Transfer

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

transfer.cgi configures the FTP/SMTP server settings to transfer images or event notification when an event occurs.

The actual transfer through FTP/SMTP is controlled by the EventAction parameter in **eventrules.cgi** for camera and smtp submenu eventactions.cgi for NVR when certain events occur such as Alarm Input, Video Analytics, Video Loss, Network Event, Face Detection, Tampering Detection, Audio Detection, Tracking and Timer.

The following submenus are used to control the transfer API:

- **ftp**: Sets the FTP (File Transfer Protocol) configuration for sending images or videos.
- **smtp**: Sets the SMTP (Simple Mail Transfer Protocol) configuration for sending notification messages. Images can be attached as email attachments.
- **smtpusers**: Sets the SMTP mail users.
- **smtpgroups**: Sets the SMTP mail user groups.
- **dataserver**: Sets the data server.

Chapter 2. FTP

2.1. Description

The **ftp** submenu configures the FTP (File Transfer Protocol) server settings.

> **NOTE** This chapter applies to network cameras only.
> Attribute to check for Feature Support: "**attributes/transfer/Support/FTP**"

Access level

Action	Camera
view	Admin
set	Admin
test	Admin

2.2. Syntax

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

2.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
view				Reads the FTP settings
set	Host	REQ, RES	<string>	Domain name of the FTP server (required) The host is specified in the format of <IP Address> or <Host name> e.g. ftp.samsung.com or 192.168.100.1
	Port	REQ, RES	<int>	FTP port number
	Username	REQ, RES	<string>	FTP user name
	Password	REQ, RES	<string>	FTP user password
	IsPasswordEncrypted	REQ	<bool>	Returns true if password sent is encrypted. Encrypted password should be sent as a post message.

Action	Parameters	Request/ Response	Type/ Value	Description
	Mode	REQ, RES	<enum> Active, Passive	FTP transmission mode
	Path	REQ, RES	<string>	The path on the FTP server where alarm images are saved. Value must be the relative path.
	ReportFileType	REQ, RES	<enum> Image, VideoClip	The type of file which is sent to the FTP server. When this parameter is not supported, only images are provided if a configured event is triggered.
	PreEventDuration	REQ, RES	<enum> 1s, 2s, 3s	Duration of pre-event recording. The unit is second. This parameter is valid when "ReportFileType" is "VideoClip"
	PostEventDuration	REQ, RES	<enum> 10s, 15s, 20s	Duration of post-event recording. The unit is second. This parameter is valid when "ReportFileType" is "VideoClip"
test	status	RES	<string>	Tries to connect to the server with the configured settings and returns the result in the form of a string

2.4. Examples

2.4.1. Getting FTP server information

REQUEST

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

TEXT RESPONSE

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

```
Host=192.168.100.1
Mode=Active
Port=22
Path=/home/test
```

```
Username=test
Password=
```

```
Host=192.168.100.1
Mode=Active
Port=22
Path=/home/test
Username=test
Password=
ReportFileType=Image
PreEventDuration=3s
PostEventDuration=10s
```

JSON RESPONSE

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

```
{
    "Host": "192.168.100.1",
    "Mode": "Active",
    "Port": 22,
    "Path": "/home/test",
    "Username": "test",
    "Password": ""
}
```

```
{
    "Host": "192.168.100.1",
    "Mode": "Active",
    "Port": 22,
    "Path": "/home/test",
    "Username": "test",
    "Password": "",
    "ReportFileType": "Image",
    "PreEventDuration": "3s",
    "PostEventDuration": "10s"
```

```
}
```

2.4.2. Setting the FTP server

The FTP host, user name, password, mode, and path can be configured as shown in the command below.

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=ftp&action=set&Password=tes&Host=223.255.254.254&P
ort=4&Username=tes&Mode=Active&Path=test1234
```

2.4.3. Setting the FTP server with an encrypted password

The FTP host, user name, mode, and path can be configured as shown in the command below. Password should be encrypted with RSA and RSA_PKCS1_PADDING (Refer to security.cgi for information on obtaining the RSA key).

Base 64 Encoded data should be sent as a POST message.

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=ftp&action=set&Host=223.255.254.254&Port=4&Usernam
e=tes&Mode=Active&Path=test1234&IsPasswordEncrypted=True
```

2.4.4. Checking the connection status of the FTP Server

REQUEST

```
http://<Device IP>/stw-cgi/transfer.cgi?msubmenu=ftp&action=test
```

TEXT RESPONSE

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

```
Status=Success
```

JSON RESPONSE

```
HTTP/1.0 200 OK
```

```
Content-type: application/json
<Body>
```

```
{
"Status": "Success"
}
```

Chapter 3. SMTP

3.1. Description

The **smtp** submenu configures the SMTP (Simple Mail Transfer Protocol) server settings.

> **NOTE** Attribute to check for Feature Support: "**attributes/transfer/Support/SMTP**"

Access level

Action	Camera	NVR
view	Admin	User
set	Admin	User
test	Admin	User

3.2. Syntax

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

3.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
view				Reads the SMTP settings.
set	Authentication	REQ, RES	<enum> None, SMTP, POPBefore SMTP	SMTP authentication
	Host	REQ, RES	<string>	Domain name of the SMTP server (Required). The host is specified in the format of <IP Address> or <Host name> e.g. smtp.samsung.com or 192.168.100.1, and the max length is 63.
	Port	REQ, RES	<int>	SMTP port number
	Encryption	REQ, RES	<enum> None, SSL	SMTP encryption type

Action	Parameters	Request/ Response	Type/ Value	Description
	Username	REQ, RES	<string>	SMTP user name **Username** is only valid if **Authentication** is set as SMTP or POPBeforeSMTP.
	Password	REQ, RES	<string>	SMTP user password **Password** is only valid if **Authentication** is set as SMTP or POPBeforeSMTP.
	IsPasswordEncrypted	REQ	<bool>	Returns true if password sent is encrypted. Encrypted password should be sent as a post message.
	Sender	REQ, RES	<string>	Email address of the sender The sender's email address is specified in the format of <Email Address>.
	Recipient	REQ, RES	<string>	Email address of the recipient The recipient's email address is specified in the format of <Email Address>. Attribute to check max recipients: "**attributes/transfer/Limit/SMTP.MaxRecipients**" CAMERA ONLY
	Subject	REQ, RES	<string>	Email title CAMERA ONLY
	Message	REQ, RES	<string>	Email content CAMERA ONLY
test	status	RES	<string>	Tries to connect to the server with the configured settings and returns the result in the form of a string.

3.4. Examples

3.4.1. Getting SMTP server information

REQUEST

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

TEXT RESPONSE

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

```
Host=192.168.100.1
Port=25
Username=test
Password=1234
Authentication=SMTP
Sender=test.a@samsung.com
Recipient=test1.a@samsung.com
Subject=hello
Message=Test message
Encryption=None
```

JSON RESPONSE

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

```
{
    "Host": "192.168.100.1",
    "Port": 25,
    "Username": "test",
    "Password": "1234",
    "Authentication": "SMTP",
    "Sender": "test.a@samsung.com",
    "Recipient": "test1.a@samsung.com",
    "Subject": "hello",
    "Message": "Test message",
    "Encryption": "None"
```

```
}
```

3.4.2. Setting the SMTP server

The SMTP host, port, user name, password, sender/recipient email addresses, email subject and message can be configured as shown in the command below.

The **Username** and **Password** parameters can be specified only when SMTP server authentication is set as SMTP or POPBeforeSMTP.

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=smtp&action=set&Authentication=SMTP&Host=www.hanwh
asecurity.com&Port=25&Encryption=None&Username=test5678&Password=pw12345&Sen
der=test@test.com&Recipient=stw@hanwhasecurity.com&Subject=test&Message=test
```

The following request example is for NVR only.

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=smtp&action=set&Authentication=SMTP&Host=192.168.1
00.1&Port=25&Encryption=None&Username=aaa&Password=4321&Sender=test@test.com
```

3.4.3. Setting the SMTP server with encrypted password

The SMTP host, port, user name, sender/recipient email addresses, email subject and message can be configured as shown in the command below. Password should be encrypted with RSA and RSA_PKCS1_PADDING (Refer to security.cgi for information on obtaining the RSA key).

Base 64 Encoded data should be sent as a POST message.

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=smtp&action=set&Authentication=SMTP&Host=www.hanwh
asecurity.com&Port=25&Encryption=None&Username=test5678&Sender=test@test.com
&Recipient=stw@hanwhasecurity.com&Subject=test&Message=test&IsPasswordEncryp
ted=True
```

3.4.4. Checking the connection status of the SMTP Server

REQUEST

```
http://<Device IP>/stw-cgi/transfer.cgi?msubmenu=smtp&action=test
```

TEXT RESPONSE

```
HTTP/1.0 200 OK

Content-type: text/plain

<Body>
```

```
Status=Success
```

JSON RESPONSE

```
HTTP/1.0 200 OK

Content-type: application/json

<Body>
```

```
{
"Status": "Success"
}
```

Chapter 4. SMTP Users

4.1. Description

The **smtpusers** submenu configures the SMTP mail users.

> **NOTE**
>
> This chapter applies to NVR only.
> Attribute to check for maximum users: "**attributes/security/Limit/MaxSMTPUser**"
> Attribute to check for maximum users in group: "**attributes/security/Limit/MaxSMTPUserPerGroup**"

Access level

Action	NVR
view	User
add, update	User
remove	User

4.2. Syntax

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

4.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
view	UserIndex	REQ	<int>	User index number
	GroupID	REQ	<string>	Group ID
add, update	UserIndex	REQ, RES	<int>	User index number
	GroupID	REQ, RES	<string>	Group ID
	UserName	REQ, RES	<string>	User name
	Recipient	REQ, RES	<string>	Email address of the recipient. The recipient's email address is specified in the format of <Email Address>.
remove	UserIndex	REQ	<int>	User index number

4.4. Examples

4.4.1. Getting the current SMTP user settings

REQUEST

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

TEXT RESPONSE

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

```
GroupID=Group 1
UserIndex=1
UserName=Recipient 1
Recipient=t1@samsung.com
GroupID=Group 1
UserIndex=2
UserName=s12
Recipient=s2@hanwhasecurity.com
GroupID=Group 1
UserIndex=3
UserName=s3
Recipient=s12@s12.com
```

JSON RESPONSE

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

```
{
    "SMTPUsers": [
        {
            "GroupID": "Group 1",
            "UserIndex": 1,
            "UserName": "Recipient 1",
            "Recipient": "t1@samsung.com"
        },
```

```
        {
            "GroupID": "Group 1",
            "UserIndex": 2,
            "UserName": "s12",
            "Recipient": "s2@hanwhasecurity.com"
        },
        {
            "GroupID": "Group 1",
            "UserIndex": 3,
            "UserName": "s3",
            "Recipient": "s12@hanwhasecurity.com"
        }
    ]
}
```

4.4.2. Getting the ‘User Index 1’ settings

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=smtpusers&action=view&UserIndex=1
```

TEXT RESPONSE

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

```
GroupID=Group 1
UserIndex=1
UserName=Recipient 1
Recipient=t1@samsung.com
```

JSON RESPONSE

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

```
{
```

```
    "SMTPUsers": [
        {
            "GroupID": "Group 1",
            "UserIndex": 1,
            "UserName": "Recipient 1",
            "Recipient": "t1@samsung.com"
        }
    ]
}
```

4.4.3. Adding a new SMTP user

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=smtpusers&action=add&GroupID=Group2&UserName=s12&R
ecipient=s12@s12
```

CAMERA RESPONSE

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

```
OK
UserIndex=13
```

The following response example is for NVR only.

NVR 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"
}
```

4.4.4. Updating the user name

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=smtpusers&action=update&UserIndex=13&UserName=t13
```

4.4.5. Removing the user

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=smtpusers&action=remove&UserIndex=7
```

Chapter 5. SMTP User Group

5.1. Description

The **smtpgroups** submenu configures the SMTP mail user groups.

> **NOTE** This chapter applies to NVR only.
> Attribute to check for maximum groups: "**attributes/security/Limit/MaxSMTPGroup**"

Access level

Action	NVR
view	User
add, update	User
remove	User

5.2. Syntax

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

5.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
view	GroupID	REQ	<string>	User group ID
add, update	GroupID	REQ, RES	<string>	User group ID
	GroupName	REQ, RES	<string>	Group name
remove	GroupID	REQ	<string>	User group ID

5.4. Examples

5.4.1. Getting the current SMTP user group settings

REQUEST

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

TEXT RESPONSE

```
HTTP/1.0 200 OK
```

```
Content-type: text/plain
<Body>
```

```
GroupID=Group 1
GroupName=Group 1
GroupID=Group2
GroupName=Group2
GroupID=Group1
GroupName=Group1
GroupID=Group 3
GroupName=Group 3
```

JSON RESPONSE

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

```
{
    "SMTPGroups": [
        {
            "GroupID": "Group 1",
            "GroupName": "Group 01"
        },
        {
            "GroupID": "Group2",
            "GroupName": "Group2"
        },
        {
            "GroupID": "Group 3",
            "GroupName": "Group 3"
        }
    ]
}
```

5.4.2. Getting the settings of ‘Group 1’

REQUEST

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

```
cgi/transfer.cgi?msubmenu=smtpgroups&action=view&GroupID=Group 1
```

TEXT RESPONSE

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

```
GroupID=Group 1
GroupName=Group 1
```

JSON RESPONSE

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

```
{
    "SMTPGroups": [
        {
            "GroupID": "Group 1",
            "GroupName": "Group 1"
        }
    ]
}
```

5.4.3. Adding an SMTP user group

When adding a group with only **GroupName**, NVR creates a group whose **GroupID** is the same as **GroupName**. If you want to create it separately, send a command with both **GroupID** and **GroupName**.

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=smtpgroups&action=add&GroupName=Group3
```

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=smtpgroups&action=add&GroupID=TestGroup&GroupName=
```

```
Group3
```

5.4.4. Updating the user group name

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=smtpgroups&action=update&GroupID=Group3&GroupName=
Group3
```

5.4.5. Removing the user group

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=smtpgroups&action=remove&GroupID=Group3
```

Chapter 6. Data Server

6.1. Description

The **dataserver** submenu configures the data server.

> **NOTE** This chapter applies to the camera only.

Access level

Action	Camera
view	Admin
set	Admin
test	Admin

6.2. Syntax

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

6.3. Parameters

Action	Parameters	Request/ Response	Type/ Value	Description
view				Reads data server settings
set	Enable	REQ, RES	<bool>	Value that shows if the data server is in active status or not
	IPv4Address	REQ, RES	<string>	IPv4 address of the data server
	Port	REQ, RES	<int>	Data server port number
	Username	REQ, RES	<string>	Data server user name
	Password	REQ, RES	<string>	Data server user password
	IsPasswordEncrypted	REQ, RES	<bool>	Returns true if password sent is encrypted. Encrypted password should be sent as a post message.
	StoreName	REQ, RES	<string>	Description of data server Up to 20 characters allowed.

Action	Parameters	Request/ Response	Type/ Value	Description
test	ConnectionState	RES	<enum> Failed, Connecting, Success	Representing value that shows if the data server is connected or not

6.4. Examples

6.4.1. Getting the current data server settings

REQUEST

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

TEXT RESPONSE

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

```
Enable=False
IPv4Address=192.168.1.1
Port=3434
UserName=admin
Password=
StoreName=testServer
```

JSON RESPONSE

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

```
{
    "Enable": false,
    "IPv4Address": "192.168.1.1",
    "Port": 3434,
    "UserName": "admin",
    "Password": "",
    "StoreName": "testServer"
```

```
}
```

6.4.2. Setting the data server

REQUEST

```
http://<Device IP>/stw-
cgi/transfer.cgi?msubmenu=dataserver&action=set&Enable=True&IPv4Address=192.
168.1.1&Port=3434&UserName=admin&Password=password&StoreName=testServer
```

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>
```

```
{
    "Status": "Success"
}
```

6.4.3. Testing the data server’s settings

REQUEST

```
http://<Device IP>/stw-cgi/transfer.cgi?msubmenu=dataserver&action=test
```

TEXT RESPONSE

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

```
ConnectionState=Connecting
```

JSON RESPONSE

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

```
{
    "ConnectionState": "Connecting"
}
```