
SERIAL_PORT	portid	i	GSET/SET
	baudrate	i	GET/SET
	databits	i	GET/SET
	parity	s	GET/SET
	stopbit	i	GET/SET
	flow_control	i	GET/SET
	silent	s	GET/SET
SERIAL_DATA	portid	i	GSET/SET
	mode	s	GET/SET
	data	b[#]	GET/SET
PTZ			
WATCHDOG	enable	i	GET/SET
	timeout	i	GET/SET
DI	ch	i	GSET
	enable	i	GET/SET
	name	s	GET/SET
	mode	s	GET/SET
DO	ch	i	GSET/SET
	enable	i	GET/SET
	name	s	GET/SET
VIDEO_OUT	ch	i	GET/SET
	outch	i	GET/SET
VIDEO_LOOPBACK	ch	i	GSET/SET
	enable	i	GET/SET

SERIAL_PORT

portid.i

[GSET/SET] Serial port ID, The port ID can be set from 0 to 2.

baudrate.i

[GET/SET] The baudrate can be set from 2400 to 115200.

databits.i

[GET/SET] Only 8 can be set as data bits.

parity.s

[GET/SET] none, even, odd.

stopbit.i

[GET/SET] This variable can be one of the following values.

stopbit	Description
0	1
1	1.5
2	2

flow_control.i

[GET/SET] This variable can be one of the following values.

flowcontrol	Description
0	No
1	Xon/Xoff
2	Hardware

silent.s

[GET/SET] This variable can be one of the following values.

silent.s	Description
all	RS-232C is used for connecting RS-232C devices.
outonly	RS-232C is used as console, but only input is available
No	RS-232C is used as console

SERIAL_DATA

portid.i

[GSET/SET] Serial port ID

mode.s

[GSET/SET] If mode is internal, the server application controls the serial port. If mode is external the external application controls the serial port.
When mode is external, the name of external application is required.

For example,
If the external application is “tcp2ser”,
Mode.s=”external tcp2ser”

data.b[#]

[GET/SET] Data to send through serial port.

WATCHDOG

enable.i

[GET/SET] Specifiy whether to enable or disable.

enable	Description
0	Disable watchdog
1	Enable watchdog

timeout.i

[GET/SET] Timeout for watchdog. If the server is halted, the system is rebooted after passing timeout. Unit is second.

DI

ch.i

[GSET] Channel number of DI.

enable.i

[GET/SET] If enable is 1, DI is on. If enable is 0, DI is off.

name.i

[GET/SET] Name of DI.

mode.i

[GET/SET] The type of D/I. It can be ‘relay’ or ‘voltage’.

DO

ch.i

[GSET/SET] Channel number of DO.

enable.i

[GET/SET] If enable is 1, DO is on. If enable is 0, DO is off.

name.i

[GET/SET] Name of DO.

VIDEO_OUT

ch.i

[GET/SET] Channel index of video out

outch.i

[GET/SET] Mode of external video out. If outch.i is set as 255, it displays the quad video on external monitor. If outch.i is set as 0,1,2 or 3, it displays the selected channel on external monitor.

enable.i

[GET/SET] If enable is 1, the video out is on. If enable is 0, the video out is off.

VIDEO_LOOPBACK

ch.i

[GET/SET] Channel number to loop back.

enable.i

[GET/SET] If enable is 1, the loop-back is on. If enable is 0, the loop-back is off.

Remarks

Dual stream can be also configured by VIDEO_LOOPBACK. For detailed information about dual stream, refer to section 3.5 at UDA5 NVE Series SDK Manual.

ch.i : 0: connect VIN1 to VIN2.
1 : connect VIN3 to VIN4

For example, if VIN1 connects with VIN2 and VIN3 connects with VIN4

```
EXT_CMD / RTSP/1.0
CSeq: 5
CmdCount: 1
SET VIDEO_LOOPBACK ch=0, enable=1, ch=1, enable=1
```

3.1.4. DDNS

Keyword	name	Type	Operation
DDNS	enable	i	GET/SET
	server_type	s	GET/SET
	address	s	GET/SET
	user_id	s	GET/SET
	user_pw	s	GET/SET
	update_time	i	GET/SET
	dns_name	s	GET/SET
	ip_type	s	GET/SET

DDNS

For using DDNS, a account must be registered before using DDNS.

enable.i

[GET/SET] If enable.i is 1, DDNS is enabled. If enable.i is 0, DDNS is disabled.

server_type.s

[GET/SET] Type of DDNS server. The default value is dyndns.

address.s

[GET/SET] Address of DDNS server.

user_id.s

[GET/SET] User ID of DDNS

user_pw.s

[GET/SET] Password of DDNS.

update_time.i

[GET/SET] Period of updating IP address. Unit is minute.

dns_name.s

[GET/SET] Name of DNS.

ip_type.i

[GET/SET] If this value is ‘real’, NVE send the IP address assigned to IP share device to DDNS server. If this value is ‘local’, NVE inform the IP address in the private address space to DDNS server.

Example
If DNS name assigned to NVE is ‘user1.dyndns.org’, the configuration of DDNS is as follow.

```
EXT_CMD / RTSP/1.0
CSeq: 1
CmdCount: 1
SET DDNS enable.i=1, server_type.s="dyndns", address.s="www.dyndns.com", user_id.s="user1",
user_pw.s="pass", update_time.i=60, dns_name.s="user1.dyndns.org"
```

3.1.5. User Management

Keyword	name	Type	Operation
ADD_USER	name	s	SET
	password	s	SET
	access_level	i	SET
DEL_USER	name	s	SET
USER_INFO	name	s	GSET/SET
	password	s	GET/SET
	access_level	i	GET/SET
USER_LIST	count	i	GET
	list	s	GET
CONNECTION_LIST	count	i	GET
	list	s	GET

ADD_USER

name.s

[SET] User ID to add

password.s

[SET] Password to add

access_level.i

[SET] Access level of a user to add.

DEL_USER

name.s

[SET] User ID to delete

USER_INFO

name.s

[GSET/SET] User ID to change or get.

password.s

[GET/SET] Password to change or get.

access_level.i

[GET/SET] Access level to change or get.

USER_LIST

count.i

[GET] Number of registered users.

list.s

[GET] List of registered users.
For example :
data="admin jbyoon user2"

CONNECTION_LIST

count.i

[GET] Number of the connected clinet.

list.s

[GET] List of the connected user and IP address.
Ex) data="jbyoon 192.168.12.2 jbyoon 192.168.20.100 ricky 192.158.23.23"

3.1.6. Event automation

3.2. A method for a channel configuration

EXT_SUBSESSION_CMD method is for configuring each channel.

Example

Request of EXT_SUBSESSION_CMD method

```
EXT_SUBSESSION_CMD /track1 RTSP/1.0
CSeq: 5
Session: 6004
CmdCount: 2
GET VIDEO_FORMAT
SET BITRATE mode=”cbr”, cbr_bitrate=1500000
```

Response of EXT_SUBSESSION_CMD method

```
RTSP/1.0 200 OK
CSeq: 5
Session: 6004
Date: Thu, Jun 01 2006 09:10:36 GMT
CmdCount: 2
GET VIDEO_FORMAT tv_mode.s="auto", tv_standard.s="ntsc", defined_imagesize="4cif",
width.i=704, height.i=480
SET BITRATE OK
```

There are keyword groups in EXT_SUBSESSION_CMD method as below.

Video Properties
Audio Properties
Motion Detection
OSD
Multicast

3.2.1. Video Properties

Keyword	name	Type	Operation
VIDEO_INFO	name	s	GET/SET
	enable	i	GET/SET
	hdelay	i	GET/SET
	vdelay	I	GET/SET
	stream	I	GET/SET
	session_id	i	GET/SET
VIDEO_FORMAT	tv_mode	s	GET/SET
	tv_standard	s	GET/SET
	defined_imagesize	s	GET/SET
	width	i	GET
	height	i	GET
	deinterlace	i	GET/SET
FRAME_RATE	method	s	GET/SET
	fps1000	i	GET/SET
	skip_frame	i	GET/SET
	max_fps	i	GET
BITRATE	mode	s	GET/SET
	cbr_bitrate	i	GET/SET
	vbr_quant	i	GET/SET
	vbr_quant_i	i	GET/SET
	vbr_quant_b	i	GET/SET
	vbr_quant_p	i	GET/SET
VIDEO_CODEC	type	s	GET/SET
	gop_size	i	GET/SET
	noisefilter	i	GET/SET
	profile_level	s	GET/SET
	video_object_type	s	GET/SET
VIDEO_COLOR_ATTR	brightness	i	GET/SET
	contrast	i	GET/SET
	saturation	i	GET/SET
	hue	i	GET/SET
	sharpness	i	GET/SET
VIDEO_STATUS	on	i	GET
VIDEO_SNAPSHOT	enablei	i	GET/SET
	num_of_picuter	i	GET/SET

Caution!
If your item is one of the IPC series or NVE100 with PAL video format, Hue value adjustment doesn't work at all. This is because of the characteristic of the decoder chip built in IPC series and NVE100. If your video format is NTSC, it has no problem in hue value adjustment.

VIDEO_INFO

name.s

[GET/SET] the nick name of channel

enable.i

[GET/SET] enable/disable the video stream

enable	Description
0	Disable the video stream
1	Enable the video stream

hdelay.i

[GET/SET] adjust the horizontal delay of the video input. The range is -32 to 32.

vdelay.i

[GET/SET] adjust the vertical delay of the video input. The range is -4 to 4.

stream.i

[GET/SET] If one of clients sets the value as '0', it makes the server to stop to send the RTSP video stream to the client only who sent the request with '0'. This command works only when the connection between the server and the client is available as this configuration is not saved.

NOTE : The other clients do not get affected by this command of separate client. Because of this, you should set the value as '0' always after the 'Setup' unless you want to get the stream anymore.

stream_id.i

[GET/SET] ‘Session ID’ is also used to enable or disable the stream because the RTSP server requires the information of session ID for the streaming connection between server and client.

VIDEO_FORMAT

tv_mode.s

[GET/SET] Mode of video input format

tv_mode	Description
autio	Auto detect mode
Fixed	User fixed mode

tv_standard.s

[GET/SET] Video input format. It can be set “ntsc” or “pal”.

Defined_imagesize.s

[GET/SET] Image size.

Defined image size	Real image size	
	NTSC	PAL
D1	720x480	720x576
HD1	720x240	720x288
VGA	640x480	640x480
QVGA	320x240	320x240
4CIF	704x480	704x576
2CIF	704x240	704x288
CIF	352x240	352x288
QCIF	176x112	176x144

width.i

[GET] Image width

height.i

[GET] Image height.

deinterlace.i

[GET] enable/disable de-interlace function.

deinterlace	Description
0	Turn off the deinterlace
1	Turn on the deinterlace

VIDEO_STATUS

on.i

[GET] Video status flag. If this flag is 1, a video signal is on. If this flag is 0, a video signal is off.

FRAME_RATE

method.s

[GET/SET] Method of frame rate. It can be set “fps1000” or “skip_frame”.

Fps1000.i

[GET/SET] value of fps1000.i is actual frame rate oordina by 1000.
For example
frame rate = 20
fps1000.i = 20 * 1000
fps1000.i = 20000

skip_frame.i

[GET/SET] the number of skipping frames.
For example
Skip frame = n
FPS = max_FPS/(1+n)

max_fps.i

[GET] Max frame rate.

BITRATE

mode.s

[GET/SET] Bitrate mode. It can be set “cbr” or “vbr”.

Cbr_bitrate.i

[GET/SET] Bitrate if mode is “cbr”. The range is 1 ~ 10000000 (10Mbps).

Vbr_quant.i

[GET/SET] Quantization value if mode is “vbr”. The range is 1 ~ 255 and lower value makes better image.

Vbr_quant_i.i

[GET/SET] Quantization value of I picture. The range is 1~255.

Vbr_quant_b.i

[GET/SET] Quantization value of B picture. The range is 1~255.

Vbr_quant_p.i

[GET/SET] Quantization value of P picture. The range is 1~255.

VIDEO_CODEC

type.s

[GET/SET] Codec type. It can be set on of mpeg4, mjpeg, h263, mpeg1.
* After changing codec type. SETUP must restart.

Gop_size.i

[GET/SET] GOP (Group Of Pictures) size. If gop_siz is 1, only I picture is used. The range is 1 ~ 255.

Noisefilter.i

[GET/SET] It can be set from 0 to 255. The default is 128.The higher number gives better image.

Profile_level.i

[GET/SET] Specifies the profile level of MPEG-44. It is not implemented yet.

Video_object_type.i

[GET/SET] Specifies the video object type of MPEG-4. It can be selected among “advanced_simple”, “simple”, “core” or “main”.. It is not implemented.

VIDEO_COLOR_ATTR

brightness.i

[GET/SET] The range is 0 ~ 255. The default value is 128.

Contrast.i

[GET/SET] The range is 0 ~ 255. The default value is 128.

Saturation.i

[GET/SET] The range is 0 ~ 255. The default value is 128.

Hue.i

[GET/SET] The range is 0 ~ 255. The default value is 128.

Sharpness.i

[GET/SET] The range is 0 ~ 255. The default value is 128.

VIDEO_SNAPSHOT

enable.i

[GET/SET] Enable/Disable the snapshot

3.2.2. Audio Properties

Keyword	Name	Type	Operation
AUDIO_IN	enable	i	GET/SET
	stream_type	s	GET/SET
	sample_rate	i	GET/SET
	data_bits	i	GET/SET
	channels	i	GET/SET
	gain	i	GET/SET
	name	s	GET/SET
AUDIO_OUT	owner	s	GET/SET
	enable	i	GET/SET
	stream_type	s	GET/SET
	sample_rate	i	GET/SET
	data_bits	i	GET/SET
	channels	i	GET/SET
	gain	i	GET/SET
	data	b[#]	SET
	name	s	GET/SET

AUDIO_IN

enable.i

[GET/SET] Specifyic whether to enable or disable audio input.

Enable	Description
0	Disable audio input
1	Enable audio input

stream_type.s

[GET/SET] Stream type. It can be set pcm, uLaw or aLaw.
* After changing stream_type, SETUP must be restart

sample_rate.i

[GET/SET] Sample per second. It can be set 8000 or 16000.

Data_bits.i

[GET/SET] Data bits. It can be set 8 or 16.

Channels.i

[GET/SET] Channel number of audio.

Channel	Description
0	mono

gain.i

[GET/SET] Gain of audio input. The range is 0 ~ 255. The default value is 128.

Name.i

[GET/SET] the name of the audio channel.

AUDIO_OUT

owner.s

[GET] Client name that uses audio output.

Enable.i

[GET/SET] Specify whether to enable or disable audio output.

Enable	Description
0	Disable audio output
1	Enable audio output

stream_type.s

[GET/SET] Stream type. It can be set pcm or uLaw.

Sample_rate.i

[GET/SET] Sample per second. It can be set 8000 or 16000.

Data_bits.i

[GET/SET] Data bits. It can be set 8 or 16.

Channels.i

[GET/SET] Channel number of audio.

Channel	Description
0	mono

gain.i

[GET/SET] Gain of audio input. The range is 0 ~ 255. The default value is 128.

Data.b[#]

[SET] Audio data encoded by base 64 for sending to server

name.i

[GET/SET] the name of audio channel

3.2.3. Motion Detection

Keyword	Name	Type	Operation
MOTION_DETECTION_PROPERTY	max_layer_count.i	i	GET
	max_rect_count.i	s	GET
	area_type_list.s	s	GET
MOTION_DETECTION	layer_id	i	GSET/SET
	enable	i	GET/SET
	acitivity	i	GET/SET
	threshold	i	GET/SET
	area_type	s	GET/SET
	area_count	i	GET/SET
	info_type	s	GET/SET
	pxl_width	i	GET/SET
	pxl_height	i	GET/SET
	bm_width	i	GET/SET
	bm_height	i	GET/SET
	bitmap_image	b[#]	GET/SET
	rect_id	i	GET/SET
	rc_x	i	GET/SET
	rc_y	i	GET/SET
	rc_width	i	GET/SET
	rc_height	i	GET/SET

MOTION_DETECTION_PROPERTY

max_layer_count.i

[GET] Maximum number of MD layer supported.

Area_type_list.s

[GET] Method list of MD area setting supported.
For example
GET MOTION_DETECTION_PROPERTY area_type_list.s=”rect bitmap”

max_rect_count.i

[GET] Maximum number of the MD rectangle supported.

MOTION_DETECTION

layer_id.i

[GSET/SET] Area ID to set motion detection

enable.i

[GET/SET]

activity.i

[GET/SET] Minimum proportion of macroblocks detection. The value of range is 1~255.
For example
macroblock = 20
activity.i = 128
proportion = 128 / 255 = 50 %
In this case, motion detection oordi when more than 10 assinged macroblocks is detected

threshold.i

[GET/SET] Sensitivity of motion detection in macroblock. The range of threshold is 1 to 255. Lower number represents more sensitivity.

Area_type.s

[GET/SET] The method of selecting area. It can be set by "rect" or "bitmap". NVE supports only "rect".

Area_count.s

[GET/SET] Number of the MD area. If area_type.s is "rect", area_count.s is the number of the MD rectangle. If area_type.s is "bitmap", area_count.s is 1.

Info_type.s

[GET/SET] The motion detection information is changed according to info_type as following table.

Info_type	Motion detection information
flag	Layer
count	Layer + Macroblock Count + Width + Height
bitmap	Layer + Macroblock Count + Width + Height + MD Bitmap Information
continuous	Layer + Macroblock Count

If info_type is set as flag, count or bitmap, NVE sends the motion detection information only when motion is detected. NVE keeps on sending the motion detection information if info_type is set as continous.

Layer : char
The layer ID where MD event occurs. It is bitwise value. E.g.) Layer is 3 when MD event occurs at both layer 0 and layer1.

Macroblcok Count : DWORD
The number of macroblock motion detected when MD event occurs.
Width : WORD
The width of a image.
Height : WORD
The height of a image.
MD Bitmap information : char array

pxl_width.i

[GET/SET]Width of image which is mapped by one pixel of bitmap image for configuring motion detection area. It is fixed as 16.

Pxl_height.i

[GET/SET] Height of image which is mapped by one pixel of bitmap image for configuring motion detection area. It is fixed as 16.

Bm_width.i

[GET/SET] Width of bitmap image for configuring motion detection area. It is only available to use with bitmap method.

Bm_height.i

[GET/SET] Height of bitmap image for configuring motion detection area. It is only available to use with bitmap method.

Bitmap_image.b[#]

[GET/SET] Motion detection bitmap image. It is only available to use with bitmap method.

Rect_id.i

[GET/SET] ID of rectangle. If area_type.s is bitmap. It is ignored.

Rc_x.i

[GET/SET] X oordinate of motion detection area. It is only available to use with rectangle method.

Rc_y.i

[GET/SET] Y oordinate of motion detection area. It is only available to use with rectangle method.

Rc_width.i

[GET/SET] Width of motion detection area. It is only available to use with rectangle method.

Rc_height.i

[GET/SET] Height of motion detection area. It is only available to use with rectangle method.

3.2.4. OSD

Keyword	name	Type	Operation
OSD_STRING	method	s	GET/SET
	x	i	GET/SET
	y	i	GET/SET
	string	s	GET/SET
	color	i	GET/SET
OSD_BITMAP	method	s	GET/SET
	x	i	GET/SET
	y	i	GET/SET
	width	i	GET/SET
	height	i	GET/SET
	bitmap	b[#]	GET/SET
	color	i	GET/SET
OSD_TIME	enable	i	GET/SET
	format	s	GET/SET
	x	i	GET/SET
	y	i	GET/SET

* OSD_BITMAP is not implemented.

OSD_STRING

method.s

[GET/SET] It must be set as string..

x.i

[GET/SET] X-coordinate of the starting point of OSD string.

y.i

[GET/SET] Y-coordinate of the starting point of OSD string.

String.s

[GET/SET] String to display

color.i

[GET/SET] Color of string.

OSD_BITMAP

method.s

[GET/SET] It must be set as bitmap.

x.i

[GET/SET] X-coordinate of the starting point of OSD string.

y.i

[GET/SET] Y-coordinate of the starting point of OSD string.

Width.i

[GET/SET] Width of OSD image

height.i

[GET/SET] Height of OSD image

bitmap.b[#]

[GET/SET] Binary data of OSD image

color.i

[GET/SET] Color of OSD image.

OSD_TIME

enable.s

[GET/SET] The time information is displayed at OSD if enable is 1.

Format.i

[GET/SET] The format of time information.

Format
YYMMDD:hhmmss, MM/DD/YYYY hh:mm:ss, DD/MM/YYYY hh:mm:ss, YYYY/MM/DD hh:mm:ss, MMM DD YYYY hh:mm:ss, DD MMM YYYY hh:mm:ss

x.i

[GET/SET] X-coordinate of the starting point of the time information string.

y.i

[GET/SET] Y-coordinate of the starting point of the time information string.

Remark

If positions of the OSD string and the OSD time are overlapped, the OSD time is displayed.
The color of OSD time is affected by the color of OSD string.

If your models is NVE 2000, NVE 4000 or any multi channel encoding model, OSD time setting is available only on 1st channel and the other channels do not support the Time OSD setting for the systematical reason of multi channel models of NVE.

3.2.5. Multicast

Keyword	name	Type	Operation
RTP_MULTICAST	type	s	GET/SET
	enable	i	GET/SET
	address	s	GET/SET
	port	i	GET/SET
	ttl	i	GET/SET

RTP_MULTICAST

type.i

[GET/SET] Data type via muticast. .It must be specified one of video, audio or event.

Enable.i

[GET/SET] Specifies whether to enable or disable multicasting.

Enable	Description
0	Disable multicasting
1	Enable multicasting

address.s

[GET/SET] Address of RTP Multicast.

Port.i

[GET/SET] Port number of RTP Multicast. Only even number is available.

Ttl.i

[GET/SET] TTL(Time To Live)

Example

```
EXT_SUBSESSION_CMD /track6 RTSP/1.0
Cseq: 1
CmdCount: 1
SET RTP_MULTICAST type.s=”video”, enable.i=1, address.s=”224.10.10.10”, port.i=6000, ttl=128
```

* Note
For now there are several limitations.
- If the one type is enabled, the other types are also enabled.

In order to retrieve via multicast, URL must include ‘multicast’ as the following example.

```
DESCRIBE /multicast RTSP/1.0
Cseq: 1
Accept: application/sdp

SETUP /multicast/track1 RTSP/1.0
Cseq: 1
Transport: RTP/AVP;multicast
```

Client can retrieve the data after joining to multicast group. IP address and the port number of Multicast group can be obtained by SDP description.

For example

```
.....
a=rtcp-unicast: reflection
.....
m=video 6000 RTP/AVP 96
c=IN IP4 224.10.10.10/128
.....
```

Port number : 6000
IP address : 224.10.10.10.
TTL : 128

3.2.6. Encryption

Keyword	name	Type	Operation
Encryption	video_encrypt_mode	s	GET/SET
	video_encrypt_key	b[#]	GET/SET
	video_encrypt_type	s	GET/SET
	video_encrypt_size	i	GET/SET
	audio_encrypt_mode	s	GET/SET
	audio_encrypt_key	b[#]	GET/SET
	audio_encrypt_size	s	GET/SET

ENCRYPTION

video_encryption_mode.i

[GET/SET] the method for encrypting video packet.

Encryption mode	Description
none	Disable encryption of video packet
aes_cbc	Encrypting video packet by AES CBC

video_encrypt_key.b[#]

[GET/SET] Specifies the key for encryption. The size of key is upto 16 bytes.

Video_encrypt_type.s

[GET/SET] Specifies which frames do you want to encrypt.

Encryption type	Description
i_only	Encrypting only I frame
p_only	Encrypting only P frames
All	Encrypting all frames.

Video_encrypt_size.s

[GET/SET] Specifies RTP video packet size to encrypt. The maximum size is the packet size except 12 bytes that is used for RTP header. The packet size is defined by **NETWORK**:***packet_size*** in 16 bytes unit. 0 means all parts of video packet except RTP header are encrypted. If the packet size -12 is not be aligned by 16 bytes, the padding data is added at end of packet.

Audio_encryption_mode.i

[GET/SET] the method for encrypting audio packet.

Encryption mode	Description
none	Disable encryption of video packet
aes_cbc	Encrypting video packet by AES CBC

audio_encrypt_key.b[#]

[GET/SET] Specifies the key for encryption. The size of key is upto 16 bytes.

Audio_encrypt_size.s

[GET/SET] Specifies RTP audio packet size to encrypt. The maximum size is the packet size except 12 bytes that is used for RTP header. The packet size is defined by **NETWORK**:***packet_size*** in 16 bytes unit. 0 means all parts of video packet except RTP header are encrypted. If the packet size -12 is not be aligned by 16 bytes, the padding data is added at end of packet.

Example

The encrypted packet is as follow.

RTP header (12 bytes)	Encryption header (4 bytes)	Size of encrypted packet. (4 bytes)	Encrypted data

The Encryption header shows the encryption method as follow.
Aes_cbc : MAKEFOURCC('A','E','S','0') (0x30534541)

```
#define MAKEFOURCC(ch0, ch1, ch2, ch3)    \
                ((unsigned int)(unsigned char)(ch0) |      \
                ((unsigned int)(unsigned char)(ch1) << 8) |        \
                ((unsigned int)(unsigned char)(ch2) << 16) |       \
                ((unsigned int)(unsigned char)(ch3) << 24 ))
```

4. Padding data

TIME STAMP (8Bytes)	FRAME SEQUENCE NUMBER(4Bytes)

Timestamp
Format : timeval structure
Size : 8 byte

Sequence number
Format : unsigned long
Size : 4 bytes

The padding data has an additional timestamp and the sequence number for the media data. The objective of using padding data is to obtain more exact time than the one you can get from the RTP header, which is calculated using the time base of RTCP. On the other hand, the timestamp in the padding points out the exact time when the media data is created.

5. RTSP over HTTP

RTSP over HTTP enables RTSP and RTP data to be placed in HTTP requests and replies. This feature is especially useful where firewalls and proxy servers are installed to protect against security attacks.

Using a Web Server

To use RTSP over HTTP, two separate sessions should be set up and they need to be bound together to form a two-way connection: GET and POST

HTTP Client Request requirement

- Specify the channel number
- Be compatible with HTTP version 1.0
- Include x-sessioncookie directive in the header
- Include Accept directive as application/x-rtsp-tunnelled

Sample Client GET Request

```
GET /0 HTTP/1.0
x-sessioncookie: 12345678
Accept: application/x-rtsp-tunnelled
```

0 is a channel number. To connect to a different channel, type in the channel number you want to use.
x-sessioncookie can be a random number.

Response to the GET request above

```
HTTP/1.0 200 OK
Server: boa
Date: Thu, 26 Jul 2007 00:52:52 GMT
Content-Type: application/x-rtsp-tunnelled
```

Sample Client POST request

```
POST /0 HTTP/1.0
x-sessioncookie: 12345678
Content-Length: 32767
Content-Type: application/x-rtsp-tunnelled
```

x-sessioncookie value should be the same with one in HTTP Client Request.

Response to the POST request above

No response will be returned when POST method is issued.

Sending RTSP command should follow the following rules.

- POST session is used to send an RTSP command and GET session is used to get the result.
- RTSP command must be 64base encoded to prevent the misinterpretation of HTTP server.

Sample RTSP command to test

```
EXT_CMD / RTSP/1.0
Cseq: 1
CmdCount: 1
GET NETWORK mode
```

RTSP command should be sent in base 64 encoded format.

```
RvhUX0NNRCAvIFJUU1AvMS4wDQpDU2VxOiAxDQpDbWRDb3VudDogMQ0KR0VUIE5F
VFdPUksgbW9kZQ0KDQo=
```

Note: In a telnet 80 port connection, return key should be pressed twice at the end of the original RTSP command lines before converting it to base64 encoded format.

You will get the result in the Get session.

```
RTSP/1.0 200 OK
Cseq: 1
Date: Wed, Aug 08 2007 10:32:03 GMT
CmdCount: 1
GET NETWORK mode.s=”static”
Using UDA5
```

Using UDA5

If you want to use UDA5 API, you just need to add **NET5_ST_HTTP_TUNNELING** as in the following example.

```
Net5ConnectEx(“192.168.0.10”, NET5_ST_HTTP_TUNNELING, 5);
```

Caution!

Using RTSP over HTTP involves longer path end-to end, encoding and decoding of data transmitted, thus making the command response speed slower. When comparing the speed with that of UDP/TCP, it usually takes more time to get the result. (Client Base64 decoding Web server Base64 decoding RTSP Server)

6. How to get an event data

The event data is controlled in the same way as video and audio. SDP description is as following.

```
……
m=event 0 RTP/AVP 100
c=IN IP4 0.0.0.0
a=rtpmap:100 EVENT-NVE/1000
a=control:track3
……
```

SETUP must be executed before retrieving the event data and it can be controlled by PLAY, PAUSE and TEARDOWN.

NVE is sending event data to the client as below situation.
- When the motion is detected.
- When the configuration is changed by the other client.
- When the sensor event occur.
- When the data of serial port is exist.
- When the video status is changed.

The data format of event is as following

```
typedef struct _NVE_EVENT {
        unsigned char   ChID;
        unsigned char   Major;
        unsigned char   Minor;
        unsigned char   Param;

        struct timeval  tTimestamp;
        int             nLength;
        void            *pData;
} NVE_EVENT;
```

Members

ChID

Channel number that the event occurs.
0xFF : all channel
0x01 ~ 0x0F : video channel
0x10 ~ 0x1F : audio channel

Major

The major number of event type

Event Type	Major Number
Motion Detection	1
DI	2

Video Status Change	3
Boot	4
Motion Info	5
Property Change	11
Serial Data	12

Minor

Minor number of event type

Event Type	Minor Number
Motion info flag	0
Motion info count	2
Motion info bitmap	4
Motion info coutinuous	8

Param

Event data

tTimestamp

Time that the event occur

nLength

Length of additional data.

pData

Pointer of additional data buffer.

Remarks

Event data is classified into two categories: one for global event, the other for channel specific event.

Global events : DI, Boot, Serial Data
Channel specific event: Motion Detection. Video Status Change, Motion Info, Property Change

The event data is retrieved via Param. If the additional event data exists, it can be transferred via pData. Because pData is void, nLength must be needed.

- Motion detection
  Param indicates the layer where MD events occur. It is a bitwise value.
  For example, if MD events occur at layer 0 and layer 2, ***Param*** is 5(00000101b).

  pData and ***nLength*** can be changed according to ***info_type***.

  - flag

nLength is 0 and pData is null.

- count
nLength is 8 and pData has as following structure.

```
Typedef struct _MD_INFO_COUNT {
        int mdCount;
        short width, height;
} MD_INFO_COUNT;
```

- bitmap

nLength is 207 and pData has the bimap information in addition to MD_INFO_COUNT structure The MD macroblock information. One bit indicates the status of one macroblock. If value is 1, it indicates motion detected.

- continuous
nLength is 16 and pData has the following structure.

```
Typedef struct _MD_INFO_CONTINUOUS {
        int mdCount;
        short width, height;
        short roiMBCount[MAX_REGIONS_OF_INTEREST];
} MD_INFO_CONTINUOUS;
```

- Property change
nLength : 4
pData : The structure is as follow.

```
Typedef struct _PROPERTY_CHANGE {
unsigned int    bImageSize      : 1;    // image size is changed
unsigned int    bVideoCodec     : 1;    // video codec is changed
unsigned int    bAudioCodec     : 1;    // audio codec is changed
unsigned int    reserved                : 29;
} PROPERTY_CHANGE;
```

Revision history

Rev.	Date	History
A	2007-08-06	Created.
B	2007-08-29	Updated for kernel 17X569
C	2007-09-05	Section 5 Event category is added. Minor errata corrected
D	2007-09-13	Updated the Motion Detection Threshold explanation
E	2008-04-15	Hue adjustment unavailable for PAL in only IPC and NVE100
F	2008-06-05	OSD Time configuration corrected
7A.00	2009-02-17	FW K641.11710 updated.
7B.00	2009-03-20	Audio output and input : spec changed
7C.00	2009-05-15	OSD_TIME – date and time formats added
7D.00	2009-06-01	Added the description of padding data

Refer to WHAT'S NEW page for more detailed update.