User manual

（RTSP Pusher）

HAPPYTIME SOFTWARE CO., LIMITED

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 Introduction 4

Chapter 2 Key features 5

Chapter 3 **Configuration** 6

3.1 Configuration Templates 6

3.2 Configuring Node Description 7

3.2.1 System parameters 7

3.2.2 Pusher node 7

Chapter 4 **Run RTSP Pusher** 13

Chapter 1 Introduction

Happytime RTSP pusher is a rtsp streaming push program that supports RTP OVER UDP, RTP OVER TCP, RTP OVER RTSP three push modes, supports streaming from living screen, camera, application windows, local audio and video files and rtsp/rtmp/srt(Secure Reliable Transport)/http mjpeg streeam, supports video format MPEG4, MJPEG, H264 and H265 and audio format G711, G722, G726, OPUS, AAC.

The audio and video parameters pushed can be set through configuration files, such as video resolution, frame rate, audio sampling rate, number of channels, etc., to support simultaneous push of multiple streams. Stable and reliable, low resource consumption.

It supports various platforms such as Windows, Linux, MAC, ARM, Android, and iOS, supports cross compilation.

It provides the live audio/video stub class, just need to implement a few interfaces to push the live audio/video to rtsp server.

Chapter 2 Key features

Support a variety of audio,video files and image files

Support push video from camera and living screen

Support push video from application windows

Support push audio from audio device

Support recording system audio on Windows

Support push data from RTSP/RTMP/SRT/HTTP MJPEG media streams

Support video codec H264,H265,MPEG4,MJPEG

Support audio codec AAC,G711A,G711U,G722,G726,OPUS

Support automatic transcoding function

Support RTMP/SRT stream to RTSP stream

Support push media data from RTSP stream

Support for configuring audio and video output parameters

Small size, suitable for embedded development

Code structure clear, easy to use

Chapter 3 Configuration

If no configuration file is specified at startup, the default configuration file rtsppusher.cfg will be used.

3.1 Configuration Templates

```
<?xml version="1.0" encoding="utf-8"?>
<config>
    <log_enable>1</log_enable>
    <log_level>1</log_level>
    <loop_nums>-1</loop_nums>
    <reconn_interval>15</reconn_interval>

    <pusher>
        <src>test.mp4</src>
        <transfer>
            <mode>udp</mode>
            <rtspurl>rtsp://192.168.1.50/myapp/live</rtspurl>
            <user></user>
            <pass></pass>
            <ip>192.168.1.50</ip>
            <vport>50003</vport>
            <aport>50004</aport>
        </transfer>
        <video>
            <codec>H264</codec>
            <width></width>
            <height></height>
            <framerate></framerate>
            <bitrate></bitrate>
        </video>
        <audio>
            <codec>G711A</codec>
            <samplerate>8000</samplerate>
```

```
                <channels>1</channels>
                <bitrate></bitrate>
            </audio>
            <metadata>0</metadata>
            <srtp>0</srtp>
        </pusher>
    </config>
```

3.2 Configuring Node Description

3.2.1 System parameters

<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

<loop_nums>

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

<reconn_interval>

When using RTSP push mode, reconnect interval after disconnection.

3.2.2 *Pusher* node

<*Pusher*> : Represents a push stream, specify the audio and video output parameters, it can configure multiple nodes.

<src>

The push stream source, it supports the following parameters:

filename: local media file name, the media files need to be placed in the directory where rtsppusher is located

rtsp, rtmp, srt or http mjpeg url: the original rtsp/rtmp/srt/http mjpeg stream address, such as

rtsp://user:pass@ip:port/url or rtmp://user:pass@ip:port/app/stream.

screenlive : stream the living screen

videodevice : stream from the camera

audiodevice: stream from the audio device

screenlive+audiodevice: stream the living screen and stream audio from the audio device

videodevice+audiodevice: stream video from the video device and stream audio from the audio device

window=[window title]: stream video from application windows

If your system have multiple audio capture device, you can use

audiodevice*N, the N to specify the audio capture device index, start from 0, such as:*

audiodevice ; stream audio from the first audio device

audiodevice1 ; stream audio from the second audio device

If your system have multiple video capture device, you can use

videodevice*N, the N to specify the video capture device index, start from 0, such as:*

videodevice ; stream video from the first video device

videodevice1 ; stream video from the second video device

If your system have multiple monitors, you can use

screenliveN, *the N to specify the monitor index, start from 0, such as:*

screenlive ; stream living screen from the first monitor

screenlive1 ; stream living screen from the second monitor

The audio index or video index represents which device can run ***rtsppusher***

-l device to view.

videodevice or audiodevice can also specify the device name, such as videodevice=testvideo

Run the ***rtsppusher -l device*** command to get the device name.

Note that there can be no spaces in the device name, if the device name contains spaces, you need to use %20 instead of spaces.

If the device name is "FaceTime HD Camera (Built-in)":

videodevice=FaceTime%20HD%20Camera%20(Built-in)

Captures the window with the specified window title, run the ***rtsppusher -l window*** command to view valid window titles

Note that there can be no spaces in the window title, if the window tilte contains spaces, you need to use %20 instead of spaces.

If the window title is "VLC media player":

window=VLC%20media%20player

<transfer> : Specify the transfer parameters

<mode>

Specify transfer mode, Support the following value:

UDP: Transfer using UDP protocol, it need to configure the <ip>, <vport> and <aport> parameters.

TCP: Transfer using TCP protocol, it need to configure the <ip>, <vport> and <aport> parameters.

RTSP: Transfer using RTSP protocol, use RTP over TCP for media transmission, it need to configure the <rtspurl>, <user> and <pass> parameters.

<rtspurl>

Pushed RTSP destination address

<user>

RTSP destination address authentication username

<pass>

RTSP destination address authentication password

<ip>

Specify the server IP address if using TCP or UDP mode.

<vport>

Specify the video data receiving port if using TCP or UDP mode

<aport>

Specify the audio data receiving port if using TCP or UDP mode

<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 it use the original video width (living screen use the screen width, camera use the default width)

<height>

Specify the output video height, if 0 it use the original video height (living screen use the screen height, camera use the default height)

<framerate>

Specify the output video framerate, if 0 it use the original video framerate (living screen and camera use the default value 25)

<bitrate>

Specify the output video bit rate, if it is 0, the bit rate is automatically calculated.

<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 it use the original audio sample rate (audio device use the default value 8000)

<channels>

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

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

<bitrate>

Specify the output audio bit rate, if it is 0, the bitrate is automatically calculated.

Note : If the data source specified by <src> has no audio, but the <audio> tag is configured, output silent audio data.

If the data source specified by <src> has audio, but no <audio> tag is configured, no audio data will be output.

<metadata> : Whether to push metadata stream data.

Note : By default, send the following test metadata stream data:

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
 <tt:Event>
     <wsnt:NotificationMessage xmlns:wsnt=\"http://docs.oasis-open.org/wsn/b-2\">
     <wsnt:Topic
Dialect="http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet">
             tns1:Device/Trigger/DigitalInput
         </wsnt:Topic>
          <wsnt:Message>
              <tt:Message PropertyOperation="Changed" UtcTime="2022-6-12T16:15:08">
              <tt:Source>
```

```
                <tt:SimpleItem Name="InputToken" Value="DIGIT_INPUT_000" />
            </tt:Source>
            <tt:Data>
                <tt:SimpleItem Name="LogicalState" Value="true" />
            </tt:Data>
            </tt:Message>
        </wsnt:Message>
    </wsnt:NotificationMessage>
</tt:Event>
</tt:MetadataStream>
```

<srtp>

Encrypt data transmission using SRTP (security RTP), 0-disable, 1-enable. It is valid when the mode is RTSP.

Chapter 4 Run RTSP Pusher

The rtsp pusher is a console application.

Windows: to run the rtsp pusher, simply type "rtsppusher".

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

rtsp pusher supports the following command line options:

-c config specify the configuration file

-c option specifies the configuration file,if not specified, the default configuration rtsppusher.cfg is used.

-l [device|videodevice|audiodevice|window]

-l device list available video and audio capture device

-l videodevice list available video capture device

-l audiodevice list available audio capture device

-l window list available application window

Below is sample output of -l device:

rtsppusher -l device

Avaiable video capture device :

index : 0, name : FaceTime HD Camera (Built-in)

Avaiable audio capture device :

index : 0, name : Headset Microphone (Apple Audio Device)

index : 1, name : Internal Digital Microphone (Apple Audio Device)

Note : The demo version has the following limitations

Maximum support two simultaneous pusher stream.