

# Network

## SUNAPI

v2.6.2

2023-04-07



## Copyright

© 2023 Hanwha Vision Co., Ltd. All rights reserved.

## Restriction

Do not copy, distribute, or reproduce any part of this document without written approval from Hanwha Vision Co., Ltd.

## Disclaimer

Hanwha Vision Co., Ltd. has made every effort to ensure the completeness and accuracy of this document, but makes no guarantee as to the information contained herein. All responsibility for proper and safe use of the information in this document lies with users. Hanwha Vision Co., Ltd. may revise or update this document without prior notice.

## Contact Information

Hanwha Vision Co., Ltd.

Hanwha Vision 6, Pangyo-ro 319beon-gil, Bundang-gu, Seongnam-si, Gyeonggi-do, 13488, KOREA  
[www.hanwhavision.com](http://www.hanwhavision.com)

Hanwha Vision America

500 Frank W. Burr Blvd. Suite 43 Teaneck, NJ 07666  
[hanwhavisionamerica.com](http://hanwhavisionamerica.com)

Hanwha Vision Europe

Heriot House, Heriot Road, Chertsey, Surrey, KT16 9DT, United Kingdom  
[hanwhavision.eu](http://hanwhavision.eu)

Hanwha Vision Middle East FZE

Jafza View 18, Office 2001-2003, Po Box 263572, Jebel Ali Free Zone, Dubai, United Arab Emirates  
[www.hanwhavision.com/ar](http://www.hanwhavision.com/ar)

# Table of Contents

1. Overview .....	9
1.1. Description .....	9
2. Network Interface .....	11
2.1. Description .....	11
2.2. Syntax .....	11
2.3. Parameters .....	11
2.4. Examples .....	14
2.4.1. Getting the system network interface settings .....	14
2.4.2. Setting a static IP .....	17
2.4.3. Setting DHCP mode .....	17
2.4.4. Setting the PPPoEPassword with encrypted password .....	18
3. DNS .....	19
3.1. Description .....	19
3.2. Syntax .....	19
3.3. Parameters .....	19
3.4. Examples .....	20
3.4.1. Getting the current DNS settings .....	20
3.4.2. Setting primary and secondary DNS server addresses .....	21
4. DDNS .....	22
4.1. Description .....	22
4.2. Syntax .....	22
4.3. Parameters .....	22
4.4. Examples .....	24
4.4.1. Getting the current DDNS configuration settings .....	24
4.4.2. Setting the DDNS type .....	26
4.4.3. Setting the PublicPassword with encrypted password .....	26
5. Bonjour .....	28
5.1. Description .....	28
5.2. Syntax .....	28
5.3. Parameters .....	28
5.4. Examples .....	28
5.4.1. Getting Bonjour settings .....	28
5.4.2. Enabling Bonjour and setting the friendly name as 'SNB6004' .....	29
6. UPnP Discovery .....	30
6.1. Description .....	30
6.2. Syntax .....	30

6.3. Parameters .....	30
6.4. Examples .....	30
6.4.1. Getting UPnP discovery settings .....	31
6.4.2. Enabling UPnP discovery and setting the friendly name as 'SNB6004' .....	31
7. Zeroconf .....	32
7.1. Description .....	32
7.2. Syntax .....	32
7.3. Parameters .....	32
7.4. Examples .....	33
7.4.1. Getting current Zeroconf settings .....	33
7.4.2. Enabling Zeroconf .....	33
8. SNMP Setup .....	35
8.1. Description .....	35
8.2. Syntax .....	35
8.3. Parameters .....	35
8.4. Examples .....	36
8.4.1. Getting current SNMP settings .....	36
8.4.2. Setting SNMP .....	37
8.4.3. Setting the Password with encrypted password .....	37
9. SNMP Trap Settings .....	38
9.1. Description .....	38
9.2. Syntax .....	38
9.3. Parameters .....	38
9.4. Examples .....	39
9.4.1. Getting current SNMP Trap settings .....	40
9.4.2. Setting an SNMP trap .....	41
9.4.3. Setting community not to use .....	41
10. QoS Setup .....	42
10.1. Description .....	42
10.2. Syntax .....	42
10.3. Parameters .....	42
10.4. Examples .....	43
10.4.1. Getting current QoS settings .....	43
10.4.2. Adding a QoS-enabled IPv4 Address .....	45
10.4.3. Updating a QoS-enabled IPv4 Address .....	45
10.4.4. Deleting a QoS IPv6 Address .....	45
11. HTTP Port .....	47
11.1. Description .....	47
11.2. Syntax .....	47
11.3. Parameters .....	47

11.4. Getting the HTTP port number .....	47
11.4.1. Getting the current HTTP port number .....	47
11.5. Setting the HTTP port .....	48
11.5.1. Setting the HTTP port to '8080' .....	48
12. HTTPS Port .....	49
12.1. Description .....	49
12.2. Syntax .....	49
12.3. Parameters .....	49
12.4. Examples .....	49
12.4.1. Getting the current HTTPS port number .....	49
12.4.2. Setting the HTTPS port to '8080' .....	50
13. RTSP .....	51
13.1. Description .....	51
13.2. Syntax .....	51
13.3. Parameters .....	51
13.4. Examples .....	51
13.4.1. Getting current RTSP settings .....	51
13.4.2. Setting the RTSP port to '1024' .....	52
14. SVN .....	53
14.1. Description .....	53
14.2. Syntax .....	53
14.3. Parameters .....	53
14.4. Examples .....	53
14.4.1. Getting current SVN settings .....	53
14.4.2. Setting the SVN port to '6002' .....	54
15. SVP .....	55
15.1. Description .....	55
15.2. Syntax .....	55
15.3. Parameters .....	55
15.4. Examples .....	55
15.4.1. Getting current SVP settings .....	55
15.4.2. Setting the SVP port to '555' .....	56
16. Bandwidth .....	57
16.1. Description .....	57
16.2. Syntax .....	57
16.3. Parameters .....	57
16.4. Examples .....	57
16.4.1. Getting the current bandwidth settings .....	57
16.4.2. Getting the bandwidth settings of the interface name 'Network 1' and 'Network 2' .....	58
16.4.3. Setting the bandwidth for the interface name 'Network3' .....	59

17. Multicast Setup .....	61
17.1. Description .....	61
17.2. Syntax .....	61
17.3. Parameters .....	61
17.4. Examples .....	61
17.4.1. Getting the current multicast settings .....	61
17.4.2. Setting the different ipaddress .....	62
18. Port Configuration .....	64
18.1. Description .....	64
18.2. Syntax .....	64
18.3. Parameters .....	64
18.4. Examples .....	65
18.4.1. Getting the port configuration information .....	65
19. Standby Device Setting .....	69
19.1. Description .....	69
19.2. Syntax .....	69
19.3. Parameters .....	69
19.4. Examples .....	69
20. MTS (mobile tracking system) settings .....	71
20.1. Description .....	71
20.2. Syntax .....	71
20.3. Parameters .....	71
20.4. Examples .....	72
20.4.1. Getting the MTS configuration information .....	72
20.4.2. Setting the MTS configuration information .....	73
21. WiFi settings .....	74
21.1. Description .....	74
21.2. Syntax .....	74
21.3. Parameters .....	74
21.4. Examples .....	75
21.4.1. Getting the WiFi configuration information .....	76
21.4.2. Scanning nearby WiFi device information .....	76
21.4.3. Connecting WiFi device .....	81
21.4.4. Enabling WiFi feature .....	83
22. DHCP clients .....	84
22.1. Description .....	84
22.2. Syntax .....	84
22.3. Parameters .....	84
22.4. Examples .....	84
22.4.1. Getting DHCP client list .....	84

23. RTSP over TLS settings .....	86
23.1. Description .....	86
23.2. Syntax .....	86
23.3. Parameters .....	86
23.4. Examples .....	86
23.4.1. Getting the RTSP over TLS configuration information .....	86
23.4.2. Setting the RTSP over TLS configuration information .....	87
24. EthStatus .....	89
24.1. Description .....	89
24.2. Syntax .....	89
24.3. Parameters .....	89
24.4. Examples .....	89
24.4.1. Getting the current inbound and outbound rates .....	89
25. PoeStatus .....	91
25.1. Description .....	91
25.2. Syntax .....	91
25.3. Parameters .....	91
25.4. Examples .....	92
25.4.1. Getting the current PoE ports' status (NVR) .....	92
25.4.2. Getting the current PoE ports' status (camera) .....	93
25.4.3. Setting the PoE port to activate (camera) .....	93
25.4.4. Setting the PoE port to activate (NVR) .....	94
26. DHCP Server .....	95
26.1. Description .....	95
26.2. Syntax .....	95
26.3. Parameters .....	95
26.4. Examples .....	96
26.4.1. Getting the current DHCP server settings .....	96
26.4.2. Setting DHCP server settings .....	98
27. ONVIF Discovery .....	99
27.1. Description .....	99
27.2. Syntax .....	99
27.3. Parameters .....	99
27.4. Examples .....	99
27.4.1. Getting the current onvif discovery settings .....	99
27.4.2. Enabling ONVIF discovery service .....	100
28. SIP Setup .....	101
28.1. Description .....	101
28.2. Syntax .....	101
28.3. Parameters .....	101

28.4. Examples	102
28.4.1. Getting the current sip setting values from cameras (this submenu only supports JSON responses)	102
28.4.2. Setting SIP configurations	103
29. SIP Account Settings	104
29.1. Description	104
29.2. Syntax	104
29.3. Parameters	104
29.4. Examples	105
29.4.1. Getting current SIP account settings (this submenu only supports JSON responses)	105
29.4.2. Adding a SIP account	107
29.4.3. Updating account settings	107
29.4.4. Remove SIP account	108
30. SIP Recipient Settings	109
30.1. Description	109
30.2. Syntax	109
30.3. Parameters	109
30.4. Examples	110
30.4.1. Getting SIP recipient settings (this submenu only supports JSON responses)	110
30.4.2. Adding a recipient by assigning them to groups 1, 2, or 3	111
30.4.3. Updating a recipient for Index 1	111
30.4.4. Selecting a target recipient and setting the Call request type to Single mode	112
30.4.5. Selecting a target recipient group and setting the Call request type to Multiple mode	112
30.4.6. Removing all recipients	113
31. SIP Call	114
31.1. Description	114
31.2. Syntax	114
31.3. Parameters	114
31.4. Examples	115
31.4.1. Getting the current SIP call state (this submenu only supports JSON responses)	115
31.4.2. Stopping a call request	115
32. NAT Traversal Settings	117
32.1. Description	117
32.2. Syntax	117
32.3. Parameters	117
32.4. Examples	118
32.4.1. Getting current NAT traversal settings (this submenu only supports JSON responses)	118
32.4.2. Changing the NAT traversal settings	118
33. P2P	120
33.1. Description	120



33.2. Syntax .....	120
33.3. Parameters .....	120
33.4. Examples .....	121
33.4.1. Retrieving the current P2P state .....	121
34. MQTT client settings .....	122
34.1. Description .....	122
34.2. Syntax .....	122
34.3. Parameters .....	122
34.4. Examples .....	124
34.4.1. Getting the MQTT client configuration information (this submenu supports only JSON response) .....	124
34.4.2. Setting the MQTT client configuration information .....	125

# Chapter 1. Overview

## 1.1. Description

This document explains network.cgi.

**network.cgi** manages the network settings of video surveillance products.

The following submenus of network.cgi are used to configure network settings:

- **interface**: Requests and configures the network interface settings.
- **dns**: Requests and configures the DNS server settings.
- **dynamicdns** : Requests and configures the DDNS server settings.
- **bonjour**: Requests and configures Bonjour discovery.
- **upnpdiscovery**: Requests and configures UPnP discovery.
- **zeroconf**: Requests and configures the zero configuration mechanism.
- **snmp** : Requests and configures the SNMP settings.
- **snmptrap**: Requests and configures the settings for SNMP trap messages.
- **qos**: Requests and configures the QoS (Quality of Service) settings
- **http**: Requests and configures the HTTP port number.
- **https**: Requests and configures the HTTPS port number.
- **rtsp**: Requests and configures the RTSP server settings.
- **svnp**: Requests and configures the SVN (Subversion) settings.
- **svp**: Requests and configures the SVP settings.
- **bandwidth**: Requests and configures the bandwidth.
- **multicastsetup**: Requests and configures multicast settings.
- **portconf**: Requests and configures all port-related settings in a device.
- **standbydeviceinfo**: Requests and configures the standby device IP address.
- **mts**: Requests and configures the MTS (mobile tracking system) configuration in NVR.
- **Wifi**: Requests and configures the WiFi configuration and WiFi scanning device.
- **dhcpcclients**: Requests and configures the DHCP client configuration.
- **poestatus**: Requests and configures the PoE ports' configuration that device supports.
- **dhcpserver**: Used to configure DHCP server on NVR device.
- **onvifdiscovery**: Requests and configures ONVIF Discovery-related settings.
- **sipsetup**: Requests and configures SIP general settings.
- **sipaccount**: Requests and configures SIP account settings.
- **siprecipients**: Requests and configures SIP recipient settings.

- **sipcall**: Requests and controls SIP call state.
- **nattraversal**: Requests and configures NAT traversal settings.
- **p2p**: Set and get the p2p activation state on the NVR device.
- **mqttclient**: Requests and configures the MQTT client settings.

# Chapter 2. Network Interface

## 2.1. Description

The **interface** submenu configures the network interface settings.

### Access level

Action	Camera	NVR	Encoder	Decoder
view	Admin	User	Admin	User
set	Admin	User	Admin	User

## 2.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?submenu=interface&action=<value>[&<parameter>=<value>...]
```

## 2.3. Parameters

Action	Parameter	Request/Response	Type/Value	Description
view				Reads the network interface settings
	InterfaceName	REQ	<csv>	Interface name
	MACAddress	RES	<string>	MAC address (read-only)
	LinkStatus	RES	<enum> Connected, Disconnect ed	Current link status (read-only)
	InterfaceType	RES	<enum> Ethernet, WiFi	Interface type (read-only)
	BroadcastAddress	RES	<string>	Broadcast address (read-only)
	IPv6DefaultAddress	RES	<string>	Default IPv6 address when IPv6 is enabled <div>CAMERA ONLY</div> <div>ENCODER ONLY</div>

Action	Parameter	Request/Response	Type/Value	Description
	InterfaceLabel	RES	<enum> All, Camera, Viewer, iSCSI, Wi-Fi	Interface type in NVR <b>NVR ONLY</b>
set	IPv4Type	REQ, RES	<enum> Manual, DHCP, PPPoE	IP Type <ul style="list-style-type: none"> <li>• Manual: Static IP. Manually set the IP, gateway, subnet mask and DNS.</li> <li>• DHCP: DHCP mode. Auto-configure the IP, gateway, and subnet mask.</li> <li>• PPPoE: PPPoE mode.</li> </ul>
	IPv4Address	REQ, RES	<string>	IPv4 address <b>IPv4Address</b> is valid only when <b>IPv4Type</b> is set to Manual.
	IPv4PrefixLength	REQ, RES	<int>	Prefix length <b>IPv4PrefixLength</b> is valid only when <b>IPv4Type</b> is set to Manual. <b>CAMERA ONLY</b> <b>ENCODER ONLY</b>
	IPv4Gateway	REQ, RES	<string>	Gateway IP address <b>IPv4Gateway</b> is valid only when <b>IPv4Type</b> is set to Manual.
	IsDefaultGateway	REQ, RES	<bool> True, False	Default gateway <b>NVR ONLY</b> <b>DECODER ONLY</b>
	HostName	REQ, RES	<string>	Host name If <b>IPv4Type</b> is not set to Manual, <b>HostName</b> will be read-only.
	IPv4SubnetMask	REQ, RES	<string>	IPv4 subnet mask. This parameter is valid only when <b>IPv4Type</b> is set to Manual.

Action	Parameter	Request/ Response	Type/ Value	Description
	PPPoEUserName	REQ, RES	<string>	User name  <b>PPPoEUserName</b> is valid only when <b>IPv4Type</b> is set to PPPoE.
	PPPoEPassword	REQ, RES	<string>	<small>Password</small> <b>PPPoEPassword</b> is valid only when <b>IPv4Type</b> is set to PPPoE.
	InterfaceName	REQ, RES	<string>	Interface name (read-only for network cameras)
	Enable	REQ, RES	<bool> True, False	Enable or Disable Network Interface  <div>NVR ONLY</div> <div>DECODER ONLY</div>
	IPv6Enable	REQ, RES	<bool> True, False	Whether to use IPv6
	IPv6Type	REQ, RES	<enum> Auto, Manual, Default	IPv6 type  <ul style="list-style-type: none"> <li>• Auto: IP address auto configuration.</li> <li>• Manual: Manual configuration.</li> <li>• Default: default IP configuration</li> </ul>
	IPv6Address	REQ, RES	<string>	IPv6 address  <b>IPv6Address</b> is valid only when <b>IPv6Type</b> is set to Manual.
	IPv6PrefixLength	REQ, RES	<int>	IPv6 prefix length  <b>IPv6PrefixLength</b> is valid only when <b>IPv6Type</b> is set to Manual.
	IPv6DefaultGateway	REQ, RES	<string>	IPv6 default gateway.  <b>IPv6DefaultGateway</b> is valid only when <b>IPv6Type</b> is set to Manual.
	IsPPPoEPasswordEncrypted	REQ	<bool> True, False	When this is set as true, password is encrypted using the public key obtained through the <b>rsa</b> submenu of security.cgi, and sent as payload content for the POST command.
	MTUSize	REQ, RES	<int>	Maximum Transmission Unit size of the network interface in bytes.  The range is from 1280 to 1500.

Action	Parameter	Request/ Response	Type/ Value	Description
	ICMPEnable	REQ, RES	<bool> True, False	Sets ICMP enable or not. <div>CAMERA ONLY</div>

## 2.4. Examples

### 2.4.1. Getting the system network interface settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?submenu=interface&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
InterfaceName=NetworkInterface1
MACAddress=00:09:18:6E:12:B0
HostName=PNM-A7083RVD-0009186E12B0
LinkStatus=Connected
InterfaceType=Ethernet
BroadcastAddress=192.168.75.255
IPv4Type=Manual
IPv4Address=192.168.75.196
IPv4PrefixLength=24
IPv4SubnetMask=255.255.255.0
IPv4Gateway=192.168.75.1
IPv6Enable=True
IPv6Type=Default
IPv6Address=fe80::209:18ff:fe6e:12b0
IPv6PrefixLength=64
IPv6DefaultGateway=
IPv6DefaultAddress=fe80::209:18ff:fe6e:12b0
MTUSize=1500
ICMPEnable=True
```

## JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "NetworkInterfaces": [
    {
      "InterfaceName": "NetworkInterface1",
      "MACAddress": "00:09:18:6E:12:B0",
      "HostName": "PNM-A7083RVD-0009186E12B0",
      "LinkStatus": "Connected",
      "InterfaceType": "Ethernet",
      "BroadcastAddress": "192.168.75.255",
      "IPv4Type": "Manual",
      "IPv4Address": "192.168.75.196",
      "IPv4PrefixLength": 24,
      "IPv4SubnetMask": "255.255.255.0",
      "IPv4Gateway": "192.168.75.1",
      "IPv6Enable": true,
      "IPv6Type": "Default",
      "IPv6Address": "fe80::209:18ff:fe6e:12b0",
      "IPv6PrefixLength": 64,
      "IPv6DefaultGateway": "",
      "IPv6DefaultAddress": "fe80::209:18ff:fe6e:12b0",
      "MTUSize": 1500,
      "ICMPEnable": true
    }
  ]
}
```

**The following sample response is for NVR.**

## TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```



```
HostName=PRN-4011
InterfaceName=Network1
MACAddress=00:09:15:05:06:26
LinkStatus=Connected
InterfaceType=Ethernet
BroadcastAddress=192.168.255.255
IPv4Type=Manual
IPv4Address=192.168.90.147
IPv4Gateway=192.168.90.1
IPv4SubnetMask=255.255.255.0
IsDefaultGateway=False
PPPoEUserName=
PPPoEPassword=
IPv6Enable=True
IPv6Type=Default
IPv6Address=fe80::209:15ff:fe05:626
IPv6PrefixLength=64
IPv6DefaultGateway=ff02::c
```

## JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "HostName": "PRN-4011",
  "NetworkInterfaces": [
    {
      "InterfaceName": "Network1",
      "MACAddress": "00:09:15:05:06:26",
      "LinkStatus": "Connected",
      "InterfaceType": "Ethernet",
      "BroadcastAddress": "192.168.255.255",
      "IPv4Type": "Manual",
      "IPv4Address": "192.168.90.147",
      "IPv4Gateway": "192.168.90.1",
      "IPv4SubnetMask": "255.255.255.0",
      "IsDefaultGateway": false,
      "PPPoEUserName": "",
```

```

        "PPPoEPassword": "",
        "IPv6Enable": true,
        "IPv6Type": "Default",
        "IPv6Address": "fe80::209:15ff:fe05:626",
        "IPv6PrefixLength": 64,
        "IPv6DefaultGateway": "ff02::c"
    }
]
}

```

### 2.4.2. Setting a static IP

Assigning a static IP manually.

To set **IPv4Type** to Manual, **IPv4Address**, **IPv4PrefixLength**, and **IPv4Gateway** must be set together.

#### REQUEST

```

http://<Device IP>/stw-
cgi/network.cgi?submenu=interface&action=set&IPv4Type=Manual&IPv4Address=19
2.168.0.101&IPv4PrefixLength=24&IPv4Gateway=192.168.0.1

```

The following request example is for NVR only.

#### REQUEST

```

http://<Device IP>/stw-
cgi/network.cgi?submenu=interface&action=set&InterfaceName=Network1&IPV4Typ
e=Manual&IPv4Address=192.168.0.101&IsDefaultGateway=true

```

### 2.4.3. Setting DHCP mode

If **IPv4Type** is set to DHCP, an available IP address on the same network is assigned to the Hanwha Vision video surveillance product.

#### REQUEST

```

http://<Device IP>/stw-
cgi/network.cgi?submenu=interface&action=set&IPv4Type=DHCP

```

The following request example is for NVR only.

#### REQUEST

```

http://<Device IP>/stw-

```

```
cgi/network.cgi?msubmenu=interface&action=set&InterfaceName=Network1&IPv4Type=DHCP
```

#### 2.4.4. Setting the PPPoEPassword with encrypted password

The PPPoEPassword can be configured as shown in the command below. PPPoEPassword should be encrypted with RSA and RSA\_PKCS1\_PADDING (Refer to security.cgi to obtain an RSA key). Base 64 encoded data should be sent as a POST message.

##### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=interface&action=set&IPv4Type=PPPoE&PPPoEUserName=User1&IsPPPoEPasswordEncrypted=True
```

# Chapter 3. DNS

## 3.1. Description

The **dns** submenu configures the DNS (Domain Name System) server settings.

### Access level

Action	Camera	NVR	Encoder	Decoder
view	Admin	User	Admin	User
set	Admin	User	Admin	User

## 3.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=dns&action=<value> [&<parameter>=<value>...]
```

## 3.3. Parameters

Action	Parameter	Request/Response	Type/Value	Description
view				Reads the DNS settings.
	InterfaceName	REQ	<string>	For filtering the response based on interface name
set	InterfaceName	REQ, RES	<string>	Interface name (read-only for network cameras)
	Type	REQ, RES	<enum> Manual, DHCP	Type
	IPType	REQ, RES	<enum> IPv4, IPv6	IP Type <div>NVR ONLY</div> <div>DECODER ONLY</div>
	PrimaryDNS	REQ, RES	<string>	Primary DNS server IP address  The address can be either IPv4 or IPv6.

Action	Parameter	Request/Response	Type/Value	Description
	SecondaryDNS	REQ, RES	<string>	Secondary DNS server IP address  The address can be either IPv4 or IPv6.  <div>CAMERA ONLY</div> <div>ENCODER ONLY</div>

## 3.4. Examples

### 3.4.1. Getting the current DNS settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?submenu=dns&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
InterfaceName=Network1
IPType=IPv4
Type=Manual
PrimaryDNS=168.126.63.1
IPType=IPv6
Type=DHCP
PrimaryDNS=
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "DNS": [
    {
```

```

    "InterfaceName": "Network1",
    "DNSByIPType": [
      {
        "IPType": "IPv4",
        "Type": "Manual",
        "PrimaryDNS": "168.126.63.1"
      },
      {
        "IPType": "IPv6",
        "Type": "DHCP",
        "PrimaryDNS": ""
      }
    ]
  }
]
}

```

### 3.4.2. Setting primary and secondary DNS server addresses

#### REQUEST

```

http://<Device IP>/stw-
cgi/network.cgi?submenu=dns&action=set&PrimaryDNS=10.1.20.12&SecondaryDNS=1
92.1.168.8

```

The following request example is for NVR only.

#### REQUEST

```

http://<Device IP>/stw-
cgi/network.cgi?submenu=dns&action=set&InterfaceName=Network1&PrimaryDNS=1.
1.1.2&Type=Manual

```

# Chapter 4. DDNS

## 4.1. Description

The **dynamicdns** submenu configures the DDNS (Dynamic DNS) server settings.

### Access level

Action	Camera	NVR	Encoder
view	Admin	User	Admin
set	Admin	User	Admin

## 4.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=dynamicdns&action=<value> [&<parameter>=<value>...]
```

## 4.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the DDNS settings.
	InterfaceName	REQ	<string>	For filtering the response based on interface name.
	SamsungServerName	RES	<string>	Samsung DDNS server URL (read-only)  <b>SamsungServerName</b> is valid only when <b>Type</b> is set to SamsungDDNS.
	Status	RES	<string>	DDNS status (read-only)

Action	Parameter	Request/Response	Type/Value	Description
	UpnpStatus	RES	<string> Success", "Fail[Need Initialization]", "Fail[Invalid Configuration]", "Fail[Router UPnP Disabled]", "Fail[Router Not Found]", "Fail[Need To Restart Router]", "Fail[UPnP Port clash]", "Fail[Quick Connect Not Supported]" "Fail[Unknown Error]"	UpnP Status (read-only) <b>Note</b> Success : Quick Connect Success  Fail[Need Initialization]: Connection Failed.  Fail[Invalid Configuration]: Invalid Network Configuration  Fail[Router UPnP Disabled] : UpnP disable state
	UpnpPortClashInfo	RES	<csv>	ClashPort List
set	InterfaceName	REQ, RES	<string>	Interface name (read-only for network cameras)
	Type	REQ, RES	<enum> Off, SamsungDDNS, PublicDDNS	Dynamic DNS type  <b>Note</b> The <b>Type</b> parameter must be sent together with the <b>set</b> action.
	SamsungProductID	REQ, RES	<string>	Product ID for Samsung DDNS  <b>SamsungProductID</b> is valid only when <b>Type</b> is set to SamsungDDNS.



Action	Parameter	Request/ Response	Type/ Value	Description
	SamsungQuickConnect	REQ, RES	<bool> True, False	Enables or disables Samsung DDNS Quick Connect  <b>SamsungQuickConnect</b> is valid only when <b>Type</b> is set to SamsungDDNS.
	PublicServiceEntry	REQ, RES	<enum> www.dyndns.org, www.no-ip.org	Public DDNS Service  <b>PublicServiceEntry</b> is valid only when <b>Type</b> is set to PublicDDNS.
	PublicHostName	REQ, RES	<string>	Public DDNS Hostname  <b>PublicHostName</b> is valid only when <b>Type</b> is set to PublicDDNS.
	PublicUserName	REQ, RES	<string>	Public DDNS user ID  <b>PublicUserName</b> is valid only when <b>Type</b> is set to PublicDDNS.
	PublicPassword	REQ, RES	<string>	Public DDNS user password  <b>PublicPassword</b> is valid only when <b>Type</b> is set to PublicDDNS.
	IsPublicPasswordEncrypted	REQ,	<bool> True, False	When this is set to true, password is encrypted using the public key obtained using the <b>rsa</b> submenu of security.cgi, and sent as payload content for the POST command.

## 4.4. Examples

### 4.4.1. Getting the current DDNS configuration settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=dynamicdns&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
InterfaceName=1a5a97d2-464a-4222-91c6-140ff36b82b6
Status=Trying
Type=SamsungDDNS
SamsungServerName=www.samsungipolis.com
SamsungProductID=snb6004gt
SamsungQuickConnect=False
PublicServiceEntry=www.dyndns.org
PublicHostName=host
PublicUserName=user
PublicPassword=pw
UpnpStatus=Fail[Router Not Found]
UpnpPortClashInfo=1024,1025,1026,1027,1028,1029,1030,1031,1032,1033
```

### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "DynamicDNS": [
    {
      "InterfaceName": "1a5a97d2-464a-4222-91c6-140ff36b82b6",
      "Status": "Trying",
      "Type": "SamsungDDNS",
      "SamsungServerName": "www.samsungipolis.com",
      "SamsungProductID": "snb6004gt",
      "SamsungQuickConnect": false,
      "PublicServiceEntry": "www.dyndns.org",
      "PublicHostName": "host",
      "PublicUserName": "user",
      "PublicPassword": "pw",
      "UpnpStatus": "Fail[Router Not Found]",
      "UpnpPortClashInfo": [1024, 1025, 1026, 1027, 1028, 1029,
1030, 1031, 1032, 1033]
    }
  ]
}
```

### 4.4.2. Setting the DDNS type

The accompanying parameters differ depending on the **Type**.

Using the Samsung DDNS service

#### REQUEST

```
http://<DeviceIP>/stw-  
cgi/network.cgi?submenu=dynamicdns&action=set&Type=SamsungDDNS&SamsungProdu  
ctID=PRODUCTID&SamsungQuickConnect=True
```

The following request example is for NVR only.

#### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=dynamicdns&action=set&InterfaceName=Network1&Type=S  
amsungDDNS&SamsungProductId=1111&SamsungQuickConnect=True
```

Using the public DDNS service

#### REQUEST

```
http://<DeviceIP>/stw-  
cgi/network.cgi?submenu=dynamicdns&action=set&Type=PublicDDNS&PublicService  
Entry=www.no-  
ip.org&PublicHostName=host_name&PublicUserName=user_name&PublicPassword=pass  
word
```

The following request example is for NVR only.

#### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi&submenu=dynamicdns&action=set&InterfaceName=Network1&Type=P  
ublicDDNS&PublicServiceEntry=www.changeip.com&PublicHostname=hostname&Public  
Username=user&PublicPassword=password
```

### 4.4.3. Setting the PublicPassword with encrypted password

The PublicPassword can be configured as shown in the command below. PublicPassword should be encrypted with RSA and RSA\_PKCS1\_PADDING (Refer to security.cgi to obtain an RSA key). Base 64 Encoded data should be sent as a POST message.

## REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=dynamicdns&action=set&Type=PublicDDNS&PublicService  
Entry=www.no-  
ip.org&PublicHostName=host_name&PublicUserName=user_name&IsPublicPasswordEnc  
rypted=True
```

# Chapter 5. Bonjour

## 5.1. Description

The **bonjour** submenu enables or disables the Bonjour discovery mechanism and sets the name for it.

The network camera that is connected to the local area network can be automatically searched on the PC with Bonjour service.

**NOTE** | This chapter applies to network cameras and encoders only.

### Access level

Action	Camera	Encoder
view	Admin	Admin
set	Admin	Admin

## 5.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=bonjour&action=<value> [&<parameter>=<value>]
```

## 5.3. Parameters

Action	Parameter	Request Response	Type/ Value	Description
view				Reads the Bonjour settings.
set	Enable	REQ, RES	<bool> True, False	Whether to use Bonjour.  <b>Note</b> <b>Enable</b> must be sent together with the <b>set</b> action.
	FriendlyName	REQ, RES	<string>	Device name or short description

## 5.4. Examples

### 5.4.1. Getting Bonjour settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=bonjour&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Enable=True
FriendlyName=SAMSUNG-SNB-6004-000918309926
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Enable": true,
  "FriendlyName": "SAMSUNG-SNB-6004-000918309926"
}
```

### 5.4.2. Enabling Bonjour and setting the friendly name as 'SNB6004'

#### REQUEST

```
http://<Device IP>/stw-
cgi/network.cgi?submenu=bonjour&action=set&Enable=True&FriendlyName=SNB6004
```

# Chapter 6. UPnP Discovery

## 6.1. Description

The **upnpdiscovery** submenu enables or disables UPnP discovery and sets the name for it.

**NOTE** | This chapter applies to network cameras and encoders only.

### Access level

Action	Camera	Encoder
view	Admin	Admin
set	Admin	Admin

## 6.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=upnpdiscovery&action=<value> [&<parameter>=<value>]
```

## 6.3. Parameters

Action	Parameter	Request/Response	Type/Value	Description
view				Reads the UPnP discovery settings.
set	InterfaceName	REQ, RES	<string>	Interface name (read-only for network cameras)
	Enable	REQ, RES	<bool> True, False	Enables or disables UPnP discovery.  If this function is enabled, cameras can be automatically searched in the client and operating system that supports the UPnP protocol.  <b>Note</b> <b>Enable</b> must be sent together with the <b>set</b> action.
	FriendlyName	REQ, RES	<string>	Device name or short description

## 6.4. Examples

### 6.4.1. Getting UPnP discovery settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=upnpdiscovery&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
InterfaceName=1a5a97d2-464a-4222-91c6-140ff36b82b6
Enable=True
FriendlyName=SAMSUNG-SNB-6004-000918309926
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "UpnpDiscovery": [
    {
      "InterfaceName": "1a5a97d2-464a-4222-91c6-140ff36b82b6",
      "Enable": true,
      "FriendlyName": "SAMSUNG-SNB-6004-000918309926"
    }
  ]
}
```

### 6.4.2. Enabling UPnP discovery and setting the friendly name as 'SNB6004'

#### REQUEST

```
http://<Device IP>/stw-
cgi/network.cgi?msubmenu=upnpdiscovery&action=set&Enable=True&FriendlyName=S
NB6004
```



# Chapter 7. Zeroconf

## 7.1. Description

The **zeroconf** submenu enables or disables Zeroconf (Zero configuration networking), which supports automatic configuration of link-local IP addresses.

### NOTE

This chapter applies to network cameras and encoders only.

### Access level

Action	Camera	Encoder
view	Admin	Admin
set	Admin	Admin

## 7.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=zeroconf&action=<value> [&<parameter>=<value>]
```

## 7.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the zero configuration settings.
	IPAddress	RES	<string>	IP address assigned by the Zeroconf protocol (read-only)  IPv4 format is allowed.
	SubnetMask	RES	<string>	Subnet mask assigned by the Zeroconf protocol (read-only)  IPv4 format is allowed.
set	InterfaceName	REQ, RES	<string>	Interface name (read-only for network cameras)
	Enable	REQ, RES	<bool> True, False	Enables or disables Zeroconf.  <b>Note</b> <b>Enable</b> must be sent together with the <b>set</b> action.

## 7.4. Examples

### 7.4.1. Getting current Zeroconf settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=zeroconf&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
InterfaceName=1a5a97d2-464a-4222-91c6-140ff36b82b6
IPAddress=169.254.4.236
SubnetMask=16
Enable=True
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "ZeroConf": [
    {
      "InterfaceName": "1a5a97d2-464a-4222-91c6-140ff36b82b6",
      "IPAddress": "169.254.4.236",
      "SubnetMask": "16",
      "Enable": true
    }
  ]
}
```

### 7.4.2. Enabling Zeroconf

#### REQUEST

```
http://<Device IP>/stw-
```

```
cgi/network.cgi?msubmenu=zeroconf&action=set&Enable=True
```

# Chapter 8. SNMP Setup

## 8.1. Description

The **snmp** submenu configures the SNMP (Simple Network Management Protocol) settings.

### Access level

Action	Camera	NVR
view	Admin	User
set	Admin	User

## 8.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=snmp&action=<value> [&<parameter>=<value>...]
```

## 8.3. Parameters

Action	Parameter	Request/Response	Type/Value	Description
view				Reads the SNMP settings.
set	Version1	REQ, RES	<bool> True, False	Enables or disables SNMP version 1.
	Version2	REQ, RES	<bool> True, False	Enables or disables SNMP version 2.
	Version3	REQ, RES	<bool> True, False	Enables or disables SNMP version 3.
	ReadCommunity	REQ, RES	<string>	SNMP read community name  <b>ReadCommunity</b> is valid only when <b>Version2</b> is set to True.
	WriteCommunity	REQ, RES	<string>	SNMP write community name  <b>WriteCommunity</b> is valid only when <b>Version2</b> is set to True.
	UserPassword	REQ, RES	<string>	User password  <b>UserPassword</b> is valid only when <b>Version3</b> is set to True.

Action	Parameter	Request/Response	Type/Value	Description
	IsPasswordEncrypted	REQ	<bool> True, False	When this is set to true, password is encrypted using the public key obtained through the <b>rsa</b> submenu of security.cgi, and sent as payload content for the POST command.

## 8.4. Examples

### 8.4.1. Getting current SNMP settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=snmp&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Version1=False
Version2=True
Version3=False
ReadCommunity=public
WriteCommunity=write
UserPassword=admin4321
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json_
<Body>
```

```
{
  "Version1": false,
  "Version2": true,
  "Version3": false,
  "ReadCommunity": "public",
  "WriteCommunity": "write",
```

```
"UserPassword": "admin4321"  
}
```

### 8.4.2. Setting SNMP

**ReadCommunity** and **WriteCommunity** values can be set only when **Version2** is set to True, as in the example below.

#### REQUEST

```
http://<DeviceIP>/stw-  
cgi/network.cgi?submenu=snmp&action=set&Version2=True&ReadCommunity=public&  
WriteCommunity=write
```

### 8.4.3. Setting the Password with encrypted password

The Password can be configured as shown in the command below. Password should be encrypted with RSA and RSA\_PKCS1\_PADDING (Refer to security.cgi to obtain an RSA key). Base 64 Encoded data should be sent as a POST message.

#### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=snmp&action=set&Version1=False&Version2=False&Versi  
on3=True& IsPasswordEncrypted=True
```

# Chapter 9. SNMP Trap Settings

## 9.1. Description

The **snmptrap** submenu configures the trap settings of SNMP (Simple Network Management Protocol).

### Access level

Action	Camera	NVR	Encoder	Decoder
view	Admin	User	Admin	User
set	Admin	User	Admin	User

## 9.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=snmptrap&action=<value>[&<parameter>=<value>...]
```

## 9.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the SNMP trap settings.
set	Enable	REQ, RES	<bool> True, False	Enables or disables trap message transmission.
	Trap.#.Address	REQ, RES	<string>	IP address  IPv4 format is allowed.  <b>Trap.#.Address</b> is valid only when <b>Enable</b> is set to True.
	Trap.#.Community	REQ, RES	<string>	SNMP community  <b>Trap.#.Community</b> is valid only when <b>Enable</b> is set to True.  <div>CAMERA ONLY</div> <div>ENCODER ONLY</div>
	Trap.#.AuthenticationFailure	REQ, RES	<bool> True, False	Enables or disables trap notification in the event of an authentication failure (read-only for NVR)
	Trap.#.ColdStart	REQ, RES	<bool> True, False	Enables or disables ColdStart.

Action	Parameter	Request/Response	Type/Value	Description
	Trap.#.LinkUp	REQ, RES	<bool> True, False	Enables or disables linkUp trap notification. (read-only for NVR)
	Trap.#.LinkDown	REQ, RES	<bool> True, False	Enables or disables linkDown trap notification. (read-only for NVR)  NVR ONLY DECODER ONLY
	Trap.#.WarmStart	REQ, RES	<bool> True, False	Enables or disables WarmStart. (read-only for NVR)  NVR ONLY DECODER ONLY
	Trap.#.AlarmInput.#	REQ, RES	<bool> True, False	Enables or disables alarm inputs  CAMERA ONLY ENCODER ONLY
	Trap.#.AlarmOutput.#	REQ, RES	<bool> True, False	Enables or disables alarm outputs  CAMERA ONLY ENCODER ONLY
	Trap.#.TamperingDetection	REQ, RES	<bool> True, False	Enables or disables tampering detection  CAMERA ONLY ENCODER ONLY
	Trap.#.UseCommunity	REQ, RES	<bool> True, False	Enables or disables Community. To disable Community feature, user can set this parameter as False. Then Community field would be erased. If user set Community field, this parameter would be set as True automatically.  CAMERA ONLY

#### NOTE

# represents the index number of the trap. Available trap numbers may vary depending on the device. Please check the device attributes using attributes.cgi.

## 9.4. Examples



### 9.4.1. Getting current SNMP Trap settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=snmptrap&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
Enable=False  
Trap.0.Address=0.0.0.0  
Trap.0.AuthenticationFailure=False  
Trap.0.ColdStart=False  
Trap.0.WarmStart=False  
Trap.0.LinkUp=False  
Trap.0.LinkDown=False  
Trap.0.UseCommunity=False
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "Enable": false,  
  "SNMPTrap": [  
    {  
      "Trap": 0,  
      "Address": "0.0.0.0",  
      "AuthenticationFailure": false,  
      "ColdStart": false,  
      "WarmStart": false,  
      "LinkUp": false,  
      "LinkDown": false,  
      "UseCommunity": false  
    }  
  ]  
}
```

```
}
```

### 9.4.2. Setting an SNMP trap

#### REQUEST

```
http://<DeviceIP>/stw-  
cgi/network.cgi?submenu=snmptrap&action=set&Enable=True&Trap.0.Address=127.  
0.0.1&Trap.0.Community=community&Trap.0.AuthenticationFailure=False&Trap.0.L  
inkUp=False
```

The following request example is for NVR only.

#### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=snmptrap&action=set&Enable=True&Trap.0.Address=127.  
0.0.1
```

### 9.4.3. Setting community not to use

#### REQUEST

```
http://<DeviceIP>/stw-  
cgi/network.cgi?submenu=snmptrap&action=set&Trap.0.UseCommunity=False
```

The following request example is for NVR only.

# Chapter 10. QoS Setup

## 10.1. Description

The **qos** submenu configures the QoS (Quality of Service) settings. It specifies the priority to secure a stable transfer rate for a specific IP.

### NOTE

This chapter applies to network cameras and encoder only.

Attribute to check for maximum IPV4QoS Addresses:

"attributes/Network/Limit/**MaxIPv4QoS**"

Attribute to check for maximum IPV6QoS Addresses:

"attributes/Network/Limit/**MaxIPv6QoS**"

### Access level

Action	Camera	Encoder
view	Admin	Admin
add, update	Admin	Admin
remove	Admin	Admin

## 10.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=qos&action=<value> [&<parameter>=<value> ...]
```

## 10.3. Parameters

Acton	Parameter	Request/ Response	Type/ Value	Description
view				Reads the QoS settings.
add, update	Enable	REQ, RES	<bool> True, False	Enables or disables QoS for registered IPv4/IPv6 addresses.  <b>Note</b> <b>IPType</b> , <b>IP Address</b> and <b>Enable</b> must be sent together with the <b>add</b> action.
	Index	REQ, RES	<int>	Index of the registered QoS setting  <b>Note</b> <b>IPType</b> and <b>Index</b> must be sent together with the <b>update</b> action.

Acton	Parameter	Request/ Response	Type/ Value	Description
	IPType	REQ, RES	<enum> IPv4, IPv6	QoS IP type
	IPAddress	REQ, RES	<string>	QoS IP address (IPv4 or IPv6 address)
	PrefixLength	REQ, RES	<int>	Prefix length for IPv4 and IPv6
	DSCP	REQ, RES	<int>	Priority of QoS DSCP (Differentiated Services Code Point)
remove	Index	REQ	<int>	Index that is to be deleted  <b>Note</b> <b>IPType</b> and <b>Index</b> must be sent together with the <b>remove</b> action.
	IPType	REQ	<enum> IPv4, IPv6	IP type that is to be deleted  <b>Note</b> <b>IPType</b> and <b>Index</b> must be sent together with the <b>remove</b> action.

## 10.4. Examples

### 10.4.1. Getting current QoS settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?submenu=qos&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
qos.Index.IPType=Enable/IPAddress/PrefixLength/DSCP
qos.1.IPv4=True/192.168.75.137/32/63
qos.2.IPv4=True/192.168.75.135/32/63
qos.1.IPv6=True/2001:1:1:1:1:1:1:1/128/63
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
```

Content-type: application/json

<Body>

```
{
  "QoS": [
    {
      "IPType": "IPv4",
      "IPList": [
        {
          "Index": 1,
          "IPAddress": "192.168.75.137",
          "PrefixLength": 32,
          "Enable": true,
          "DSCP": 63
        },
        {
          "Index": 2,
          "IPAddress": "192.168.75.135",
          "PrefixLength": 32,
          "Enable": true,
          "DSCP": 63
        }
      ]
    },
    {
      "IPType": "IPv6",
      "IPList": [
        {
          "Index": 1,
          "IPAddress": "2001:1:1:1:1:1:1:1",
          "PrefixLength": 128,
          "Enable": true,
          "DSCP": 63
        }
      ]
    }
  ]
}
```

### 10.4.2. Adding a QoS-enabled IPv4 Address

To add a QoS IP with the **add** action, the **IPType**, **IP Address**, and **Enable** parameters must be set.

#### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=qos&action=add&IPType=IPv4&IPAddress=10.10.10.10&DS  
CP=1&Enable=True
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
OK  
Index=1
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "Response": "Success"  
}
```

### 10.4.3. Updating a QoS-enabled IPv4 Address

To edit the QoS IP/DSCP settings with the **update** action, the **IPType** and **Index** parameters must be set.

#### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=qos&action=update&Index=1&Enable=False&IPType=IPv4&  
IPAddress=10.10.10.10&PrefixLength=5
```

### 10.4.4. Deleting a QoS IPv6 Address

To remove the QoS IP/DSCP settings with the **remove** action, the **IPType** and **Index** parameters must be

set.

#### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=qos&action=remove&Index=2&IPType=IPv6
```

# Chapter 11. HTTP Port

## 11.1. Description

The **http** submenu configures the HTTP port number.

### Access level

Action	Camera	NVR	Encoder	Decoder
view	Admin	User	Admin	User
set	Admin	User	Admin	User

## 11.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=http&action=<value> [&<parameter>=<value>...]
```

## 11.3. Parameters

Action	Parameter	Request/Response	Type/Value	Description
view				Reads the HTTP port settings
set	Port	REQ, RES	<int>	HTTP port number

**Note**  
**Port** must be sent together with the **set** action.

## 11.4. Getting the HTTP port number

### 11.4.1. Getting the current HTTP port number

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=http&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```



```
Port=80
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "Port": 80  
}
```

## 11.5. Setting the HTTP port

### 11.5.1. Setting the HTTP port to '8080'

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=http&action=set&Port=8080
```

# Chapter 12. HTTPS Port

## 12.1. Description

The **https** submenu configures the HTTPS (Secure Hypertext Transfer Protocol) port number.

### Access level

Action	Camera	NVR	Encoder	Decoder
view	Admin	User	Admin	User
set	Admin	(Not supported)	Admin	(Not supported)

## 12.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=https&action=<value> [&<parameter>=<value>...]
```

## 12.3. Parameters

Action	Parameter	Request/Response	Type/Value	Description
view				Reads the HTTPS port settings.
set	Port	REQ, RES	<int>	HTTPS port number (read-only for NVR)  <b>Note</b> <b>Port</b> must be sent together with the <b>set</b> action.

## 12.4. Examples

### 12.4.1. Getting the current HTTPS port number

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=https&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain
```

```
<Body>
```

```
Port=443
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "Port": 443  
}
```

### 12.4.2. Setting the HTTPS port to '8080'

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=https&action=set&Port=8080
```

# Chapter 13. RTSP

## 13.1. Description

The **rtsp** submenu configures the RTSP (Real Time Streaming Protocol) server settings.

### Access level

Action	Camera	NVR	Encoder
view	Admin	User	Admin
set	Admin	User	Admin

## 13.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=rtsp&action=<value> [&<parameter>=<value>...]
```

## 13.3. Parameters

Action	Parameter	Request/Response	Type/Value	Description
view				Reads the RTSP settings.
	MobilePort	RES	<int>	Mobile port <b>NVR ONLY</b>
set	Port	REQ, RES	<int>	RTSP port number
	Timeout	REQ, RES	<enum> 0s, 60s	RTSP timeout (seconds) <b>CAMERA ONLY</b> <b>ENCODER ONLY</b>
	ProfileSessionPolicy	REQ, RES	<enum> Continue, Disconnect	Keep connection when profile setting is changed <b>CAMERA ONLY</b> <b>ENCODER ONLY</b>

## 13.4. Examples

### 13.4.1. Getting current RTSP settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=rtsp&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
Port=554  
Timeout=60s  
ProfileSessionPolicy=Continue
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "Port": 554,  
  "Timeout": "60s",  
  "ProfileSessionPolicy": "Continue"  
}
```

### 13.4.2. Setting the RTSP port to '1024'

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=rtsp&action=set&Port=1024
```

# Chapter 14. SVN

## 14.1. Description

The **svn** submenu configures the SVN (Samsung Video Network Protocol) settings. SVN is Samsung's own video network protocol for integrating Hanwha network cameras with 3<sup>rd</sup> party applications.

**NOTE** | This chapter applies to network cameras only.

### Access level

Action	Camera
view	Admin
set	Admin

## 14.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=svn&action=<value> [&<parameter>=<value>...]
```

## 14.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the SVN settings.
set	Port	REQ, RES	<int>	SVN port number

## 14.4. Examples

### 14.4.1. Getting current SVN settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?submenu=svn&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
Port=4520
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json_  
<Body>
```

```
{  
  "Port": 4520  
}
```

#### 14.4.2. Setting the SVN port to '6002'

##### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=svnp&action=set&Port=6002
```

# Chapter 15. SVP

## 15.1. Description

The **svp** submenu configures the SVP (Smart Viewer Protocol) port number.

**NOTE** | This chapter applies to NVR only.

### Access level

Action	NVR
view	User
set	User

## 15.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=svp&action=<value> [&<parameter>=<value> ...]
```

## 15.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the SVP (Smart Viewer Protocol) settings
set	Port	REQ, RES	<int>	SVP port number

## 15.4. Examples

### 15.4.1. Getting current SVP settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=svp&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```



Port=554

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Port": 554
}
```

#### 15.4.2. Setting the SVP port to '555'

##### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=svp&action=set&Port=555
```

# Chapter 16. Bandwidth

## 16.1. Description

The **bandwidth** submenu configures the bandwidth.

### NOTE

This chapter applies to NVR only.

### Access level

Action	NVR
view	User
set	User

## 16.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=bandwidth&action=<value>[&<parameter>=<value>...]
```

## 16.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the bandwidth settings
	InterfaceName	REQ	<csv>	Interface name
set	InterfaceName	REQ, RES	<string>	Interface name
	Bandwidth	REQ, RES	<int>	Bandwidth

## 16.4. Examples

### 16.4.1. Getting the current bandwidth settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=bandwidth&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
InterfaceName=Network1
Bandwidth=400
InterfaceName=Network2
Bandwidth=0
InterfaceName=Network3
Bandwidth=0
InterfaceName=Network4
Bandwidth=0
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "NetworkBandwidths": [
    {
      "InterfaceName": "Network1",
      "Bandwidth": 400
    },
    {
      "InterfaceName": "Network2",
      "Bandwidth": 0
    },
    {
      "InterfaceName": "Network3",
      "Bandwidth": 0
    },
    {
      "InterfaceName": "Network4",
      "Bandwidth": 0
    }
  ]
}
```

### 16.4.2. Getting the bandwidth settings of the interface name 'Network 1' and 'Network 2'

## REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=bandwidth&action=view&InterfaceName=Network1,Networ  
k2
```

## TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
InterfaceName=Network1  
Bandwidth=400  
InterfaceName=Network2  
Bandwidth=0
```

## JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json_  
<Body>
```

```
{  
  "NetworkBandwidths": [  
    {  
      "InterfaceName": "Network1",  
      "Bandwidth": 400  
    },  
    {  
      "InterfaceName": "Network2",  
      "Bandwidth": 0  
    }  
  ]  
}
```

### 16.4.3. Setting the bandwidth for the interface name 'Network3'

## REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=bandwidth&action=set&InterfaceName=Network3&Bandwid  
th=200
```

# Chapter 17. Multicast Setup

## 17.1. Description

The **multicastsetup** submenu can be used to configure the multicast configuration of a streaming server. Using this interface client, the multicast IP, port and TTL values can be set.

**NOTE** | This chapter applies to NVR only.

### Access level

Action	NVR
view	User
set	Admin

## 17.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=multicastsetup&action=<value>[&<parameter>=<value>.  
..]
```

## 17.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the multicast settings
	PortRange	REQ, RES	<enum>	Port range
set	TTL	REQ, RES	<int>	TTL setting in seconds
	IPAddress	REQ, RES	<string>	Multicast IP address used for streaming
	PortStart	REQ, RES	<int>	Starting this port number any port would be used for multicast streaming

## 17.4. Examples

### 17.4.1. Getting the current multicast settings

The following command gets the current multicast settings.

## REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=multicastsetup&action=view
```

## TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
PortRange=8000~8159  
TTL=5  
IPAddress=224.126.63.1
```

## JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "PortRange": "8000~8159",  
  "TTL": 5,  
  "IPAddress": "224.126.63.1"  
}
```

## 17.4.2. Setting the different ipaddress

The following command sets a new multicast address.

## REQUEST

```
http://<Device IP>/ stw-  
cgi/network.cgi?msubmenu=multicastsetup&action=set&IPAddress=224.126.64.122
```

## TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

OK

#### JSON RESPONSE

HTTP/1.0 200 OK  
Content-type: application/json\_  
<Body>

```
{  
  "Response": "Success"  
}
```



# Chapter 18. Port Configuration

## 18.1. Description

The **portconf** submenu can be used to view and set the port settings of the device.

### Access level

Action	Camera	NVR	Encoder	Decoder
view	User	User	User	User
set	Admin	Admin	Admin	Admin

## 18.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=portconf&action=<value>[&<parameter>=<value>...]
```

## 18.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
view	FixedPorts	RES	<csv>	List of fixed ports used by the device
	UsedPorts	RES	<csv>	List of ports used by the device
	RouterUsedPorts	RES	<csv>	Provides a list of ports used by the router when using UPNP
set	ProtocolType	REQ, RES	<enum> TCP, UDP- UNICAST, UDP- MULTICAST	Transmission mode used by SVP server <div>NVR ONLY</div> <div>DECODER ONLY</div>
	HTTPPort	REQ, RES	<int>	HTTP port used by the device
	HTTPSPort	REQ, RES	<int>	HTTPs port used in the device.
	RTSPPort	REQ, RES	<int>	RTSP server port
	DevicePort	REQ, RES	<int>	Used to configure SVNp port for camera and SVP port for NVR
	RTSPTimeout	REQ, RES	<int>	RTSP timeout setting, in seconds <div>CAMERA ONLY</div> <div>ENCODER ONLY</div>

## 18.4. Examples

### 18.4.1. Getting the port configuration information

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?submenu=portconf&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
FixedPorts=22,25,110,111,123,443,3260,3702,4520,9923
UsedPorts=80,554,555,556,557,558,7001,7002,7003,7004,7005,7006,7007,7008,700
9,7010,7011,7012,7013,7014,7015,7016,7017,7018,7019,7020,7021,7022,7023,7024
,7025,7026,7027,7028,7029,7030,7031,7032,7033,7034,7035,7036,7037,7038,7039,
7040,7041,7042,7043,7044,7045,7046,7047,7048,7049,7050,7051,7052,7053,7054,7
055,7056,7057,7058,7059,7060,7061,7062,7063,7064,10001,10002,10003,25000
RouterUsedPorts=22,25,110,111,123,443,3260,3702,4520,9923
ProtocolType=UDP-UNICAST
HTTPPort=80
HTTPSPort=443
RTSPPort=558
DevicePort=554
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "FixedPorts": [
    22,
    25,
    110,
    111,
    123,
    443,
```

```
    3260,  
    3702,  
    4520,  
    9923  
],  
"UsedPorts": [  
    80,  
    554,  
    555,  
    556,  
    557,  
    558,  
    7001,  
    7002,  
    7003,  
    7004,  
    7005,  
    7006,  
    7007,  
    7008,  
    7009,  
    7010,  
    7011,  
    7012,  
    7013,  
    7014,  
    7015,  
    7016,  
    7017,  
    7018,  
    7019,  
    7020,  
    7021,  
    7022,  
    7023,  
    7024,  
    7025,  
    7026,  
    7027,  
    7028,  
    7029,
```

```
7030,  
7031,  
7032,  
7033,  
7034,  
7035,  
7036,  
7037,  
7038,  
7039,  
7040,  
7041,  
7042,  
7043,  
7044,  
7045,  
7046,  
7047,  
7048,  
7049,  
7050,  
7051,  
7052,  
7053,  
7054,  
7055,  
7056,  
7057,  
7058,  
7059,  
7060,  
7061,  
7062,  
7063,  
7064,  
10001,  
10002,  
10003,  
25000  
],  
"RouterUsedPorts":  [
```

```
    22,  
    25,  
    110,  
    111,  
    123,  
    443,  
    3260,  
    3702,  
    4520,  
    9923  
],  
"ProtocolType": "UDP-UNICAST",  
"HTTPPort": 80,  
"HTTPSPort": 443,  
"RTSPPort": 558,  
"DevicePort": 554  
}
```

# Chapter 19. Standby Device Setting

## 19.1. Description

The **standbydeviceinfo** submenu is used to set the standby device IP address.

### NOTE

This chapter applies to NVR only.

### Access level

Action	NVR
view	User
set	Admin

## 19.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=standbydeviceinfo&action=<value>[&<parameter>=<valu  
e>...]
```

## 19.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				Reads the standby device IP address
set	IPAddress	REQ, RES	<string>	Sets the standby device IP address

## 19.4. Examples

### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=standbydeviceinfo&action=set&IPAddress=192.168.90.1  
22
```

### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

OK

#### JSON RESPONSE

HTTP/1.0 200 OK  
Content-type: application/json\_  
<Body>

```
{  
  "Response": "Success"  
}
```

# Chapter 20. MTS (mobile tracking system) settings

## 20.1. Description

The **mts** submenu is used to configure the MTS (mobile tracking system) in NVR

**NOTE** | This chapter applies to NVR only.

### Access level

Action	NVR
view	User
set	User

## 20.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=mts&action=<value> [&<parameter>=<value>...]
```

## 20.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view	BaseURL	RES	<string>	URL to access the server
set	Enable	REQ, RES	<bool> True, False	Enable MTS
	Version	REQ, RES	<string>	MTS version
	Periodicity	REQ, RES	<enum>	Periodicity of updating the GPS location
	EnableEventNotification	REQ, RES	<bool> True, False	Enable Event Notification to server.
	EnableJPEGPUSH	REQ, RES	<bool> True, False	Enable JPEG snapshot transmission.
	IPType	REQ, RES	<enum>	Server IP type
	IPAddress	REQ, RES	<string>	Server IP address.
	Port	REQ, RES	<int>	Server port.



## 20.4. Examples

### 20.4.1. Getting the MTS configuration information

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=mts&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Enable=False
Version=v1
Periodicity=5s
EnableEventNotification=False
EnableJPEGPush=False
IPType=IPv4
IPAddress=129.123.123.12
Port=3000
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Enable": false,
  "Version": "v1",
  "Periodicity": "5s",
  "EnableEventNotification": false,
  "EnableJPEGPush": false,
  "IPType": "IPv4",
  "IPAddress": "129.123.123.12",
  "Port": 3000
}
```

## 20.4.2. Setting the MTS configuration information

### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=mts&action=set&Enable=True&Version=V2&Periodicity=1  
0s&EnableEventNotification=True&EnableJPEGPush=True&IPType=IPv4&IPAddress=25  
5.255.255.255&Port=2555
```

### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
OK
```

### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json_  
<Body>
```

```
{  
  "Response": "Success"  
}
```

# Chapter 21. WiFi settings

## 21.1. Description

The **wifi** submenu is used to set and get WiFi connection configuration

**NOTE** | This chapter applies to NVR only.

### Access level

Action	NVR
view	User
set	User
control	User

## 21.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=  
wifi&action=<value>[&<parameter>=<value>...]
```

## 21.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view	ConnectedSSID	RES	<string>	Reads the SSID on the device
	SignalStrength	RES	<int>	Reads the Signal Strength
	WPSEnable	RES	<bool> True, False	Reads the WPS (WiFi Protected Setup).
	WiFiMode	REQ	<enum> Station, Direct	Reads the WiFi Mode.
set	Enable	REQ, RES	<bool> True, False	Enabling/Disabling WiFi feature.
control	Mode	REQ	<enum> Scan, Connect	Mode to scan for available WiFi devices, or to connect to the WiFi device
	WiFiMode	REQ	<enum> Station, Direct	Station mode

Action	Parameter	Request/ Response	Type/ Value	Description
	SSID	REQ	<string>	<p>WiFi device ID</p> <p>The <b>SSID</b> parameter is valid only when <b>Mode</b> is set to Connect.</p> <p>The <b>SSID</b> and <b>Password</b> parameters should be sent along with the <b>SecurityMode</b> parameter.</p> <p>The maximum length of the string should be less than 32 characters.</p>
	Password	REQ	<string>	<p>WiFi device password</p> <p>The <b>Password</b> parameter is valid only when <b>Mode</b> is set to Connect, and <b>SecurityMode</b> is not None.</p> <p>The <b>SSID</b> and <b>Password</b> parameters should be sent along with the <b>SecurityMode</b> parameter.</p>
	IsPasswordEncrypted	REQ	<bool> True, False	<p>True if password sent is encrypted</p> <p>Encrypted password should be sent as post message.</p>
	SecurityMode	REQ	<enum> None, WEP, PSK, Dot1X, Extended	<p>Encoding type</p> <p>The <b>SecurityMode</b> parameter is valid only when <b>Mode</b> is set to Connect.</p>
	AvailableNetworks.#.SSID	RES	<string>	<p>WiFi device ID</p> <p>The maximum length of the string should be less than 32 characters.</p>
	AvailableNetworks.#.SecurityMode	RES	<enum> None, WEP, WPA/WPA2 PSK	<p>Encoding type</p>
	AvailableNetworks.#.SignalStrength	RES	<int>	<p>Signal strength of the WiFi connection</p> <p>The range should be 0 to 100.</p>

## 21.4. Examples

### 21.4.1. Getting the WiFi configuration information

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=wifi&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
ConnectedSSID=  
SignalStrength=0  
WiFiMode=  
WPSEnable=False  
Enable=True
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "ConnectedSSID": "",  
  "SignalStrength": 0,  
  "WiFiMode": "",  
  "WPSEnable": false,  
  "Enable": true  
}
```

### 21.4.2. Scanning nearby WiFi device information

#### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=wifi&action=control&Mode=Scan&WiFiMode=Station
```

## TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
AvailableNetworks.1.SSID=iptime_pky
AvailableNetworks.1.SecurityMode=WPA/WPA2PSK
AvailableNetworks.1.SignalStrength=5
AvailableNetworks.2.SSID=kuhyeon_iptime
AvailableNetworks.2.SecurityMode=WPA/WPA2PSK
AvailableNetworks.2.SignalStrength=4
AvailableNetworks.3.SSID=iptime5G_pky
AvailableNetworks.3.SecurityMode=None
AvailableNetworks.3.SignalStrength=5
AvailableNetworks.4.SSID=iMacT
AvailableNetworks.4.SecurityMode=WPA/WPA2PSK
AvailableNetworks.4.SignalStrength=3
AvailableNetworks.5.SSID=TP-LINK_A3AF00
AvailableNetworks.5.SecurityMode=WPA/WPA2PSK
AvailableNetworks.5.SignalStrength=3
AvailableNetworks.6.SSID=SYG IPT 2g
AvailableNetworks.6.SecurityMode=WEP
AvailableNetworks.6.SignalStrength=3
AvailableNetworks.7.SSID=iptime
AvailableNetworks.7.SecurityMode=None
AvailableNetworks.7.SignalStrength=4
AvailableNetworks.8.SSID=NBAndroidHotspot2608
AvailableNetworks.8.SecurityMode=WPA/WPA2PSK
AvailableNetworks.8.SignalStrength=3
AvailableNetworks.9.SSID=B2C_PART4-5G
AvailableNetworks.9.SecurityMode=WPA/WPA2PSK
AvailableNetworks.9.SignalStrength=1
AvailableNetworks.10.SSID=B2C_PART1-5G
AvailableNetworks.10.SecurityMode=WPA/WPA2PSK
AvailableNetworks.10.SignalStrength=1
AvailableNetworks.11.SSID=NETGEAR27
AvailableNetworks.11.SecurityMode=WPA/WPA2PSK
AvailableNetworks.11.SignalStrength=3
AvailableNetworks.12.SSID=iptime_br
AvailableNetworks.12.SecurityMode=WPA/WPA2PSK
```

```
AvailableNetworks.12.SignalStrength=2
AvailableNetworks.13.SSID=KJK
AvailableNetworks.13.SecurityMode=WPA/WPA2PSK
AvailableNetworks.13.SignalStrength=3
AvailableNetworks.14.SSID=B2C_PART2-5G
AvailableNetworks.14.SecurityMode=WPA/WPA2PSK
AvailableNetworks.14.SignalStrength=1
AvailableNetworks.15.SSID=iptime_neighbor
AvailableNetworks.15.SecurityMode=None
AvailableNetworks.15.SignalStrength=4
AvailableNetworks.16.SSID=Mobile_Guest
AvailableNetworks.16.SecurityMode=WPA/WPA2PSK
AvailableNetworks.16.SignalStrength=2
AvailableNetworks.17.SSID=Mobile88_Guest
AvailableNetworks.17.SecurityMode=WPA/WPA2PSK
AvailableNetworks.17.SignalStrength=2
AvailableNetworks.18.SSID=MobileASUS_88
AvailableNetworks.18.SecurityMode=WPA/WPA2PSK
AvailableNetworks.18.SignalStrength=2
AvailableNetworks.19.SSID=iptime_sooj
AvailableNetworks.19.SecurityMode=WPA/WPA2PSK
AvailableNetworks.19.SignalStrength=2
AvailableNetworks.20.SSID=iptime_hwan
AvailableNetworks.20.SecurityMode=WPA/WPA2PSK
AvailableNetworks.20.SignalStrength=2
AvailableNetworks.21.SSID=UCOMM_IPTIME(Smart_Cam)
AvailableNetworks.21.SecurityMode=WPA/WPA2PSK
AvailableNetworks.21.SignalStrength=2
AvailableNetworks.22.SSID=MVipTime
AvailableNetworks.22.SecurityMode=WPA/WPA2PSK
AvailableNetworks.22.SignalStrength=2
```

## JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "AvailableNetworks": [
    {
```

```

        "SSID": "iptime_pky",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 5
    },
    {
        "SSID": "kuhyeon_iptime",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 4
    },
    {
        "SSID": "iptime5G_pky",
        "SecurityMode": "None",
        "SignalStrength": 5
    },
    {
        "SSID": "iMacT",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 3
    },
    {
        "SSID": "TP-LINK_A3AF00",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 3
    },
    {
        "SSID": "SYG IPT 2g",
        "SecurityMode": "WEP",
        "SignalStrength": 3
    },
    {
        "SSID": "iptime",
        "SecurityMode": "None",
        "SignalStrength": 4
    },
    {
        "SSID": "NBAndroidHotspot2608",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 3
    },
    {
        "SSID": "B2C_PART4-5G",

```



```

        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 1
    },
    {
        "SSID": "B2C_PART1-5G",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 1
    },
    {
        "SSID": "NETGEAR27",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 3
    },
    {
        "SSID": "iptime_br",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 2
    },
    {
        "SSID": "KJK",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 2
    },
    {
        "SSID": "B2C_PART2-5G",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 1
    },
    {
        "SSID": "iptime_neighbor",
        "SecurityMode": "None",
        "SignalStrength": 4
    },
    {
        "SSID": "Mobile_Guest",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 2
    },
    {
        "SSID": "Mobile88_Guest",
        "SecurityMode": "WPA/WPA2PSK",

```

```

        "SignalStrength": 2
    },
    {
        "SSID": "MobileASUS_88",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 2
    },
    {
        "SSID": "iptime_sooj",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 2
    },
    {
        "SSID": "iptime_hwan",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 2
    },
    {
        "SSID": "UCOMM_IPTIME(Smart_Cam)",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 2
    },
    {
        "SSID": "MVipTime",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 2
    },
    {
        "SSID": "iptime_yjcho",
        "SecurityMode": "WPA/WPA2PSK",
        "SignalStrength": 3
    }
]
}

```

### 21.4.3. Connecting WiFi device

#### REQUEST

```

http://<Device IP>/stw-
cgi/network.cgi?msubmenu=wifi&action=control&Mode=Connect&SSID=Samsung_5G&Se

```

```
curityMode=WEP&Password=5tkatjd!
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
OK
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json_  
<Body>
```

```
{  
  "Response": "Success"  
}
```

#### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=wifi&action=control&Mode=Connect&SSID=Samsung_5G&Se  
curityMode=None
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
OK
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json_
```

```
<Body>
```

```
{  
  "Response": "Success"  
}
```

#### 21.4.4. Enabling WiFi feature

##### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=wifi&action=set&Enable=False
```

##### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
OK
```

##### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json_  
<Body>
```

```
{  
  "Response": "Success"  
}
```

# Chapter 22. DHCP clients

## 22.1. Description

The **dhcpclients** submenu is used to get DHCP clients' information.

### NOTE

This chapter applies to NVR only.

### Access level

Action	NVR
view	User

## 22.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=
dhcpclients&action=view[&<parameter>=<value>...]
```

## 22.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view	InterfaceName	REQ, RES	<csv>	Requests information about specific interface name
	IPType	REQ, RES	<enum> IPv4, IPv6	Requests information about specific IP type
	Index	RES	<int>	Shows index of the response list.
	IPAddress	RES	<string>	Shows IP address of the client
	MacAddress	RES	<string>	Shows Mac address of the client

## 22.4. Examples

### 22.4.1. Getting DHCP client list

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=dhcpclients&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
```

<Body>

```
Index.1.InterfaceName=Network1
Index.1.IPType=IPv4
Index.1.IPAddress=192.168.111.18
Index.1.MACAddress=e1:6c:d6:ae:52:90
Index.2.InterfaceName=Network1
Index.2.IPType=IPv4
Index.2.IPAddress=192.168.111.19
Index.2.MACAddress=00:09:18:ff:ff:ff
Index.3.InterfaceName=Network1
Index.3.IPType=IPv4
```

### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json_
<Body>
```

```
{
  "DHCPClients": [
    {
      "Index": 1,
      "InterfaceName": "Network1",
      "IPType": "IPv4",
      "IPAddress": "192.168.111.18",
      "MACAddress": "e1:6c:d6:ae:52:90"
    },
    {
      "Index": 2,
      "InterfaceName": "Network1",
      "IPType": "IPv4",
      "IPAddress": "192.168.111.19",
      "MACAddress": "00:09:18:ff:ff:ff"
    }
  ]
}
```

# Chapter 23. RTSP over TLS settings

## 23.1. Description

The **rtspovertls** submenu supports RTSP over TLS (RTSPS).

**NOTE** | This chapter applies to NWC only.

### Access level

Action	Camera
view	User
set	User

## 23.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=rtspovertls&action=<value>[&<parameter>=<value>...]
```

## 23.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view				
set	Enable	REQ, RES	<bool> True, False	It is rtsp over tls
	Port	REQ, RES	<int>	It is rtsp over tls

## 23.4. Examples

### 23.4.1. Getting the RTSP over TLS configuration information

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?submenu=rtspovertls&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
Enable=True  
Port=1024
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  
  "Enable": true,  
  
  "Port": 1024  
  
}
```

### 23.4.2. Setting the RTSP over TLS configuration information

#### REQUEST

```
http://<Device IP>/stw-cgi  
/network.cgi?msubmenu=rtspovertls&action=set&Enable=True&Port=1024
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
OK
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{
```



```
"Response": "Success"
```

```
}
```

# Chapter 24. EthStatus

## 24.1. Description

The **dhcpcclients** submenu is used to get the inbound and outbound traffic info in real time.

**NOTE** | This chapter applies to NVR only.

### Access level

Action	NVR
view	User
set	User

## 24.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=ethstatus&action=<value>[&<parameter>=<value>...]
```

## 24.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view	EthIndex	RES	<int>	Ethernet Index
	InBound	RES	<float>	Inbound data rate in MBs
	OutBound	RES	<float>	Outbound rate in MBs

## 24.4. Examples

### 24.4.1. Getting the current inbound and outbound rates

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?submenu=ethstatus&action=view
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{
  "EthernetStaus": [
    {
      "EthIndex": 0,
      "InBound": 8.181055,
      "OutBound": 1.010742
    }
  ]
}
```

# Chapter 25. PoeStatus

## 25.1. Description

The **poestatus** submenu is used to get or change the settings of POE ports that device supports.

For camera users, they can find out whether their cameras support POE ports or not by referring to **POEExtender** parameter in attributes (attributes/network/support).

Action	Camera	NVR
view	User	User
set	Admin	Admin

## 25.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=poestatus&action=<value> [&<parameter>=<value>...]
```

## 25.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view	Port	REQ, RES	<csv>	PoE port number <b>CAMERA ONLY</b>
	Port.#.PowerConsumption	RES	<float>	Shows power consumption of each port.
	Total	RES	<float>	Sum of all PoE ports' power consumption
	Port.#.Summary	RES	<enum> NoFault, VoltageFault, ThermalShutdownFault, LoadDisconnect, OverloadTotalPower	Shows each port's status. <b>NVR ONLY</b>
set	Enable	REQ, RES	<bool>	Configures or sees activation status <b>CAMERA ONLY</b>

Action	Parameter	Request/Response	Type/Value	Description
	Port.#.Enable	REQ	<bool>	Configures each port's activation status  <b>NVR ONLY</b>

## 25.4. Examples

### 25.4.1. Getting the current PoE ports' status (NVR)

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=poestatus&action=view
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json_
<Body>
```

```
{
  "Total": 0,
  "Ports": [
    {
      "Port": 0,
      "PowerConsumption": 0,
      "Enable": true,
      "Summary": "NoFault"
    },
    {
      "Port": 1,
      "PowerConsumption": 0,
      "Enable": true,
      "Summary": "NoFault"
    },
    {
      "Port": 2,
      "PowerConsumption": 0,
      "Enable": true,
      "Summary": "NoFault"
    }
  ],
}
```

```

    {
      "Port": 3,
      "PowerConsumption": 0,
      "Enable": true,
      "Summary": "NoFault"
    },
    {
      "Port": 4,
      "PowerConsumption": 0,
      "Enable": true,
      "Summary": "NoFault"
    }
  ]
}

```

### 25.4.2. Getting the current PoE ports' status (camera)

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=poestatus&action=view
```

#### JSON RESPONSE

```

HTTP/1.0 200 OK
Content-type: application/json_
<Body>

```

```

{
  "Total": 0,
  "Ports": [
    {
      "Port": 1,
      "Enable": true,
      "PowerConsumption": 0
    }
  ]
}

```

### 25.4.3. Setting the PoE port to activate (camera)

#### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=poestatus&action=set&Port=1&Enable=True
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json_  
<Body>
```

```
{  
  "Response": "Success"  
}
```

### 25.4.4. Setting the PoE port to activate (NVR)

#### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=poestatus&action=set&Port.0.Enable=True
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json_  
<Body>
```

```
{  
  "Response": "Success"  
}
```

# Chapter 26. DHCP Server

## 26.1. Description

The **dhcpserver** submenu is used to configure DHCP server on device for each network interface.

### NOTE

This chapter applies to NVR only and IPv6 DHCP server can be configured only after enabling IPv6 on the interface.

### Access level

Action	NVR
view	User
set	User

## 26.2. Syntax

```
http://<Device IP>/stw-  
cgi/network.cgi?submenu=dhcpserver&action=<value> [&<parameter>=<value>...]
```

## 26.3. Parameters

Action	Parameter	Request/ Response	Type/ Value	Description
view	InterfaceName	REQ	<enum> Network1, Network2, Network3	Network interface name
	CheckServer	REQ	<bool> True, False	Check if another DHCP is running on the network  <b>Note</b> Response can take some time in this case.
	ExternalDhcpServer	RES	<bool> True, False	If True, another DHCP server is already running on the same network.  If checkserver parameter is delivered through the request, this parameter is added to the response.



Action	Parameter	Request/Response	Type/Value	Description
set	InterfaceName	REQ, RES	<enum> Network1, Network2, Network3	Network interface name used as reference for set operation (Fixed value cannot be changed)
	Enable	REQ, RES	<bool> True, False	Enable DHCP server
	IPType	REQ, RES	<enum> IPv4, IPv6	IPType Selection  <b>Note</b> Both IPv4 and IPv6 DHCP servers can be enabled on the same interface.
	IPRangeFrom	REQ, RES	<string>	IP Allocation starting range
	IPRangeTo	REQ, RES	<string>	IP Allocation limit
	IPLeaseTime	REQ, RES	<int>  secs	Lease duration in seconds (valid range : 3600 - 172800)

## 26.4. Examples

### 26.4.1. Getting the current DHCP server settings

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?submenu=dhcpserver&action=view
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json_
<Body>
```

```
{
  "DHCPServer": [
    {
      "InterfaceName": "Network1",
      "IPType": "IPv4",
      "Enable": true,
      "IPRangeFrom": "192.168.75.2",
      "IPRangeTo": "192.168.75.254",
```

```

        "IPLeaseTime": 86400
    },
    {
        "InterfaceName": "Network1",
        "IPType": "IPv6",
        "Enable": false,
        "IPRangeFrom": "",
        "IPRangeTo": "",
        "IPLeaseTime": 86400
    },
    {
        "InterfaceName": "Network2",
        "IPType": "IPv4",
        "Enable": false,
        "IPRangeFrom": "192.168.2.2",
        "IPRangeTo": "192.168.2.254",
        "IPLeaseTime": 86400
    },
    {
        "InterfaceName": "Network2",
        "IPType": "IPv6",
        "Enable": false,
        "IPRangeFrom": "",
        "IPRangeTo": "",
        "IPLeaseTime": 86400
    },
    {
        "InterfaceName": "Network3",
        "IPType": "IPv4",
        "Enable": false,
        "IPRangeFrom": "192.168.3.2",
        "IPRangeTo": "192.168.3.254",
        "IPLeaseTime": 86400
    },
    {
        "InterfaceName": "Network3",
        "IPType": "IPv6",
        "Enable": false,
        "IPRangeFrom": "",
        "IPRangeTo": "",
        "IPLeaseTime": 86400
    }

```

```
}  
]  
}
```

## 26.4.2. Setting DHCP server settings

### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=dhcpserver&action=set&Enable=True&InterfaceName=Net  
work2&IPType=IPv4
```

### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json_  
<Body>
```

```
{  
  "Response": "Success"  
}
```

# Chapter 27. ONVIF Discovery

## 27.1. Description

The **onvifdiscovery** submenu enables or disables ONVIF Discovery mechanism. In factory default, ONVIF discovery is enabled. When it is enabled, camera can be discovered using the ONVIF ws-discovery protocol.

NOTE

This chapter applies to network cameras and NVR only.

### Access level

Action	Camera	NVR
view	Admin	Admin
set	Admin	Admin

## 27.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=onvifdiscovery&action=<value> [&<parameter>=<value>]
```

## 27.3. Parameters

Action	Parameter	Request/Response	Type/Value	Description
view				Reads the onvifdiscovery settings.
	InterfaceName	REQ	<string>	Requests information about specific interface name
set	InterfaceName	REQ, RES	<string>	Interface name
	Enable	REQ, RES	<bool> True, False	Enables or disables ONVIF discovery service

Note

**Enable** must be sent together with the **set** action.

## 27.4. Examples

### 27.4.1. Getting the current onvif discovery settings

## REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=onvifdiscovery&action=view
```

## TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
InterfaceName=NetworkInterface1  
Enable=True
```

## JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "OnvifDiscovery": [  
    {  
      "InterfaceName": "NetworkInterface1",  
      "Enable": true  
    }  
  ]  
}
```

## 27.4.2. Enabling ONVIF discovery service

### REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=onvifdiscovery&action=set&Enable=True
```

# Chapter 28. SIP Setup

## 28.1. Description

The **sipsetup** submenu is used to configure SIP general settings.

### NOTE

This chapter is applicable to intercom cameras.

Attribute to check for feature support: "attributes/Network/Support/SIP"

### Access level

Action	Camera
view	Admin
set	Admin

## 28.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=  
sipsetup&action=<value> [&<parameter>=<value>]
```

## 28.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
view	SIPEnable	REQ, RES	<bool> True, False	Whether to use SIP
set	AllowIncomingCalls	REQ, RES	<bool> True, False	Whether to allow incoming calls
	SIPPort	REQ, RES	<int>	Port number to be used by SIP
	SIPTLSPort	REQ, RES	<int>	Port number to be used by SIPoverTLS
	RTSPStartPort	REQ, RES	<int>	Initial port number to be used by RTSP
	TransportAutoSwitchEnable	REQ, RES	<bool> True, False	Whether to use transport auto switch
	AllowContactRewrite	REQ, RES	<bool> True, False	Whether to use contact rewrite
	CallDurationLimitEnable	REQ, RES	<bool> True, False	Whether to use call duration limit
	CallDurationLimit	REQ, RES	<int>	Call duration limit (minutes)
	RegistrationInterval	REQ, RES	<int>	Registration interval (seconds)

Action	Parameters	Request/Response	Type/Value	Description
	AudioDirection	REQ, RES	<enum> ReceiveOnly, SendOnly, SendAndReceive	Audio direction
	CallStopEnable	REQ, RES	<bool> True, False	If enabled, user can stop the call with the call button

## 28.4. Examples

### 28.4.1. Getting the current sip setting values from cameras (this submenu only supports JSON responses)

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?submenu=sipsetup&action=view
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "SIPEnable": true,
  "AllowIncomingCalls": false,
  "SIPPort": 5060,
  "SIPTLSPort": 5061,
  "RTPStartPort": 4000,
  "TransportAutoSwitchEnable": true,
  "AllowContactRewrite": true,
  "CallDurationLimitEnable": false,
  "CallDurationLimit": 60,
  "RegistrationInterval": 300,
  "AudioDirection": "SendAndReceive",
  "CallStopEnable": true
}
```

## 28.4.2. Setting SIP configurations

### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=sipsetup&action=set&
SIPEnable=True&SIPPort=4521&SIPTLSPort=5001&AllowIncomingCalss=False&RTSPSta
rtPort=4000&TransportAutoSwitchEnable=True&AllowContactRewrite=True
```

### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Response": "Success"
}
```



# Chapter 29. SIP Account Settings

## 29.1. Description

The **sipaccount** submenu configures credential and connection settings of the SIP registrar

### NOTE

This chapter is applicable to intercom cameras.

Attributes to check for feature support: "Attributes/Network/Support/SIP"

### Access level

Action	Camera
view	Admin
add/update	Admin
remove	Admin

## 29.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=  
sipaccount&action=<value> [&<parameter>=<value>]
```

## 29.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
view	Index	REQ	<int>	Account index
	Accounts.#.Status	RES	<enum> Unavailable , Available	Account status
	Accounts.#.Type	RES	<enum> PeerToPeer , Registrar	Account type
	Accounts.#.IsFixed	RES	<bool> True, False	Whether the default account is fixed

Action	Parameters	Request/ Response	Type/ Value	Description
	Accounts.#.ConnectionTypeInUse	RES	<enum> Domain, BackupDomain, Registrar, BackupRegistrar, PeerToPeer	Connection the account is using
add/update	Index	REQ, RES	<int>	Account index
	Name	REQ, RES	<string>	Account name
	Enable	REQ, RES	<bool> True, False	Whether to use account
	UserID	REQ, RES	<string>	SIP user ID
	AuthenticationID	REQ, RES	<string>	SIP authentication ID
	Password	REQ, RES	<string>	SIP password
	CallerID	REQ, RES	<string>	SIP Caller ID
	DomainName	REQ, RES	<string>	SIP Domain name
	RegistrarAddress	REQ, RES	<string>	SIP Registrar address
	BackupDomainName	REQ, RES	<string>	SIP backup domain name
	BackupRegistrarAddress	REQ, RES	<string>	SIP backup registrar address
	SIPProxyAddress	REQ, RES	<string>	SIP Proxy address
	TransportMode	REQ, RES	<enum> UDP, TCP, TLS	SIP transport mode
	IsPasswordEncrypted	REQ	<bool> True, False	When this is set to true, the password is encrypted using the public key obtained through the <b>rsa</b> submenu of security and cgi, and it is sent as payload content for the POST command.
remove	Index	REQ	<csv>	Account index

## 29.4. Examples

### 29.4.1. Getting current SIP account settings (this submenu only supports JSON responses)

## REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?submenu=sipaccount&action=view
```

## JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Accounts": [
    {
      "Index": 0,
      "Status": "Available",
      "Name": "peer-to-peer",
      "Enable": true,
      "Type": "PeerToPeer",
      "UserID": "peer-to-peer",
      "AuthenticationID": "",
      "Password": "",
      "CallerID": "",
      "DomainName": "192.168.71.90",
      "RegistrarAddress": "192.168.71.90",
      "BackupDomainName": "",
      "BackupRegistrarAddress": "",
      "SIPProxyAddress": "",
      "TransportMode": "UDP",
      "IsFixed": true,
      "ConnectionTypeInUse": "PeerToPeer"
    },
    {
      "Index": 1,
      "Status": "Unavailable",
      "Name": "account1",
      "Enable": true,
      "Type": "Registrar",
      "UserID": "admin",
      "AuthenticationID": "0001",
      "Password": "",
      "CallerID": "00001",
    }
  ]
}
```

```

        "DomainName": "192.168.125.213",
        "RegistrarAddress": "192.168.125.213",
        "BackupDomainName": "192.168.125.213",
        "BackupRegistrarAddress": "1",
        "SIPProxyAddress": "",
        "TransportMode": "UDP",
        "IsFixed": false,
        "ConnectionTypeInUse": "Domain"
    }
]
}

```

### 29.4.2. Adding a SIP account

#### REQUEST

```

http://<Device IP>/stw-
cgi/network.cgi?submenu=sipaccount&action=add&Name=test&Enable=True&UserID=
test&AuthenticationID=1234&Password=1122&CallerID=0011&DomainName=domainName
&RegistrarAddress=addr&TransportMode=TLS

```

#### JSON RESPONSE

```

HTTP/1.0 200 OK
Content-type: application/json
<Body>

```

```

{
    "Response": "Success",
    "Index": 1
}

```

### 29.4.3. Updating account settings

#### REQUEST

```

http://<Device IP>/stw-cgi/network.cgi?submenu=
sipaccount&action=update&Index=1&Name=name&Enable=True&UserID=admin&Password
=pass&DomainName=192.168.125.213&TransportMode=UDP&AuthenticationID=0001&Cal
lerID=0001&BackupDomainName=192.168.125.213&RegistrarAddress=192.168.125.213
&BackupRegistrarAddress=192.168.125.214

```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Response": "Success"
}
```

#### 29.4.4. Remove SIP account

##### REQUEST

```
http://<Device IP>/stw-
cgi/network.cgi?submenu=sipaccount&action=remove&Index=1
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Response": "Success"
}
```

# Chapter 30. SIP Recipient Settings

## 30.1. Description

The **siprecipients** submenu is used to manage SIP recipients. It can add, update, or delete SIP recipients, and you can choose the type of call requests.

### NOTE

This chapter is applicable to intercom cameras.

Attributes to check for feature support: "Attributes/Network/Support/SIP"

### Access level

Action	Camera
view	Admin
set	Admin
add	Admin
update	Admin
remove	Admin

## 30.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=  
siprecipients&action=<value> [&<parameter>=<value>]
```

## 30.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
view	Index	REQ	<int>	Recipient index
	Group	REQ	<enum> None, 1~5	Group number for filter of view
set	CallRequestType	REQ, RES	<enum> Single, Multiple	Recipient calling type when a CallRequest event occurs
	RecipientInSingleMode	REQ, RES	<enum> None, #	Recipient when the <b>CalRequestType</b> is in Single mode  It can set the recipient index

Action	Parameters	Request/ Response	Type/ Value	Description
	RecipientInMultipleMode	REQ, RES	<enum> None, 1~5	Recipient when the <b>CallRequestType</b> is in Multiple mode  It can set the group number
add	Name	REQ	<string>	Recipient name
	TransportMode	REQ	<enum> Registrar, PeerToPeer	Recipient transport mode
	Address	REQ	<string>	Recipient address
	Group	REQ	<enum> None, 1~5	Recipient group
update	Recipients.#.Name	REQ, RES	<string>	Recipient name
	Recipients.#.TransportM ode	REQ, RES	<enum> Registrar, PeerToPeer	Recipient transport mode
	Recipients.#.Address	REQ, RES	<string>	Recipient address
	Recipients.#.Group	REQ, RES	<enum> None, 1~5	Recipient group
remove	Index	REQ	<csv> All, #	Recipient index

## 30.4. Examples

### 30.4.1. Getting SIP recipient settings (this submenu only supports JSON responses)

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?submenu=siprecipients&action=view
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "CallRequestType": "Single",
  "RecipientInSingleMode": "None",
```

```

"RecipientInMultipleMode": "None",
"Recipients": [
  {
    "Index": 1,
    "Name": "recipient1",
    "TransportMode": "Registrar",
    "Address": "192.168.125.123",
    "Group": [
      "1",
      "2",
      "3"
    ]
  }
]
}

```

### 30.4.2. Adding a recipient by assigning them to groups 1, 2, or 3

#### REQUEST

```

http://<Device IP>/stw-cgi/
network.cgi?msubmenu=siprecipients&action=add&Name=recipient1&TransportMode=
Registrar&Address=192.168.125.123&Group=1,2,3

```

#### JSON RESPONSE

```

HTTP/1.0 200 OK
Content-type: application/json
<Body>

```

```

{
  "Response": "Success",
  "Index": 1
}

```

### 30.4.3. Updating a recipient for Index 1

#### REQUEST

```

http://<Device IP>/stw-cgi/network.cgi?
msubmenu=siprecipients&action=update&Recipients.1.Name=recipient2&Recipients

```



```
.1.TransportMode=PeerToPeer&Recipients.1.Address=192.168.125.123&Recipients.1.Group=1,2
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Response": "Success"
}
```

### 30.4.4. Selecting a target recipient and setting the Call request type to Single mode

The group you are setting up must contain at least one recipient.

#### REQUEST

```
http://<Device IP>/stw-cgi/
network.cgi?submenu=siprecipients&action=set&CallRequestType=Single&RecipientInSingleMode=1
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Response": "Success"
}
```

### 30.4.5. Selecting a target recipient group and setting the Call request type to Multiple mode

You can set a group index of previously registered recipients to a group.

## REQUEST

```
http://<Device IP>/stw-cgi/  
network.cgi?msubmenu=siprecipients&action=set&CallRequestType=Multiple&Recip  
ientInMultipleMode=1
```

## JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "Response": "Success"  
}
```

### 30.4.6. Removing all recipients

## REQUEST

```
http://<Device IP>/stw-  
cgi/network.cgi?msubmenu=siprecipients&action=remove&Index=All
```

## JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "Response": "Success"  
}
```

# Chapter 31. SIP Call

## 31.1. Description

The **sipcall** submenu is used for getting SIP call states and controlling SIP calls.

### NOTE

This chapter is applicable to intercom cameras.

Attributes to check for feature support: "Attributes/Network/Support/SIP"

### Access level

Action	Camera
check	Admin
control	Admin

## 31.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=  
sipcall&action=<value> [&<parameter>=<value>]
```

## 31.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
check	CallState	RES	<enum> Idle, Call, Ringing	SIP call state
	TestCallState	RES	<enum> Idle, Trying, Success, Failure	State for test call control commands
control	StopCallRequest	REQ	<bool> True, False	SIP call stop request  <b>Note</b> Deprecated parameter, Use <b>CallRequest</b> instead of this parameter

Action	Parameters	Request/Response	Type/Value	Description
	CallRequest	REQ	<enum> StopVMS, StartVMS, StopCall, TestCall	control the callrequest <ul style="list-style-type: none"> <li>• StopVMS: Stop VMS call session</li> <li>• StartVMS: Stop CallRequest event to connect from VMS, return error if SIP is already connected</li> <li>• StopCall: Stop all call session</li> <li>• TestCall: Start test call</li> </ul>

## 31.4. Examples

### 31.4.1. Getting the current SIP call state (this submenu only supports JSON responses)

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=sipcall&action=check
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "CallState": "Idle",
  "TestCallState": "Idle"
}
```

### 31.4.2. Stopping a call request

This command is used to mean that the VMS has accepted the CallRequest. If you request this command, CallRequest and SIP Calling will be aborted.

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=sipcall&action=control&CallRequest=StartVMS
```

## JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "Response": "Success"  
}
```

# Chapter 32. NAT Traversal Settings

## 32.1. Description

The **nattraversal** submenu provides settings for the NAT Traversal features.

### NOTE

This chapter is applicable to intercom cameras.

Attributes to check for feature support: "Attributes/Network/Support/SIP"

### Access level

Action	Camera
view	Admin
set	Admin

## 32.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=  
nattraversal&action=<value>[&<parameter>=<value>]
```

## 32.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
view				
set	ICEEnable	REQ, RES	<bool> True, False	Whether to use ICE
	STUNEnable	REQ, RES	<bool> True, False	Whether to use STUN
	STUNAddress	REQ, RES	<string>	STUN address
	TURNEnable	REQ, RES	<bool> True, False	Whether to use TURN
	TURNAddress	REQ, RES	<string>	TURN address
	TURNUserID	REQ, RES	<string>	TURN user ID
	TURNUserPassword	REQ, RES	<string>	TURN user password

Action	Parameters	Request/Response	Type/Value	Description
	IsPasswordEncrypted	REQ	<bool> True, False	When this is set to true, the password is encrypted using the public key obtained through the <b>rsa</b> submenu of security and cgi, and is sent as payload content for the POST command.

## 32.4. Examples

### 32.4.1. Getting current NAT traversal settings (this submenu only supports JSON responses)

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=nattraversal&action=view
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "ICEEnable": true,
  "STUNEnable": true,
  "STUNAddress": "192.168.71.27",
  "TURNEnable": true,
  "TURNAddress": "192.168.71.27",
  "TURNUserID": "user1",
  "TURNUserPassword": ""
}
```

### 32.4.2. Changing the NAT traversal settings

#### REQUEST

```
http://<Device IP>/stw-cgi/
network.cgi?msubmenu=nattraversal&action=set&ICEEnable=True&STUNEnable=True&
STUNAddress=192.168.71.27&TURNEnable=True&TURNAddress=192.168.71.27&TURNUser
ID=user1&TURNUserPassword=pass
```

## JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "Response": "Success"  
}
```



# Chapter 33. P2P

## 33.1. Description

The **p2p** submenu is used to set and retrieve the p2p activation state.

**NOTE** | This chapter applies to NVR only

### Access level

Action	NVR
view	User
set	User

## 33.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=
p2p&action=<value>[&<parameter>=<value>]
```

## 33.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
view	Status	RES	<enum> Success, NoCertificate, NetworkUnavailable,, DeviceKeyIDFail,, ConfigDownloadFail,, ConnectionFail, Unknown	Success - Succeeded  NoCertificate - No public certificate installed. DeviceKeyIDFail - Failed to create device ID and device key.  ConfigDownloadFail - Failed to download P2P setting information.  ConnectionFail - Failed to connect server.  NetworkUnavailable - Network settings are wrong.  Unknown - Unknown Errors
set	Enable	REQ, RES	<bool> True, False	P2P use, not used.

## 33.4. Examples

### 33.4.1. Retrieving the current P2P state.

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=p2p&action=view
```

#### TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
Enable=True  
Status=Success
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "P2P": {  
    "Enable": true,  
    "Status": "Unknown"  
  }  
}
```

# Chapter 34. MQTT client settings

## 34.1. Description

The **mqttclient** submenu is used to configure the MQTT client

**NOTE** | This chapter applies to network cameras only.

### Access level

Action	Camera
view	Admin
set	Admin
check	Admin

## 34.2. Syntax

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=
mqttclient&action=<value> [&<parameter>=<value>...]
```

## 34.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
view				
set	Enable	REQ, RES	<bool> True, False	Configures whether to enable the MQTT client.
	Username	REQ, RES	<string>	Username to use to connect to the MQTT broker
	Password	REQ, RES	<string>	Password to use to connect to the MQTT broker
	Address	REQ, RES	<string>	Address of the MQTT broker
	Port	REQ, RES	<int>	Port of the MQTT broker
	Transport	REQ, RES	<enum> TCP, TLS, WebSocket, WebSocket Secure	Transport type to connect to the MQTT broker

Action	Parameters	Request/ Response	Type/ Value	Description
	BasePath	REQ, RES	<string>	BasePath for WebSocket  <b>Note</b> This parameter is valid only when <b>Transport</b> set to <b>WebSocket</b> or <b>WebSocketSecure</b>
	CleanSessionEnable	REQ, RES	<bool> True, False	Whether the CleanSession flag is enabled when connecting to an MQTT broker
	CustomClientIDEnable	REQ, RES	<bool> True, False	Whether to use a custom client ID when connecting to an MQTT broker
	CustomClientID	REQ, RES	<string>	Custom client ID  <b>Note</b> This parameter is only valid when <b>CustomClientIDEnable</b> is set to <b>True</b> , and if it is an empty string value, it is used an auto-generated id.
	KeepAliveInterval	REQ, RES	<int>	Keep alive interval
	ConnectionTimeout	REQ, RES	<int>	Connection timeout
	CACertificateName	REQ, RES	<string>	The name of the CA certificate installed in the device  <b>Note</b> For a list of CA certificates, check the <b>cacertificate</b> submenu of security.cgi.
	ClientCertificateName	REQ, RES	<string>	The name of the client certificate installed in the device  <b>Note</b> For a list of client certificates, check the <b>ssl</b> submenu of security.cgi.
	VerifyServerCertification	REQ, RES	<bool> True, False	Whether to verify the MQTT broker's certificate
	ALPN	REQ, RES	<string>	ALPN (Application Layer Protocol Negotiation) protocol ID

Action	Parameters	Request/Response	Type/Value	Description
	AutoReconnectEnable	REQ, RES	<bool> True, False	Whether to automatically reconnect to the MQTT broker
	DefaultTopicPrefix	REQ, RES	<string>	Default prefix for MQTT topics
	ConnectionMessageIndex	REQ, RES	<int>	Index of the message to be sent when connected to the MQTT broker. If the value is set to 0, it is unspecified.  <b>Note</b> To set this item, <b>MQTT Publication</b> message must be set in advance
	LWTMessageIndex	REQ, RES	<int>	Message index to be sent when MQTT broker abnormal disconnection. If the value is set to 0, it is unspecified.  <b>Note</b> To set this item, <b>MQTT Publication</b> message must be set in advance
	IsPasswordEncrypted	REQ, RES	<bool> True, False	When this is set to true, the password is encrypted using the public key obtained through the <b>rsa</b> submenu of security and cgi, and is sent as payload content for the POST command.
check	Status	RES	<enum> Disconnected, Connected	Current connection status of MQTT client

## 34.4. Examples

### 34.4.1. Getting the MQTT client configuration information (this submenu supports only JSON response)

#### REQUEST

```
http://<Device IP>/stw-cgi/network.cgi?msubmenu=mqttclient&action=view
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
```

Content-type: application/json

<Body>

```
{
  "Enable": true,
  "Username": "user",
  "Password": "",
  "Address": "192.168.0.2",
  "Port": 1883,
  "Transport": "TCP",
  "BasePath": "",
  "CleanSessionEnable": false,
  "CustomClientIDEnable": false,
  "CustomClientID": "00:09:18:FF:FF:FF",
  "KeepAliveInterval": 30,
  "ConnectionTimeout": 60,
  "CACertificateName": "HTW_rootca",
  "VerifyServerCertification": false,
  "ClientCertificateName": "HTW_default",
  "ALPN": "",
  "AutoReconnectEnable": false,
  "DefaultTopicPrefix": "TestPrefix",
  "ConnectionMessageIndex": 0,
  "LWTMessageIndex": 0
}
```

### 34.4.2. Setting the MQTT client configuration information

#### REQUEST

```
http://<Device IP>/stw-
cgi/network.cgi?submenu=mqttclient&action=set&Enable=True&Username
=user&Password=pass&Address=192.168.0.2&Port=1883&Transport=TCP&CleanSession
Enable=False&CustomClientIDEnable=False&KeepAliveInterval=30&ConnectionTimeo
ut=30&CACertificateName=HTW_rootca&VerifyServerCertification=False&ClientCer
tificateName=HTW_default&AutoReconnectEnable=False&DefaultTopicPrefix=TestPr
efix
```

#### JSON RESPONSE

```
HTTP/1.0 200 OK
```

Content-type: application/json\_  
<Body>

```
{  
  "Response": "Success"  
}
```