

В браузере с помощью Delight Player

Описание

Поток, опубликованный на WCS сервере, можно воспроизвести в браузерном VR-плеере, например, [Delight Player](#). Таким образом можно проигрывать поток в устройствах виртуальной и смешанной реальности, если на этом устройстве работает один из поддерживаемых браузеров. Отметим, что качество воспроизведения потока в устройстве VR будет тем выше, чем выше качество публикуемого потока.

Поддерживаемые платформы и браузеры

	Chrome	Firefox	Safari	Edge
Windows	✓	✓	✗	✓
Mac OS	✓	✓	✓	✓
Android	✓	✓	✗	✓
iOS	✓	✓	✓	✓

Технологии

- WebRTC
- HLS

Использование WebRTC

Поток в Delight Player можно воспроизвести двумя способами:

1. С помощью WebSDK
2. С помощью только JavaScript и HTML5

Использование возможностей WebSDK

Чтобы воспроизвести поток по WebRTC в Delight Player или любом другом стороннем JavaScript плеере, видеоэлемент страницы, в котором будет воспроизводиться поток, передается параметром `remoteVideo` в функцию WebSDK `session.createStream()`

```

session.createStream({
  name: document.getElementById('playStream').value,
  display: display,
  remoteVideo: video
})
...

```

Тестирование

1. Для теста возьмем:
2. WCS сервер
3. веб-приложение [Media Devices](#) для публикации потока в высоком разрешении
4. VR-плеер [Delight](#) для воспроизведения потока
5. Установим разрешение публикуемого потока 1920x1080

The screenshot shows a configuration panel with the following settings:

- Screen share:** off
- Size:** 1920 (width) and 1080 (height)
- FPS:** 30
- Bitrate:** min: 0, max: 0

6. Нажмем **Connect**, укажем имя потока **test** и нажмем **Publish**

The screenshot shows the **Media Devices** interface with the following details:

- Local (Left):**
 - Video stats:** Bytes sent: 8460431, Packets sent: 7719, Frames encoded: 1274
 - Audio stats:** Bytes sent: 200204, Packets sent: 2237
 - Resolution: 1920x1080
- Preview (Right):**
 - Video stats:** Bytes received: 8422719, Packets received: 6870, Frames decoded: 1247
 - Audio stats:** Bytes received: 197275, Packets received: 2202
 - Resolution: 1920x1080
- WCS URL:** wss://test1.flashphoner.com:8
- Status:** PUBLISHING

7. Воспроизводим поток в VR-плеере



Угол зрения в браузере на ПК можно менять мышью, на iOS или специализированном VR-устройстве угол зрения меняется в зависимости от положения в пространстве.

Пример кода страницы с плеером

1. Объявление видеоэлемента для воспроизведения потока, поля ввода имени потока и кнопок запуска и остановки воспроизведения

```
<div style="width: 50%;" id="display">
  <dl8-live-video id="remoteVideo" format="STEREO_TERPON">
    <source>
  </dl8-live-video>
</div>
<input class="form-control" type="text" id="playStream"
placeholder="Stream Name">
<button id="playBtn" type="button" class="btn btn-default"
disabled>Play</button>
<button id="stopBtn" type="button" class="btn btn-default"
disabled>Stop</button>
```

2. Обработка события готовности плеера к воспроизведению

```
document.addEventListener('x-dl8-evt-ready', function () {
  dl8video = document.getElementById('remoteVideo');
  $('#playBtn').prop('disabled', false).click(function() {
    playStream();
  });
});
```

3. Установка соединения с сервером и создание потока

```
var video = dl8video.contentElement;
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED,
function (session) {
    var session = Flashphoner.getSessions()[0];
    session.createStream({
        name: document.getElementById('playStream').value,
        display: display,
        remoteVideo: video
    }).on(STREAM_STATUS.PLAYING, function (stream) {
        ...
    }).play();
})
```

4. Запуск воспроизведения в VR-плеере и обработка нажатия кнопки остановки воспроизведения

```
...
session.createStream({
    ...
}).on(STREAM_STATUS.PLAYING, function (stream) {
    dl8video.start();
    $('#stopBtn').prop('disabled', false).click(function() {
        $('#playBtn').prop('disabled', false);
        $('#stopBtn').prop('disabled', true);
        stream.stop();
        dl8video.exit();
    });
}).play();
...
```

 [Full source code of the sample VR player page](#)



Использование возможностей JavaScript и HTML5

Чтобы воспроизвести поток по WebRTC в Delight Player или любом другом стороннем JavaScript плеере, на странице создается псевдоэлемент для вывода потока

```
var mockRemoteDisplay = $('<div></div>');
var mockRemoteVideo = $('<video></video>', {id: 'mock-REMOTE_CACHED_VIDEO'});
mockRemoteDisplay.append(mockRemoteVideo);
```

Псевдоэлемент `mockRemoteDisplay` передается параметром `display` в функцию WebSDK `session.createStream()`, а элемент `mockRemoteVideo` передается как источник потока в VR-плеер

```
session.createStream({
    name: $('#streamName').val(),
```



```

display: mockRemoteDisplay.get(0)
}).on(STREAM_STATUS.PLAYING, function (stream) {
    var srcObject = mockRemoteVideo.get(0).srcObject;
    video.srcObject = srcObject;
    dl8video.start();
    ...
}).play();

```

Тестирование

1. Для теста возьмем:

- WCS сервер - веб-приложение [Media Devices](#) для публикации потока в высоком разрешении - VR-плеер [Delight](#) для воспроизведения потока

1. Установим разрешение публикуемого потока 1920x1080

The screenshot shows a configuration panel with the following settings:

- Screen share:** off
- Size:** 1920 (width) and 1080 (height)
- FPS:** 30
- Bitrate:** min (0) and max (0)

2. Нажмем **Connect**, укажем имя потока **test** и нажмем **Publish**

The screenshot shows the Media Devices interface with the following details:

- Local (Left):**
 - Video stats:** Bytes sent: 8460431, Packets sent: 7719, Frames encoded: 1274
 - Audio stats:** Bytes sent: 200204, Packets sent: 2237
 - Resolution: 1920x1080
- Preview (Right):**
 - Video stats:** Bytes received: 8422719, Packets received: 6870, Frames decoded: 1247
 - Audio stats:** Bytes received: 197275, Packets received: 2202
 - Resolution: 1920x1080
- WCS:** wss://test1.flashponer.com:8
- Status:** PUBLISHING

3. Воспроизводим поток в VR-плеере



Пример кода страницы с плеером

1. Объявление видеоэлемента для воспроизведения потока, поля ввода имени потока и кнопок запуска и остановки воспроизведения

```
<div style="width: 50%;">
  <dl8-live-video id="remoteVideo" format="STEREO_TERPON" muted="true">
    <source>
  </dl8-live-video>
</div>
<input class="form-control" type="text" id="streamName"
placeholder="Stream Name">
<button id="playBtn" type="button" class="btn btn-default"
disabled>Play</button>
<button id="stopBtn" type="button" class="btn btn-default"
disabled>Stop</button>
```

2. Обработка события готовности плеера к воспроизведению

```
document.addEventListener('x-dl8-evt-ready', function () {
  dl8video = $('#remoteVideo').get(0);
  $('#playBtn').prop('disabled', false).click(function() {
    publishStream();
  });
});
```

3. Создание псевдоэлементов для воспроизведения потока

```
var mockRemoteDisplay = $('<div></div>');
var mockRemoteVideo = $('<video></video>', {id: 'mock-
REMOTE_CACHED_VIDEO'});
mockRemoteDisplay.append(mockRemoteVideo);
```

4. Установка соединения с сервером и создание потока

```
var video = dl8video.contentElement;
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED,
function (session) {
  var session = Flashphoner.getSessions()[0];
  session.createStream({
    name: $('#streamName').val(),
    display: mockRemoteDisplay.get(0)
  }).on(STREAM_STATUS.PLAYING, function (stream) {
    ...
  }).play();
})
```

5. Запуск воспроизведения в VR-плеере и обработка нажатия кнопки остановки воспроизведения

```
...
session.createStream({
  ...
}).on(STREAM_STATUS.PLAYING, function (stream) {
  var srcObject = mockRemoteVideo.get(0).srcObject;
  video.srcObject = srcObject;
  dl8video.start();
  mockRemoteVideo.get(0).pause();
  mockRemoteVideo.get(0).srcObject = null;
  $('#stopBtn').prop('disabled', false).click(function() {
    stream.stop();
    $('#playBtn').prop('disabled', false);
    $('#stopBtn').prop('disabled', true);
    dl8video.exit();
  });
}).play();
...
```

 [Full source code of the sample VR player page](#)



Использование HLS

В тех случаях, когда с воспроизведением потока в Delight Player по WebRTC возникают проблемы, можно проиграть поток по HLS

Тестирование

1. Для теста возьмем:

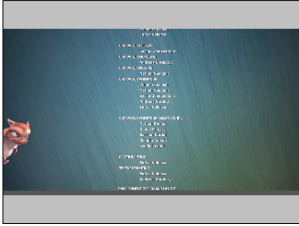
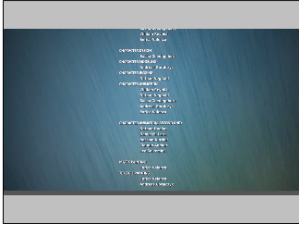
- WCS сервер - веб-приложение [Media Devices](#) для публикации потока в высоком разрешении - VR-плеер [Delight](#) для воспроизведения потока

1. Установим разрешение публикуемого потока 1920x1080

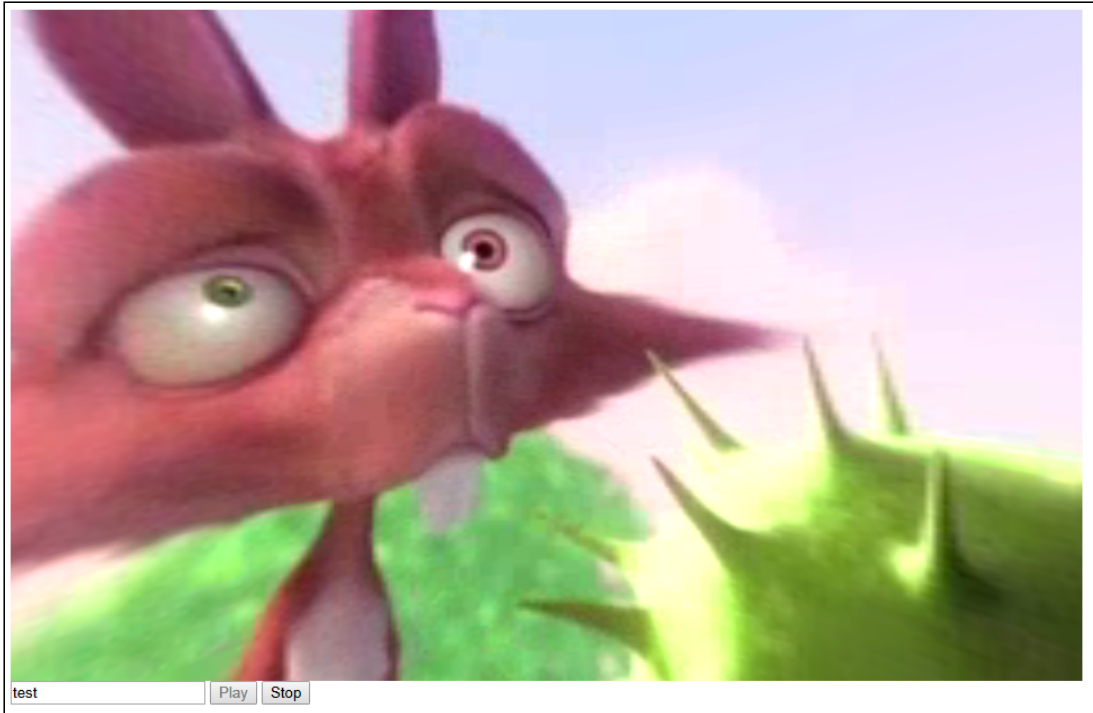
Screen share	<input type="checkbox"/>	<input type="button" value="off"/>
Size	<input type="text" value="1920"/>	<input type="text" value="1080"/>
FPS	<input type="text" value="30"/>	
Bitrate	<input type="text" value="min"/>	<input type="text" value="max"/>
	<input type="text" value="0"/>	<input type="text" value="0"/>

2. Нажмем **Connect**, укажем имя потока **test** и нажмем **Publish**

Media Devices

Video stats Bytes sent: 8460431 Packets sent: 7719 Frames encoded: 1274 Audio stats Bytes sent: 200204 Packets sent: 2237	Local  1920x1080	➔	Preview  1920x1080	Video stats Bytes received: 8422719 Packets received: 6870 Frames decoded: 1247 Audio stats Bytes received: 197275 Packets received: 2202
	WCS <input type="text" value="wss://test1.flashponer.com:8443"/> PUBLISHING			

3. Воспроизводим поток в VR-плеере



Пример кода страницы с плеером

1. Объявление видеоэлемента для воспроизведения потока, поля ввода имени потока и кнопок запуска и остановки воспроизведения

```
<div style="width: 50%;" id="display">
  <d18-live-video id="remoteVideo" format="MONO_360">
    <source type="application/x-mpegurl" id="hlsSource"/>
  </d18-live-video>
</div>
<input class="form-control" type="text" id="playStream"
placeholder="Stream Name">
<button id="playBtn" type="button" class="btn btn-default"
disabled>Play</button>
<button id="stopBtn" type="button" class="btn btn-default"
disabled>Stop</button>
```

2. Обработка события готовности плеера к воспроизведению

```
document.addEventListener('x-d18-evt-ready', function () {
  d18video = document.getElementById('remoteVideo');
  $('#playBtn').prop('disabled', false).click(playStream);
});
```

3. Получение URL сервера для воспроизведения HLS

```
var hlsUrl = getHLSUrl();
```

4. Запуск воспроизведения в VR-плеере и обработка нажатия кнопки остановки воспроизведения

```
...
var video = dl8video.contentElement;
var streamName = document.getElementById('playStream').value;
$('#hlsSource').attr("src",hlsUrl + "/" + streamName + "/" + streamName +
".m3u8");
dl8video.start();
$('#stopBtn').prop('disabled', false).click(function() {
    $('#playBtn').prop('disabled', false);
    $('#stopBtn').prop('disabled', true);
    dl8video.exit();
});
```

 [Full source code of the sample VR player page](#)



Известные проблемы

1. Поток не воспроизводится в Delight Player по WebRTC или воспроизводится с фризами.

Симптомы

Поток в Delight Player не воспроизводится вообще (MS Edge) либо воспроизводится с постоянными фризами (iOS Safari)

Решение

Использовать HLS для воспроизведения потока

2. VR-отображение не работает при воспроизведении потока в Delight Player по HLS в MS Edge на Windows 10 Mobile.

Симптомы

Поток в Delight Player воспроизводится по HLS, но картинка плоская

✓ **Решение**

Использовать устройство на актуальной операционной системе с поддержкой большего количества браузеров.

3. Поток не воспроизводится в браузере Safari по HLS

 **Симптомы**

Поток в Delight Player не воспроизводится, индикатор загрузки отображает 99%, затем черный экран, либо отображается ошибка CORS

✓ **Решение**

Использовать [nginx](#) в качестве обратного прокси для воспроизведения потока по HLS в Safari