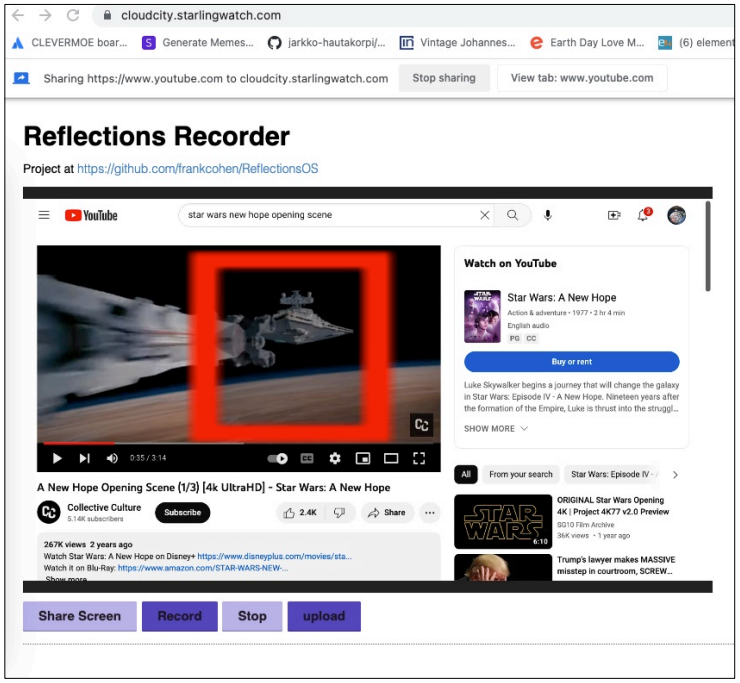
Reflections Recorder 4

Version: May 15, 2023

fcohen@starlingwatch.com, (408) 364-5508



Recorder APIs:

https://server/ - recorder

https://server/files - show list of files in html view

https://server/listfiles - show json format of available files

https://server/<filename.tar> - serve the tar file

https://server/touch/<filename.tar> - touch the file

https://server/delete/<filename.tar>

Reflections Recorder is a utility to capture and create media archive files for playing on the Reflections entertainment platform. It records video and audio using WebRTC in a browser window and uploads it to a server over HTTPS. Recorder's server side component processes the video into 240x240 pixel uncompressed mjpeg format, packs the video and audio into a TAR file, creates a Show manifest, and provides file transfer services to the Reflections device.

Recorder source is in Github at https://github.com/frankcohen/ReflectionsOS/tree/main/Experiments/Recorder4 and on the AWS EC2 server at /var/www/node-api/public/ See Reflections Cloud City 4 Setup document for details.

Reflections Recorder 4 project changes:

1) Uses browser cookie to remember the location and size for selection area between sessions.

2) Finishes server migration to a new AWS EC2 instance (from Lightsail): Node.js based file services work again, pm2 does not restart the node app on upload or file changes, install any missing server side libraries.

3) Adds /delete/filename.tar to delete files.

4) Displays selection area with new styling and provides new control to change selection are size and location.

5) Restores aspect ratio support (see commented-out source code)

New selection area styling and control

Selection box appears at 40 x, 40 y, and 240 height and 240 width by default.

Selection box is fill 25% opaque white, with stroke 3 pixel of #FEFACD

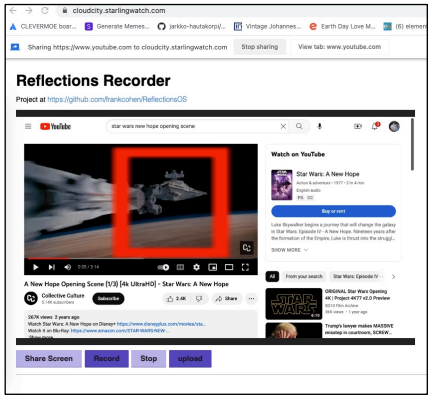
Click within the selection box to drag to new x and y position

Click and drag sizing icon (lower right) to change selection area size, selection area is always a square (x = y). Minimum selection area size is 50x50 pixels.

Architecture

Version: May 15, 2023

fcohen@starlingwatch.com, (408) 364-5508



Notes:
This page uses the WebRTC API to capture video
https://www.w3.org/TR/mediacapture-streams/

Example code to capture video is at:
https://simpl.info/mediarecorder/

Note: Chrome only allows this to work over HTTPS https://developer.mozilla.org/docs/Web/API/MediaDevices/getUserMedia#Browser_compatibility.

User chooses the browser window, tab, or application window, and defines the cropping area to record.

Recorder's Web component uploads the entire recorded video and audio, then uses ffmeg to crop the video to the selection area.

Recorder is intended to run at a minimum on Chrome/Mac and Windows and AWS Linux.

https://cloudcity.starlingwatch.com/

Javascript-based HTTP Post to host://reflections/recorder
Uploads starwars.mjpeg (video and audio)

Stores starswars.mjpeg to the file system

Uses ffmpeg to split the audio into its own file, including downsampling the audio to:

sample rate of 8000 kHz
16 bits
stereo format
WAV format

ffmpeg -i "$i" -c:a libfdk_aac -b:a 8k "starwars.m4a"

and down-converting video to:
240x240 pixels, uncompressed

ffmpeg -i "$i" -vf "fps=15,scale=240:240:flags=lanczos,crop=240:in_h:(in_w-240)/2:0" -q:v 9 "starwars.mjpeg"

Creates TAR archive of video file, audio file, and show file (JSON format)

```
{
  "ReflectionsShow": {
    "title": "Reflections Recorder",
    "showname": "recorder",
      "version": "1",
        "events": {

      "1": {
        "type": "event",
        "name": "First sequence",
        "comment": "Plays audio and video",
        "sequence": {
          "playaudio": "StarWars.m4a",
          "playvideo": "StarWars.mjpeg"
          }
        }
      }
    }
}
```

http://starlingwatch.com/reflections/getshow

StarWars.tar

Reflections Recorder project records video and audio using WebRTC in a browser window, uploads it to a server over HTTP, processes the video, packs the video and audio into a TAR file, and provides file services.

Use case:

1) Browse https://cloudcity.starlingwatch.com
2) Enter a URL to any Web page. For example, https://www.youtube.com/watch?v=vLgsf8Pei6Q
3) A user changeable selection box appears in the capture area. Use the mouse to drag the box to determine the recording area. Use the mouse to click and drag the size icon (in the lower right corner of the capture area) to resize the capture area. The capture area is always square.
4) Click Record
5) Later click Stop
6) Click Save. The browser executes an HTTP Post to the Cloud City REST service to upload an MJPEG file (containing video and audio). REST service stores the MJPEG file to the server local file system.
7) REST service splits the audio into its own file (starwars.m4a) and video file (starwars.mjpeg).
8) Service creates a TAR file containing the audio, video, and show script.
9) Wrist watch makes HTTP REST getshow service requests. Cloud City replies with the newest TAR file (starwars.tar)
10) Wrist watch decodes the tar file and runs the audio and video files.