

# Перехват и обработка декодированных кадров

- [Описание](#)
- [Реализация перехватчика](#)
  - [Отдельный каталог для собственных Java библиотек](#)
- [Тестирование](#)

## Описание

При [включенном транскодинге](#) существует возможность перехватывать декодированные кадры опубликованного потока в формате YUV и изменять их содержимое попиксельно на стороне сервера. Кадр, в который были внесены изменения, будет закодирован и отправлен в выходной поток транскодера.

## Реализация перехватчика

Для перехвата декодированных кадров необходимо разработать класс на языке Java, реализующий интерфейс `IDecodedFrameInterceptor`. Функция этого класса `frameDecoded()` будет получать декодированные кадры в формате YUV, например

### TestInterceptor.java

```
// Package name should be strictly defined as com.flashphoner.frameInterceptor
package com.flashphoner.frameInterceptor;

// Import decoded frame interceptor interface
import com.flashphoner.sdk.media.IDecodedFrameInterceptor;
// Import YUV frame description
import com.flashphoner.sdk.media.YUVFrame;

/**
 * Custom decoded frames interceptor implementation example
 * The example draws a cross over the picture
 */
public class TestInterceptor implements IDecodedFrameInterceptor {

    // Constants to parse pixel
    private final int Y = 0;
    private final int U = 1;
    private final int V = 2;

    // Dark colored pixel
    private final byte[] DarkPixel = new byte []{42, -128, -128};

    /**
     * Function to handle decoded frame
     * @param streamName - stream name
     * @param frame - decoded YUV frame
     */
    @Override
    public void frameDecoded(String streamName, YUVFrame frame) {
        // Get frame height
        int frameHeight = frame.getHeight();
        // Get frame width
        int frameWidth = frame.getWidth();
        // Declare cross lines padding
        int PADDING = 4;
        // Define frame center
        int frameCenterX = frameWidth / 2;
        int frameCenterY = frameHeight / 2;
        // Define vertical line bounds
        int leftBound = frameCenterX - PADDING;
        int rightBound = frameCenterX + PADDING;
        // Define horizontal line bounds
        int topBound = frameCenterY - PADDING;
        int bottomBound = frameCenterY + PADDING;

        // Walk through the frame pixels and draw a cross
```

```

        for (int x = 0; x < frameWidth; x++) {
            for (int y = 0; y < frameHeight; y++) {
                if (validateCoord(x, leftBound, rightBound) || validateCoord(y, topBound, bottomBound)) {
                    // Read the pixel
                    byte[] pixel = frame.readPixel(x, y);
                    // Modify the pixel
                    pixel[Y] = DarkPixel[Y];
                    pixel[U] = DarkPixel[U];
                    pixel[V] = DarkPixel[V];
                    // Write the pixel back
                    frame.writePixel(x, y, pixel);
                }
            }
        }
    }

    /**
     * Helper function to validate pixel drawing
     * @param coord - pixel coordinate
     * @param low - low coordinate bound
     * @param high - high coordinate bound
     * @return true if coordinate is valid
     */
    private boolean validateCoord(int coord, int low, int high) {
        return (coord > low && coord < high);
    }
}

```

Затем следует скомпилировать класс в байт-код. Для этого создаем дерево каталогов, соответствующее названию пакета написанного класса

```
mkdir -p com/flashphoner/frameInterceptor
```

и выполняем команду

```
javac -cp /usr/local/FlashphonerWebCallServer/lib/wcs-core.jar ./com/flashphoner/frameInterceptor
/TestInterceptor.java
```

Теперь упакуем скомпилированный код в jar-файл

```
jar -cf testlayout.jar ./com/flashphoner/frameInterceptor/TestInterceptor.class
```

и скопируем его в каталог, где размещены библиотеки WCS сервера

```
cp testinterceptor.jar /usr/local/FlashphonerWebCallServer/lib
```

Для того, чтобы использовать разработанный класс, необходимо указать имя его пакета в настройках файла [flashphoner.properties](#)

```
decoded_frame_interceptor=com.flashphoner.frameInterceptor.TestInterceptor
```

и перезапустить WCS.

## Отдельный каталог для собственных Java библиотек

Начиная со сборки [5.2.1512](#), Java библиотеки (jar файлы) должны помещаться в каталог `/usr/local/FlashphonerWebCallServer/lib/custom`

```
cp testlayout.jar /usr/local/FlashphonerWebCallServer/lib/custom
```

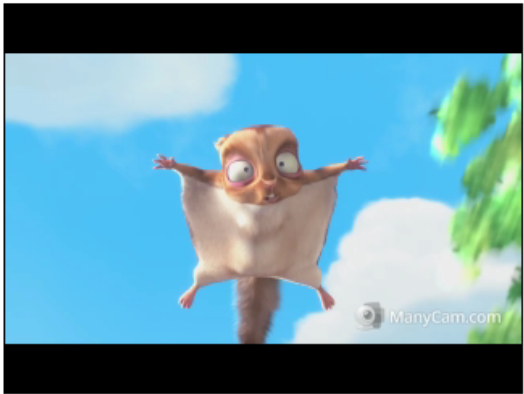
Этот каталог сохраняется при дальнейших обновлениях сервера к более новым сборкам. Таким образом, нет необходимости снова копировать jar файлы после установки обновления.

# Тестирование

1. Опубликуйте поток в примере Two Way Streaming [https://test1.flashphoner.com:8444/client2/examples/demo/streaming/two\\_way\\_streaming/two\\_way\\_streaming.html](https://test1.flashphoner.com:8444/client2/examples/demo/streaming/two_way_streaming/two_way_streaming.html), где test1.flashphoner.com - адрес WCS сервера


## Two-way Streaming

Local



testStop

Player



0f45PlayAvailable

PUBLISHING

wss://test1.flashphoner.com:8443Disconnect

ESTABLISHED

2. Проиграйте поток в примере Player с указанием разрешения, чтобы включился транскодинг, например <https://test1.flashphoner.com:8444/client2/examples/demo/streaming/player/player.html?resolution=320x240>, где test1.flashphoner.com - адрес WCS сервера

## Player



WCS URL

wss://test1.flashphoner.com:8443

Stream

test

Volume



Full Screen



PLAYING

Stop

На изображении будут видны измененные пиксели.