Stream transcoding

The cases when transcoding is enabled

Video stream transcoding will be enabled automatically in one of the following cases:

1. Streamer and player codecs do not match by name.
For example, streamer publishes H.264 stream and player tries to play VP8.

2. H.264 codecs are differ bypacketization-mode parameter
For example, streamer publishes stream with packetization-mode=1 (default value) and player explicitly sets packetization-mode=0.The situation is quite rare, because almost all players support packetization-mode=1

3. Player resolution is explicitly set.
Example:

```
session.createStream({name:"stream1", constraints:{audio:true, video:{width:640,height:480}}}).play();
```

If the player explicitly sets the resolution desired, transcoding will be enabled even when the player resolution exactly matches the publisher one. This is done because WebRTC browser can change video resolution while publishing stream. To adapt the stream to the resolution that is specified by player, the stream should be transcoded.

4. Player bitrate is explicitly set.

Example

```
session.createStream({name:"stream1", constraints:{audio:true, video:{bitrate:300}}}).play();
```

In this case transcoder will be enabled to encode the stream to the bitrate specified.

Besides, transcoding can be forcibly enabled on server using this parameter inflashphoner.propertiesfile

```
disable_streaming_proxy=true
```

Transcoding dramatically increases the server resources consumption (CPU cores). Therefore, use it carefully!

Force transcoding disabling

Transcoding may be fully disabled on serverusing this parameter inflashphoner.propertiesfile

```
transcoding_disabled=true
```

If trascoding is forcefully disabled, in all four cases described above theTRANSCODING_REQUIRED_BUT_DISABLEDerror will be returned to client.

Transcoding disabling does not affectsstream mixer, transcoding will be enabled automatically when mixer is used.

Transcoding management with REST API

Obsolete REST API version (server builds before5.2.898)

REST query should be HTTP/HTTPS POST request as:

- HTTP:http://test.flashphoner.com:8081/rest-api/transcoder/startup
- HTTPS:https://test.flashphoner.com:8444/rest-api/transcoder/startup

Where:

- test.flashphoner.com is WCS server address
- 8081 is a standard REST / HTTP port of WCS server
- 8444 is a standardHTTPS port
- rest-api is mandatory URL prefix
- /transcoder/startup is REST query

REST queries and response states

REST query	Request example	Response example	Response states	Description

/transcoder/startup	{
	 "uri": "transcoder://tcode1",
	 "remoteStreamName": "test",
	 "localStreamName": "testT",
	 "encoder": {
	  "width": 640,
	  "height": 480,
	  "keyFrameInterval": 30,
	  "fps": 30,
	  "watermark": "Test.png"
	 }
	}
			400 - Bad request
			409 - Conflict
			500 - Internal error
				Create transcoder with defined parameters for certain stream

/transcoder/find	{
	 "remoteStreamName": "test"
	}
		[
		    {
		        "localMediaSessionId": "42a92132-bcd1-4436-a96f-3fec36b32b37",
		        "localStreamName": "testT",
		        "remoteStreamName": "test",
		        "uri": "transcoder://tcode1",
		        "status": "PROCESSED_LOCAL",
		        "hasAudio": true,
		        "hasVideo": true,
		        "record": false,
		        "encoder": {
		            "width": 640,
		            "height": 480,
		            "keyFrameInterval": 30,
		            "fps": 30,
		            "watermark": "Test.png"
		        }
		    }
		]
			200 – Transcoders found
			404 – Transcoders not found
				Find the transcoder by certain criteria

/transcoder
/find_all
		[
		{
		"localMediaSessionId": "42a92132-bcd1-4436-a96f-3fec36b32b37",
		"localStreamName": "testT",
		"remoteStreamName": "test",
		"uri": "transcoder://tcode1",
		"status": "PROCESSED_LOCAL",
		"hasAudio": true,
		"hasVideo": true,
		"record": false,
		"encoder": {
		"width": 640,
		"height": 480,
		"keyFrameInterval": 30,
		"fps": 30
		}
		}
		]
			200 – Transcoders found
			404 – Transcoders not found
				Find all transcoders

/transcoder
/terminate
	{
	"uri":"transcoder://tcode1"
	}
			200 - Transcoders is terminated
			404 - Transcoder not found
				Stop transcoder and its output stream

/transcoder
/set_watermark
	{
	"uri":"transcoder://tcode1",
	"watermark":"/opt/media/logo.png",
	"x":10,
	"y":10,
	"marginTop":5,
	"marginLeft":5,
	"marginBottom":5,
	"marginRight":5
	}
			200 - OK
			400 - Bad request
			404 - Not found
				Add watermark to transcoder output stream

Parameters

Name	Description	Example
uri	Transcoder URL	transcoder://tcode1
localStreamName	Transcoder output stream name	testT
remoteStreamName	Stream name to transcode	test
localMediaSessionId	Transcoder media session Id	42a92132-bcd1-4436-a96f-3fec36b32b37
status	Transcoder state	PROCESSED_LOCAL
hasAudio	Output stream has audio	true
hasVideo	Output stream has video	true

record	Output stream is recorded	`false`
Encoder parameters		
width	Picture width	`640`
height	Picture height	`480`
keyFrameInterval	Key frame generation interval (GOP)	`30`
fps	Frames per second	`30`
bitrate	Bitrate, in kbps	`500`
type	Codec	`OPENH264`
watermark	Watermark file	Test.png

Known limits

1.Transcoder cannot be created by REST API for audio only stream. In response to /transcoder/startup query for such stream, server returns 400 Bad request with message "Can't start transcoder for audio only stream"

2. If neither width nor height are specified when creating a transcoder by REST API, transcoding will not be enabled, the incoming stream will be copied without reencoding.

3. If only height is specified, the incoming stream will be transcoded with aspect ratio preserving if enabled.

4, If only width is specified, the quey return 400 Bad request with message "Height is not specified"

REST API version 2 (server builds since5.2.898)

REST query should be HTTP/HTTPS POST request as:

- HTTP:http://test.flashphoner.com:8081/rest-api/transcoder2/startup
- HTTPS:https://test.flashphoner.com:8444/rest-api/transcoder2/startup

Where:

- test.flashphoner.com is WCS server address
- 8081 is a standard REST / HTTP port of WCS server
- 8444 is a standardHTTPS port
- rest-api is mandatory URL prefix
- /transcoder2/startup is REST query

REST queries and response states

REST query	Request example	Response example	Response states	Description

/transcoder2
/startup
	{
	 "uri": "transcoder2://tcode2",
	 "remoteStreamName": "test",
	 "localStreamName": "testT",
	 "encoder": {
	  "videoCodec": "H264",
	  "audioCodec": "mpeg4-generic",
	  "width": 320,
	  "height": 240,
	  "keyFrameInterval": 60,
	  "fps": 30,
	  "audioRate": 44100,
	  "audioBitrate": 64000
	 }
	}
			200 - OK
			400 - Bad request
			409 - Conflict
			500 - Internal error
				Create transcoder with defined parameters for certain stream

/transcoder2
/find
	{
	 "remoteStreamName": "test"
	}
		[
		  {
		    "localMediaSessionId": "82ad5545-e11e-4f0f-801a-49e69d8c38f2",
		    "localStreamName": "testT",
		    "remoteStreamName": "test",
		    "uri": "transcoder2://tcode2",
		    "status": "PROCESSED_LOCAL",
		    "hasAudio": true,
		    "hasVideo": true,
		    "record": false,
		    "encoder": {
		      "width": 320,
		      "height": 240,
		      "keyFrameInterval": 60,
		      "fps": 30,
		      "audioRate": 44100,
		      "audioCodec": "mpeg4-generic",
		      "videoCodec": "H264",
		      "videoRate": 90000
		    }
		  }
		]
			200 – OK
			404 – Not found
				Find the transcoder by certain criteria

/transcoder
/find_all
		[
		  {
		    "localMediaSessionId": "82ad5545-e11e-4f0f-801a-49e69d8c38f2",
		    "localStreamName": "testT",
		    "remoteStreamName": "test",
		    "uri": "transcoder2://tcode2",
		    "status": "PROCESSED_LOCAL",
		    "hasAudio": true,
		    "hasVideo": true,
		    "record": false,
		    "encoder": {
		      "width": 320,
		      "height": 240,
		      "keyFrameInterval": 60,
		      "fps": 30,
		      "audioRate": 44100,
		      "audioCodec": "mpeg4-generic",
		      "videoCodec": "H264",
		      "videoRate": 90000
		    }
		  }
		]
			200 – OK
			404 – Not found
				Find all transcoders

/transcoder
/terminate
	{
	 "uri":"transcoder2://tcode2"
	}
			200 – OK
			404 – Not found
				Stop transcoder and its output stream

/transcoder2
/set_watermark
	{
	 "uri":"transcoder2://tcode1",
	 "watermark":"/opt/media/logo.png",
	 "x":10,
	 "y":10,
	 "marginTop":5,
	 "marginLeft":5,
	 "marginBottom":5,
	 "marginRight":5
	}
			200 - OK
			400 - Bad request
			404 - Not found
				Add watermark to transcoder output stream

Parameters

Name	Description	Example
uri	Transcoder URL	`transcoder2://tcode2`
localStreamName	Transcoder output stream name	`testT`
remoteStreamName	Stream name to transcode	`test`
localMediaSessionId	Transcoder media session Id	`82ad5545-e11e-4f0f-801a-49e69d8c38f2`
status	Transcoder state	`PROCESSED_LOCAL`
hasAudio	Output stream has audio	`true`

hasVideo	Output stream has video	`true`
record	Output stream is recorded	`false`
Encoder parameters		
width	Picture width	`320`
height	Picture height	`240`
audioCodec	Audio codec	mpeg4-generic
audioRate	Audio sample rate, Hz	44100
audioChannels	Audio channels	2
audioBitrate	Audio bitrate, bps	64000
videoCodec	Video codec	H264
keyFrameInterval	Key frame generation interval (GOP)	`30`
fps	Frames per second	`30`
bitrate	Video bitrate, in kbps	`500`
type	Encoder type	`OPENH264`
watermark	Watermark file	Test.png
videoRate	Video sample rate, Hz	90000

Known limits

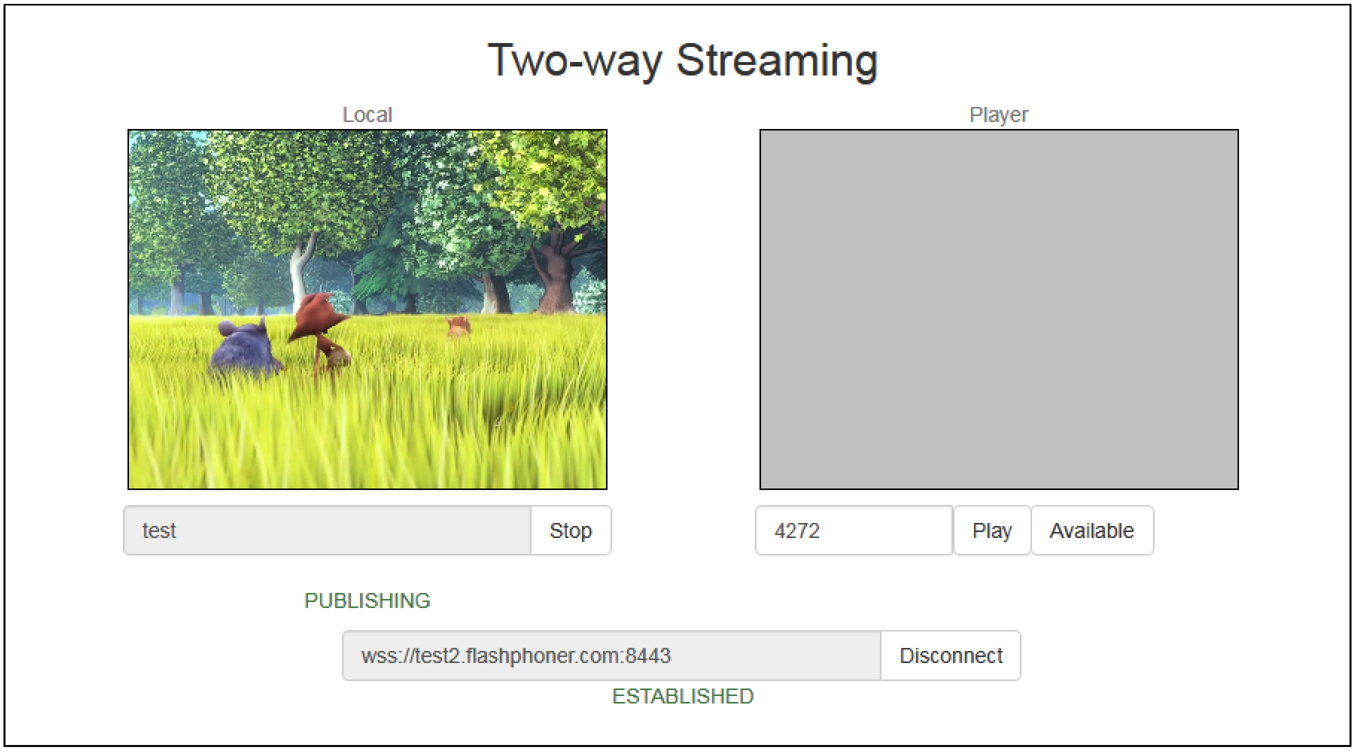
1. If video transcoding parameters are passed for audio only stream, or audio transcoding parameters are passed for video only stream, 400 Bad request will return

Quick manual for testing

1. For test we use

- WCS server;
- Two Way Streamingweb application to publish a stream;
- Playerweb application to play an output stream;
- Chrome browser withREST clientto send REST queries to server

2. Open Two Way Streaming application and publish stream named test



test

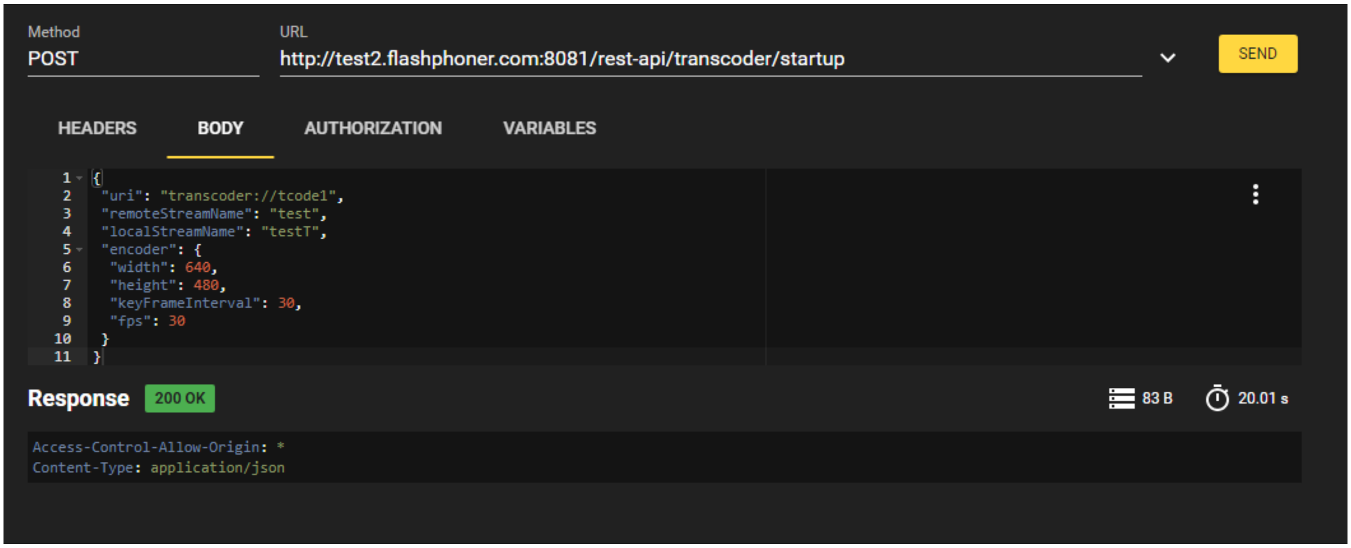
4272

PUBLISHING

wss://test2.flashphoner.com:8443

ESTABLISHED

3. Open REST client and send REST query /transcoder/startup



Method: POST

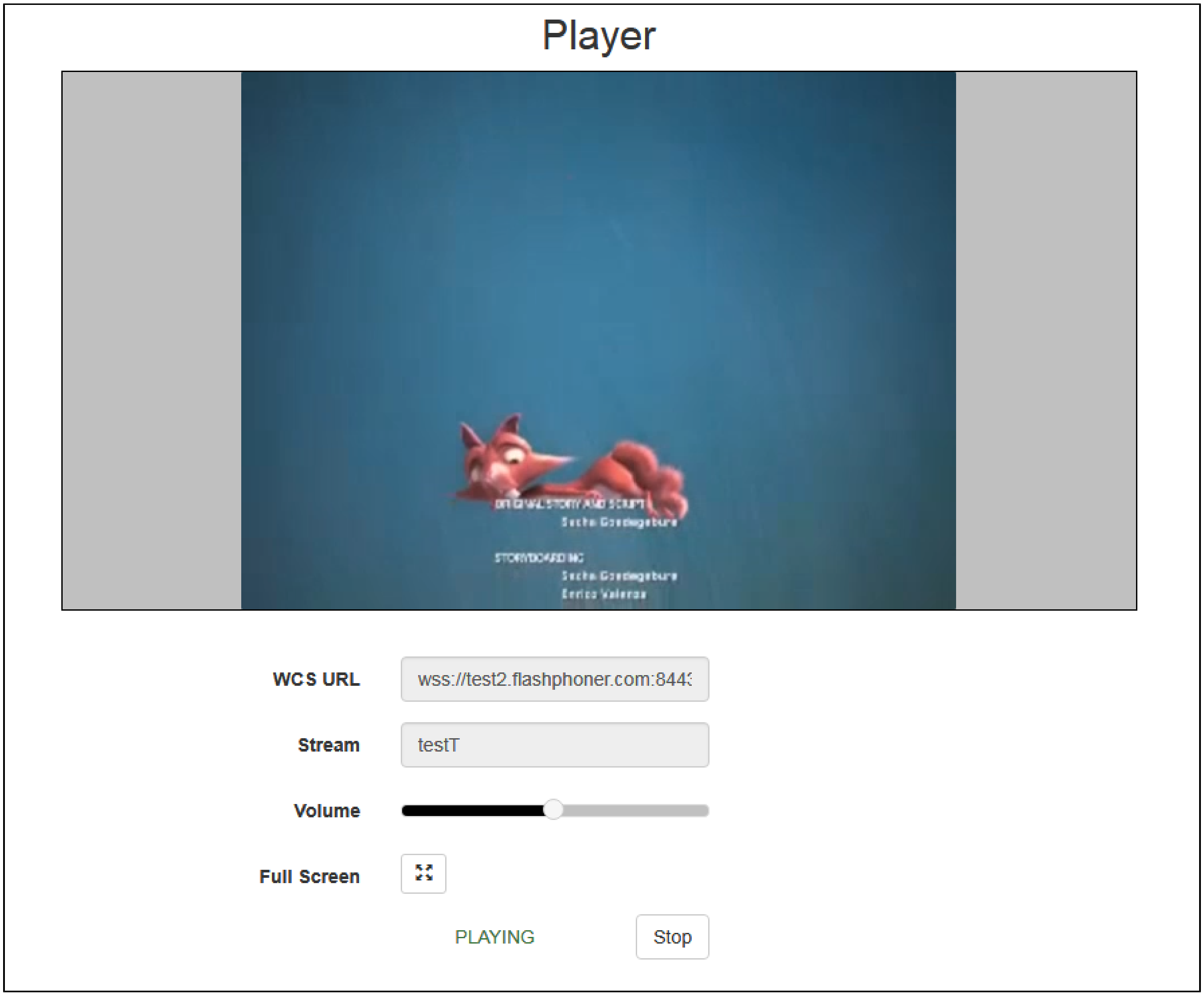
URL: http://test2.flashphoner.com:8081/rest-api/transcoder/startup

```
{
 "uri": "transcoder://tcode1",
 "remoteStreamName": "test",
 "localStreamName": "testT",
 "encoder": {
  "width": 640,
  "height": 480,
  "keyFrameInterval": 30,
  "fps": 30
 }
}
```

200 OK

```
Access-Control-Allow-Origin: *
Content-Type: application/json
```

4. Open Player application, set testT to Stream field and click Start

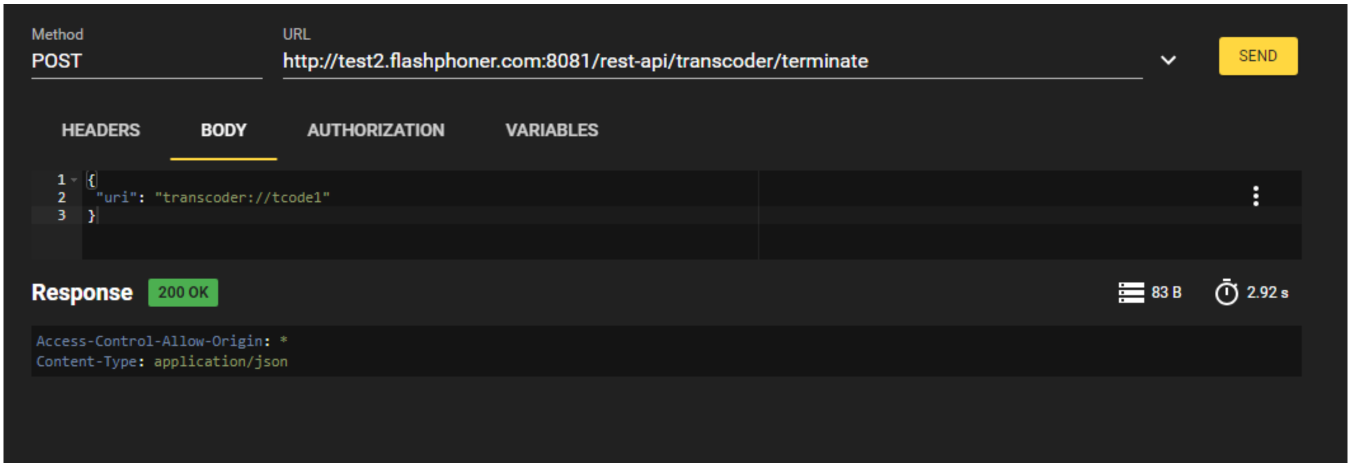


Player

WCS URL: wss://test2.flashphoner.com:844

Stream: testT

5.Open REST client adn send REST query /transcoder/terminate



Method: POST

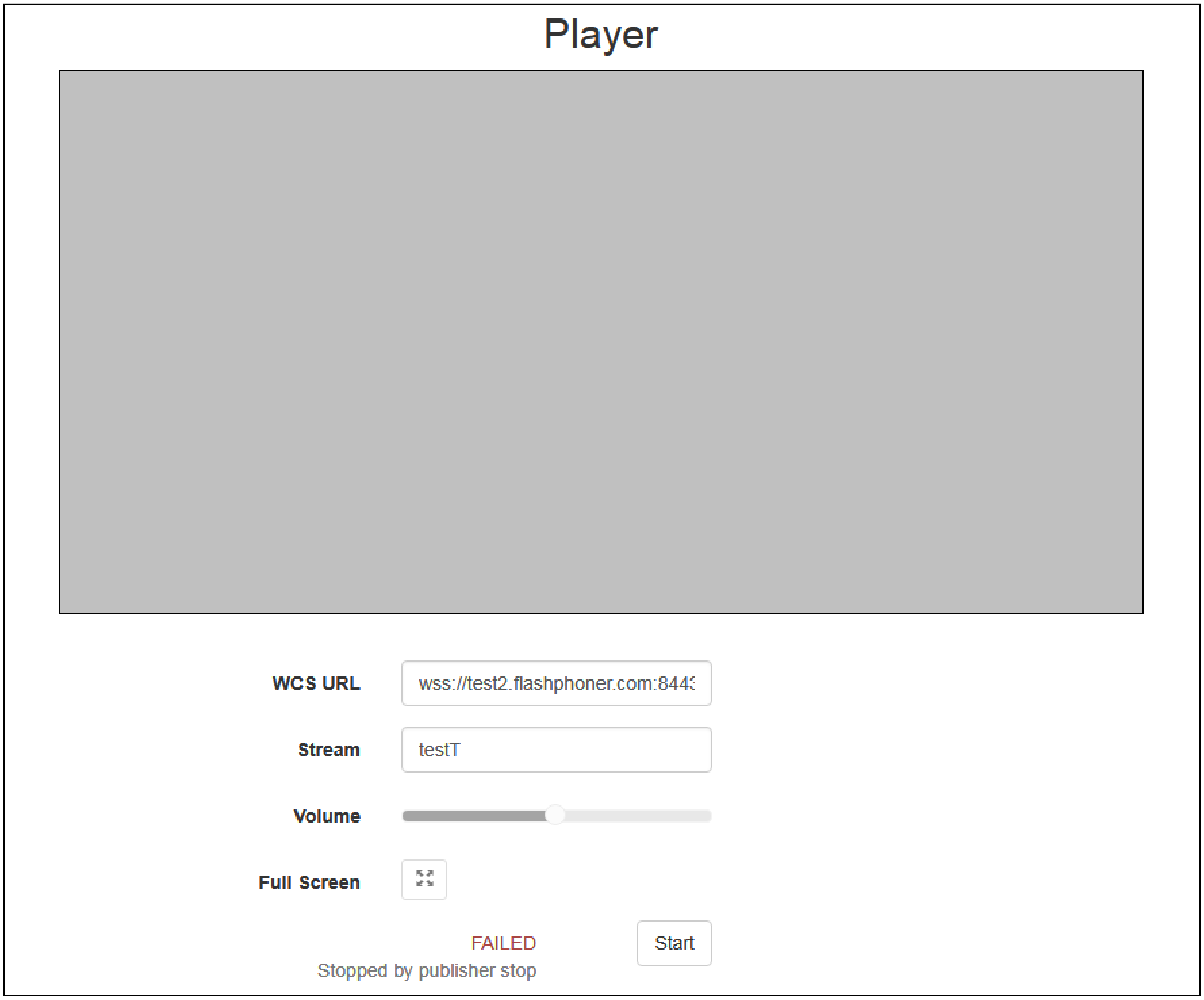
URL: http://test2.flashphoner.com:8081/rest-api/transcoder/terminate

```
{
 "uri": "transcoder://tcode1"
}
```

Response 200 OK

```
Access-Control-Allow-Origin: *
Content-Type: application/json
```

6. Playback will be stopped due to transcoder stop



WCS URL: wss://test2.flashphoner.com:844

Stream: testT

Volume

Full Screen

FAILED

Stopped by publisher stop

Picture aspect ratio preserving

By default, if the stream is published with one picture resolution and is requested to play with another resolution, WCS tries to preserve picture aspect ratio. For example, if stream is published on server with resolution 640x360, aspect ratio 16:9, and subscriber requests to play it with resolution 320x240 (4:3), the stream will be transcoded to resolution 320x180 (16:9). If subscriber requests picture height only without setting width, aspect ratio will also be preserved.

To disable picture aspect ratio preserving, the following parameter sho;ud be set inflashphoner.propertiesfile

```
video_transcoder_preserve_aspect_ratio=false
```

In this case stream will be transcoded to picture width and height that are requested by subscriber. If subscriber does not set picture height, it wiil be set to 120.If subscriber does not set picture width, it wiil be set to 160.

Transcoder output stream audio and video synchronization

By default transcoder does not synchronize output stream audio and video, leaving sinchronization value as is. This can lead to out of audio and video sync in stream transcoded. To prevent this, the pacer buffer is added in build5.2.543which can be enabled with the following parameter

```
av_paced_sender=true
```

Pacer buffer maximum size is set in frames by the following parameter

```
av_paced_sender_max_buffer_size=5000
```

By default pacer buffer size is 5000 frames.

The statistics information received by the following query is uused to control pacer buffer usage

```
curl -s 'http://localhost:8081/?action=stat&format=json&groups=buffer_stats'
```

A certain stream watermarking

Since build5.2.693it is possible to add watermark to transcoded stream when creating transcoder using REST API, for example

```
{
 "uri": "transcoder://tcode1",
 "remoteStreamName": "test",
 "localStreamName": "testT",
 "encoder": {
  "width": 640,
  "height": 480,
  "keyFrameInterval": 30,
  "fps": 30,
  "watermark": "Test.png"
 }
}
```

By default, if file name only is passed, watermark picture file should be placed to /usr/local/FlashphonerWebCallServer/conf folder. The full path to the file can also be passed, for example

```
{
 "uri": "transcoder://tcode1",
 "remoteStreamName": "test",
 "localStreamName": "testT",
 "encoder": {
  "width": 640,
  "height": 480,
  "keyFrameInterval": 30,
  "fps": 30,
  "watermark": "/opt/media/Test.png"
 }
}
```

Adding and changing stream watermark dynamically

Since build 5.2.1349 in is possible to dynamically add or change stream watermark without stopping the transcoder. A watermark can be added, changed or moved to another picture location according to coordinates defined using REST API query `/transcoder2/set_watermark`

```
{
 "uri":"transcoder2://tcode1",
 "watermark":"/opt/media/logo.png",
 "x":10,
 "y":10,
 "marginTop":5,
 "marginLeft":5,
 "marginBottom":5,
 "marginRight":5
}
```

Where

- watermark - watermark file name
- x, y - top left watermark corner coordinates on the stream picture

- marginTop, marginLeft, marginBottom, marginRignt - watermark margins from stream picture borders

If watermark coordinates are out of stream picture bounds, the watermark will be scaled to the bounds using margins.

To move watermark to another location on the stream picture, send the query with the same file name and a new coordinates. To remove watermark from the stream picture, send the query with empty `watermark` field

```
{
 "uri":"transcoder2://tcode1",
 "watermark":""
}
```

Multithreaded encoding

Since build5.2.816multithreaded strems encoding is supported using OpenH264 encoder. Encoder threads count can be set with the following parameter

```
video_encoder_max_threads=2
```

By default, streams will be encoded in 2 threads.

Multi threaded encoding is enabled depending on transcoder output stream resolution. The threshold can be set with the following parameter

```
video_encoder_second_thread_threshold=777000
```

The threshold value is the product of the picture width multiplication to the height. Therefore, 720p and higher resolutions wiil be encoded in multiple threads. This threshold can be lowered if necessary. For example, to encode 480p pictures in multiple threads, set the following value

```
video_encoder_second_thread_threshold=408950
```

Known issues

1. Encoding quality settings cannot be applied if OpenH264 is used

CSymptoms: picture quality is not changing when using different`constraints.video.quality` values, for example

```
constraints.video.quality=5
```

does not differ from

```
constraints.video.quality=20
```

Solution: do not use OpenH264 encoder because it does not support CRF

```
encoder_priority=FF
```