User manual

（Onvif Rtsp Server）

Happytimesoft Technology Co., LTD

Declaration

Since the product version upgrade or other reasons, this manual will subsequently be updated. Unless otherwise agreed, this manual only as a guide, this manual all statements, information, recommendations do not constitute any express or implied warranties。

www.happytimesoft.com

Table of Contents

Chapter 1 ONVIF SERVER

1.1 Configuration

1.1.1 Configuration Templates

```xml
<?xml version="1.0" encoding="utf-8"?>
<config>
    <server_ip></server_ip>
    <server_port>8000</server_port>
    <http_max_users>16</http_max_users>
    <https_enable>0</https_enable>
    <need_auth>0</need_auth>
    <log_enable>1</log_enable>
    <log_level>1</log_level>
    <information>
        <Manufacturer>Happytimesoft</Manufacturer>
        <Model>IP Camera</Model>
        <FirmwareVersion>2.4</FirmwareVersion>
        <SerialNumber>123456</SerialNumber>
        <HardwareId>1.0</HardwareId>
    </information>
    <user>
        <username>admin</username>
        <password>admin</password>
        <userlevel>Administrator</userlevel>
    </user>
    <user>
        <username>test</username>
        <password>123456</password>
        <userlevel>User</userlevel>
    </user>
    <profile>
        <video_source>
            <width>1280</width>
            <height>720</height>
        </video_source>
        <video_encoder>
```

```
        <width>160</width>
        <height>128</height>
        <quality>4</quality>
        <session_timeout>10</session_timeout>
        <framerate>25</framerate>
        <encoding_interval>50</encoding_interval>
        <bitrate_limit>2048</bitrate_limit>
        <encoding>H264</encoding>
        <h264>
            <gov_length>50</gov_length>
            <h264_profile>Main</h264_profile>
        </h264>
    </video_encoder>
    <audio_source></audio_source>
    <audio_encoder>
        <session_timeout>10</session_timeout>
        <sample_rate>8</sample_rate>
        <bitrate>64</bitrate>
        <encoding>G711</encoding>
    </audio_encoder>
    <stream_uri></stream_uri>
</profile>
<profile>
    <video_source>
        <width>1280</width>
        <height>720</height>
    </video_source>
    <video_encoder>
        <width>640</width>
        <height>480</height>
        <quality>4</quality>
        <session_timeout>10</session_timeout>
        <framerate>25</framerate>
        <encoding_interval>50</encoding_interval>
        <bitrate_limit>2048</bitrate_limit>
        <encoding>H264</encoding>
        <h264>
            <gov_length>50</gov_length>
```

```xml
                <h264_profile>Main</h264_profile>
            </h264>
        </video_encoder>
        <audio_source></audio_source>
        <audio_encoder>
            <session_timeout>10</session_timeout>
            <sample_rate>8</sample_rate>
            <bitrate>64</bitrate>
            <encoding>G711</encoding>
        </audio_encoder>
        <stream_uri></stream_uri>
    </profile>
    <scope>onvif://www.onvif.org/Profile/Streaming</scope>
    <scope>onvif://www.onvif.org/location/country/china</scope>
    <scope>onvif://www.onvif.org/type/video_encoder</scope>
    <scope>onvif://www.onvif.org/name/IP-Camera</scope>
    <scope>onvif://www.onvif.org/hardware/HI3518C</scope>
    <event>
        <renew_interval>60</renew_interval>
        <simulate_enable>1</simulate_enable>
    </event>
</config>
```

1.1.2 Configuring Node Description

<server_ip>

Specify the IP address onvif server bindings, if not specified, the onvif server will bind to the default routing interface IP address.

<server_port>

Specify the port onvif server binding, providing web service service on this port, the default is 8000.

<http_max_users>

Maximum supported HTTP client

<https_enable>

Indicates whether enable https connection, 0 is disable, 1 enable

<need_auth>

Indicates whether authentication is required, 0 don't require, 1 require.

<log_enable>

Indicates whether logging is enabled, 0 is not enabled, 1 enable.

<log_level>

The log level:

TRACE	0
DEBUG	1
INFO	2
WARN	3
ERROR	4
FATAL	5

<information> : Config the ONVIF device basic information

<Manufacturer>

The manufactor of the device

<Model>

The device model

<FirmwareVersion>

The firmware version in the device

<SerialNumber>

The serial number of the device

<HardwareId>

The hardware ID of the device

<user> : Contains a list of the onvif users, it can configure multiple nodes

<username>

Username string

<password>

Password string

<userlevel>

User level string, The following values can be configured:

Administrator

Operator

User

Anonymous

<profile> : A media profile maps a video and/or audio source to a video and/or an audio encoder, configurations.

<video_source> : If the media profile contains a video, the video source configuration

<width>

The video source width

<height>

The video source height

<video_encoder>: If the media profile contains a video, the video encoder configuration

<width>

Encoded video width

<height>

Encoded video height

<quality>

Relative value for the video quantizers and the quality of the video. A high value within supported quality range means higher quality

<session_timeout>

The rtsp session timeout for the related video stream

<framerate>

Maximum output framerate in fps

<encoding_interval>

Interval at which images are encoded and transmitted. (A value of 1 means that every frame is encoded, a value of 2 means that every 2nd frame is encoded ...)

<bitrate_limit>
The maximum output bitrate in kbps

<encoding>
Used video codec, either JPEG, MPEG4, H264 or H265

<h264>: Configure H.264 related parameters

<gov_length>

Group of Video frames length. Determines typically the interval in which the I-Frames will be coded. An entry of 1 indicates I-Frames are continuously generated. An entry of 2 indicates that every 2nd image is an I-Frame, and 3 only every 3rd frame, etc. The frames in between are coded as P or B Frames

<h264_profile>

The H.264 profile, either Baseline, Main, Extended or High

<h265>: Configure H.265 related parameters

<gov_length>

Group of Video frames length. Determines typically the interval in which the I-Frames will be coded. An entry of 1 indicates I-Frames are continuously generated. An entry of 2 indicates that every 2nd image is an I-Frame, and 3 only every 3rd frame, etc. The frames in between are coded as P or B Frames

<h265_profile>

The H.265 profile, either Main or Main10

<mpeg4>: Configure MPEG4 related parameters

<gov_length>

Determines the interval in which the I-Frames will be coded. An entry of 1 indicates I-Frames are continuously generated. An entry of 2 indicates that every 2nd image is an I-Frame, and 3 only every 3rd frame, etc. The frames in between are coded as P or B Frames

<mpeg4_profile>

The Mpeg4 profile, either simple profile (SP) or advanced simple profile (ASP)

<audio_source> : If the media profile contains a audio, the audio source configuration

<audio_encoder>:If the media profile contains a audio, the audio encoder configuration

<session_timeout>

The rtsp session timeout for the related audio stream

<sample_rate>

The output sample rate in kHz

<bitrate>

The output bitrate in kbps

<encoding>

Audio codec used for encoding the audio input (either G711, G726 or AAC)

<stream_uri>

The profile RTSP stream address, if not specify, it default to **rtsp://yourip//test.264**

<scope>

Contains a list of URI definining the device scopes

<event> : Event Configuration parameters

<renew_interval>

Event renew interval

<simulate_enable>

Specifies whether to generate simulation event

1.2 Configuration file

When running onvif server for the first time, use the default configuration file **onvif.cfg**, which sets 2 profiles.

When stop onvif server, it writes the runtime configuration into the **onvifrun.cfg** file, and the configuration in the onvifrun.cfg file will be load at the next time it runs.

1.3 Compatibility test

1. ONVIF SERVER PROFILE T passed the compatibility test version

Windows version download from:

http://www.happytimesoft.com/downloads/happytime-onvif-rtsp-server-profilet.zip

Linux version download from:

http://www.happytimesoft.com/downloads/happytime-onvif-rtsp-server-profilet.tar.gz

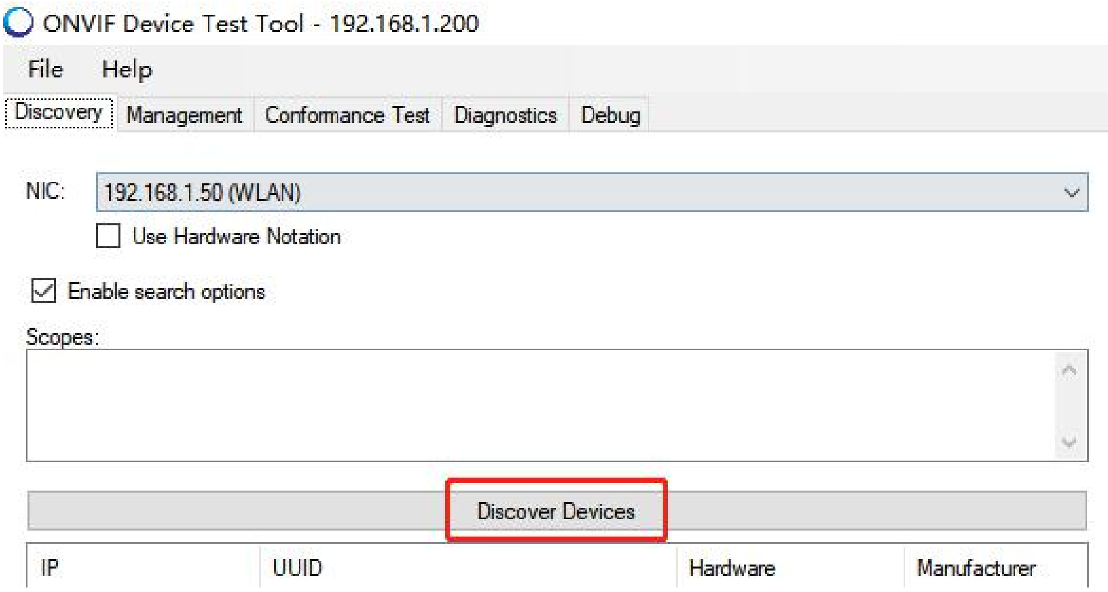
1. Modify the ONVIF SERVER configuration file onvif.cfg and specify the <need_auth> value as 1; Modify the RTSP SERVER configuration file rtsp.cfg and specify the <metadata> value as 1.

2. Run the onvifrtspserver.

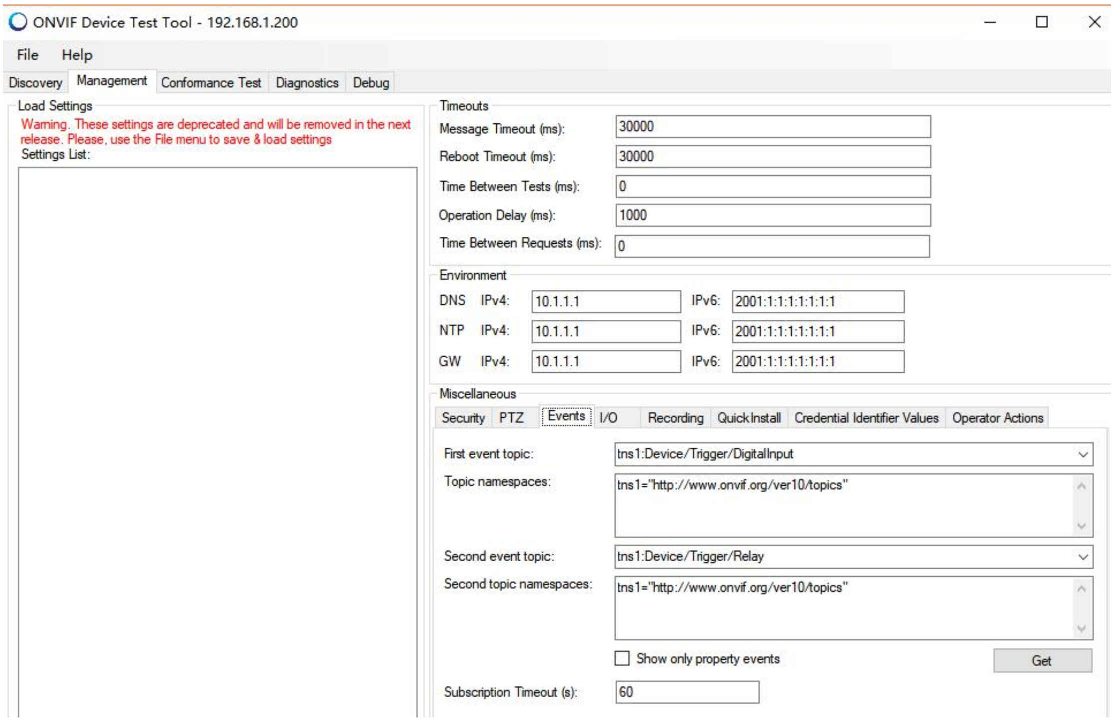
3. Run the ONVIF Device Test Tool.

Note: onvif rtsp server and test tools should run on different computers

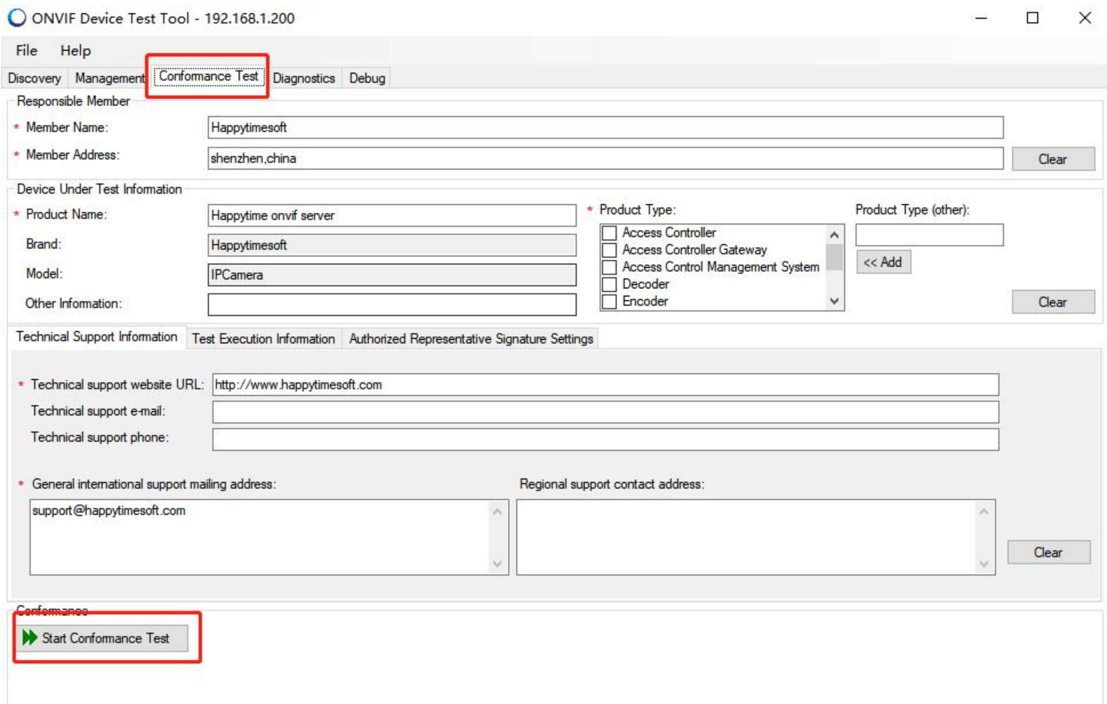
4. Click “Discover Devices” button, as the following:



5. Switch to “Management” tab, select “Events” tab at “miscellaneous”, then click “Get” button, as the following:



6. Switch to “Conformance Test” tab, click “Start Conformance Test” button:



1.4 ONIVF features

The onvif rtsp server supports the onvif features listed in the following table:

Features		
Security	WS-Username Token	
	Digest	

Discovery	BYE Message	
	Types	tds:Device
		dn:Network Video Transmitter
Device Service	Capabilities	GetCapabilities
		GetService
	Network	Zero Configuration
		NTP
		Dynamic DNS
		IP Filter
		HTTPS
	System	System Logging
		HTTP System Logging
		HTTP Firmware Upgrade
		HTTP Support Information
		HTTP System Backup
	Security	Default Access Policy
		Maximum Users
		Remote User Handling
		Maximum Username Length
		Maximum Password Length
	I/O	Relay outputs
Event Service	WS Basic Notification	
	Message Content Filter	ONVIF Message Content Filter Dialect
	Get Service Capabilities	MaxPullPoints capability
	Pull-Point Notification	
Media Service	Video	JPEG
		H.264
		MPEG4
	Audio	G.711
		G.726
		AAC
	Audio Output	G.711
		AAC
	Real-time Streaming	RTP/UDP
		RTP/RTSP/HTTP
		RTP/RTSP/TCP

		RTP-Multicast/UDP
	Snapshot URI	
Media2 Service	Video	H.265
		H.264
	Audio	G.711
		AAC
	Audio outputs	G.711
		AAC
	Real-time Streaming	RTP/UDP
		RTP/RTSP/HTTP
		RTP/RTSP/TCP
		RTP-Multicast/UDP
	RTSP WebSocket	
	Snapshot URI	
	Video Source Mode	
	OSD	
	Analytics	
	Metadata	
	Media2 Events	Media/ProfileChanged
		Media/ConfigurationChanged
PTZ Service	Absolute move	Pan/Tilt movement
		Zoom movement
	Relative move	Pan/Tilt movement
		Zoom movement
	Continuous move	Pan/Tilt movement
		Zoom movement
	Presets	
	Home position	Configuration
	Auxiliary operations	
	Speed	Speed for Pan/Tilt
		Speed for Zoom
	Move Status	
	Status Position	
	Get Compatible Configurations	
Device IO Service	Relay outputs	Bistable Mode
		MonoStable Mode
	Digital Inputs	Digital Input Options

Imaging Service	IrCutfilter Configuration	
	Tampering Events	Image Too Blurry
		Image Too Dark
		Image Too Bright
		Global Scene Change
	Motion Alarm	
	Focus Control	
Analytics Service	Rule Engine	Rule Options
		Motion Region Detector Rule
	Analytics Modules	Analytics Module Options
Recording Control Service	Dynamic Recordings	
	Dynamic Tracks	
	Audio Recording	
	Recording Options	
	tns1:RecordingCofig/DeleteTrackData	
	Metadata Recording	
	Encoding	JPEG
		H264
		MPEG4
Recording Search Service	Metadata Search	
	PTZ Position Search	
Door Control Service	Door Entity	Access Door
		Lock Door
		Double Lock Door
		Block Door
		Lock Down Door
		Lock Open Door
		Door Monitor
		Double Lock Monitor
		Alarm
		Tamper
		Fault
	Door Control Events	
	Door Management	
	Client Supplied Token	
Access Control Service	Area Entity	

	Access Point Entity	Enable/Disable Access Point
		Duress
		Access Taken
		Anonymous Access
	Access Point Management	
	Area Management	
	Access Control Events	
Replay Service	RTP/RTSP/TCP	
Receiver Service		
Credential Service	Credential Validity	
	Credential Access Profile Validity	
	pt:Card	
	pt:PIN	
	pt:Fingerprint	
	Reset Antipassback Violation	
	Client Supplied Token	
	Whitelist	
	Blacklist	
	Validity Supports Time Value	
Access Rules Service	Multiple Schedules Access Point	
	Client Supplied Token	
Schedule Service		
Thermal Service		

1.5 ONVIF Version

The onvif rtsp server implements the following ONVIF service:

ONVIF Service	Prefix	Url	version
device	tds	http://www.onvif.org/ver10/device/wsdl	20.06
event	tev	http://www.onvif.org/ver10/events/wsdl	20.06
media	trt	http://www.onvif.org/ver10/media/wsdl	20.06
media 2	tr2	http://www.onvif.org/ver20/media/wsdl	20.06
ptz	tptz	http://www.onvif.org/ver20/ptz/wsdl	18.12
image	timg	http://www.onvif.org/ver20/imaging/wsdl	19.06
analytics	tan	http://www.onvif.org/ver20/analytics/wsdl	20.06
recording control	trc	http://www.onvif.org/ver10/recording/wsdl	19.06

search	tse	http://www.onvif.org/ver10/search/wsdl	18.12
replay	trp	http://www.onvif.org/ver10/replay/wsdl	18.06
access control	tac	http://www.onvif.org/ver10/accesscontrol/wsdl	20.06
door control	tdc	http://www.onvif.org/ver10/doorcontrol/wsdl	19.12
device IO	tmd	http://www.onvif.org/ver10/deviceIO/wsdl	19.12
thermal	tth	http://www.onvif.org/ver10/thermal/wsdl	17.06
credential	tcr	http://www.onvif.org/ver10/credential/wsdl	19.12
access rules	tar	http://www.onvif.org/ver10/accessrules/wsdl	19.06
schedule	tsc	http://www.onvif.org/ver10/schedule/wsdl	18.12
receiver	trv	http://www.onvif.org/ver10/receiver/wsdl	18.12
provisioning	tpv	http://www.onvif.org/ver10/provisioning/wsdl	18.12

1.6 Supports multiple channels

The onvif rtsp server supports multi channel by modifying configuration file (onvif.cfg).

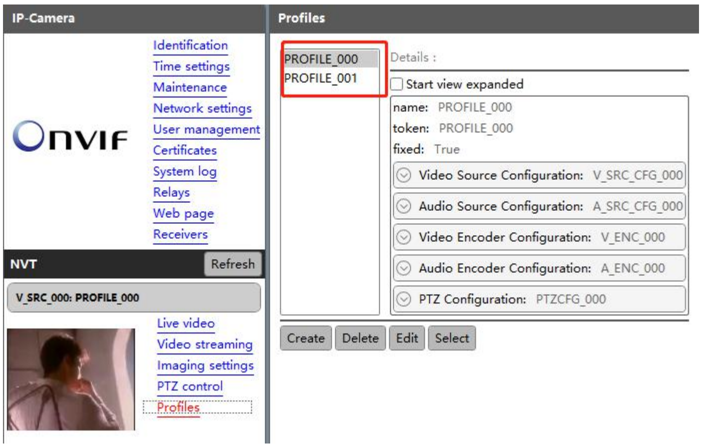
Each <profile> tag represents a channel in the configuration file.

The default configuration file supports 2 channels, you can add <profile> tag to support more channels.

Note : If <video_source>.width and <video_source>.height of multiple <profile> tags are the same, example:

```
<profile>
   <video_source>
      <width>1280</width>
      <height>720</height>
   </video_source>
....
</profile>

<profile>
   <video_source>
      <width>1280</width>
      <height>720</height>
   </video_source>
....
</profile>
```

The onvif device manager will show the profiles as the following:



If <video_source>.width and <video_source>.height of multiple <profile> tags are not the same, example:

```
<profile>
    <video_source>
        <width>1280</width>
        <height>720</height>
    </video_source>
....
</profile>

<profile>
    <video_source>
        <width>640</width>
        <height>480</height>
    </video_source>
....
</profile>
```

The onvif device manager will show the profiles as the following:

1.7 Modify RTSP stream address

If the value of <stream_uri> in the <profile> tag in the onvif server configuration file is not modified, the RTSP stream address provided by the onvif server by default is rtsp://ip/test.mp4, and the user can modify the <stream_uri> in <profile> tag to specify the rtsp stream address provided by the onvif server. such as:

<profile>

...

<stream_uri>rtsp://192.168.3.27/live</stream_uri>

</profile>

Chapter 2 RTSP SERVER

2.1 Introduction

Happytime RTSP Server is a complete RTSP server application. It can stream audio and video files in various formats.

It can also stream video from camera and live screen, stream audio from audio device.

It can stream H265, H264, MP4, MJPEG video stream and G711, G722, G726, AAC, OPUS audio stream.

These streams can be received/played by standards-compliant RTSP/RTP media clients.

It support rtsp proxy function.

It support audio back channel function.

It support rtsp over http function.

It support rtp multicast function.

Support for data pusher function.

Enjoying multimedia content from your computer can be a pleasant way for you to spend your free time. However, sometimes you might need to access it from various locations, such as a different computer or a handheld device, Happytime RTSP Server, that can help you achieve quick and efficient results.

2.2 Key features

The server can transmit multiple streams concurrently

It can stream audio and video files in various formats

It can stream audio from audio device

It can stream video from camera and live screen

It can stream H265, H264, MP4, MJPEG video stream

It can stream G711, G722, G726, AAC, OPUS audio stream

It supports rtsp over http function

It supports rtsp over websocket function

It supports rtp multicast function

It supports data pusher function

It supports RTSP proxy function, as the following:

Support Audio Backchannel

Happytime rtsp server comply with ONVIF backchannel specification, the url is :

https://www.onvif.org/specs/stream/ONVIF-Streaming-Spec-v1706.pdf

5.3.2.1 Example 1: Server without backchannel support:

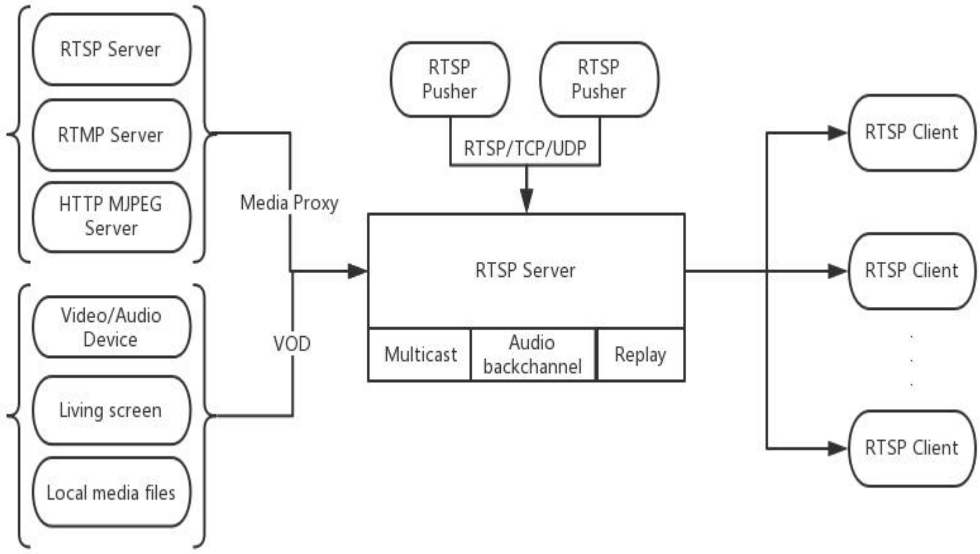
```
Client - Server:            DESCRIBE rtsp://192.168.0.1 RTSP/1.0
                            Cseq: 1
                            User-Agent: ONVIF Rtsp client
                            Accept: application/sdp
                            Require: www.onvif.org/ver20/backchannel

Server - Client:            RTSP/1.0 551 Option not supported
                            Cseq: 1
                            Unsupported: www.onvif.org/ver20/backchannel
```

5.3.2.2 Example 2: Server with Onvif backchannel support:

```
Client - Server:            DESCRIBE rtsp://192.168.0.1 RTSP/1.0
                            Cseq: 1
                            User-Agent: ONVIF Rtsp client
                            Accept: application/sdp
                            Require: www.onvif.org/ver20/backchannel

Server - Client:            RTSP/1.0 200 OK
                            Cseq: 1
                            Content-Type: application/sdp
                            Content-Length: xxx

                            v=0
                            o= 2890842807 IN IP4 192.168.0.1
                            s=RTSP Session with audiobackchannel
                            m=video 0 RTP/AVP 26
                            a=control:rtsp://192.168.0.1/video
                            a=recvonly
                            m=audio 0 RTP/AVP 0
                            a=control:rtsp://192.168.0.1/audio
                            a=recvonly
                            m=audio 0 RTP/AVP 0
                            a=control:rtsp://192.168.0.1/audioback
                            a=rtpmap:0 PCMU/8000
                            a=sendonly
```

```
Client - Server:          SETUP rtsp://192.168.0.1/video RTSP/1.0
                          Cseq: 2
                          Transport: RTP/AVP;unicast;client_port=4588-4589

Server - Client:    RTSP/1.0 200 OK
                          Cseq: 2
                          Session: 123124;timeout=60
                          Transport:RTP/AVP;unicast;client_port=4588-4589;
                          server_port=6256-6257

Client - Server:          SETUP rtsp://192.168.0.1/audio RTSP/1.0
                          Cseq: 3
                          Session: 123124
                          Transport: RTP/AVP;unicast;client_port=4578-4579

Server - Client:          RTSP/1.0 200 OK
                          Cseq: 3
                          Session: 123124;timeout=60
                          Transport:RTP/AVP;unicast;client_port=4578-4579;
                          server_port=6276-6277

Client - Server:          SETUP rtsp://192.168.0.1/audioback RTSP/1.0
                          Cseq: 4
                          Session: 123124
                          Transport: RTP/AVP;unicast;client_port=6296-6297
                          Require: www.onvif.org/ver20/backchannel

Server - Client:          RTSP/1.0 200 OK
                          Cseq: 4
                          Session: 123124;timeout=60
                          Transport:RTP/AVP;unicast;client_port=6296-6297;
                          server_port=2346-2347


Client - Server:          PLAY rtsp://192.168.0.1 RTSP/1.0
                          Cseq: 5
                          Session: 123124
                          Require: www.onvif.org/ver20/backchannel

Server - Client:          RTSP/1.0 200 OK
                          Cseq: 5
                          Session: 123124;timeout=60
```

2.3 Function chart



2.4 Configuration

2.4.1 Configuration Templates

```
<?xml version="1.0" encoding="utf-8"?>
<config>
	<serverip></serverip>
	<serverip></serverip>
	<serverport>554</serverport>
	<loop_nums>1</loop_nums>
	<multicast>0</multicast>
	<metadata>1</metadata>
	<rtsp_over_http>1</rtsp_over_http>
	<http_port>8080</http_port>
	<need_auth>0</need_auth>
	<log_enable>1</log_enable>
	<log_level>1</log_level>

	<user>
		<username>admin</username>
		<password>123456</password>
	</user>
	<user>
		<username>user</username>
```

```
		<password>123456</password>
	</user>

	<output>
		<url>screenlive</url>
		<video>
			<codec>H264</codec>
			<width></width>
			<height></height>
			<framerate></framerate>
			<bitrate></bitrate>
		</video>
		<audio>
			<codec>G711U</codec>
			<samplerate>8000</samplerate>
			<channels>1</channels>
			<bitrate></bitrate>
		</audio>
	</output>

	<output>
		<url></url>
		<video>
			<codec>H264</codec>
			<width></width>
			<height></height>
			<framerate></framerate>
			<bitrate></bitrate>
		</video>
		<audio>
			<codec>G711U</codec>
			<samplerate></samplerate>
			<channels></channels>
			<bitrate></bitrate>
		</audio>
	</output>

	<proxy>
```

```xml
        <suffix>proxy</suffix>
        <url></url>
        <user></user>
        <pass></pass>
        <transfer>TCP</transfer>
        <output>
            <video>
                <codec>H264</codec>
                <width>640</width>
                <height>480</height>
                <framerate>25</framerate>
                <bitrate></bitrate>
            </video>
            <audio>
                <codec>AAC</codec>
                <samplerate>32000</samplerate>
                <channels>2</channels>
                <bitrate></bitrate>
            </audio>
        </output>
    </proxy>

    <pusher>
        <suffix>pusher</suffix>
        <video>
            <codec>H264</codec>
        </video>
        <audio>
            <codec>G711U</codec>
            <samplerate>8000</samplerate>
            <channels>1</channels>
        </audio>
        <transfer>
            <mode>UDP</mode>
            <ip></ip>
            <vport>50001</vport>
            <aport>50002</aport>
        </transfer>
```

<output>
<video>
<codec>H264</codec>
<width>640</width>
<height>480</height>
<framerate>25</framerate>
<bitrate></bitrate>
</video>
<audio>
<codec>AAC</codec>
<samplerate>32000</samplerate>
<channels>2</channels>
<bitrate></bitrate>
</audio>
</output>
</pusher>

<backchannel>
<codec>G711U</codec>
<samplerate>8000</samplerate>
<channels>1</channels>
</backchannel>
</config>

2.5 Configuring Node Description

2.5.1 System parameters

<serverip>

Specify the IP address RTSP server bindings, if not specified, the RTSP server will bind to the default routing interface IP address.

Note: This node can configure multiple instances,meaning that the server can bind multiple IP addresses or domain names.

<serverport>

Specify the port RTSP server binding, the default is 554.

<loop_nums>

When streaming video files, specify the number of loop playback,-1 means infinite loop.

<multicast>

Whether to enable rtp multicast function, 0-disable, 1-enable.

<metadata>

Whether to enable the meta data stream, 0-disable, 1-enable.

<rtsp_over_http>

Whether to enable rtsp over http function, 0-disable,1-enable.

<http_port>

Specify the HTTP service port for rtsp over http function.

<need_auth>

Whether enable the user authentication function,0-disable,1-enable

<log_enable>

Whether enable the log function,0-disable,1-enable

<log_level>

The log level:

TRACE	0
DEBUG	1
INFO	2
WARN	3
ERROR	4
FATAL	5

2.5.2 *User* node

<user> : Specify the login username password, it can configure multiple nodes

<username>

The login username

<password>

The login password

2.5.3 *Output* node

<output> : Specify the audio and video output parameters, it can configure multiple nodes

<url>

Match URL address, it can be filename, or file extension name. Such as:

screenlive : match live screen stream

videodevice : match camera video stream

*.mp4 : match all mp4 media file

sample.flv : match sample.flv file

If not config this node, it will match all url as the audio/video default output parameters.

The match order from top to bottom, therefore the default output configuration should be placed in the last.

<video> : Specify the video output parameters

<codec>

Specify the video stream codec, it can specify the following value:

H264 : output H264 video stream

H265 : output H265 video stream

MP4: output MP4 video stream

JPEG: output MJPEG video stream

<width>

Specify the output video width, If 0 use the original video width (live screen stream use the screen width, camera stream use the default width)

<height>

Specify the output video height, If 0 use the original video height (live screen stream use the screen height, camera stream use the default height)

<framerate>

Specify the output video framerate, If 0 use the original video framerate (live screen use the default value 15, camera stream use the default value 25)

<bitrate>

Specify the output video bit rate, if 0, automatically calculate the output bit rate, the unit is kb/s.

Note: This parameter is valid only if encoding is required (eg screenlive, videodevice) or if transcoding is required.

<audio> : Specify the audio output parameters

<codec>

Specify the audio stream codec, it can specify the following value:

G711A: output G711 a-law audio stream

G711U: output G711 mu-law audio stream

G722: output G726 audio stream

G726: output G726 audio stream

AAC: output AAC audio stream

OPUS: output OPUS audio stream

<samplerate>

Specify the audio sample rate, it can specify the following values:

8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100, 48000

If 0 use the original audio sample rate (audio device stream use the default value 8000)

<channels>

Specify the audio channel number, 1 is mono, 2 is stereo

If 0 use the original audio channel number (audio device stream use the default value 2)

Note : G726 only support mono.

<bitrate>

Specify the output audio bit rate, if 0, automatically calculate the output bit rate, the unit is kb/s.

Note: This parameter is valid only if encoding is required (such as screenlive, videodevice) or if transcoding is required.

2.5.4 *Proxy* node

<proxy> : Specify the rtsp proxy parameters, it can configure multiple nodes

<suffix>

Specify the rtsp stream suffix, you can play the proxy stream from:

rtsp://youip/suffix

<url>

The original rtsp/rtmp/http mjpeg stream address.

<user> <pass>

Specify the original rtsp/rtmp/http mjpeg stream address login user and password information

<transfer>

Specify the rtsp client transfer protocol:

TCP: rtsp client uses RTP over TCP

UDP: rtsp client uses RTP over UDP

MULTICAST: rtsp client uses multicast

<output>

Specify the stream output parameter. If the parameter does not appear, use the parameters of the original RTSP stream. If it appears and the configured parameters are inconsistent with the parameters of the original RTSP stream, then the transcode output is performed.

The child nodes under this node are consistent with the meaning of the <output> node.

2.5.5 *Pusher* node

<pusher> : Specify the data pusher parameters, it can configure multiple nodes

<suffix>

Specify the rtsp stream suffix, you can play the pusher stream from:

rtsp://youip/suffix

<video> : Specify the the input video data parameters

<codec>

Specify the video codec, it can specify the following value:

H264 : H264 video stream

H265 : H265 video stream

MP4: MP4 video stream

JPEG: MJPEG video stream

<audio> : Specify the input audio data parameters

<codec>

Specify the audio codec, it can specify the following value:

G711A: G711 a-law audio stream

G711U: G711 mu-law audio stream

G722: G726 audio stream

G726: G726 audio stream

OPUS: OPUS audio stream

AAC: AAC audio stream

<samplerate>

Specify the audio sample rate, it can specify the following values:

8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100, 48000

<channels>

Specify the audio channel number, 1 is mono, 2 is stereo

Note : G726 only support mono.

<transfer>： Specify the data transfer parameters

<mode>：Specify the data transer protocol, it can specify the following value:

TCP: use TCP connection to transfer the data

UDP: use UDP connection to transfer the data

RTSP: use RTSP connection to transfer the data, it support FFMPEG rtsp pusher.

<ip>: Specified data receiving IP address, if there is no configuration, the default IP address is used (valid in TCP or UDP mode).

<vport>: Specify the video data receiving port (valid in TCP or UDP mode)

<aport>: Specify the audio data receiving port (valid in TCP or UDP mode)

<output>

Specify the stream output parameter. If the parameter does not appear, use the parameters of the original pusher stream. If it appears and the configured parameters are inconsistent with the parameters of the original pusher stream, then the transcode output is performed.

The child nodes under this node are consistent with the meaning of the <output> node.

2.5.6 *Backchannel* node

<backchannel> : specify the audio back channel parameters

<codec>

Specify the audio back channel stream codec, it can specify the following value:

G711A: G711 a-law audio stream

G711U: G711 mu-law audio stream

G722: G726 audio stream

G726: G726 audio stream

OPUS: OPUS audio stream

<samplerate>

Specify the audio back channel sample rate, it can specify the following values:

8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100, 48000

If 0 use the default value 8000

<channels>

Specify the audio channel number, 1 is mono, 2 is stereo

If 0 use the default value 1

Note : G726 only support mono.

2.6 Data pusher

Data pusher means that RTSP server receives external data sources and then sends them out as RTSP streams.

The data pusher support TCP, UDP and RTSP mode.

Audio and video data are packaged and sent in RTP format.

If it is TCP mode, you need to add 4 bytes in front of the RTP header, as the following:

```
typedef struct
{
    uint32    magic      : 8;
    uint32    channel    : 8;
    uint32    rtp_len    : 16;
} RILF;
```

magic: 0x24

channel: 0

ttp_len: the RTP load length, including RTP header,

You can download the examples of sending H264 data from the following link:

http://happytimesoft.com/downloads/happytime-rtsp-h264-data-pusher-example.zip

Note: If you use TCP or UDP mode data push, you need to add <pusher>tag in the rtsp server configuration file, specify the push audio and video parameters and push port, etc.

If you use RTSP mode to push data, no configuration is required. The url suffix of the pushed RTSP address can be any legal string.

If it is RTSP mode, it supports standard RTSP push stream, such as FFMPEG rtsp pusher.

FFMPEG rtsp over UDP:

ffmpeg -re -i test.mp4 -vcodec libx264 -acodec copy -preset ultrafast -f rtsp rtsp://yourip/pusher

FFMPEG rtsp over TCP:

ffmpeg -re -i test.mp4 -vcodec libx264 -acodec copy -preset ultrafast -f rtsp -rtsp_transport tcp rtsp://yourip/pusher

Examples of protocols pushed by RTSP are as follows:

C->S:

OPTIONS rtsp://192.168.3.27/mypusher RTSP/1.0
CSeq: 1
User-Agent: happytimesoft rtsp client

S->C:
RTSP/1.0 200 OK
Server: happytime rtsp server V5.0
CSeq: 1
Date: Tue, Sep 15 2020 00:45:17 GMT
Public: DESCRIBE, SETUP, PLAY, PAUSE, OPTIONS, TEARDOWN, GET_PARAMETER, SET_PARAMETER, ANNOUNCE, RECORD

C->S:
ANNOUNCE rtsp://192.168.3.27/mypusher RTSP/1.0
CSeq: 2
User-Agent: happytimesoft rtsp client
Content-type: application/sdp
Content-Length: 476

v=0
o=- 0 0 IN IP4 192.168.3.27
s=session
c=IN IP4 192.168.3.27
t=0 0
a=control:*
m=video 0 RTP/AVP 96
a=rtpmap:96 H264/90000
a=fmtp:96 packetization-mode=1;profile-level-id=000015;sprop-parameter-sets=Z2QAFaw07AoDmwEQAAADABAAAAMDCPFi04A=,aO+8sA==
a=control:realvideo
m=audio 0 RTP/AVP 97
a=rtpmap:97 MPEG4-GENERIC/8000/2
a=fmtp:97 streamtype=5;profile-level-id=1;mode=AAC-hbr;sizelength=13;indexlength=3;indexdeltalength=3;config=159056E500
a=control:realaudio

S->C:

RTSP/1.0 200 OK
Server: happytime rtsp server V5.0
CSeq: 2
Date: Tue, Sep 15 2020 00:45:17 GMT

C->S:
SETUP rtsp://192.168.3.27/mypusher/realvideo RTSP/1.0
CSeq: 3
Transport: RTP/AVP/TCP;unicast;interleaved=0-1
User-Agent: happytimesoft rtsp client

S->C:
RTSP/1.0 200 OK
Server: happytime rtsp server V5.0
CSeq: 3
Date: Tue, Sep 15 2020 00:45:17 GMT
Session: 41
Transport: RTP/AVP/TCP;unicast;interleaved=0-1

C->S:
SETUP rtsp://192.168.3.27/mypusher/realaudio RTSP/1.0
CSeq: 4
Session: 41
Transport: RTP/AVP/TCP;unicast;interleaved=2-3
User-Agent: happytimesoft rtsp client

S->C:
RTSP/1.0 200 OK
Server: happytime rtsp server V5.0
CSeq: 4
Date: Tue, Sep 15 2020 00:45:17 GMT
Session: 41
Transport: RTP/AVP/TCP;unicast;interleaved=2-3

C->S:
RECORD rtsp://192.168.3.27/mypusher RTSP/1.0
CSeq: 5
Session: 41

Range: npt=0.0-
User-Agent: happytimesoft rtsp client

S->C:
RTSP/1.0 200 OK
Server: happytime rtsp server V5.0
CSeq: 5
Date: Tue, Sep 15 2020 00:45:17 GMT
Session: 41

2.7 RTSP over HTTP

The key of RTSP over HTTP is to allow RTSP packets to communicate via HTTP port.

We know that the standard port of RTSP is 554, but due to various security policy configurations such as firewalls, there may be restrictions when the client accesses port 554, which prevents the normal transmission of RTSP packets.

But the HTTP port (port 80) is generally open, so there is the idea of letting RTSP packets pass through port 80, namely RTSP over HTTP

The details of RTSP over HTTP are as follows:

First, the client opens two socket connect to the rtsp server HTTP ports. We call these two sockets "data socket" and "command socket".

Step 1. The client sends an HTTP GET command through the "data socket" to request an RTSP connection.

Step 2. The server responds to the HTTP GET command through the "data socket" and responds with success/failure.

Step 3. The client creates a "command socket" and sends an HTTP POST command through the "command socket" to establish an RTSP session.

At this point, the auxiliary function of HTTP is completed, and the server does not return the client's HTTP POST command. Next is the standard process of RTSP on the HTTP port, but it needs to be completed through two sockets. The "command socket" is only responsible for sending, and the "data socket" is only responsible for receiving.

Step 4. The client sends RTSP commands (BASE64 encoding) through the "command socket".

Step 5. The server responds to the RTSP command (in plain text) through the "data socket".

Step 6. Repeat Step4-Step5 until the client sends the RTSP PLAY command and the server responds to the RTSP PLAY command.Step 6. Repeat Step4-Step5 until the client sends the RTSP PLAY command and the server responds to the RTSP PLAY command.

Step 7. The server transmits audio and video data to the client through the "data socket"

After the data exchange is complete...

Step 8. The client sends the RTSP TEARDOWN command (BASE64 encoding and) through the "command socket"

Step 9. The server responds to the RTSP TEARDOWN command (in plain text) through the "data socket".

Step 10. Close the two sockets.

2.8 RTSP over Websocket

First establish an HTTP connection, and then upgrade to the websocket protocol, RTSP over websocket protocol upgrade process:

```
C-->S:
GET /websocket HTTP/1.1
Host: 192.168.3.27
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: KSO+hOFs1q5SkEnx8bvp6w==
Origin: http://192.168.3.27
Sec-WebSocket-Protocol: rtsp.onvif.org
Sec-WebSocket-Version: 13
```

S-->C:

HTTP/1.1 101 Switching Protocols

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Accept: G/cEt4HtsYEnP0MnSVkKRk459gM=

Sec-WebSocket-Protocol: rtsp.onvif.org

Sec-WebSocket-Version: 13

After the protocol upgrade is successful, perform normal rtsp protocol exchange, and send and receive data through websocket connection.

2.9 Multiple capture devices support

1. If your system have multiple audio capture device, you can use rtsp://yourip:port/audiodevice*N, the N to specify the audio capture device index, start from 0, such as:*

rtsp://192.168.0.100/audiodevice ; stream audio from the first audio device

rtsp://192.168.0.100/audiodevice1 ; stream audio from the second audio device

2. If your system have multiple video capture device, you can use rtsp://yourip:port/videodevice*N, the N to specify the video capture device index, start from 0, such as:*

rtsp://192.168.0.100/videodevice ; stream video from the first video device

rtsp://192.168.0.100/videodevice1 ; stream video from the second video device

3. If your system have multiple monitors, you can use rtsp://yourip:port/screenlive*N, the N to specify the monitor index, start from 0, such as:*

rtsp://192.168.0.100/screenlive ; stream living screen from the first monitor

rtsp://192.168.0.100/screenlive1 ; stream living screen the second monitor

Chapter 3 Run ONVIF RTSP Server

The server is a console application.

Windows: to run the server, simply type "onvifrtspserver".

Linux: to run the server, type "./start.sh", on linux platform, the server run as deamon by default.

Note : The demo version supports up to 4 concurrent streams.

The release version supports up to 100 concurrent streams