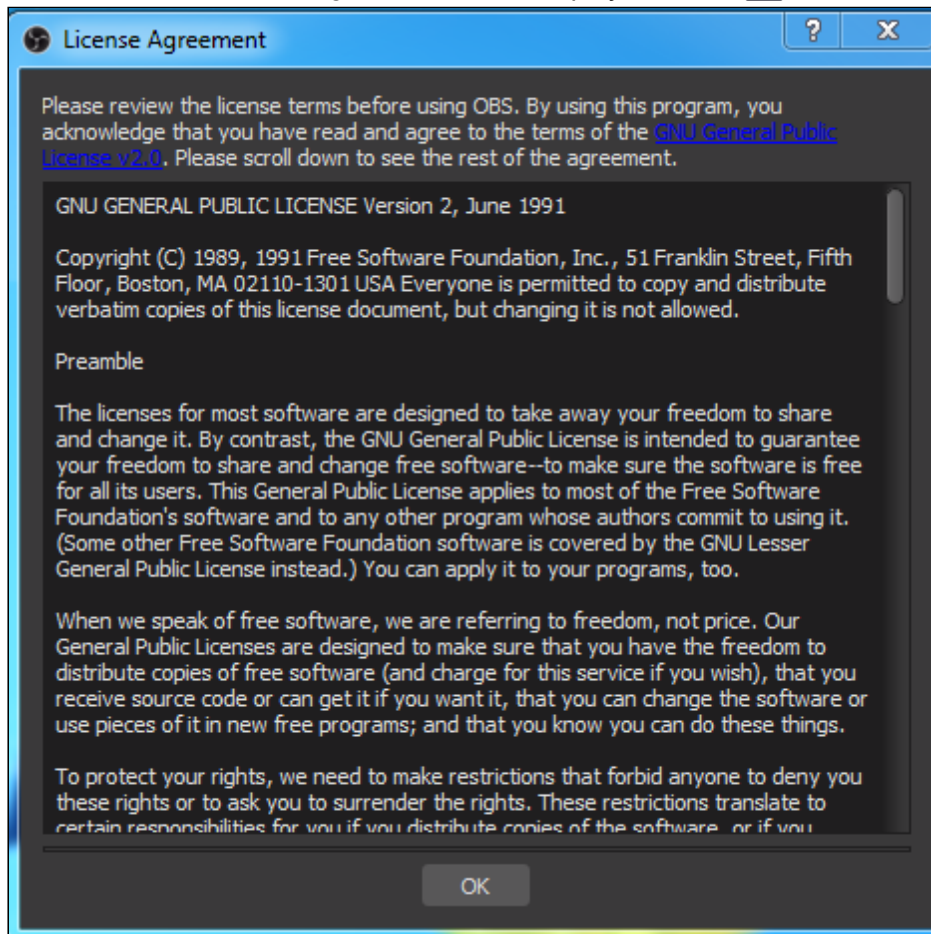


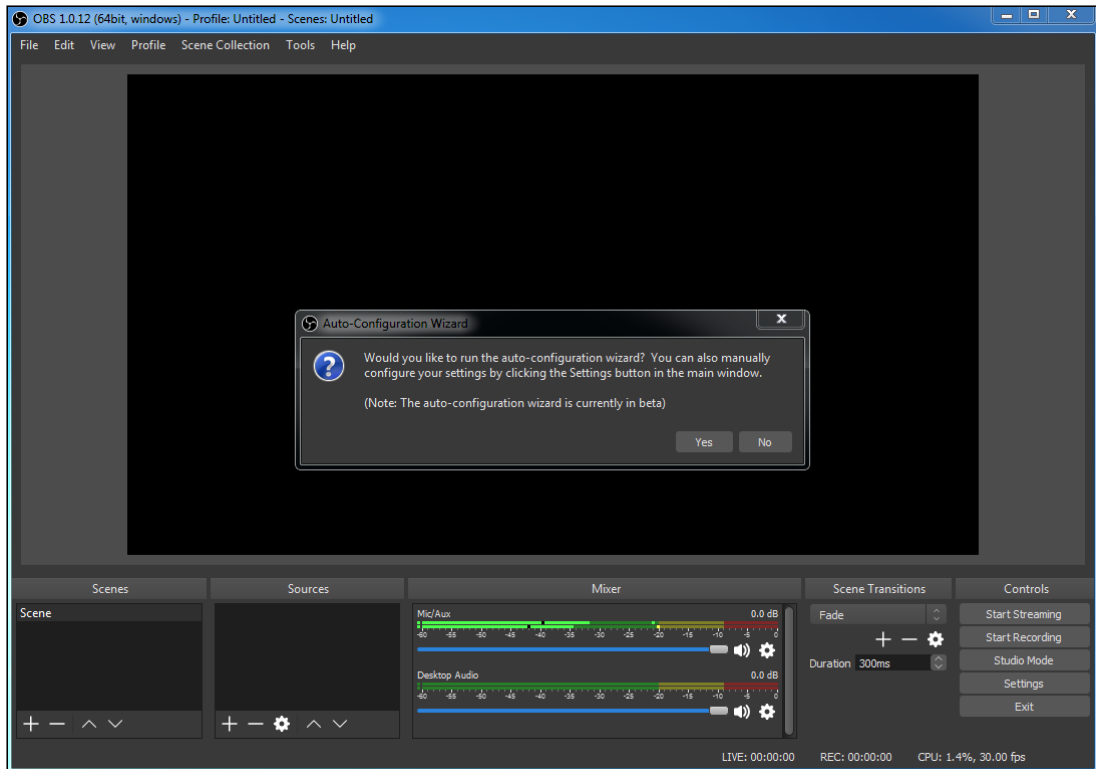
Setup and testing

First client launch and setup

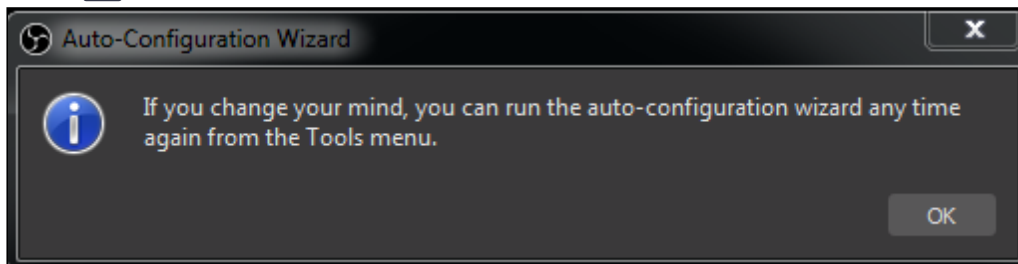
1. On first launch, a license agreement will be displayed. Press **OK**



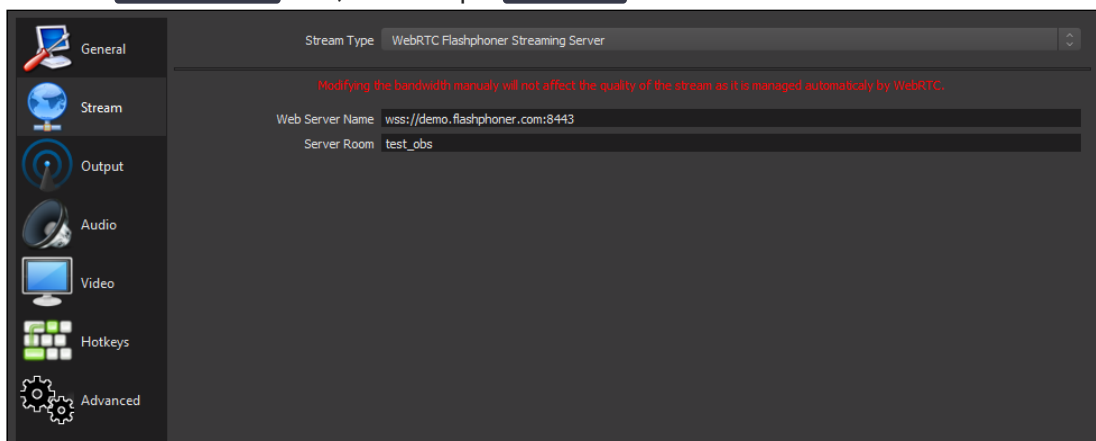
2. Then automatic setup wizard will be supposed to launch. Press **No**



3. Press **OK** in the next window

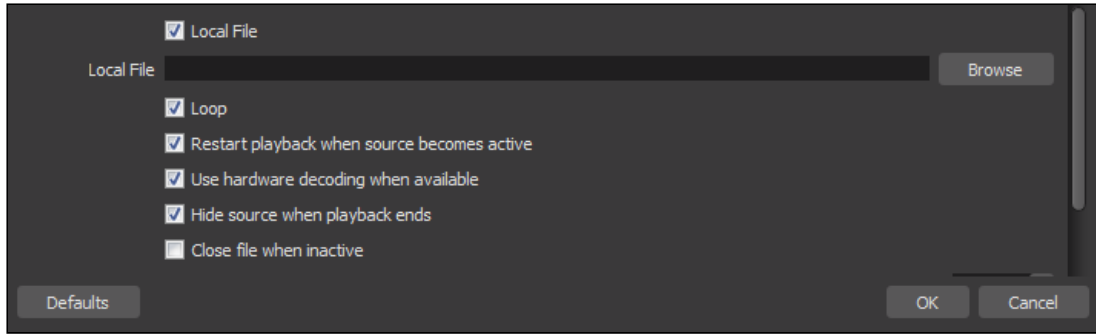


4. Press **Settings** button. On **Stream** tab choose **WebRTC Flashponer Streaming Server** in **Stream Type** drop-down list. Enter Secure Websocket server URL to **Web Server Name** field, for example **wss://demo.flashponer.com:8443**, and set stream name to **Server Room** field, for example **test_obs**

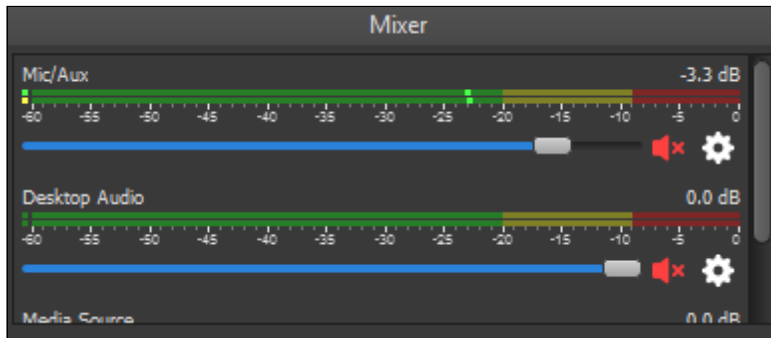


If necessary, set maximum video resolution on **Video** tab, the settings are the same as OBS Studio. Press **Ok** to save and apply settings.

5. Add media source, local file for example



6. To escape echo while stream publishing, mute all other sound sources excluding media source chosen



Translation is ready to start.

Server setup

Enforce bitrate increasing

To prevent translation quality decreasing while Full HD stream is publishing, bitrate increasing should be enforced on server side with the following parameters in [flashphoner.properties](#) file

```
webrtc_sdp_min_bitrate_bps=3000000  
webrtc_sdp_max_bitrate_bps=7000000
```

In this case publishing bitrate will be held between 3000 kbps and 7000 kbps.

Playback codec configuration

By default, for playing WebRTC video stream the same video codec is used with which it is published (VP8), and transcoding on server is not performed.

To use H.264 codec for playback, VP8 can be excluded on browser side using Web SDK `stipCodecs` option:

```
playStream = session.createStream({
  ...
  stripCodecs: "VP8"
}).on(STREAM_STATUS.PENDING, function (publishStream) {
  ...
});
playStream.play();
```

If H.264 is used for playback, VP8 to H.264 transcoding is performed on server.

WebRTC transport setup

By default, OBS WebRTC Client publish a stream using UDP transport. Enable TCP transport to prevent packet loss using the following WCS settings:

```
ice_tcp_transport=true
```

Quick testing

1. Press Start Streaming button in OBS WebRTC Client, allow Windows Firewall bypassing to application if necessary. Green square in bottom right corner shows that stream is published successfully



2. Open [Player](#) example page on server, set stream name `test_obs` in `Stream` field and press `Start`. The stream playback starts

Player



WCS URL

Stream

Volume

Full Screen

PLAYING