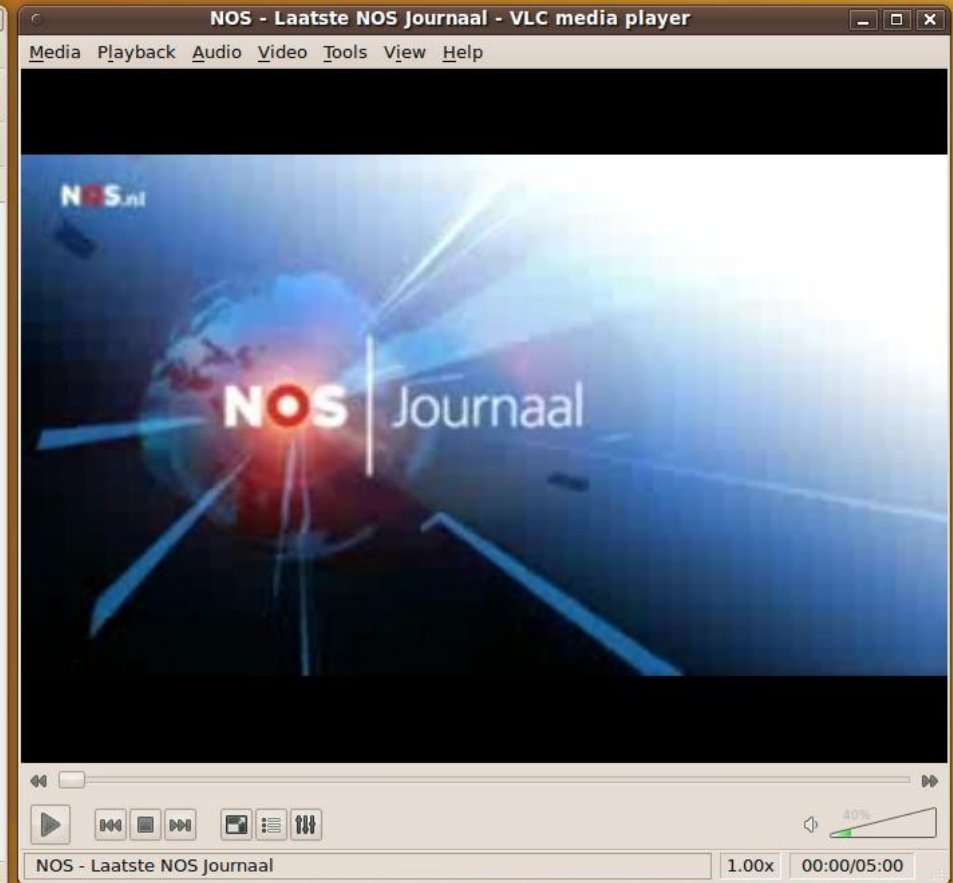
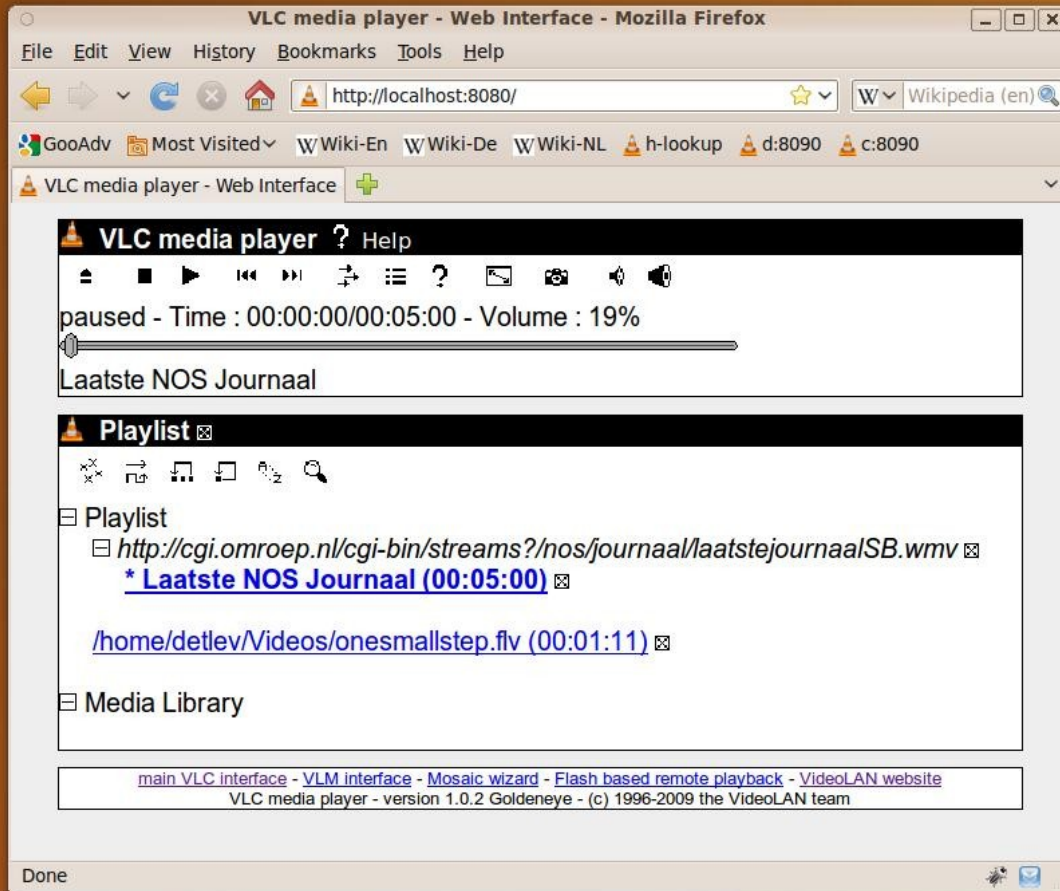


# VLC http interface

## Browser Interface vs. normal interface



# *Controlling VLC from Perl via http*

```
# Wrote a little module to do things like:
my $vlc = VlcHttp->new({host=>'localhost:8080'})

$vlc->in_play($file);           # start playing file
$vlc->wait_for_state('playing');

$vlc->pl_pause;                 # pause playback
$vlc->wait_for_state('paused');

$vlc->seek(23);                 # seek to 23s
$vlc->pl_pause;
$vlc->sleep(4);                 # play for 4s
$vlc->pl_pause;

$vlc->volume("30%");
$vlc->fullscreen; $vlc->snapshot;

# (Not yet definitive form here: VlcHttp.pm)
```

# *How does it work?*

- VLC has built-in http server
- nicely documented on:
  - <http://www.videolan.org/doc/vlc-user-guide/en/ch05.html>
- to enable the http interface:
  - `vlc -I http` (http instead of normal GUI)
  - `vlc --extraintf http` (http + normal GUI)
- by default on `http://localhost:8080`
- `--http-host <host>:<port>` for other url

## *Security: .hosts, .access*

- Remote access: requires entry in .hosts
  - <vlc-dir>/http/.hosts
  - add line like: 192.168.0.0/16
  - sometimes required reboot... (?)
- *Quite important*: allowed hosts will get some access to whole file system through the browser when vlc is running with http interface!
- .access for password protection by directory (line: "detlev:secret")

# *VLC as http server*

- document-root:
  - `<vlc-dir>/http`
- with `<vlc-dir>` typically being:
  - `/usr/share/vlc`
  - `C:\Program Files\VideoLAN\VLC`
- GUI is defined in:
  - `<vlc-dir>/http/index.html`
  - `<vlc-dir>/http/js/functions.js`

# *How GUI sends commands (AJAX)*

- index.html has special tags:
  - `<vlc id="include" param1="dialogs/main" />`
- expanded by vlc to dialogs containing buttons like:
  - `<button id="btn_pause" onclick="pl_pause();" ... title="Pause" >`
- java-script in functions.js makes request to:
  - `http://localhost:8080/requests/status.xml?command=pl_pause`

# *http-response*

- For requests to:
- <http://localhost:8080/requests/status.xml>  
(including any commands sent)
  - xml listing of **current status** (stop/playing/paused, position, length etc.)
- <http://localhost:8080/requests/playlist.xml>
  - xml listing of **currently loaded playlist** (including id number of clips, needed for some calls)
- List of main functions: [vlc\\_calls.html](#)

# *All the perl it takes to play clip 26*

demo03.pl:

```
use LWP::Simple;

$id = $ARGV[0] || 26;

$status_url =
    "http://localhost:8080/requests/status.xml";

$xml = LWP::Simple::get($status_url
    . "?command=pl_play&id=$id");
```

# *All the perl it takes to find current playback position*

demo04.pl:

```
use LWP::Simple;
use XML::DOM;
use XML::DOM::XPath;

$status_url =
    "http://localhost:8080/requests/status.xml";
$xml = LWP::Simple::get($status_url);

$parser = new XML::DOM::Parser;
$doc = $parser->parse($xml);
@nodes = $doc->findnodes( '/root/time' );
$time = $nodes[0]->getFirstChild->getNodeValue;

printf "time=%s\n", $time;
```

# *Uses...?*

- Personally I use it for:
  - language practice: quickly go back to positions I want to study / ask questions about
  - start playback of tv channels (online streams or DVB-T) on other computers from command line
- Might use it for:
  - taking snapshots at computed times
  - controlling volume / fullscreen mode remotely and/or from command line

*<The End>*