

Запись трансляции

- [Описание](#)
- [Краткое руководство по тестированию](#)
 - [Запись трансляции](#)
- [Настройка](#)
 - [Серверная часть](#)
 - [Включение и отключение записи потоков](#)
 - [Формирование имени файла записи потока](#)
 - [Ротация файлов записей](#)
 - [Скрипт обработки записанных файлов](#)
 - [Настройка частоты дискретизации аудио при записи](#)
 - [Клиентская часть](#)
- [Запись потоков по требованию](#)
 - [REST-методы и статусы ответа](#)
 - [Параметры](#)
- [Известные проблемы](#)

Описание

Медиапоток, захваченный WCS, может быть записан при публикации.

Поддерживаемые протоколы:

- WebRTC
- RTMP
- RTSP

Форматы записи:

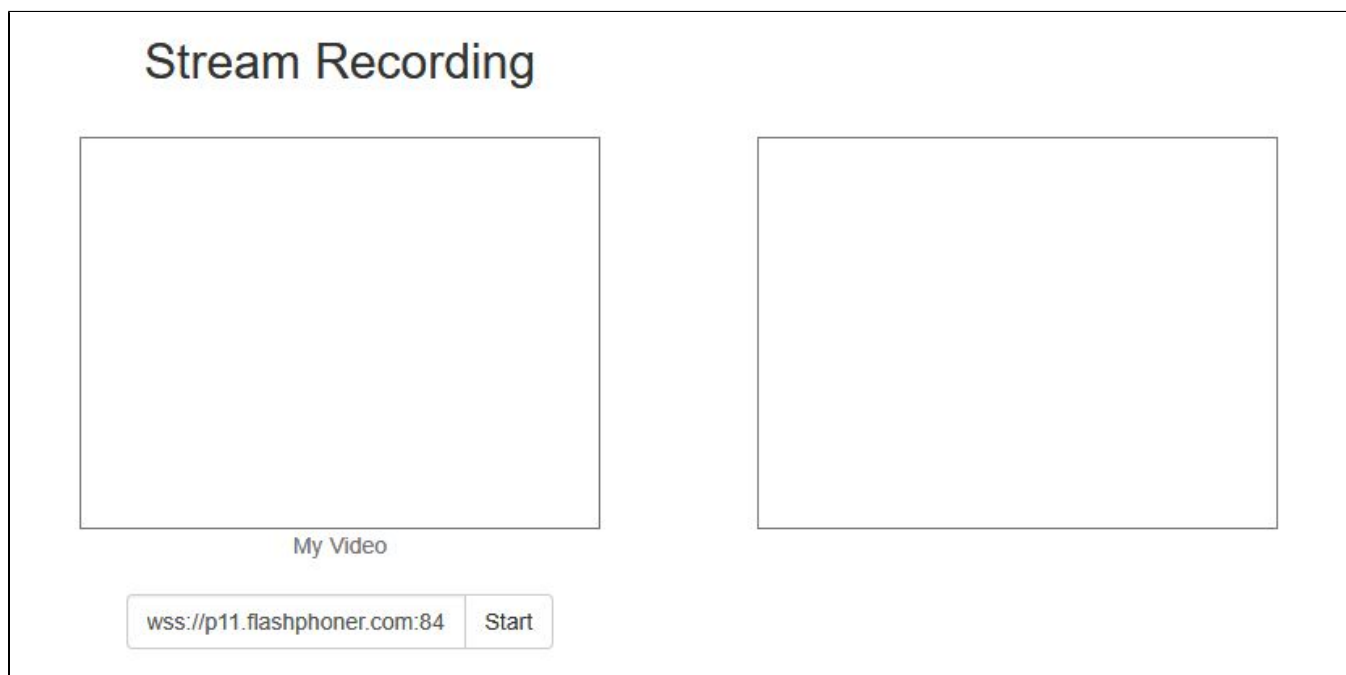
- MP4 для кодеков H.264 + AAC
- WebM для кодека VP8 + Vorbis

Краткое руководство по тестированию

Запись трансляции

1. Для теста используем демо-сервер demo.flashphoner.com и веб-приложение Stream Recording

https://demo.flashphoner.com/client2/examples/demo/streaming/stream_recording/recording.html



2. Нажмите кнопку "Start". Начнется захват и трансляция потока.

Stream Recording



My Video

wss://p11.flashphoner.com:84

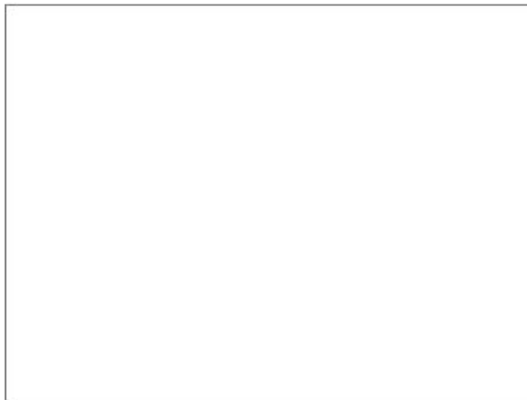
Stop

PUBLISHING



3. Нажмите кнопку "Stop". Трансляция остановится, отобразится ссылка на воспроизведение и скачивание записанного фрагмента

Stream Recording



My Video

wss://p11.flashphoner.com:84

Start

UNPUBLISHED



Download recorded stream

Настройка

Серверная часть

Включение и отключение записи потоков

По умолчанию запись потоков включена на стороне WCS-сервера.

Для отключения в конфиг /usr/local/FlashphonerWebCallServer/conf/flashphoner.properties следует добавить

```
record_streams=false
```

Настройка

```
record_flash_published_streams=true
```

включает запись потоков, опубликованных при помощи Flash, RTMP-кодировщика или с другого RTMP-сервера.

Настройка

```
record_rtsp_streams=true
```

включает запись потоков, захваченных с RTSP IP-камер.

Формирование имени файла записи потока

Настройка stream_record_policy определяет способ формирования имени файла записи потока. Например,

```
stream_record_policy=streamName
```

означает, что имя файла будет соответствовать имени потока. Поток, опубликованный при помощи ffmpeg

```
ffmpeg -re -i BigBuckBunny.mp4 -preset ultrafast -acodec aac -vcodec h264 -strict -2 -f flv rtmp://test1.  
flashphoner.com:1935/live/stream_ffmpeg
```

будет записан в файл stream_ffmpeg.mp4.

По умолчанию, имя файла формируется по шаблону

```
stream_record_policy=template
```

В свою очередь, шаблон задается настройкой stream_record_policy_template. По умолчанию

```
stream_record_policy_template=stream-{mediaSessionId}-{login}
```

Доступны следующие элементы шаблона:

Элемент	Описание	Максимальный размер
{streamName}	Имя потока	
{startTime}	Время начала записи	20 символов
{sessionId}	Идентификатор сессии в кодировке BASE64	60 символов
{mediaSessionId}	Идентификатор медиасессии	36 символов
{login}	Логин	32 символа
{audioCodec}	Аудиокодек	4 символа
{videoCodec}	Видеокодек	4 символа

Если имя файла создается из имени потока, в нем могут быть символы, недопустимые к использованию в именах, например, прямой слэш '/'. В этом случае имя файла должно быть закодировано при помощи настройки

```
encode_record_name=true,HEX
```

При этом имя файла будет закодировано шестнадцатиричным числом. Настройка

```
encode_record_name=true,BASE64
```

кодирует имя файла при помощи BASE64.

Другой способ экранирования недопустимых символов - их удаление при помощи параметра `exclude_record_name_characters`. По умолчанию

```
exclude_record_name_characters=
```

Например, для исключения двоеточия, запятой, точки и слэша необходимо указать

```
exclude_record_name_characters=:.,/
```

Ротация файлов записей

Потоки могут записываться частями заданной длительности при помощи параметра `record_rotation`. Например, настройка

```
record_rotation=20
```

определяет длительность фрагмента в 20 секунд.

Скрипт обработки записанных файлов

Настройка `on_record_hook_script` указывает на shell-скрипт, который вызывается по завершении записи потока.

По умолчанию скрипт располагается в каталоге `/usr/local/FlashphonerWebCallServer/bin`:

```
on_record_hook_script=/usr/local/FlashphonerWebCallServer/bin/on_record_hook.sh
```

но может быть размещен в любом другом месте под другим именем, например:

```
on_record_hook_script=/opt/on_record.sh
```

Этот скрипт можно использовать для копирования или перемещения записи потока из директории `WCS_HOME/records` в другое место по завершении записи.

Пример:

```
STREAM_NAME=$1
SRC_FILE=$2
SRC_DIR="/usr/local/FlashphonerWebCallServer/records/"
REPLACE_STR="/var/www/html/stream_records/${STREAM_NAME}-"
DST_FILE="${SRC_FILE}/${SRC_DIR}/${REPLACE_STR}"
cp $SRC_FILE $DST_FILE
```

Здесь

- \$1 - имя потока
- \$2 - абсолютное имя файла записи потока
- по завершении записи потока файл записи копируется в директорию `/var/www/html/stream_records/`

Необходимо учитывать длину абсолютного имени файла (с учетом имени каталога), которое будет получено при копировании. Если абсолютное имя целевого файла превышает 255 символов, команда копирования завершится с ошибкой, и скрипт не сработает в соответствии с ожиданиями.

Настройка частоты дискретизации аудио при записи

По умолчанию, запись звука ведется с частотой дискретизации 44.1 кГц. При необходимости, это значение можно изменить при помощи параметра

```
record_audio_codec_sample_rate=48000
```

В данном случае частота дискретизации будет установлена в 48 кГц.

Клиентская часть

При включении записи потоков на сервере, будет ли записан поток, или нет, зависит от значения параметра `record`, переданного функции `createStream` в скрипте публикующего клиента:

- `true` - поток, опубликованный с использованием этого клиента, будет записан;
- `false` - поток не будет записан.

Например, в скрипте веб-приложения Stream Recording [recording.html](#), [recording.js](#), содержится следующий код:

```
function publishStream(session) {
  var streamName = $('#url').val().split('/')[3];
  session.createStream({
    name: streamName,
    display: localVideo,
    record: true,
    receiveVideo: false,
    receiveAudio: false
    ...
  }).publish();
}
```

Запись потоков по требованию

В некоторых случаях, необходимо записать поток, который уже транслируется сервером, например, выходной поток микшера. Это можно сделать при помощи REST API. Обратите внимание, что только потоки в статусе "PUBLISHING" могут быть записаны.

REST-запрос должен быть HTTP/HTTPS POST запросом в таком виде:

- HTTP: <http://streaming.flashphoner.com:8081/rest-api/stream/startRecording>
- HTTPS: <https://streaming.flashphoner.com:8444/rest-api/stream/startRecording>

Здесь:

- streaming.flashphoner.com - адрес WCS-сервера
- 8081 - стандартный REST / HTTP порт WCS-сервера
- 8444 - стандартный HTTPS порт
- `rest-api` - обязательный префикс
- `/stream/startRecording` - используемый REST-вызов

REST-методы и статусы ответа

REST-метод	Пример тела REST-запроса	Пример тела REST-ответа	Статусы ответа	Описание
<code>/stream /startRecording</code>	<pre>{ "mediaSessionId": "5a072377-73c1-4caf-abd3" }</pre>		404 - Not found 500 - Internal error	Начать запись потока в указанной медиасессии
<code>/stream /stopRecording</code>	<pre>{ "mediaSessionId": "5a072377-73c1-4caf-abd3" }</pre>		404 - Not found 500 - Internal error	Завершить запись потока в указанной медиасессии

Параметры

Имя параметра	Описание	Пример
mediaSessionId	Идентификатор сессии	5a072377-73c1-4caf-abd3

Известные проблемы

1. Максимальная длина имени файла во всех актуальных файловых системах Linux ограничена 255 символами. При создании файла записи, имя будет сокращено до данного предела, включая расширение и номер части, если включена ротация.
2. При записи потоков, опубликованных в конференции, ротация будет автоматически отключена, в противном случае полученные файлы будет невозможно объединить.
3. Скрипт обработки записанных файлов требует повышения прав для копирования и других операций над файлами записей на виртуальных машинах Amazon

Симптомы: операции над записанными файлами не выполняются

Решение: использовать `sudo` для файловых операций и вызова внешних скриптов, если WCS установлен на виртуальной машине Amazon, например

```
sudo cp $SRC_FILE $DST_FILE
```