Bypass

Hanwha

SUNAPI

v2.6.2
2023-04-07

Contact Information

Hanwha Vision Co., Ltd.
Hanwha Vision 6, Pangyo-ro 319beon-gil, Bundang-gu, Seongnam-si, Gyeonggi-do, 13488, KOREA
www.hanwhavision.com

Hanwha Vision America
500 Frank W. Burr Blvd. Suite 43 Teaneck, NJ 07666
hanwhavisionamerica.com

Hanwha Vision Europe
Heriot House, Heriot Road, Chertsey, Surrey, KT16 9DT, United Kingdom
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

Table of Contents

Chapter 1. Overview

1.1. Description

bypass.cgi is used to send SUNAPI commands to the connected cameras (Registered using SUNAPI protocol only) via NVR, and to get the response for the corresponding command.

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

Chapter 2. bypass

2.1. Description

The **bypass** submenu in NVR bypasses the commands sent by SUNAPI client applications directly to the camera, and gets the response from the camera.

> **NOTE**
>
> This submenu supports all SUNAPI commands except for multi-part response commands such as intermediate events for firmware update, monitor and monitordiff actions in eventstatus.cgi.
> For the Firmware update command, only the final response will be sent to the requested client.
> For other commands, the response sent by the camera will be given as the response for a bypass command.

Access level

Action	Camera
control	User

2.2. Syntax

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

2.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
control	Channel	REQ	<int>	Channel number
	HeadersToSend	REQ	<csv>	Extra headers to be sent to camera such as cookies, response format etc.
	BypassURI	REQ	<string>	SUNAPI command to be sent to camera

2.4. Examples

2.4.1. Getting Camera Attributes

REQUEST

```
http://<Device IP>/stw-cgi/bypass.cgi?msubmenu=bypass&action=control&Channel=0&BypassURI=/stw-
```

```
cgi/attributes.cgi/eventsources/videoanalysis/view/DetectionType
```

TEXT RESPONSE

```
<parameter response="false" request="true" name="DetectionType">
    <dataType>
        <enum>
            <entry value="Off"/>
            <entry value="MotionDetection"/>
            <entry value="IntelligentVideo"/>
            <entry value="MDAndIV"/>
        </enum>
    </dataType>
</parameter>
```

2.4.2. Getting camera events in text format

REQUEST

```
http://<Device IP>/stw-
cgi/bypass.cgi?msubmenu=bypass&action=control&Channel=0&BypassURI=/stw-
cgi/eventstatus.cgi?msubmenu=eventstatus&action=check
```

TEXT RESPONSE

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

```
Channel.0.MotionDetection=True
Channel.0.MotionDetection.RegionID.1=True
Channel.0.MotionDetection.RegionID.2=False
Channel.0.MotionDetection.RegionID.3=False
Channel.0.MotionDetection.RegionID.4=False
Channel.0.Tampering=False
Channel.0.DefocusDetection=False
Channel.0.AudioDetection=False
Channel.0.VideoAnalytics.Passing=False
Channel.0.VideoAnalytics.Entering=False
Channel.0.VideoAnalytics.Exiting=False
Channel.0.VideoAnalytics.Appearing=False
```

```
Channel.0.VideoAnalytics.Disappering=False
AlarmInput.1=False
AlarmOutput.1=False
```

2.4.3. Getting camera events in JSON format

REQUEST

```
http://<Device IP>/stw-
cgi/bypass.cgi?msubmenu=bypass&action=control&Channel=0&
HeadersToSend=Accept:application/json&BypassURI=/stw-
cgi/eventstatus.cgi?msubmenu=eventstatus&action=check
```

JSON RESPONSE

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

```
{
    "ChannelEvent": [
        {
            "Channel": 0,
            "MotionDetection": true,
            "MotionDetectionRegions": {
                "1": true,
                "2": false,
                "3": false,
                "4": false
            },
            "Tampering": false,
            "DefocusDetection": false,
            "AudioDetection": false,
            "VideoAnalytics": {
                "Passing": false,
                "Entering": false,
                "Exiting": false,
                "Appearing": false,
                "Disappering": false
            }
        }
```

```
    ],
    "AlarmInput": {
        "1": false
    },
    "AlarmOutput": {
        "1": false
    }
}
```

NOTE HeadersToSend is not mandatory; this parameter is used to send any special headers to the camera. "Accept:application/json" header can also be sent to NVR to get the response from the camera in JSON format.

2.4.4. Configuration backup

REQUEST

```
curl --digest -u <userid>:<password> "http://<Device IP>/stw-
cgi/bypass.cgi?msubmenu=bypass&action=control&Channel=0&BypassURI=/stw-
cgi/system.cgi?msubmenu=configbackup&action=control" > config.bin
```

2.4.5. Configuration restore

Encode to base64

```
openssl base64 -in config.bin -out encoded.bin
```

REQUEST

```
curl --digest -u <userid>:<password> "http://<Device IP>/stw-
cgi/bypass.cgi?msubmenu=bypass&action=control&Channel=0&BypassURI=/stw-
cgi/system.cgi?msubmenu=configrestore&action=control&ExcludeSettings=Network
,Camera" -H "Expect:" --data-urlencode @encoded.bin
```

2.4.6. Firmware update

REQUEST

```
curl --digest -u <userid>:<password> "http://<Device IP>/stw-
cgi/bypass.cgi?msubmenu=bypass&action=control&Channel=0&BypassURI=/stw-
cgi/system.cgi?msubmenu=firmwareupdate&action=control&Type=Normal" -H
"Expect:" -F uploadFile=@srn-4000-pkg_v2.00_150114103354.img
```