

Stream watermarking

- [Overview](#)
- [Watermark picture mixing with alpha channels support](#)
- [Known issues](#)

Overview

A watermark can be applied to stream captured while translating it to subscribers. To do this, the following parameter should be added to [flashphoner.properties](#) file to specify watermark PNG file name:

```
custom_watermark_filename=watermark.png
```

Watermark PNG file should be placed to `/usr/local/FlashphonerWebCallServer/conf` catalog.

Watermark can be added only using the commercial license because Flashphoner watermark will be added to all streams when trial license is used.

If watermark picture contains black regions, those regions will not be displayed because black is background color for video playback.

Stream transcoding must be enabled to add watermark, because stream must be decoded and then encoded with picture applied for further broadcasting. To enable transcoding, the stream resolution can be set explicitly when publishing

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

and playing the stream

```
session.createStream({name: 'stream1', constraints: {audio: true, video: {width: 320, height: 240}}}).play();
```

In this case all the streams are publishing on server in 640x480 resolution and are playing in 320x240. That automatically enables transcoding, rescaling from 640x480 to 320x240 and watermark applying.

Stream watermarking will increase the server load approximately 0.5-1 core for each incoming video stream depending on video transcoded resolution. For example, 16 incoming streams handling may require 8-16 core server. In this regard, it is recommended to apply watermark on stream publisher source if possible.

Watermark picture mixing with alpha channels support

Some watermark picture colors may be distorted or invisible (transparent) when applying this picture as watermark. It depends on alpha channel values of the picture applied. Since build [5.2.1064](#) the following parameter is added to enable watermark picture mixing with alpha channels support

```
custom_watermark_mix=true
```

Note that alpha channels support implementation today executes a many subsequential operations. This can lead to CPU load increasing and low quality of the resulting image. In the future, it is planned to be implemented using parallel mixing on GPU.

Known issues

1. Default watermark (black picture) will be used if watermark file is damaged or absent

Symptoms: black picture in output stream when watermark is added, there is a message in server log

```
Wrong watermark file format. Should be PNG.
```

Solution: use only correct PNG file to add watermark