

# RTP stream publishing via RTSP

## Overview

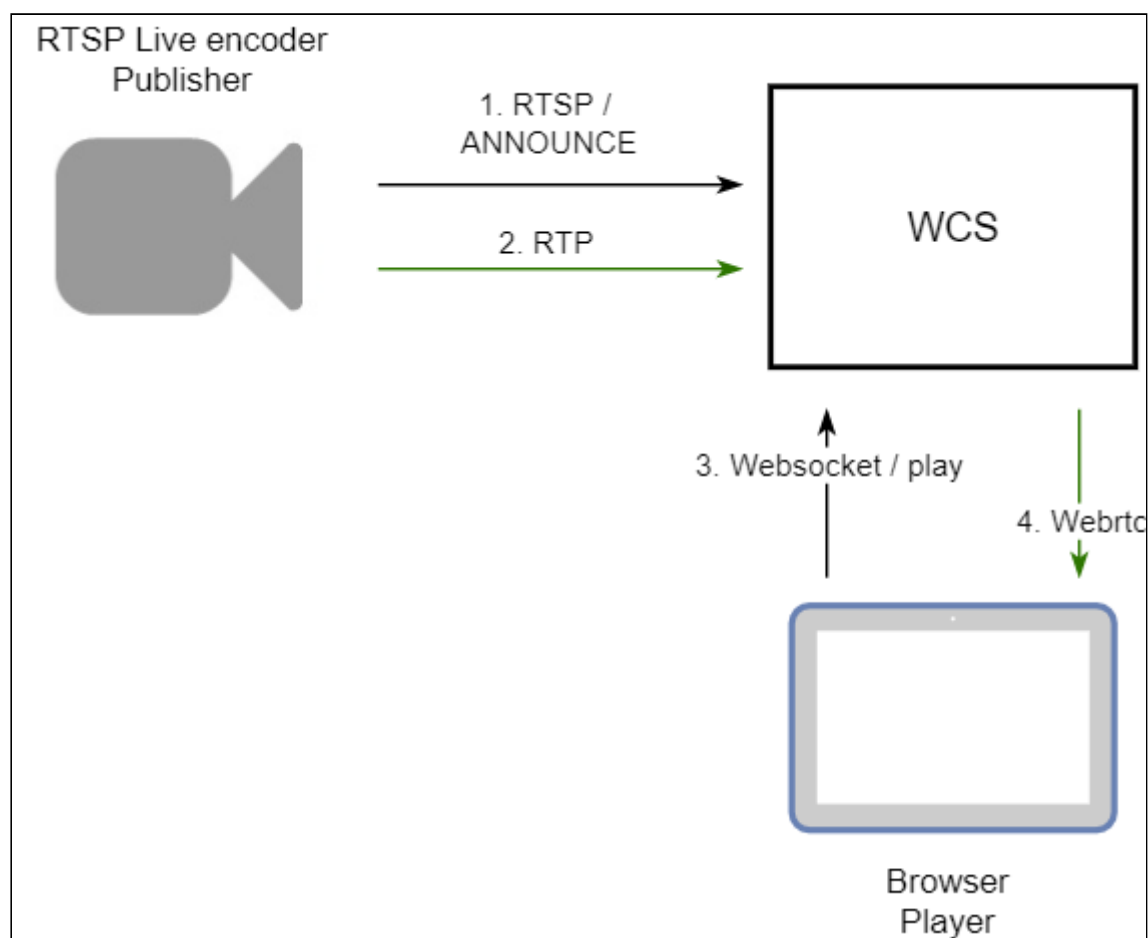
Since build [5.2.902](#) it is possible to publish RTP stream via RTSP directly to WCS. Both TCP interleaved and UDP transports are supported. This can be useful to publish H264+Opus streams from hardware or software encoder supporting RTSP. Since build [5.2.1584](#), H265 codec is also supported for publishing.

## Codecs supported

Video: - H264 - H265 (since build [5.2.1584](#)) - VP8

Audio: - AAC - Opus

## Operation flowchart



1. Live Encoder connects to the server via RTSP and sends **ANNOUNCE** command.
2. Live Encoder sends RTP stream to the server.
3. The browser establishes a connection via Websocket and sends the **playStream** command
4. The browser receives the WebRTC stream and plays that stream on the page.

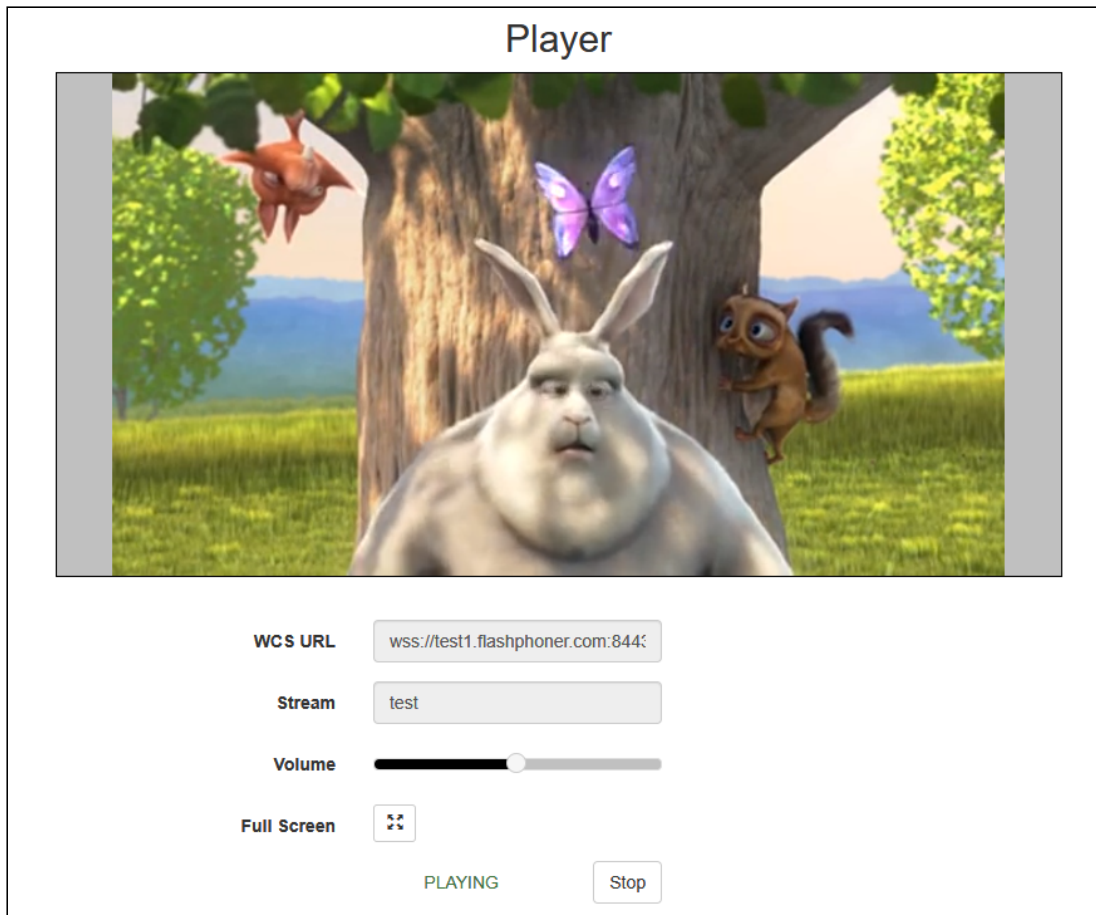
## Quick manual on testing

1. For test we use:
2. WCS server
3. ffmpeg to publish RTP stream via RTSP
4. [Player](#) web application in Chrome browser to play the stream
5. Start RTSP H264+Opus stream publishing using ffmpeg

```
ffmpeg -stream_loop -1 -re -i bunny360p.mp4 -c:a libopus -ac 2 -ar 48000 -c:v copy -b:a 96K -b:v 500K -f rtsp -rtsp_transport tcp rtsp://test1.flashphoner.com:554/test
```

```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'bunny360p.mp4':
  Metadata:
    major_brand      : isom
    minor_version    : 512
    compatible_brands: isomiso2avc1mp41
    encoder          : Lavf58.12.100
  Duration: 00:09:56.46, start: 0.000000, bitrate: 631 kb/s
  Stream #0:0(eng): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 640x360, 499 kb/s, 24 fps, 24 tbr, 12288 tbn, 48 tbc (default)
  Metadata:
    handler_name     : VideoHandler
  Stream #0:1(eng): Audio: aac (LC) (mp4a / 0x61347060), 48000 Hz, stereo, fltp, 128 kb/s (default)
  Metadata:
    handler_name     : SoundHandler
Stream mapping:
  Stream #0:0 -> #0:0 (copy)
  Stream #0:1 -> #0:1 (aac (native) -> opus (libopus))
Press [q] to stop, [?] for help
Output #0, rtsp, to 'rtsp://test1.flashphoner.com:554/test':
  Metadata:
    major_brand      : isom
    minor_version    : 512
    compatible_brands: isomiso2avc1mp41
    encoder          : Lavf58.45.100
  Stream #0:0(eng): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 640x360, q=2-31, 500 kb/s, 24 fps, 24 tbr, 90k tbn, 24 tbc (default)
  Metadata:
    handler_name     : VideoHandler
  Stream #0:1(eng): Audio: opus (libopus), 48000 Hz, stereo, flt, 96 kb/s (default)
  Metadata:
    handler_name     : SoundHandler
    encoder          : Lavc58.91.100 libopus
frame= 170 fps= 24 q=-1.0 size=N/A time=00:00:07.05 bitrate=N/A speed=0.998x
```

6. Open Player application. Set stream name to **Stream** field and click **Start**. The stream published playback begins



## H265 publishing

Since build [5.2.1584](#), RTP stream in H265 codec may be published via RTSP

```
ffmpeg -re -i source.mp4 -c:v libx265 -c:a aac -b:a 160k -bsf:v  
hevc_mp4toannexb -keyint_min 60 -profile:v main -preset veryfast -x265-params  
crf=23:bframes=0 -f rtsp -rtsp_transport tcp  
rtsp://test1.flashphoner.com:554/test
```

To do this, H265 should be added to codecs supported list

```
codecs=opus,alaw,ulaw,g729,speex16,g722,mpeg4-generic,telephone-  
event,h264,vp8,flv,mpv,h265
```

and to exclusion lists

```
codecs_exclude_sip=mpeg4-generic,flv,mpv,h265  
codecs_exclude_sip_rtmp=opus,g729,g722,mpeg4-generic,vp8,mpv,h265  
codecs_exclude_sfu=alaw,ulaw,g729,speex16,g722,mpeg4-generic,telephone-  
event,flv,mpv,h265
```

 **Warning**

H265 stream will be transcoded to H264 or VP8 to play from server!

## RTSP port setup

TCP port 554 is used by default to publish RTP stream via RTSP

```
rtsp.port=554
```

However, the port is privileged and is available to listen to applications launched by `root` only. If WCS is starting from `flashphoner` user (the default launch mode), the port should be changed, for example

```
rtsp.port=5554
```