

User manual

(RTMP Pusher)

Declaration

All rights reserved. No part of this publication may be excerpted, reproduced, translated, annotated or edited, in any form or by any means, without the prior written permission of the copyright owner.

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 <i>System parameters</i>	7
3.2.2 <i>Pusher node</i>	7
Chapter 4 Run RTMP Pusher	9

Chapter 1 Introduction

Happytime RTMP pusher is a high-efficiency RTMP pusher app, it support push the local media files, audio/video devices, living screen and the rtsp stream, support multiple pushers at the same time, support for up to 100 push streams, stable and reliable.

It support for most rtmp servers, such as Wowza、Red5、nginx_rtmp、crtmpserver etc. It can be perfectly applied to live broadcast requirements in various industries, desktop live broadcast, live camera, live broadcast, etc.

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

It provides the live audio/video stub process class, just need to implement a few interfaces to push the live audio/video RTMP stream.

Chapter 2 Key features

Support a variety of audio and video files

Support push video from camera and living screen

Support push audio from audio device

Support RTSP stream to RTMP stream

Support for configuring audio and video output parameters

Small size, suitable for embedded development

Code structure clear, easy to use

Chapter 3 Configuration

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>
  <pusher>
    <srcurl>1.mp4</srcurl>
    <dsturl>rtmp://192.168.1.102/myapp/mystream2</dsturl>
    <video>
      <width></width>
      <height></height>
      <framerate></framerate>
    </video>
    <audio>
      <samplerate>8000</samplerate>
      <channels>2</channels>
    </audio>
  </pusher>
  <pusher>
    <srcurl>rtsp://192.168.1.6/test.mp4</srcurl>
    <dsturl>rtmp://192.168.1.102/myapp/mystream3</dsturl>
    <user></user>
    <pass></pass>
    <video>
      <width></width>
      <height></height>
      <framerate></framerate>
    </video>
    <audio>
      <samplerate>8000</samplerate>
      <channels>2</channels>
    </audio>
  </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.

3.2.2 *Pusher* node

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

<srcurl>

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 rtmppusher is located

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

rtspaddr : specify the rtsp stream address as the push source, such as rtsp://127.0.0.1/live

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

videodeviceN, 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*

The audio device and screen similar to the video device

<dsturl>: Specify the push destination address, it should be a RTMP stream address, the RTMP client (such as VLC) can watch the audio/video by this RTMP stream address.

<user>: if the **<srcurl>** is a RTSP stream address, specify the rtsp server's login user name

<pass>: if the **<srcurl>** is a RTSP stream address, specify the rtsp server's login password

<video> : Specify the video output parameters

<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 use the default value 15, camera use the default value 25)

<audio> : Specify the audio output parameters

<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)

Chapter 4 Run RTMP Pusher

The rtmp pusher is a console application, to run the pusher, simply type "rtmppusher".

Note : The demo version has the following limitations

Maximum support two simultaneous pusher stream.