

# Maintenance Manual

## Synchro in the Wild

The project contains three main parts : aside scripts, Server and Web client.

### 1 Scripts

This part is composed of 5 files:

- exTimecode.sh
- startServer.sh
- parseTimecode.cpp
- parseTimecode.h
- Makefile

The first script extracts the informations from ffmpeg and stores them in a temporary file.

The makefile compiles the parser.

The parser scans the 3 files extracted from the script above and stores the following values in the file 'data.xml' placed in the 'public' repository:

- video name
- frame rate for each video
- time code for each video
- duration for each video

If you want to get more information, change the file "parseTimecode.cpp" and "parseTimecode.h". To have a model of ffmpeg information text, run the command "ffmpeg -i <video name>".

Currently parser uses 4 arguments, the 3 video files and an another file. The fourth argument is used to retrieve the video name. (used exclusively for server) This argument could be delete if you change method to access video name.

The script "info.sh" automatically starts all the steps mentioned above. If you change the access method for the name, you should check if command rm is necessary.

## 2 Server

The server use express, a framework for nodejs. If you choose to don't use it, you should change all structure of "app.js".

The file "app.js" is the main file for the server. It configures all routes. To add a new route or modify a route, you should modify this file. 4 routes exist: '/' to access upload page, '/synchro' to access video player page, '/upload' to upload video in server, '/preview' to access preview page.

To modify upload method, change the route '/upload'. You could change path to upload video.

## 3 Web Client

The Web Client has three views : upload.html, player.html and preview.html. You will find the upload functions used in upload.html in the file 'public/javascript/upload.js'. For the two other pages, the functions are in 'public/javascript/controls.js'. These functions are principally needed to control the videos but some help to load the data.xml to get the informations about videos (ie frame rate, name ...) or to save the data in a new XML file.

The design of the pages are all in one CSS file, 'public/css/display.css'.

A specific point is the function save() in 'controls.js', if you want to get more informations in the final xml file than just name, frame rate, synchronization frame, starting frame and ending frame, you will need to modify it. The data are often stored in the localStorage which is a storage in the client side on the browser. After getting the informations you need, just modify the xml text.

The last point about this part is that you can test the views without any localhost server by changing the different paths with the ones commented underneath in each view. For instance '<link href="css/display.css" rel="stylesheet">' must become '<link href="../../public/css/display.css" rel="stylesheet">'.