

VLC Play Howto

Alexis de Lattre

Anil Daoud

Benjamin Pracht

Clément Stenac

Jean-Paul Saman

VLC Play Howto

by Alexis de Lattre, Anil Daoud, Benjamin Pracht, Clément Stenac, and Jean-Paul Saman

Copyright © 2002-2006 the VideoLAN project

This document is the complete user guide for the client aspects of VLC.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. The text of the license can be found in the appendix. *GNU General Public License*.

Table of Contents

1. Introduction to VLC.....	1
Overview of the VideoLAN project	1
VLC media player	1
Getting some help	1
2. Installing VLC.....	3
Installing VLC.....	3
Uninstalling VLC	4
3. Basic use of VLC.....	6
General interface description	6
Simple play	8
Using the playlist	18
Hotkeys	23
Using subtitles	26
Video and audio filters	30
Basic troubleshooting.....	34
4. Advanced use of VLC.....	37
Use the command line	37
Advanced use of filters	41
The HTTP interface.....	43
Other control interfaces	44
The Mozilla plugin	46
A. GNU General Public License.....	54
Preamble.....	54
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION	54
How to Apply These Terms to Your New Programs.....	58
B. Building pages for the HTTP interface	60
Introduction	60
VLC macros	60
The RPN evaluator	61
The macros	63

Chapter 1. Introduction to VLC

Overview of the VideoLAN project

VideoLAN is a complete software solution for video streaming and playback, developed by students of the Ecole Centrale Paris (<http://www.ecp.fr>) and developers from all over the world, under the GNU General Public License (<http://www.gnu.org/copyleft/gpl.html>) (GPL). VideoLAN is designed to stream MPEG videos on high bandwidth networks.

VideoLAN was originally designed for network streaming but VideoLAN's main software, VLC media player has evolved to become a full-featured cross-platform media player.

More details about the project can be found on the VideoLAN Web site (<http://www.videolan.org/>).

VLC media player

Originally called VideoLAN Client, VLC media player is the main software of the VideoLAN solution.

VLC works on many platforms: Linux, Windows, Mac OS X, BeOS, *BSD, Solaris, Familiar Linux, Yopy/Linupy and QNX. It can play:

- MPEG-1, MPEG-2 and MPEG-4 / DivX files from a hard disk, a CD-ROM drive, ...
- DVDs, VCDs, and Audio CDs
- from satellite card (DVB-S),
- Several types of network stream : RTP/UDP Unicast, RTP/UDP Multicast, HTTP, RTSP, MMS, etc .
- From acquisition or encoding cards (on GNU/Linux and Windows only)

VLC can also be used as a streaming server. These aspects are detailed in the Streaming Howto (<http://www.videolan.org/doc/streaming-howto>)

The present guide describes all the client (playback) aspects of VLC

Getting some help

Documentation

The user documentation of VideoLAN is made up of 4 documents:

- the *Play-Howto*. This document is the complete guide of VLC as a client.
- the *Streaming Howto*. This document is the complete guide of VLC as a streaming server. It will give you practical examples to set up your streaming solution.
- the *VLS user guide*. This document is the complete guide for VLS (VideoLAN's legacy streaming server)
- The *VideoLAN FAQ*. This document contains Frequently Asked Questions about VideoLAN.

The latest version of these documents can be found on the documentation page (<http://www.videolan.org/doc/>).

User support

There are different methods to get support from The VideoLAN team. For more details, please visit our Support page (<http://www.videolan.org/support/>).

Chapter 2. Installing VLC

Installing VLC

There are VLC binaries available for the many OSes, but not for all supported ones. If there are no binaries for your OS or if you want to change the default settings, you can compile VLC from sources.

Windows

VLC works under Windows 95/98/ME/2000/XP. Download the self-extracting package from the VLC Windows download page (<http://www.videolan.org/vlc/download-windows.html>). Launch the `.exe` to install VLC.

Mac OS X

Download the Mac OS X package from the VLC MacOS X download page (<http://www.videolan.org/vlc/download-macosx.html>). Double-click on the icon of the package : an icon will appear on your Desktop, right beside your drive(s). Open it and drag the VLC application from the resulting window to the place where you want to install it (it should be `/Applications`).

BeOS

Download the Zip file from the VLC BeOS download page (<http://www.videolan.org/vlc/download-beos.html>). Unzip the file in a directory to install VLC.

Debian GNU/Linux

Debian stable (woody)

Add the following lines to your `/etc/apt/sources.list`:

```
deb http://download.videolan.org/pub/videolan/debian woody main
deb-src http://download.videolan.org/pub/videolan/debian woody main
```

Then, for a normal install, do:

```
# apt-get update
# apt-get install gnome-vlc libdvdcss2
```

Debian unstable (sid)

Add the following lines to your `/etc/apt/sources.list`:

```
deb http://download.videolan.org/pub/videolan/debian sid main
deb-src http://download.videolan.org/pub/videolan/debian sid main
```

Then, for a normal install, do:

```
# apt-get update
```

```
# apt-get install wxvlc libdvdcss2
```

Debian testing (sarge)

You should not be using Debian testing unless you perfectly know what you are doing. It is almost impossible to support Debian testing and there are no plans to do it. For more informations on Debian testing, please look: testing page (<http://www.debian.org/devel/testing>)

Linux Mandrake

There are VLC packages for Mandrake 9.1 and cooker.

To install them, add the following sources for either Mandrake 9.1 or Cooker (you can use Easy urpmi (<http://plf.zarb.org/~nanardon/>) for that): *contrib* from the core distribution and *plf* (Penguin Liberation Front) from the external add-ons.

Then install the required packages with **urpmi**:

```
# urpmi libdvdcss2 libdvdpplay0 wxvlc vlc-plugin-a52 vlc-plugin-ogg vlc-plugin-mad
```

Linux Redhat

Download the RPM package *vlc* and the packages listed in the *required libraries and codecs* section (the other packages are optional) from the VLC Red Hat download page (<http://www.videolan.org/vlc/download-redhat.html>) and put them all into the same directory.

Then install the RPM packages you have downloaded:

```
# rpm -U *.rpm
```

If you have not installed all the RPM packages included with your distribution, you may be asked to install a few of them first.

Compile the sources by yourself (for every other OS)

For more detailed information on compiling VLC by yourself, please have a look at the documentation on our developers website (<http://developers.videolan.org>).

Uninstalling VLC

Windows

Click on the *Uninstall VLC* icon that was created during installation

Mac OS X

Drag the VLC application to your trash can. You can also remove the configuration file and the cache files in `~/Library/Preferences/VLC/`. There is an AppleScript on the disk-image which lets you do this automatically.

BeOS

Delete the `vlc-version` directory. You can also remove the configuration file `/boot/home/config/settings/vlcrc`.

Debian GNU/Linux

Remove the packages that you installed:

```
# apt-get remove --purge wxvlc vlc-mad libdvdcss2 libdvbpsil
```

GNU/Linux Redhat, Mandrake and SuSE

Uninstall the RPM packages that you installed:

```
# rpm -e vlc-version vlc-mad-version wxvlc-version
libdvdcss2-version libdvbpsil-version
```

If you compiled VLC from sources

Go to the directory containing VLC sources and execute

```
# make uninstall
```

You can then remove the VLC sources.

Chapter 3. Basic use of VLC

General interface description

VLC has several interfaces: a cross-platform interface, for Windows and GNU/Linux, called wxWidgets, a MacOS X native interface and a skinnable interface for Windows and GNU/Linux.

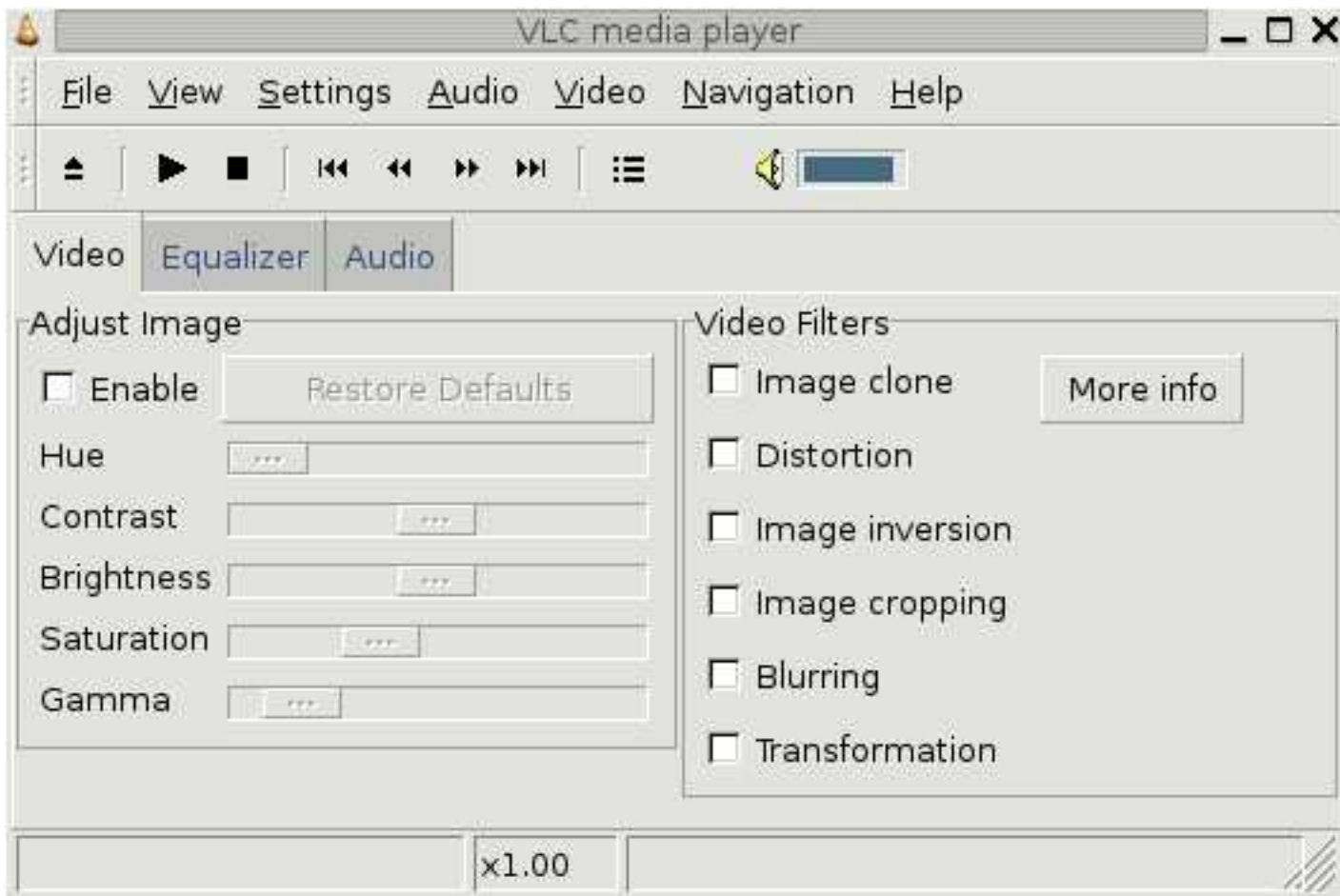
Windows and GNU/Linux

This is the default interface on Windows and GNU/Linux (the screenshot is done on GNU/Linux, but it would look quite the same on Windows).



The wxWidgets interface

This interface also features an *Extended GUI* which contains many additional features. To display or hide it, go to the *Settings* menu and click *Extended GUI*.



The wxWidgets interface with extended GUI

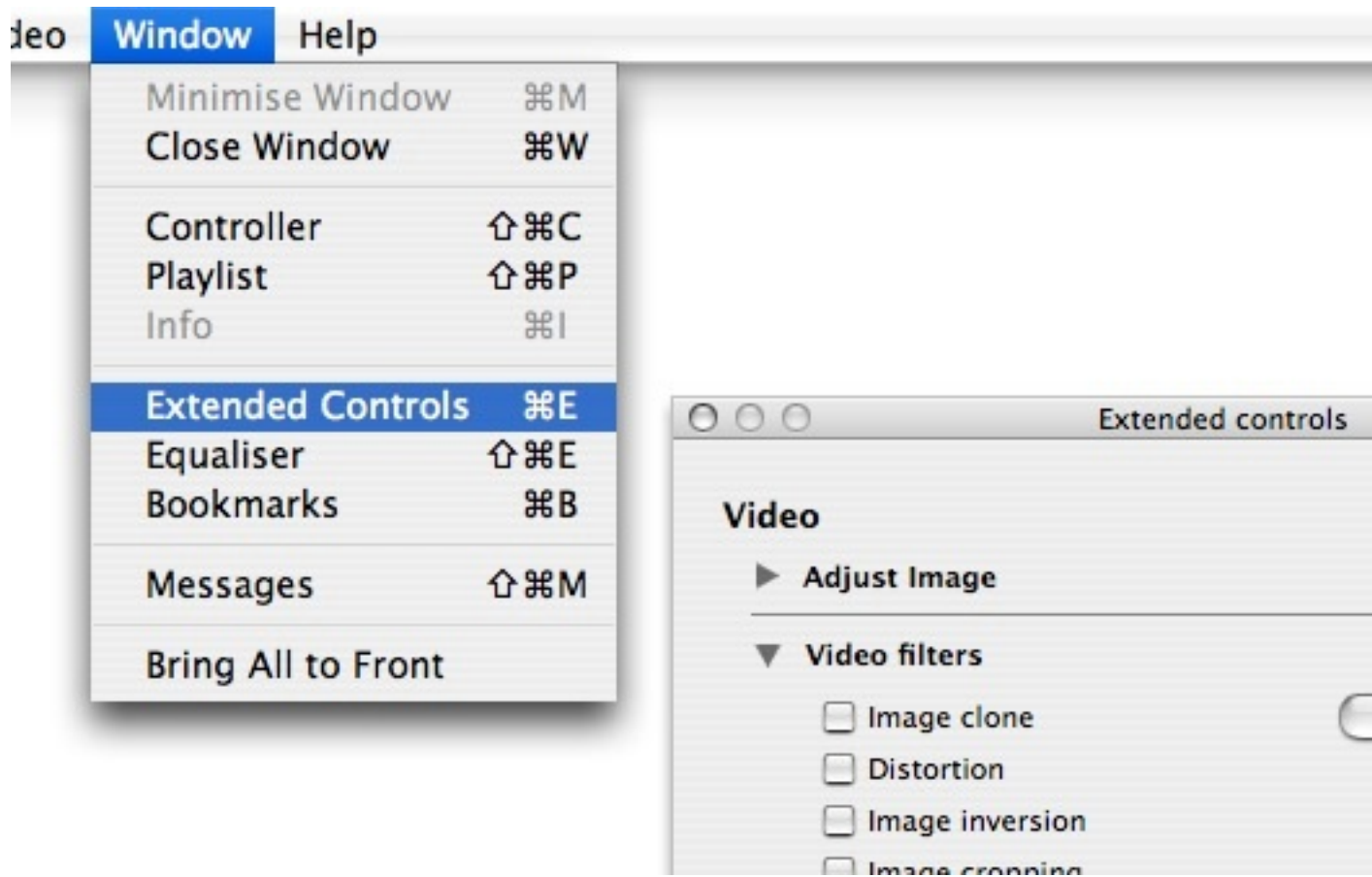
Native Mac OS X (Cocoa)

This is the default interface on Mac OS X.



The Mac OS X interface

This interface features an *Extended GUI* as well. It is called "Extended Controls" and can be opened through the Window-menu.



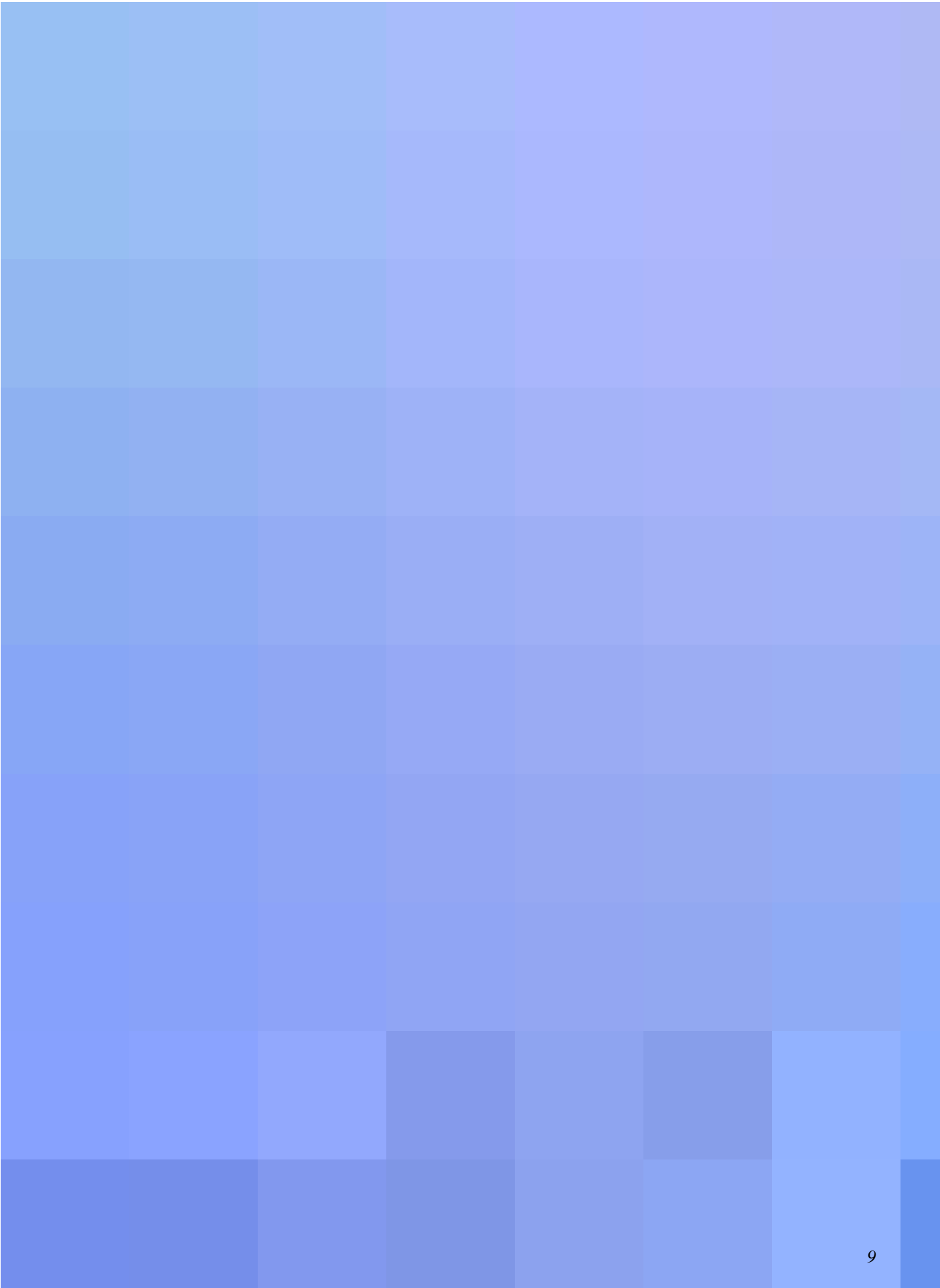
The Mac OS X interface with the extended controls panel

Simple play

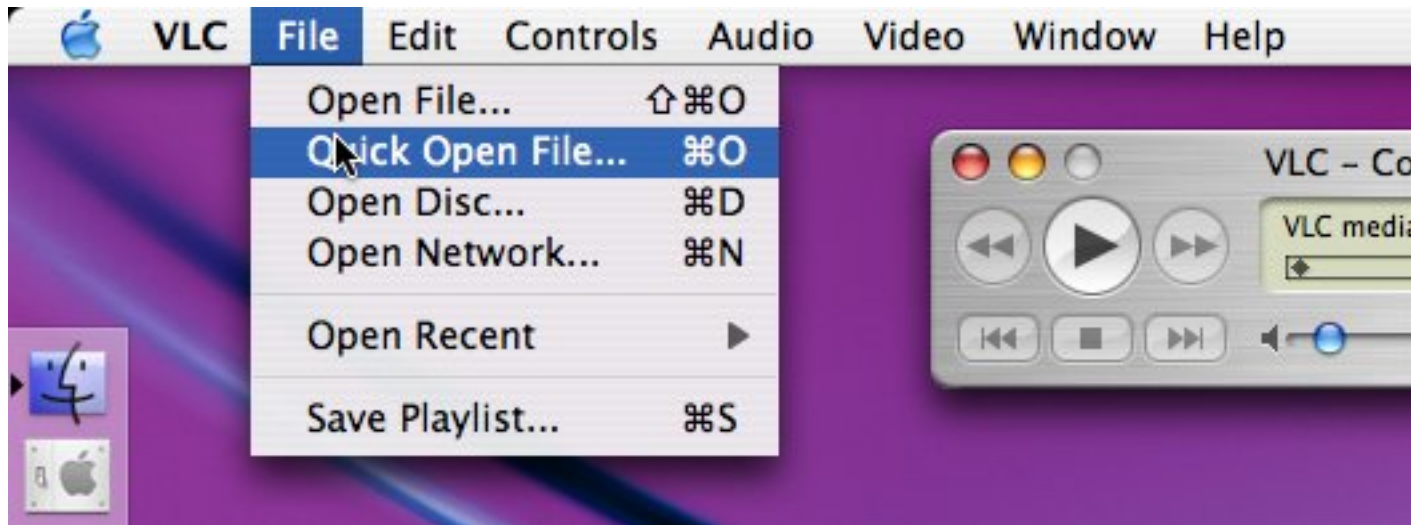
Play a file

To open a file, open the *File* menu, and select the *Quick Open File* menu item. An Open File dialog box will appear. Select the file you want to open, and select Open. VLC will start playing the selected file.

An alternative is to Drag'n'Drop your file on the VLC main interface or playlist window from the file explorer (Finder on MacOS X).

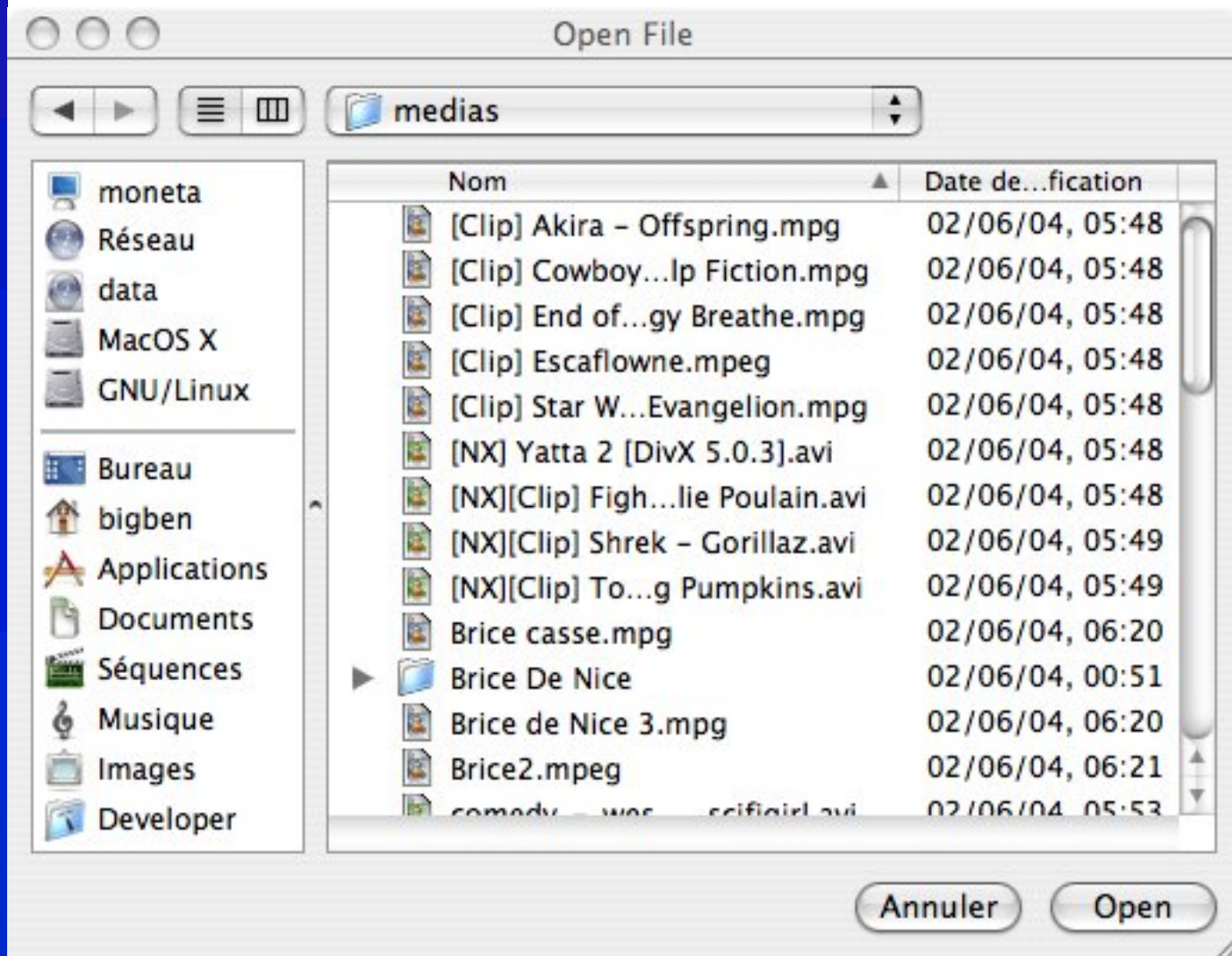


The File menu - wxWidgets interface



The File menu - MacOS X interface

The Open file dialog - wxWidgets interface



The Open file dialog - MacOS X interface

Play a CD/DVD/VCD

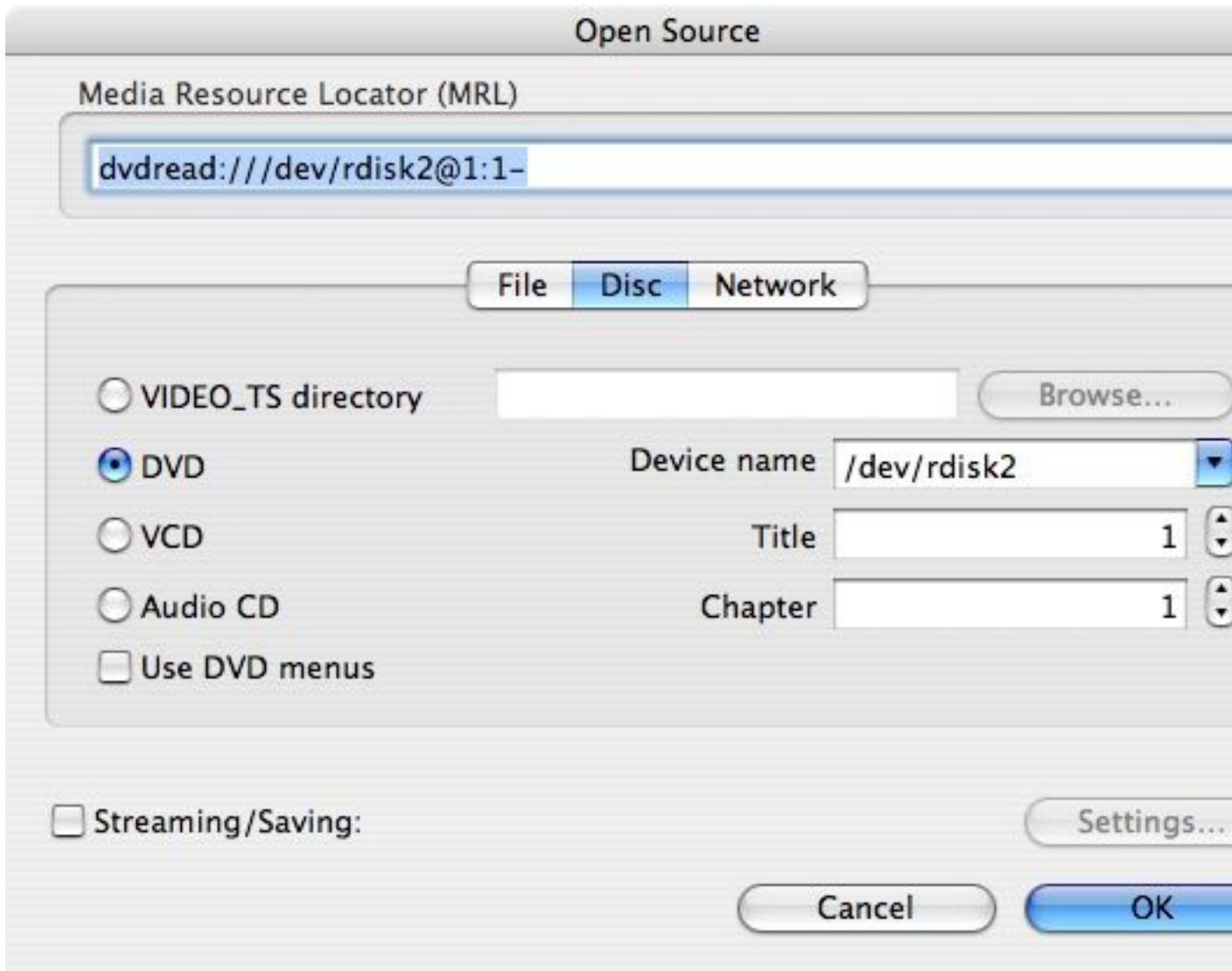
To Play a CD, VCD or a DVD, open the *File* menu, and select the *Open Disc...* menu item. In the Open Disk Dialog Box, select the type of media (DVD, VCD or Audio CD). When reading a DVD, you can enable DVD menus by selecting the *DVD (menus)* Disc type in the wxWidgets Interface. In the MacOS X interface, this can be done by selecting the "Use DVD menus" dialog box.

You can select the drive from which the media should be read by giving the appropriate drive letter or device name in the "Device Name" text input. This should be autodetected on MacOS X.

If you want to start the DVD or VCD playback from a given title and chapter instead of from the beginning, you can set it using the *Title* and *Chapter* selectors.

You can start playback by selecting the *Ok* button.

The Open disk dialog - wxWidgets interface



The Open disk dialog - MacOS X interface

Play a network stream (WebRadio, WebTV, ...)

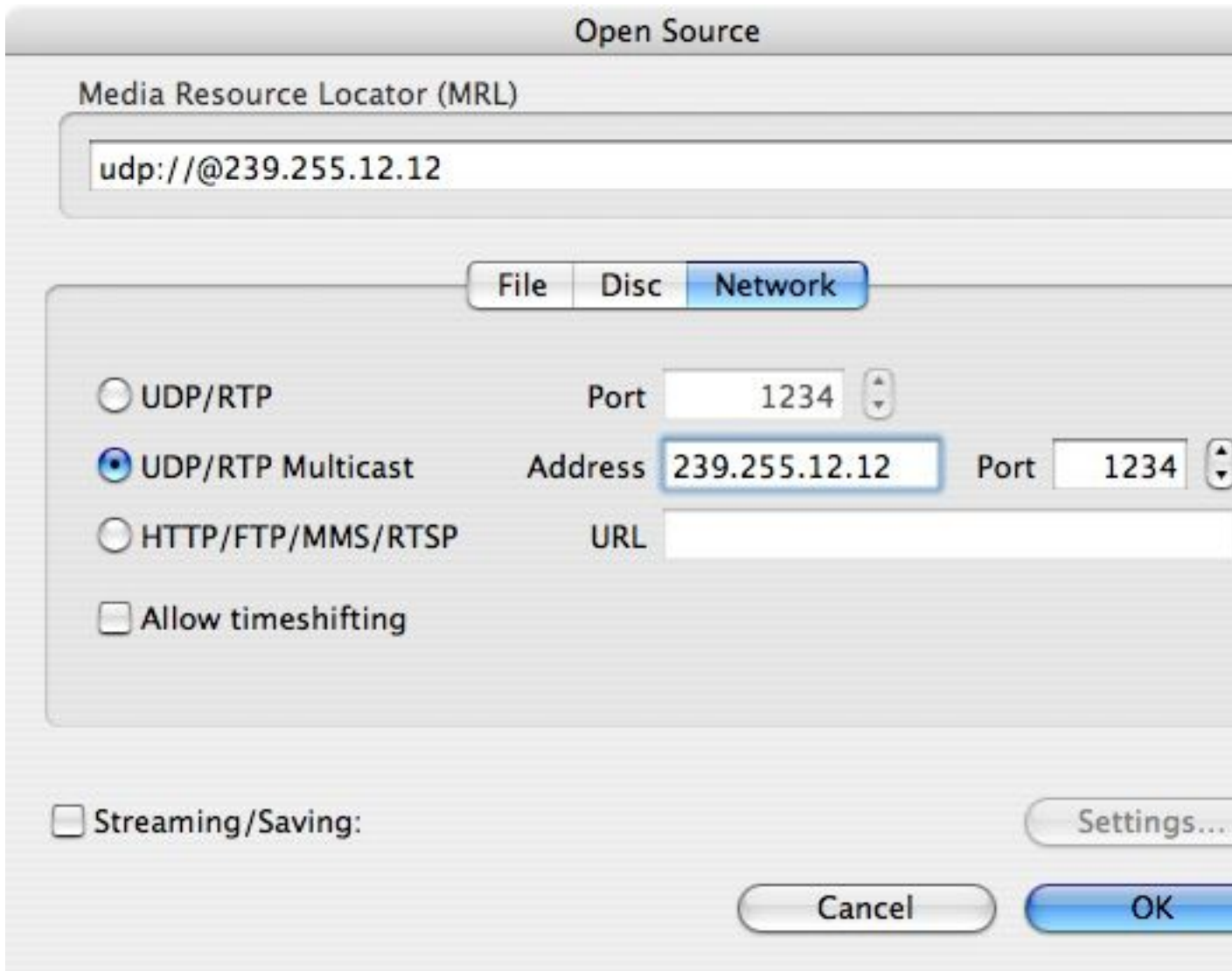
To open a network stream, open the "File" menu and select the "Open Network Stream" menu item.

- To open a UDP unicast stream, select *UDP/RTP*, and set the appropriate UDP port in the selector (it is 1234 for streams sent by a VLC or VLS server).
- To open a UDP multicast stream, select *UDP/RTP multicast*. Give the address of the multicast group in the "Address" text input, and select the appropriate UDP port.
- To open a stream sent over http (Webradios, WebTVs, Shoutcast, Icecast...), ftp, or mms (Microsoft Media Server), select "HTTP/FTP/MMS", and give the corresponding complete URL, (such as `http://live.stream.org:8080/live` or `mms://live.ms.stream.net:8080/live.asf`) in the corresponding text input. This also the way to open a RTSP stream with the MacOS X interface.
- To open a RTSP stream (sent by Darwin Streaming Server, VLC, etc), in the wxWidgets interface, select "RTSP" and give the URL in the text input.

You can start playback by selecting the *Ok* button.

If you get some stuttering during playback, you can try to increase the size of the read buffer. This can be done in the *Open Network Stream* dialog box, by selecting the *Caching* box. You can then choose the amount time (in milliseconds) VLC should store data in its buffer before starting playback.

The Open network dialog - wxWidgets interface



The Open network dialog - MacOS X interface

Play from an acquisition card

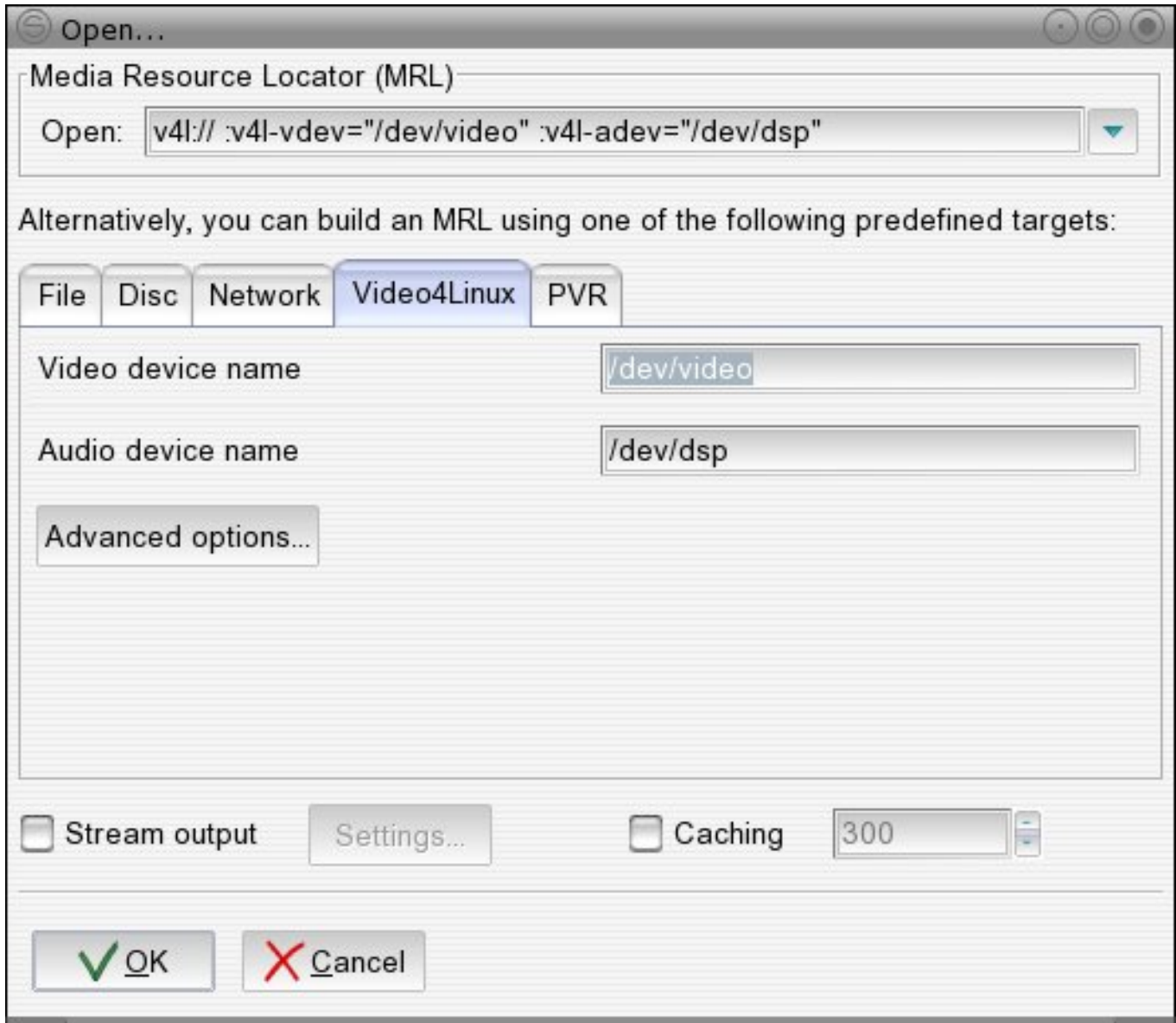
This currently only possible on Linux and Windows. Open the File menu, and select "Open Capture Device..."

On Windows, supported cards include webcams, TV cards, acquisition cards... provided they come with directshow compatible drivers (Almost all acquisition cards do). You can choose the device to use for video and audio capture using the "Video device name" and "Audio device name" selectors. If your device doesn't appear in the list, try to select the "Refresh list" button. You can access the settings of your acquisition device by selecting the *configure* button. Options here depend on the driver of the device. You can select the "Device Proprieties" box if you want the configuration dialog box of every device to be displayed after having pressed the *Ok* button. Select the *Tuner properties* box to be prompted for tuner settings (PAL/NTSC standard, frequency...) for TV cards. The *Advanced options...* button allows to select some further settings useful in some rare cases, such as the chroma of the input (the way colors are encoded) and the size of the input buffer.

The Open Capture device dialog and a device configuration windows - wxWidgets interface

On Linux, supported cards include webcams, TV cards, acquisition cards, provided they are supported by the Video4Linux architecture. Hauppauge PVR 250/350 cards are also supported, using the IVTV drivers (<http://ivtv.sourceforge.net/>).

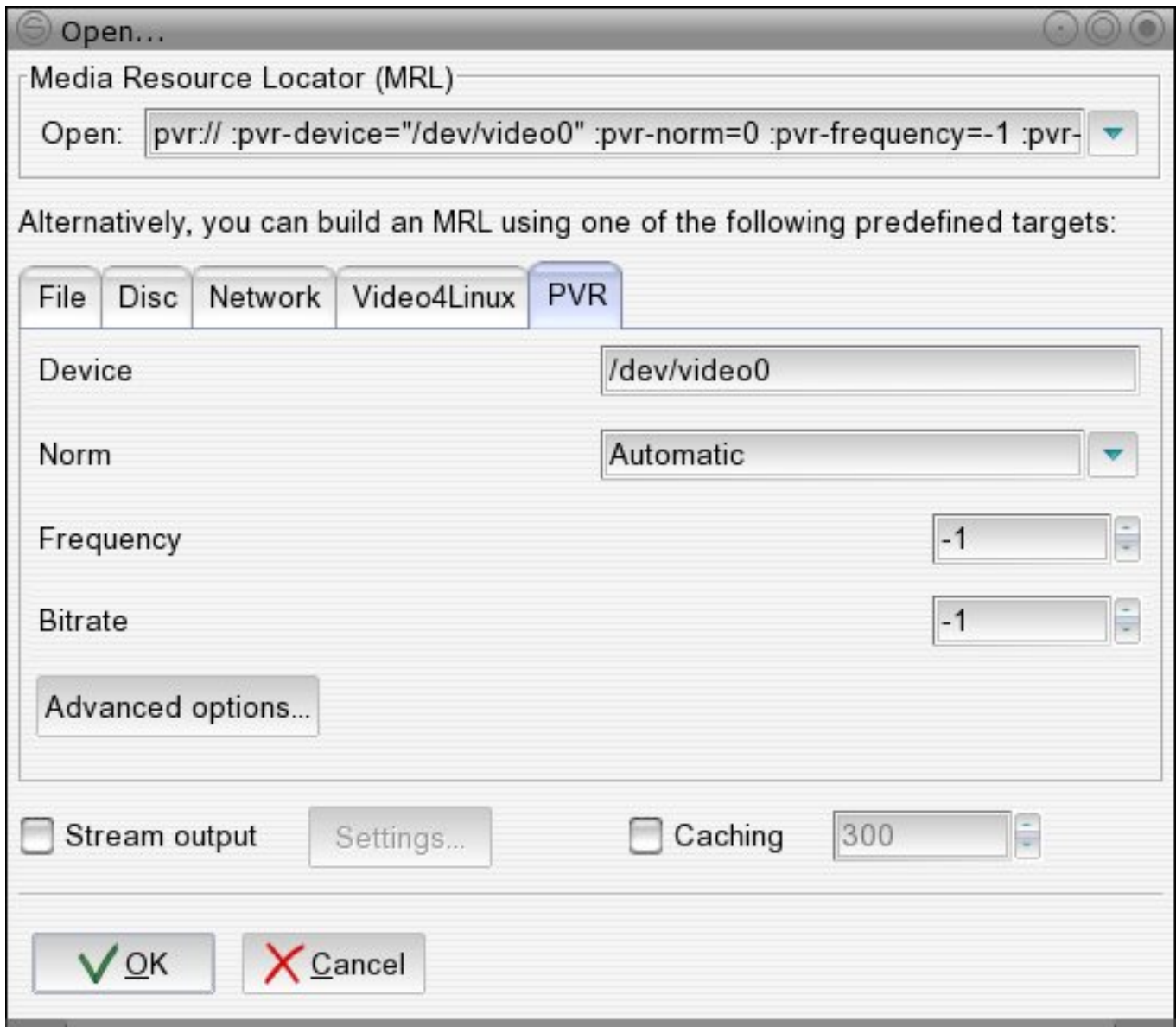
- For Video4Linux devices, you can set the name of the video and audio devices using the "Video device name" and "Audio device name" text inputs. The "Advanced options..." button allows to select some further settings useful in some rare cases, such as the chroma of the input (the way colors are encoded) and the size of the input buffer.



The Open Video4Linux dialog - wxWidgets interface

- To use a Hauppauge PVR card, select the PVR tab in the "Open" dialog box. Use the "Device" text input to set the device of the card you want to use. You can set the Norm of the tuner (PAL, SECAM or NTSC) by using the "Norm" Drop Down. The Frequency selector allows you to set the frequency of the tuner (in kHz), the bitrate selector to set the bitrate of the resulting encoded stream (in bit/s). The "Advanced Options" button allows to set some more settings, such as the size of the encoded video (in pixels), its framerate (in frame per second), the interval between 2 key frames, etc.

After having set all the required parameters, you can start the capture by selecting the "Ok" button.



The Open PVR dialog - wxWidgets interface

Using the playlist

VLC can of course store a list several files to play one after the other, using its playlist system. To access the playlist, click on the *Playlist* button on the main interface (see *General interface description*).

Each time you use the Open dialog box, the stream you select is appended at the end of the playlist and started.

The playlist window shows all the streams available in the playlist. Double-click one to have it played.

The Playlist - wxWidgets interface



The Playlist - MacOS X interface

Adding items, saving and loading playlists

In the wxWidgets interface, the *Manage* menu allows you to append an item at the end of the playlist (its playback won't start immediately), to save the playlist as a M3U or PLS file, or to import a playlist file.

In the MacOS X interface, saving a playlist can be done using the *Save Playlist...* function in the *File* menu. To import a playlist file, open it the same way as any other media file, using the *Quick Open File...* menu item.

Sorting

In the wxWidgets interface, *Sort* allows you to sort the playlist according to several criteria, or to shuffle it. You can also sort by clicking the header of the column

In the MacOS X interface, sorting can be done by clicking the header of the column matching the criteria you want to use for sorting.

Playlist modes

The playlist supports several playback modes.

In the wxWidgets interface, the toolbar contains three playlist mode buttons. They allow to enable random mode, to repeat the whole playlist or to repeat one item.

In the MacOS X interface, random mode can be enabled by selecting the *Random* box. A drop down menu allows you to enable playlist and item repeat modes.

Misc

Search

You also have a search tool. Enter a search string and hit search. The next item to match the string will be highlighted. Keep hitting Search to cycle between all matching items.

Moving items

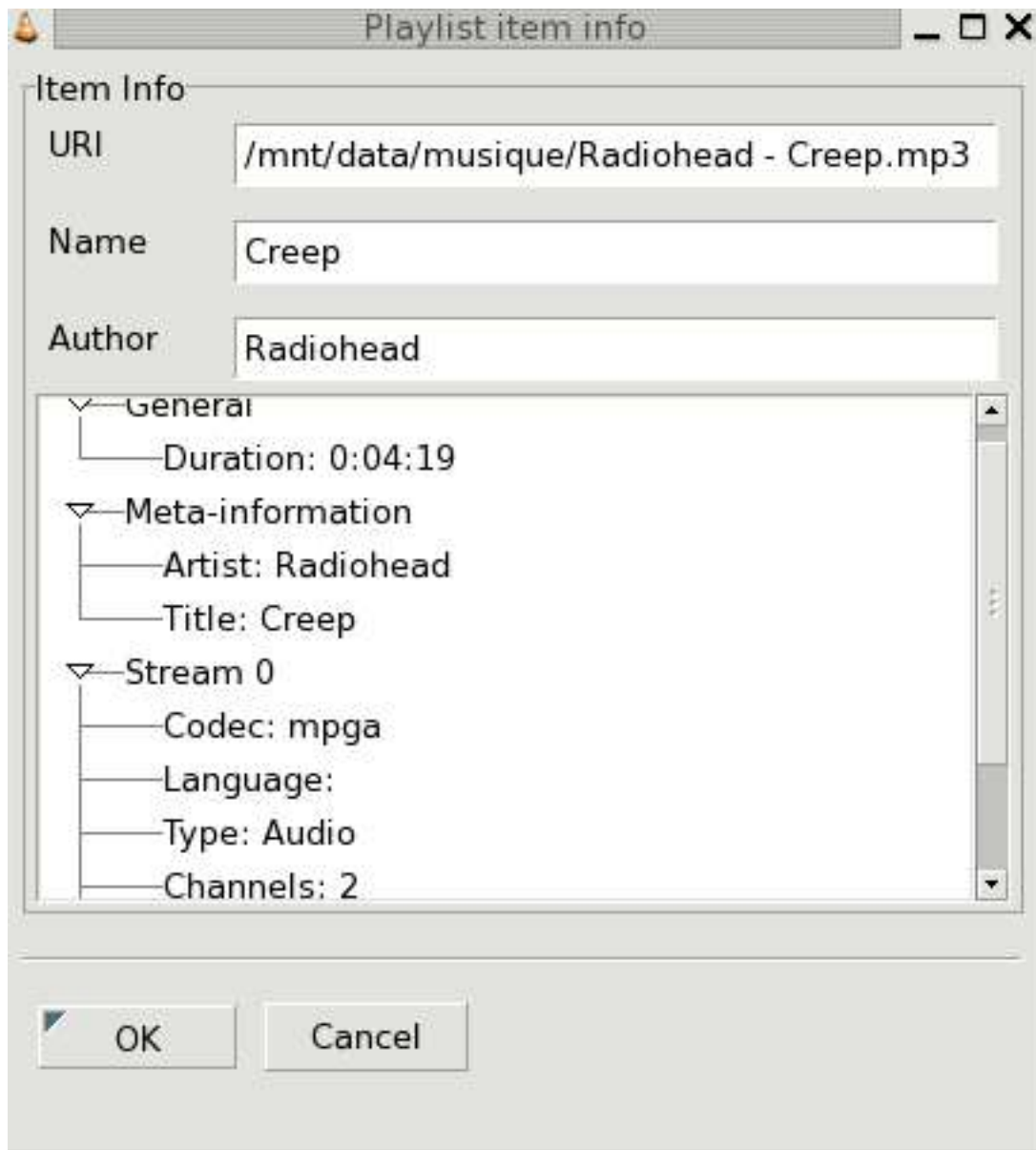
In the wxWidgets interface, the *Up* and *Down* buttons at the bottom of the playlist window allow you to move an item. Select an item and use these buttons to move it.

In the MacOS X interface, you can move an item with the mouse, easily using a *Drag'n'Drop*.

Contextual menu

By right-clicking an item, a contextual menu will appear, giving access to a number of functions (play the item, disable it, delete it, or get info on it).

If you ask for info, an *item info* dialog box will appear. This dialog box also allows you to change the name, the author and the location of the item to play.



Item Info Dialog - wx Interface



Item Info Dialog - MacOS X interface

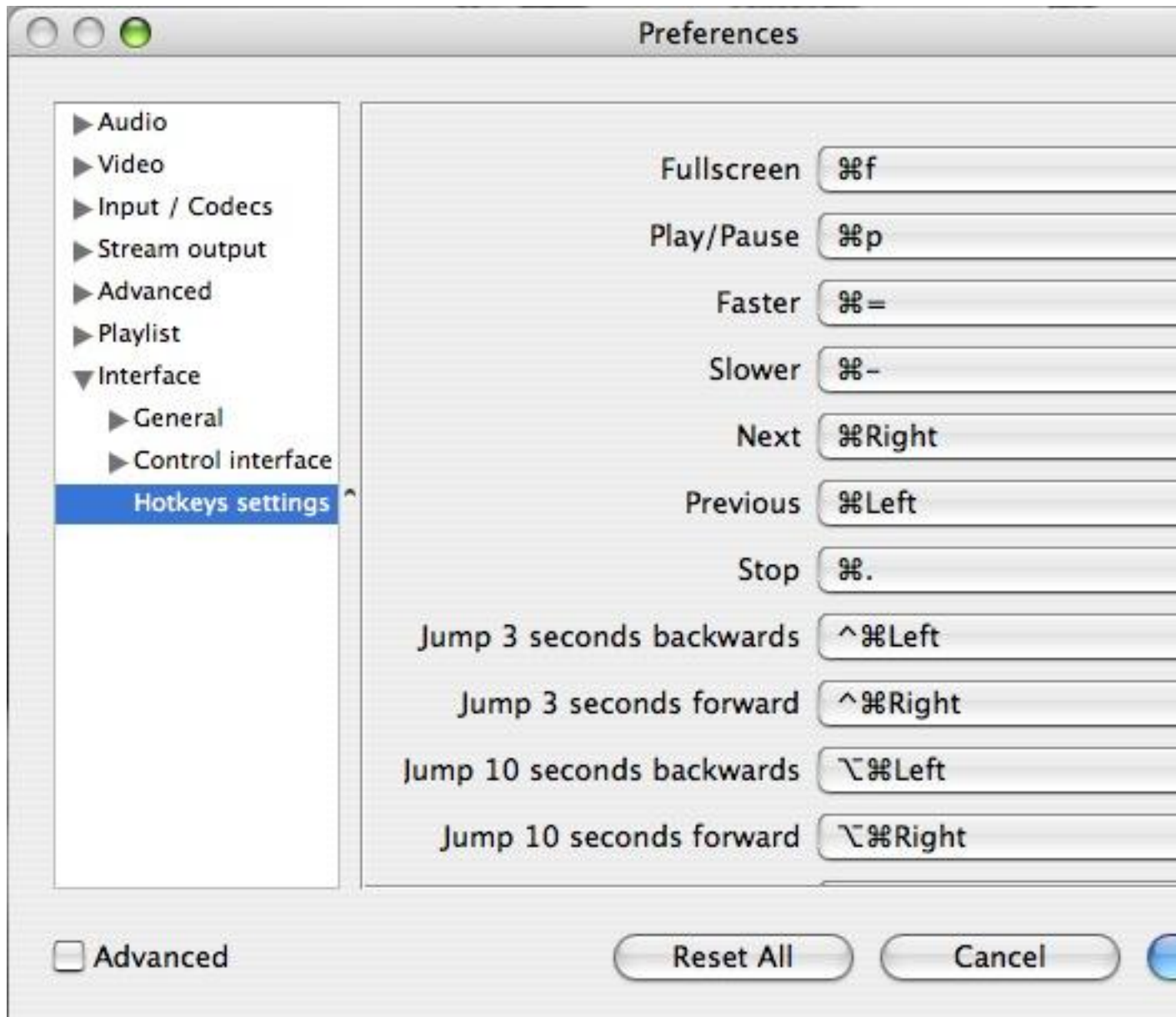
Hotkeys

Most of VLC functions are accessible using hotkeys

The list of the available hotkeys and their binding can be retrieved and altered in the preferences panel of the player. In the wxWidgets interface, preferences are available in the "Settings" menu, "Preferences" menu item. In the MacOS X interface, open the "VLC" menu, and select "Preferences". Select the "Hot keys" panel in the dialog. For each function of VLC, you get the bound hotkey in the dropdown in the right part of the panel. Several boxes gives the list of modifiers for the hotkey. To trigger an action using a hotkey, you need to press at the same time the keys corresponding to the different selected modifiers as well as the key set in the dropdown.

To change the binding of a hotkey, select or deselect boxes corresponding to the different modifiers, and change the key by using the dropdown. Select the *Save* button to apply the changes.

The Hotkeys Panel - wxWidgets interface



The Hotkeys Panel - MacOS X interface

Using subtitles

VLC supports many kinds of subtitles.

Media with included subtitles

Many types of media can have embedded subtitles. VLC can read subtitles for the following media:

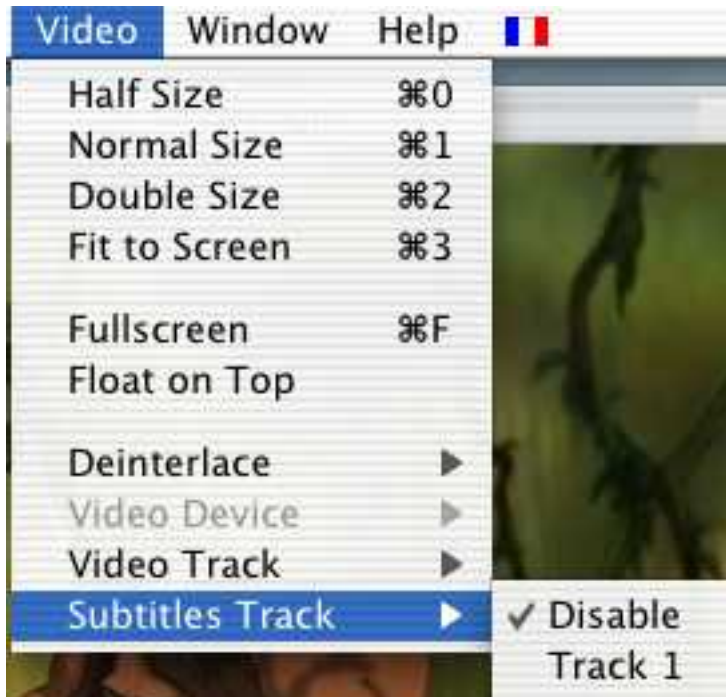
- DVD
- SVCD
- OGM files

- Matroska (MKV) files

Subtitles are disabled by default. To enable them, go to the *Video* menu, and to *Subtitles track*. All available subtitles tracks will be listed. Select one to get the subtitles. Depending on the media, a description (language, for example) might be available for the track.



Select a subtitles track under Windows or Linux



Select a subtitles track under MacOS X

DVD and SVCD subtitles are merely images, so you won't be able to change anything for them. OGM and Matroska subtitles are rendered text, so you will be able to change several options.

Text rendering options can be changed in the Preferences. In the *Modules* section, *text renderer* subsection, open the *freetype* page. You can then set the font and its size. For the font, you have to select a font file. Under Windows, they can be found in *C:\Windows\Fonts*. Under MacOS X, they are in */System/Library/Fonts*. Size can be set either relatively or as a number of pixels.

You need to restart your stream for the font modifications to take effect.

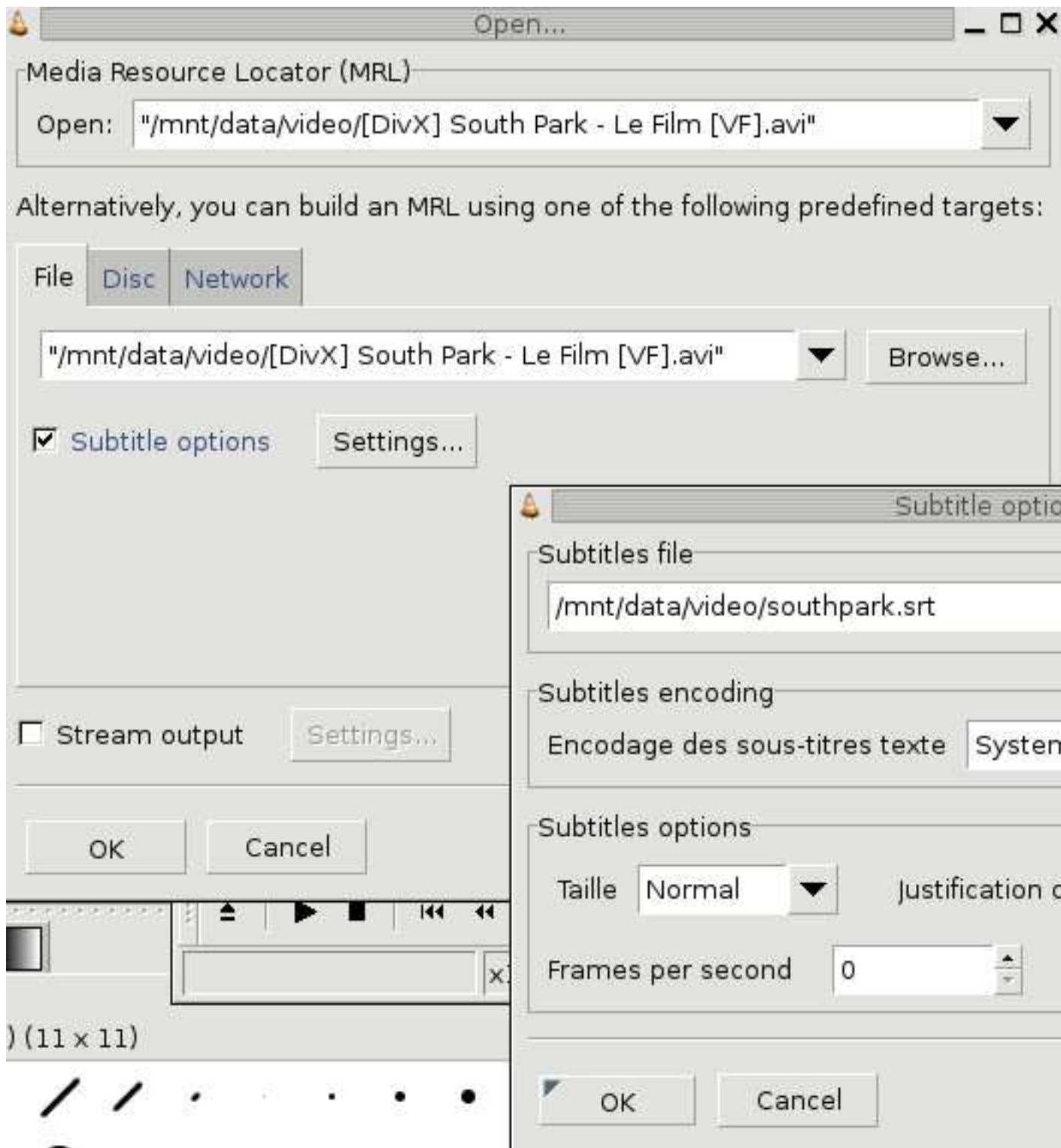
Subtitles files

While modern file formats like Matroska or OGM can handle subtitles directly, older formats like AVI can't. Therefore, a number of subtitles files formats have been created. You need two files: the video file and the subtitles files that only contains the text of the subtitles and timestamps.

VLC can handle these types of subtitles files:

- MicroDVD
- SubRIP
- SubViewer
- SSA
- Sami
- Vobsub (this one is quite special: it is not made from text but from images, which means that you can't change the fonts)

To open a subtitles file, use the Advanced Open dialog box (Menu File, Open file). Select your file by clicking on the *Browse* button. Then, check the *Subtitle options* checkbox and click on the Settings button.



Select a subtitles file under Windows or Linux

You can then select the subtitles file by clicking the *Browse* button. You can also set a few options like character encoding, alignment and size. The delay option allows you to delay the subtitles against the video if they are not in sync. If they are not at the same speed, you might also want to adjust the subtitles framerate.

Note: For Vobsub subtitles, you need to select the `.idx` file, not the `.sub` file

Encoding, alignment and size won't have any effect for Vobsub subtitles.

Font can be changed as explained in the previous section.

Video and audio filters

VLC includes a system of *filters* that allow you to modify the audio and video.

Deinterlacement and Post Processing

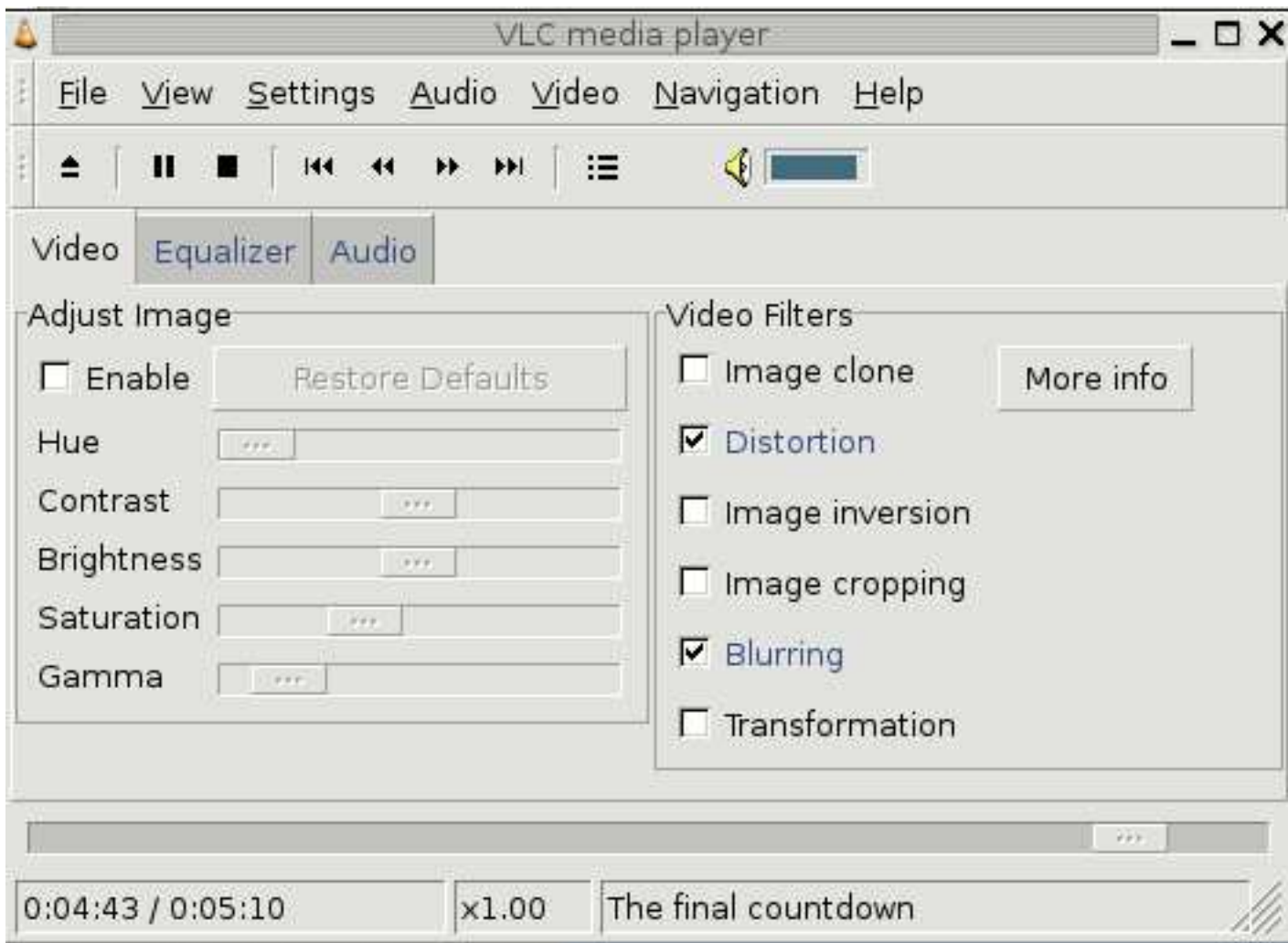
VLC is able to deinterlace a video stream using different deinterlacement methods. Deinterlacement can be enabled in the *Video* menu, *Deinterlacement* menu item. The *Blend* methods gives the best results in most cases. The *discard* method is a less resource consuming alternative.

On some particular streams (MPEG 4, DIVX, XVID, Sorenson, ...), some additional image filtering can be applied to the video before display, improving its quality in some cases. This can be enabled in the *Video* menu, *Post processing* menu item. Different levels of post processing can be chosen here. A higher level means more filtering.

Video filters

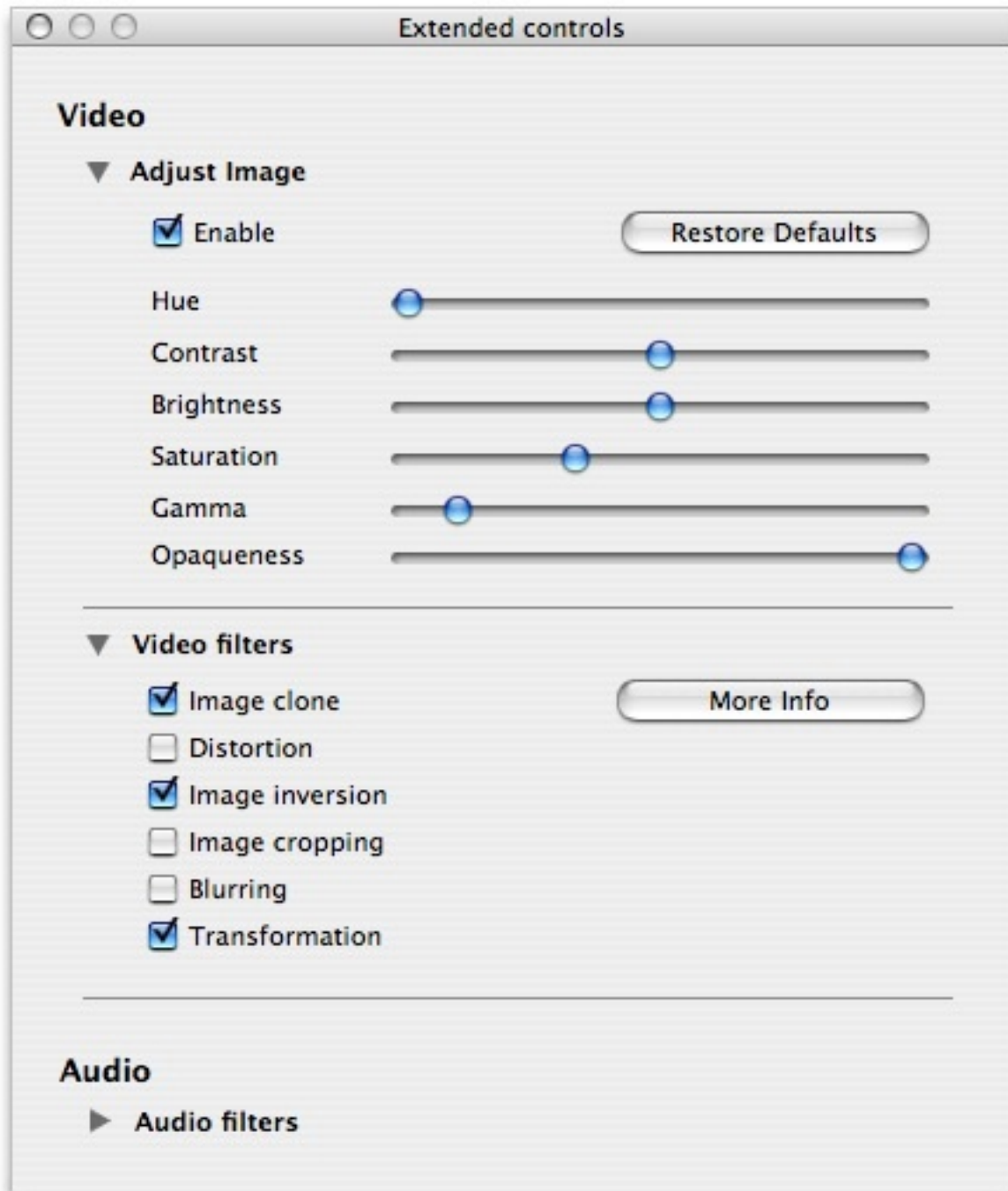
VLC features several filters able to change the video (distortion, brightness adjustment, motion blurring, ...).

With the wxWidgets interface, filters can be easily enabled using the Extended GUI. In the *Video* tab, simply select the filters to enable. Image settings can be easily adjusted.



Video filters selection in the wxWidgets interface

You can enable these filters through the *Extended Controls panel* on Mac OS X. Click on the triangle next to *Video filters* to select your filters or expand the *Adjust Image* section to change the contrast, hue, etc.



Video filters selection in the Mac OS X interface

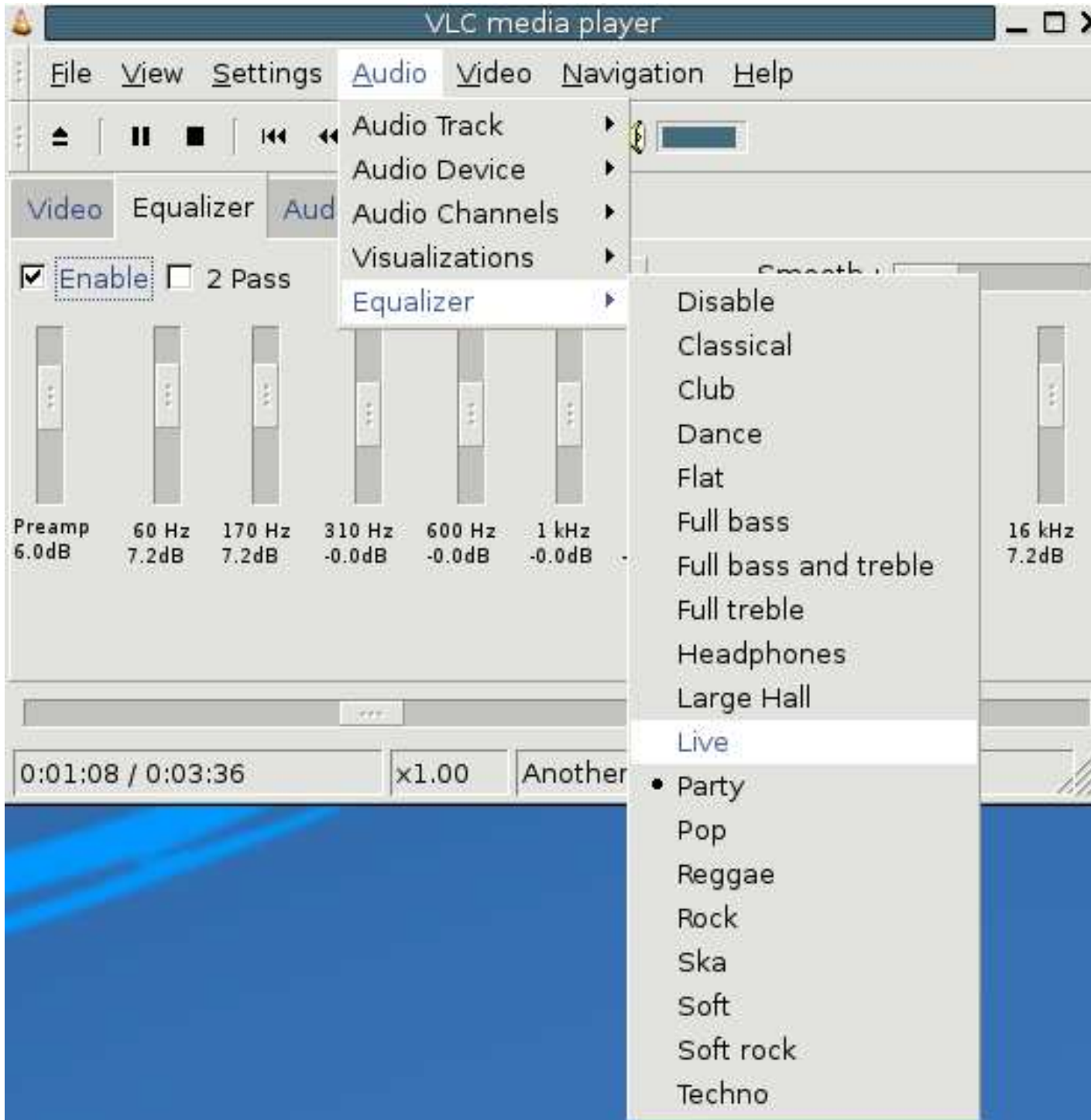
For better control, you need to go to the preferences. To select the filters to be enabled, go to *Video*, then to *Filters*. In the "video filter module" box, enter the names of the filters to enable, separated by semicommas. Filters will be applied in the selected order. Valid names are "clone", "wall", "transform", "adjust", "crop", "deinterlace", "distort", "motionblur" and "logo".

If you want to tune the behavior of these filters, go to *Video, Filters, [your filter]*. For each filter, you will find a short description and the options

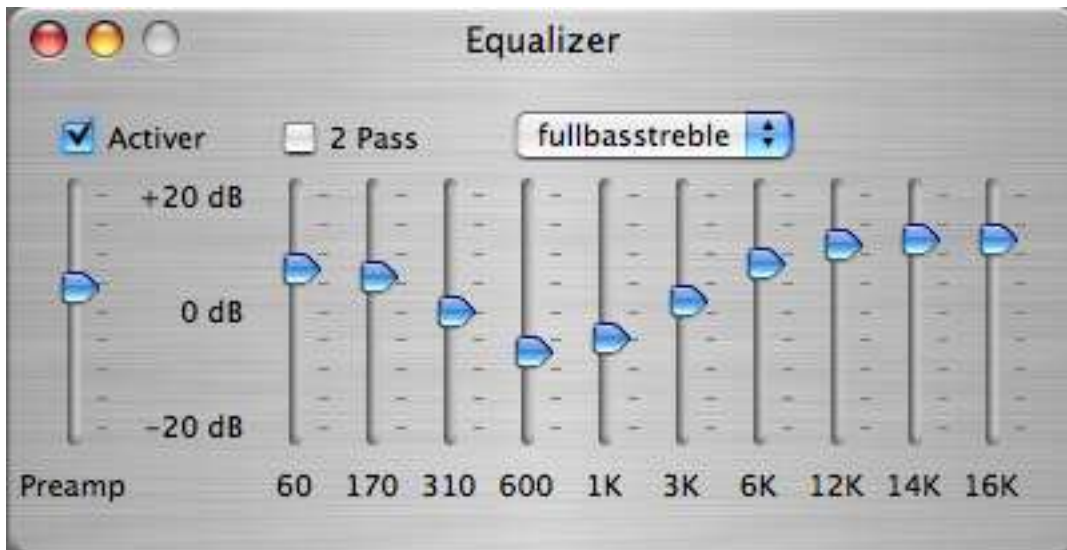
Audio filters

Equalizer

VLC features a 10-band graphical equalizer. You can display it by activating the advanced GUI on wxWidgets or by clicking the *Equalizer* button on the MacOS X interface.



The equalizer in the wxWidgets interface



The equalizer in the MacOS X interface

Presets are available in the Audio menu in wxWidgets, or in the Equalizer window in the MacOS X interface

Other audio filters

At the moment, VLC features two other audio filters: a volume normalizer and a filter providing sound spatialization with a headphone. They can be enabled in the Audio tab of the extended GUI for the wxWidgets interface and in the Audio section of the Extended Controls panel of the Mac OS X interface.

For better control, you need to go to the preferences. To select the filters to be enabled, go to *Audio*, then to *Filters*. In the "audio filters" box, enter the names of the filters to enable, separated by commas. Valid names are "equalizer", "normvol" and "headphone".

If you want to tune the behavior of these filters, go to *Audio, Filters, [your filter]*. The equalizer and headphone filters can be tuned.

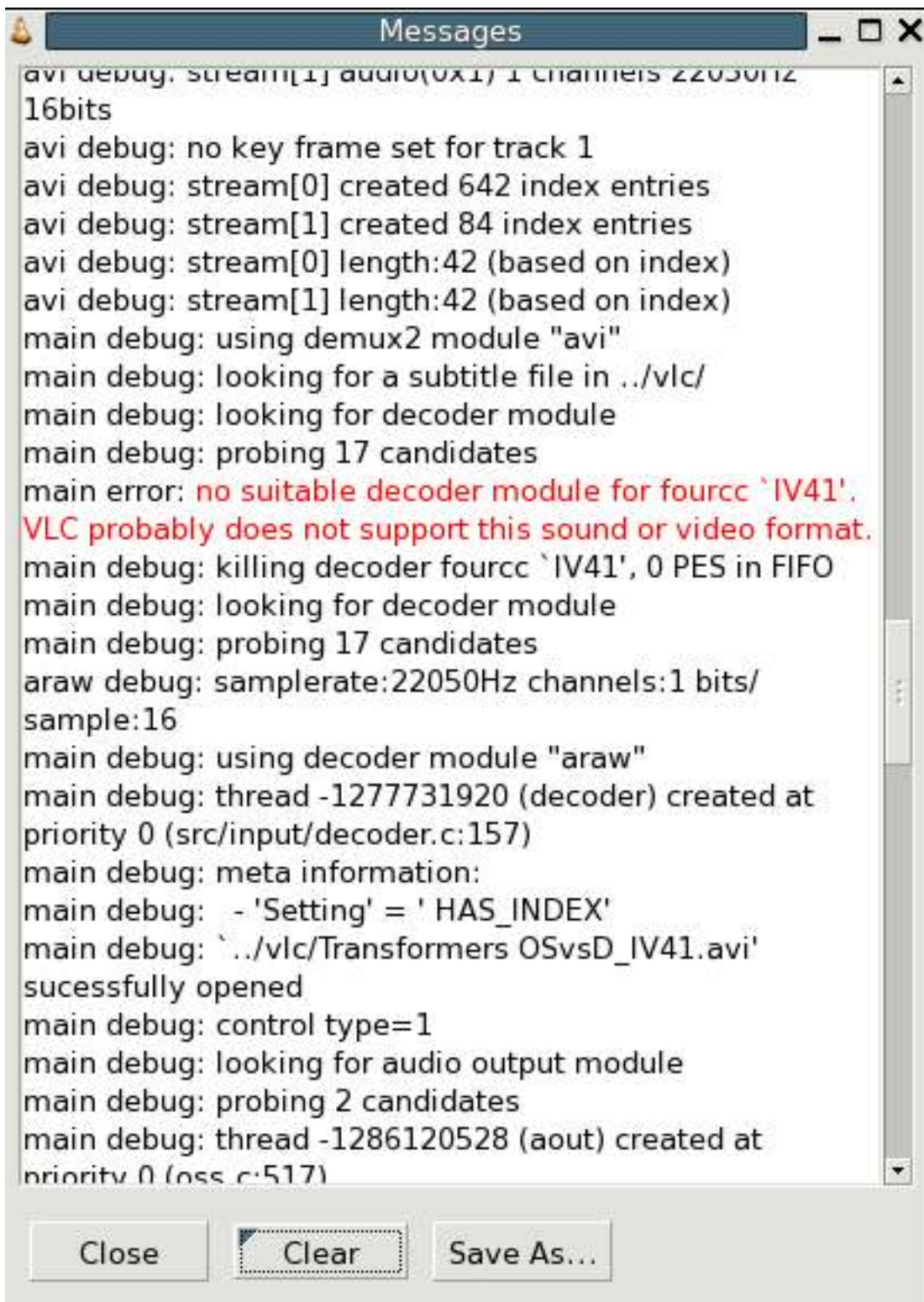
Basic troubleshooting

Note: This is very incomplete

File does not play, only sound or only video

Maybe the file you are trying to read is not fully supported. VLC does not use the codec packs you might have installed. It comes with its own codecs. If there is no open-source decoder for the format you are trying to read, it won't be supported. (There is an exception, under Windows, for codecs that use the DirectShow framework).

To find out, open the Messages Window (View menu) and restart your stream. Look for error messages (red messages)



The wxWidgets messages window

In this example, the file contains a IV41 video stream, a codec that is not supported by VLC.

You may of course have other messages. If you post to a VideoLAN mailing list or in the forum, please include such a log, it is very valuable.

Weird VLC behavior ("stopped working")

A very common thing is a bogus preferences file. Don't hesitate to delete it if problems appear suddenly. You will find in the FAQ details on how to delete your preferences file (<http://www.videolan.org/doc/faq/en/index.html#id2470084>).

Computer crashes / Video is corrupted

Another common problem is buggy video drivers. Try upgrading them from the website of your video card's manufacturer.

Also, you can try disabling Overlay (Preferences/General/Video, untick "Overlay video output")

Chapter 4. Advanced use of VLC

Use the command line

All standard operations of VLC should be available from the GUI. However, some complex operations can only be done from the command line and there are situations in which you don't need or want a GUI. Here is the complete description of VLC's command line and how to use it

You need to be quite comfortable with command line usage to use this.

Note: Windows users have to use the `--option-name="value"` syntax instead of the `--option-name value` syntax.

Getting help

VLC uses a modular structure. The core mainly manages communication between modules. All the multimedia processing is done by modules. There are input modules, demultiplexers, decoders, video output modules, ...

This chapter will only describe the "general" options, i.e. the core options. Each module adds new options. For example, the HTTP input module will add options for caching, proxy, authentication, ...

By using `vlc --help`, you will get the basic core options. `vlc --longhelp` will give all the basic options (core + modules). Adding `--advanced` will give the "advanced options" (for advanced users). So `vlc --longhelp --advanced` will give you all options.

Also, you might want to get debug informations. To do this, use `-v` or `-vv` (this will show lower severity messages). If your console supports it, you can add `--color` to get messages in color.

Opening streams

The following commands start VLC and start reading the given element(s)

Opening a file

Start VLC with:

```
% vlc my_file
```

VLC should be able to recognize the file type. If it does not, you can force demultiplexer and decoder (see below).

A list of all video and audio codecs supported by VLC is available on the VLC features list (<http://www.videolan.org/vlc/features.html>).

Opening a DVD or VCD, or an audio CD

Start VLC with:

For a DVD with menus :

```
% vlc dvd://[device] [@raw_device] [@[title] [: [chapter] [:angle]]]
```

In most cases, `vlc dvd://` or `vlc dvd://[device]` will do. [device] is for example `/dev/dvd` on GNU/Linux or `D:` on Windows (complete path to your DVD drive).

or (DVD without menus) :

```
% vlc dvdsimple://[device] [@raw_device] [@[title] [: [chapter] [: angle]]]
```

or (VCD):

```
% vlc vcd://[device] [{E|P|E|T|S} [number]]
```

or (Audio CD):

```
% vlc cdda://[device] [@[track]]
```

Receiving a network stream

To receive an unicast RTP/UDP stream (sent by VLC's stream output), start VLC with:

```
% vlc rtp://@:5004
```

If 5004 is the port to which packets are sent. 1234 is another commonly used port number. For more information, look at the Streaming Howto

To receive an multicast RTP/UDP stream (sent by VLC's stream output), start VLC with:

```
% vlc rtp://@multicast_address:port
```

To receive a SSM (source specific multicast) stream, you can use:

```
% vlc rtp://server_address@multicast_address:port
```

This only works on OSs that support SSM (Windows XP and Linux)

To receive a HTTP stream, start VLC with :

```
% vlc http://www.example.org/your_file.mpg
```

To receive a RTSP stream, start VLC with:

```
% vlc rtsp://www.example.org/your_stream
```

Modules selection

VLC always try to select the most appropriate interface, input and output modules, among the ones available on the system, according to the stream it is given to read. However, you may wish to force the use of a specific module with the following options.

- `--intf <module>` allows you to select the interface module.
- `--extraintf <module>` allows you to select extra interface modules that will be launched in addition to the main one. This is mainly useful for special *control* interfaces, like HTTP, RC (Remote Control), ... (see below)
- `--aout <module>` allows you to select the audio output module.
- `--vout <module>` allows you to select the video output module.
- `--memcpy <module>` allows you to choose a memory copy module. You should probably never touch that.

You can get a listing of the available modules by using `vlc -l`

Stream Output

The Stream output system allows vlc to become a streaming server.

For more details on the stream output system, please have a look at the Streaming Howto.

Other Options

Audio options

- `--noaudio` disables audio output.
- `--mono` forces VLC to treat the stream in mono audio.
- `--volume <integer>` sets the level of audio output (between 0 and 1024)
- `--aout-rate <integer>` sets the audio output frequency (Hz). By default, VLC will try to autodetect this.
- `--desync <integer>` compensates desynchronization of audio (ms). (If audio and video streams are not synchronized, use this setting to delay the audio stream)
- `--audio-filter <string>` adds audio filters to the processing chain. Available filters are visual (visualizer with spectrum analyzer and oscilloscope), headphone (virtual headphone spatialization) and normalizer (volume normalizer)

Video options

- `--novideo` disables video output.
- `--greyscale` turns video output into grayscale mode.
- `--fullscreen` (or `-f`) sets fullscreen video.
- `--nooverlay` disables hardware acceleration for the video output.
- `--width, --height <integer>` sets the video window dimensions. By default, the video window size will be adjusted to match the video dimensions.
- `--zoom <float>` adds a zoom factor.
- `--aspect-ratio <mode>` forces source aspect ratio. Modes are 4x3, 16x9, ...
- `--spumargin <integer>` forces SPU subtitles position.
- `--filter <string>` adds video filters to the processing chain. You can add several filters, separated by commas
- `--sout-transcode-sfilter <string>` adds video subpictures filter to the processing chain.

Playlist options

- `--random` plays files randomly forever.
- `--loop` loops playlist on end.
- `--repeat` repeats the current item until another item is forced
- `--play-and-stop` stops the playlist after each played item.

Network options

- `--server-port <integer>` sets server port.
- `--iface <string>` specifies the network interface to use.
- `--iface-addr <string>` specifies your network interface IP address.
- `--mtu <integer>` specifies the MTU of the network interface.
- `--ipv6` forces IPv6.
- `--ipv4` forces IPv4.

CPU options

You should probably not touch these options unless you know what you are doing.

- `--nommx` disables the use of MMX CPU extensions.
- `--no3dn` disables the use of 3D Now! CPU extensions.
- `--nommxext` disables the use of MMX Ext CPU extensions.
- `--nosse` disables the use of SSE CPU extensions.
- `--noaltivec` disables the use of AltiVec CPU extensions.

Miscellaneous options

- `--quiet` deactivates all console messages.
- `--color` displays color messages.
- `--search-path <string>` specifies interface default search path.
- `--plugin-path <string>` specifies plugin search path.
- `--no-plugins-cache` disables the plugin cache (plugins cache speeds up startup)
- `--dvd <string>` specifies the default DVD device.
- `--vcd <string>` specifies the default VCD device.
- `--program <integer>` specifies program (SID) (for streams with several programs, like satellite ones).
- `--audio-type <integer>` specifies the default audio type to use with dvds.
- `--audio-channel <integer>` specifies the default audio channel to use with dvds.
- `--spu-channel <integer>` specifies the default subtitle channel to use with dvds.
- `--version` gives you information about the current VLC version.
- `--module <module>` displays help about specified module. (Shortcut : `-p`)

Item-specific options

There are many options that are related to items (like `--novideo`, `--codec`, `--fullscreen`).

For all of these, you have the possibility to make them item-specific, using ":" instead of "--" and putting the option just after the concerned item

Examples:

- `% vlc file1.mpg :fullscreen file2.mpg`
will play file1.mpg in fullscreen mode and file2.mpg in the default mode (which is generally no fullscreen), whereas
- `% vlc --fullscreen file1.mpg file2.mpg`
will play both files in fullscreen mode
- `% vlc --fullscreen file1.mpg :sub-file=file1.srt :no-fullscreen file2.mpg :filter=distort`
will play file1.mpg in windowed (no-fullscreen) mode with the subtitles file file1.srt and will play file2.mpg with video filter distort enabled in fullscreen mode (item-specific options override global options)

Advanced use of filters

Filters

These are the old style VLC filters. They only apply to on screen display and thus cannot be streamed.

Deinterlacing video filter

Module name : *deinterlace*

- `--deinterlace-mode {discard,blend,mean,bob,linear}` choose a deinterlacing mode.

Invert video filter

Module name : *invert*

Image properties filter

Module name : *adjust*

- `--contrast <float>` Image contrast in the 0-2 range.
- `--brightness <float>` Image brightness in the 0-2 range.
- `--hue <integer>` Image hue in the 0-360 range
- `--saturation <float>` Image saturation in the 0-3 range.
- `--gamma <float>` Image gamma in the 0-10 range.

Wall video filter

Module name : *wall*

This filter splits the output in several windows.

- `--wall-cols <integer>` Number of columns.
- `--wall-rows <integer>` Number of rows.

- `--wall-active <string>` Select the windows you want to display. To select windows 2 and 4 you would write `--wall-active 2,4`. When this option isn't specified, all windows are displayed.

Video transformation filter

Module name : *transform*

- `--transform-type {90,180,270,hflip,vflip}` Select rotation angle or symmetry.

Distort video filter

Module name : *distort*

Clone video filter

This filter clones the output window.

Module name : *clone*

- `--clone-count <integer>` Number of clones.
- `--clone-vout-list <string>` Comma separated string of video output modules.

Crop video filter

Module name : *crop*

- `--crop-geometry <string>` Set the geometry of the zone to crop. This is set as `<width> x <height> + <left offset> + <top offset>`.
- `--autocrop` Enable automatic black border cropping.

Motion blur filter

Module name : *motionblur*

- `--blur-factor <integer>` Blur factor in the 1-127 range.

Video pictures blending

Module name : *blend*

Video scaling filter

Module name : *scale*

Subpictures Filters

These are the new VLC filters. They can be streamed.

Time display sub filter

Module name : *time*

- *--time-format* <string> Time format string. You can use the following special characters %Y = year, %m = month, %d = day, %H = hour, %M = minute, %S = second.
- *--time-x* <integer> X offset from the left in pixels.
- *--time-y* <integer> Y offset from the top in pixels.

Marque display sub filter

Module name : *marq*

- *--marq-marquee* <string> Marquee text to display.
- *--marq-x* <integer> X offset from the left in pixels.
- *--marq-y* <integer> Y offset from the top in pixels.
- *--marq-timeout* <timeout> Defines the time the marquee must remain displayed in milliseconds. Default value is 0 (remain forever).

Logo video filter

Module name : *logo*

This filter can be used both as an old style filter or a subpictures filter.

- *--logo-file* <string> Full path of the PNG file to use.
- *--logo-x* <integer> X offset from the left in pixels.
- *--logo-y* <integer> Y offset from the top in pixels.
- *--logo-transparency* <integer> You can set the logo transparency value here (from 0 for full transparency to 255 for full opacity).

Note: You can move the logo by left-clicking on it.

The HTTP interface

VLC ships with a little HTTP server integrated. It is used both to stream using HTTP, and for the HTTP remote control interface.

To start VLC with the HTTP interface, use:

```
% vlc -I http [--http-src /directory/ --http-host host:port]
```

If you want to have both the "normal" interface and the HTTP interface, use `vlc --extraintf http`.

The HTTP interface will start listening at host:port (<all interfaces>:8080 if omitted), and will reproduce the structure of /directory at http://host:port/ (vlc_source_path/share/http if omitted).

Use a browser to go to http://your_host_machine:port. You should be taken to the main page

VLC is shipped with a set of files that should be enough for generic needs. It is also possible to customize pages. See Appendix B.

Available pages are :

- `/index.html` gives access to the playlist, allows you to add or delete items and to control playback.
- `/info.html` gives you statistics and data about the usage of the server.
- `/admin/`. This folder gives more power and is therefore protected by a login/password. The default couple is admin/admin. The passwords and users can be changed in the `.access` file in the `VLC_directory/share/http/admin` folder. (Each folder can be protected, see Appendix B for more details).
- `/vlm/`. This folder allows you to control the VideoLAN manager (VLM). The VLM allows you to stream several input streams with only one VLC instance. For more information about VLM, please look at the Streaming Howto (<http://www.videolan.org/doc/streaming-howto/en>).

Other control interfaces

VLC includes a number of so-called interfaces that are not really interfaces, but means of controlling VLC. Nevertheless, they are enabled by setting them as interface or extra interface, either in the Preferences, in General/Interface, or using `-I` or `--extraintf` on the command line.

Hotkeys

This module allows you to control VLC and playback via hotkeys. It is always enabled by default. You can use hotkeys in the video output window.

Hotkeys can be hacked by :

```
% vlc --key-<function> '<code>'
```

Code is composed by modifiers keys (Alt, Shift, Ctrl, Meta, Command) separated by a dash (-) and terminated by a key (a...z, +, =, -, ', +, <, >, ', /, :, ', \,], *, Left, Right, Up, Down, Space, Enter, F1...F12, Home, End, Menu, Esc, Page Up, Page Down, Tab, Backspace, Mouse Wheel Up and Mouse Wheel Down). Main controls are available from hotkeys, such as : fullscreen, play-pause, faster, slower, next, prev, stop, quit, vol-up, etc. (use the `--longhelp` option for full list of functions). For example, for binding fullscreen to Ctrl-f, run :

```
% vlc --key-fullscreen 'Ctrl-f'
```

RC and RTCI

These two interfaces allow you to control VLC from a command shell (possibly using a remote connexion or a Unix socket).

Start VLC with `-I rc` or `--extraintf rc`. When you get the **Remote control interface initialized**, `'h'` for **help** message, press h and Enter to get help about available commands.

To be able to remote connect to your VLC using a TCP socket (telnet-like connexion), use `--rc-host your_host:port`. Then, by connecting (using telnet or netcat) to the host on the given port, you will get the command shell.

To use a UNIX socket (local socket, this does not work for Windows), use `--rc-unix /path/to/socket`. Commands can then be passed using this UNIX socket.

The RTCI interface gives you more advanced options, such as marquee control for the marquee subpicture filter (See filter section).

Ncurses

This is a text interface, using ncurses library.

Start VLC with `-I ncurses` or `--extraintf ncurses`. You will then get something like that:

```

VLC media player (ncurses interface) [ h for help ]

Source   : BG/05 - Cup Of Coffee.mp3
State    : Playing
Position : 0:01:42/0:04:31 (37.89%)
Volume   : 62%

[Progress bar]

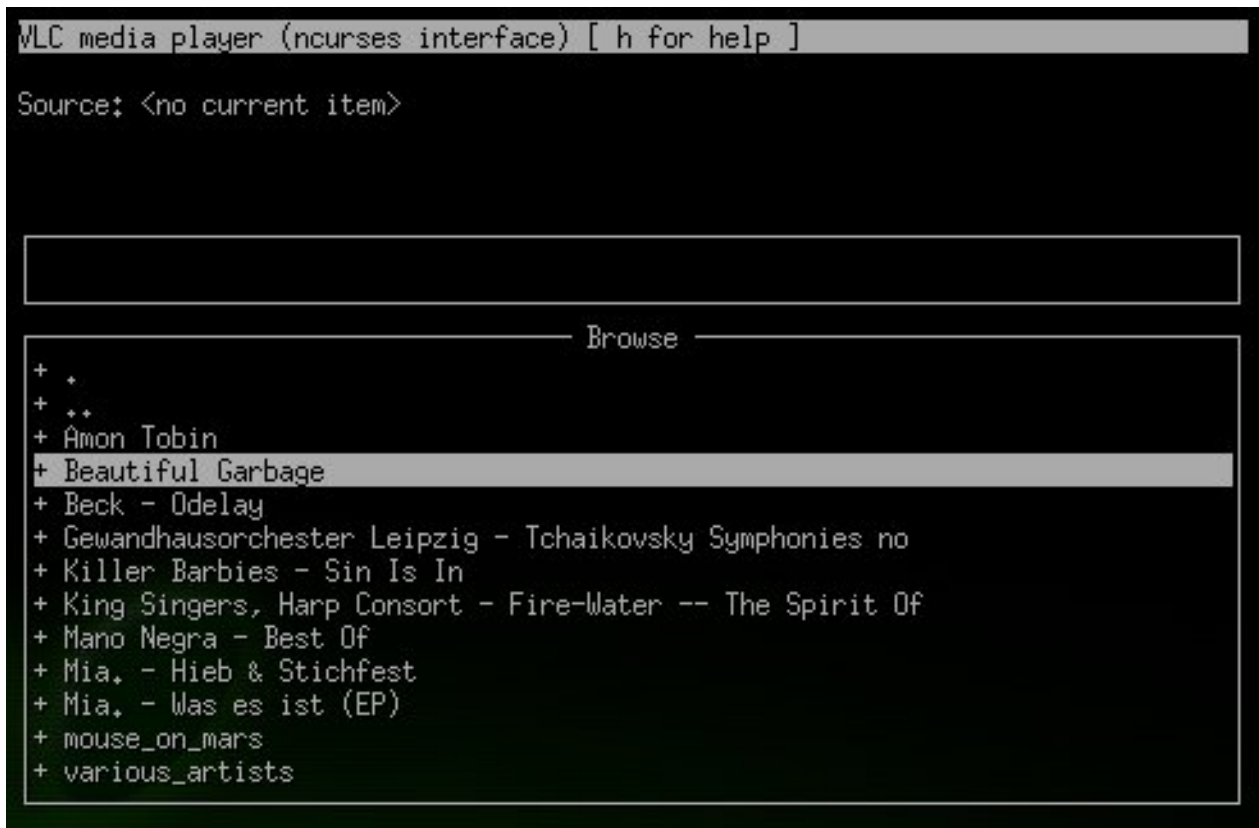
----- Playlist -----
0 - 'BG/01 - Shut Your Mouth.mp3' (01 - Shut Your Mouth.mp3)
1 - 'BG/02 - Androgyny.mp3' (02 - Androgyny.mp3)
2 - 'BG/03 - Can't Cry These Tears.mp3' (03 - Can't Cry These Tears.mp3)
3 - 'BG/04 - Til The Day I Die.mp3' (04 - Til The Day I Die.mp3)
> 4 - 'BG/05 - Cup Of Coffee.mp3' (05 - Cup Of Coffee.mp3)
5 - 'BG/06 - Silence Is Golden.mp3' (06 - Silence Is Golden.mp3)
6 - 'BG/07 - Cherry Lips (Go Baby Go...) (07 - Cherry Lips (Go Baby Go!).mp3)
7 - 'BG/08 - Breaking Up The Girl.mp3' (08 - Breaking Up The Girl.mp3)
8 - 'BG/09 - Drive You Home.mp3' (09 - Drive You Home.mp3)
9 - 'BG/10 - Parade.mp3' (10 - Parade.mp3)
10 - 'BG/11 - Nobody Loves You.mp3' (11 - Nobody Loves You.mp3)
11 - 'BG/12 - Untouchable.mp3' (12 - Untouchable.mp3)
12 - 'BG/13 - So Like A Rose.mp3' (13 - So Like A Rose.mp3)

```

The ncurses interface

Press `h` to get the list of all available commands, with a short description.

There is also a filebrowser available for the ncurses interface in order to add playlist items. Press `'B'` to use it.



The ncurses filebrowser

You can set the filebrowser starting point by launching vlc with the `--browse-dir` option:

```
% vlc -I ncurses --browse-dir /filebrowser/starting/point/
```

Gestures

Gestures provide a simple mouse gestures control. TODO

The Mozilla plugin

VLC can also be embedded in a web browser ! The following browsers are supported Mozilla (<http://www.mozilla.org/>), Firefox (<http://www.mozilla.org/products/firefox/>) and Safari (<http://www.apple.com/macosx/features/safari>).

Install the plugin

GNU/Linux Debian

You should already have the following lines in your `/etc/apt/sources.list` file, if you run Debian Woody:

```
deb http://download.videolan.org/pub/videolan/debian woody main
deb-src http://download.videolan.org/pub/videolan/debian woody main
```

VLC's latest packaged version is always in the official Debian unstable branch. However you should still use our apt-get line if you need libdvdcss in its latest version:

```
deb http://download.videolan.org/pub/videolan/debian sid main
deb-src http://download.videolan.org/pub/videolan/debian sid main
```

Install the *mozilla-plugin-vlc* package:

```
# apt-get update
# apt-get install mozilla-plugin-vlc
```

Windows

Quit Firefox or Mozilla

Copy the two files in *VLC_Installation_folder\mozilla* (usually *C:\Program Files\VideoLAN\VLC\mozilla*) to your mozilla plugins directory (Usually *C:\Program Files\Mozilla\plugins* or *C:\Program Files\Mozilla Firefox\plugins*).

Restart Firefox or Mozilla

MacOS X

The Mozilla/Safari plugin for MacOS X is only available from vlc version 0.8.5.1 and onwards.

Quit Safari browser

Download the Mozilla/safari plugin package from MacOS X download page (<http://www.videolan.org/vlc/download-macosx.html>).

Run the installer from the dmg image.

Compile the sources yourself

Please look at the developers page (<http://developers.videolan.org>) for information on how to do this.

Use the plugin

If you open a link to a video file handled by the VLC plugin (To get the list of handled types, browse to **about:plugins**) or a page with an embedded video, the plugin should open and read the video.

Build HTML pages that use the plugin (version <= 0.8.5)

The following API description is only valid uptill version 0.8.5 of the mozilla plugin.

Additionally to viewing video on all pages, you can build custom pages that will use the advanced features of the plugin, using Javascript functions to control playback or extract information from the plugin.

The vlc plugin supports the following function calls:

- **play()** : Start playing media in the plugin.
- **pause()** : Pause playback.
- **stop()** : Stop media playback.

- **fullscreen()** : Switch the video to full screen.
- **set_volume(vol)** : Set the volume. *vol* has to be an int in the 0-200 range.
- **get_volume()** : Get the current volume setting.
- **mute()** : Toggle volume muting.
- **set_int_variable(var_name, value)** :
- **set_bool_variable(var_name, value)** :
- **set_str_variable(var_name, value)** :
- **get_int_variable(var_name)** :
- **get_bool_variable(var_name)** :
- **get_str_variable(var_name)** :
- **clear_playlist()** : Clear the playlist.
- **add_item(mrl)** : Append an item whose location is given by the Media Resource Locator to the playlist.
- **next()**
- **previous()**
- **isplaying()** : return true if the plugin is playing something.
- **get_length()** : Get the media's length in seconds.
- **get_position()** : Get the current position in the media in percent.
- **get_time()** : Get the current position in the media in seconds.
- **seek(seconds,is_relative)** : If *is_relative* is true, seek relatively to current time, else seek from beginning of the stream. Seek time is specified in seconds.

Here are a few examples of HTML pages that use the Mozilla plugin.

Example 1

In this example, the plugin will read an HTTP stream inside the web page. If the user goes fullscreen, he will have to press **f** to go back in normal view.

```
<html>
<head><title>Demo of VLC mozilla plugin</title></head>

<body>

<h1>Demo of VLC mozilla plugin - Example 1</h1>

<embed type="application/x-vlc-plugin"
        name="video1"
        autoplay="no" loop="yes" width="400" height="300"
        target="http://server.example.org/video1.vob" />
<br />
    <a href="javascript:;" onclick='document.video1.play()''>Play video1</a>
    <a href="javascript:;" onclick='document.video1.pause()''>Pause video1</a>
    <a href="javascript:;" onclick='document.video1.stop()''>Stop video1</a>
    <a href="javascript:;" onclick='document.video1.fullscreen()''>Fullscreen</a>

</body>
</html>
```

Example 2

In this example, the plugin will read a multicast RTP/UDP stream in a dedicated video output window.

```
<html>
<head><title>Demo of VLC mozilla plugin</title></head>

<body>

<h1>Demo of VLC mozilla plugin - Example 2</h1>

<embed type="application/x-vlc-plugin"
        name="video2"
        autoplay="no" loop="no" hidden="yes"
        target="rtp://@239.255.12.42:5004" />
<br />
  <a href="javascript:;" onclick='document.video2.play()''>Play video2</a>
  <a href="javascript:;" onclick='document.video2.stop()''>Stop video2</a>
  <a href="javascript:;" onclick='document.video2.fullscreen()''>Fullscreen</a>

</body>
</html>
```

Build HTML pages that use the plugin (version > 0.8.5.1)

Additionally to viewing video on all pages, you can build custom pages that will use the advanced features of the plugin, using Javascript functions to control playback or extract information from the plugin.

The vlc plugin exports several objects that can be accessed for setting and getting information. When used improperly the API's will throw an exception that includes a string that explains what happened. For example when asking for `vlc.input.length` when there is no playlist item playing.

The vlc plugin knows the following objects:

- **audio** : Access audio properties.
- **input** : Access input properties.
- **playlist** : Access playlist properties.
- **video** : Access video properties.
- **log** : Access log properties.
- **messages** : Access to log message properties.
- **iterator** : Access to log iterator properties.

The following JavaScript code shows how to get a reference to the vlc plugin. This reference can then be used to access the objects of the vlc plugin.

```
<html>
<title>VLC Mozilla plugin test page</title>
<body>
<embed type="application/x-vlc-plugin" pluginspage="http://www.videolan.org" version="VideoLAN.VLC"
        width="640"
        height="480"
        id="vlc">
</embed>
<script language="Javascript">
```

```

<!--
var vlc = document.getElementById("vlc");
vlc.audio.toggleMute();
!-->
</script>
</body>
</html>

```

Root object

readonly properties

- **VersionInfo** : returns version information string

read/write properties

- **none**

methods

- **vlc.versionInfo()** : returns version information string

Audio object

readonly properties

- **none**

read/write properties

- **vlc.audio.mute** : boolean value to mute and unmute the audio
- **vlc.audio.volume** : a value between [0-200] which indicates a percentage of the volume.
- **vlc.audio.track** : a value between [0-65535] which indicates the audio track to play or that is playing. (supported in vlc version > 0.8.6)
- **vlc.audio.channel** : an integer between 1 and 5 that indicates which audio channel mode is used, values can be: "1=stereo", "2=reverse stereo", "3=left", "4=right", "5=dolby". Use vlc.audio.channel to check if setting of the new audio channel has succeeded. (supported in vlc version > 0.8.6)

Audio Channel:

```

<SELECT readonly onChange='doAudioChannel(this.value)'+>
  <OPTION value=1>Stereo</OPTION>
  <OPTION value=2>Reverse stereo</OPTION>
  <OPTION value=3>Left</OPTION>
  <OPTION value=4>Right</OPTION>
  <OPTION value=5>Dolby</OPTION>
</SELECT>
<SCRIPT language="javascript">
<!--
function doAudioChannel(value)
{
  var vlc = getVLC("vlc");
  vlc.audio.channel = parseInt(value);
  alert(vlc.audio.channel);
};

```

--' >

methods

- **vlc.audio.toggleMute()** : boolean toggle that mutes and unmutes the audio based upon the previous state

Input object

readonly properties

- **vlc.input.length** : length of the input file in number of milliseconds
- **vlc.input.fps** : frames per second returned as a float
- **vlc.input.hasVout** : a boolean that returns true when the video is being displayed, it returns false when video is not displayed

read/write properties

- **vlc.input.position** : normalized position in multimedia stream item given as a float value between [0.0 - 1.0]
- **vlc.input.time** : the absolute position in time given in milliseconds, this property can be used to seek through the stream

```
<!-- absolute seek in stream !-->
vlc.input.time = <absolute seek>
<!-- relative seek in stream !-->
vlc.input.time = vlc.input.time + <relative seek>
```

- **vlc.input.state** : current state of the input chain given as enumeration (IDLE/CLOSE=0, OPENING=1, BUFFERING=2, PLAYING=3, PAUSED=4, STOPPING=5, ERROR=6)
- **vlc.input.rate** : input speed given as float (1.0 for normal speed, 0.5 for half speed, 2.0 for twice as fast, etc.)

methods

- **none**

Playlist object

readonly properties

- **vlc.playlist.itemCount** : number that returns the amount of items currently in the playlist
- **vlc.playlist.isPlaying** : a boolean that returns true if the current playlist item is playing and false when it is not playing

read/write properties

- **none**

methods

- **vlc.playlist.add(mrl)** : add a playlist item as MRL (Multimedia Resource Locator). The MRL must be given as a string.
- **vlc.playlist.add(mrl,name,options)** : add a playlist item as MRL (Multimedia Resource Locator), with metaname 'name' and options 'options'. All input values must be given as string.
- **vlc.playlist.play()** : start playing the current playlist item
- **vlc.playlist.togglePause()** : toggle the pause state for the current playlist item

- **vlc.playlist.stop()** : stop playing the current playlist item
- **vlc.playlist.next()** : iterate to the next playlist item
- **vlc.playlist.prev()** : iterate to the previous playlist item
- **vlc.playlist.clear()** : empty the current playlist, all items will be deleted from the playlist
- **vlc.playlist.removeItem(number)** : remove the given item number (which cannot be greater than vlc.playlist.itemCount)

Video object

readonly properties

- **vlc.video.width** : returns the horizontal size of the video
- **vlc.video.height** : returns the vertical size of the video

read/write properties

- **vlc.video.fullscreen** : when set to true the video will be displayed in fullscreen mode, when set to false the video will be shown inside the video output size. The property takes a boolean as input.
- **vlc.video.aspectRatio** : get and set the aspect ratio to use in the video screen. The property takes a string as input value. Valid values are: "1:1", "4:3", "16:9", "16:10", "221:100" and "5:4"

methods

- **vlc.video.toggleFullscreen()** : toggle the fullscreen mode based on the previous setting

Log object

readonly properties

- **vlc.log.messages** : returns a messages object

read/write properties

- **vlc.log.verbosity** : write number [-1,0,1,2,3] for changing the verbosity level of the log messages. The numbers have the following meaning: -1 disable, 0 info, 1 error, 2 warning, 3 debug.

methods

- **none**

Messages object

readonly properties

- **messages.count** : returns number of messages in the log
- **messages.severity** : number that indicates the severity of the log message (0 = info, 1 = error, 2 = warning, 3 = debug)
- **message.name** : name of VLC module that printed the log message (e.g: main, http, directx, etc...)
- **message.type** : type of VLC module that printed the log message (eg: input, access, vout, sout, etc...)
- **message.message** : the message text

read/write properties

- **none**

methods

- **messages.clear()** : clear the current log buffer. It should be called as frequently as possible to not overflow the plugins logging buffer. Call this method after the log messages of interest are read.
- **messages.iterator()** : returns a messages iterator object, which is used to iterate over the messages in the log. *Don't clear the log buffer while holding an iterator object.*

Iterator object

readonly properties

- **iterator.hasNext** : returns a boolean that indicates when true, that wheter `vlc.log.messages.next()` returns the next message.

read/write properties

- **none**

methods

- **iterator.next()** : returns the next message object it the log

Appendix A. GNU General Public License

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software - to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps:

1. copyright the software, and
2. offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

Section 0

This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a

work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

Section 1

You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Section 2

You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

1. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
2. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
3. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License.

Exception:: If the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

Section 3

You may copy and distribute the Program (or a work based on it, under Section 2 in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

1. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
2. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
3. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

Section 4

You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

Section 5

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

Section 6

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

Section 7

If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

Section 8

If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

Section 9

The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

Section 10

If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY Section 11

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

Section 12

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Appendix B. Building pages for the HTTP interface

Introduction

This appendix describes the language used for writing dynamic web pages for the HTTP interface.

Pages must be placed in the share/http folder in either VLC's folder (Windows, Mac) or /usr/share/vlc/share/http or /usr/local/share/vlc/share/http (or wherever vlc's shared files are installed)

Some files are handled a bit specially :

- Files beginning with '.' are not exported.
- A '.access' file will be opened and the http interface will expect to find at the first line a login/password (written as login:password). This login/password will be used to protect all files in this directory. Be careful that only files in this directory will be protected. (sub-directories won't be protected.)
- A '.hosts' file will be opened and the http interface will expect to find a list of network/mask pairs separated by new line, for instance 192.168.0.0/255.255.255.0. If this file is present, then the default behaviour is to deny access from hosts which don't match any of the network/mask pairs to all the files of the directory. If the file is not present, then any host has access to the files of the directory. Be careful that only files in this directory will be protected. (sub-directories won't be protected.)
- The file <dir>/index.html will be exported as <dir> and <dir>/ and not as index.html.

The MIME type is set by looking at the file extension and cannot be specified nor modified for a specific file. Unknown extensions will have "application/octet-stream" as MIME type.

You should avoid exporting big files. Each file is indeed first loaded into the memory before being sent to the client, so please be careful.

VLC macros

Each time a .html/.htm page is requested, it is parsed by VLC before being sent. The parser searches for the VLC macros, and executes or substitutes them. Moreover, URL arguments received by the GET method can be interpreted .

A VLC macro looks like: `<vlc id="macro-name" param1="macro-parameters1" param2="macro-parameters2" />`.

"id" is the only mandatory field, param1 and param2 may or may not be present and depend on the value of "id".

You should take care that you _have to_ respect this syntax, VLC won't like invalid syntax. (It could easily lead to crashes).

Examples :

Correct: `<vlc id="value" param1="version" />`

Incorrect: `<vlc id="value" param1="version" >` (missing tag ending), `<vlc id=value param1="version" />` (missing "")

Valid macros are:

- *control* (1 optional parameter)
- *include* (1 parameter)
- *get* (2 parameters)
- *set* (2 parameters)
- *rpn* (1 parameter)

- *if* (1 optional parameter)
- *else* (no parameter)
- *end* (no parameter)
- *value* (1 optional parameter)
- *foreach* (2 parameters)

For powerful macros, you may use these tools :

- RPN Evaluator (see part 2)
- Stacks: The stack is a place where you can push numbers and strings, and then pop them backs. It's used with the little RPN evaluator.
- Local variables: You can dynamically create new variables and changes their values. Some local variables are predefined:
 - *url_value* : parameter of the URL
 - *url_param* : 1 if *url_value* isn't empty else 0
 - *version* : the VLC version
 - *copyright* : the VLC copyright
 - *vlc_compile_time*, *vlc_compile_by*, *vlc_compile_host*, *vlc_compile_domain*, *vlc_compiler*, *vlc_changeset* : information on the VLC build
 - *stream_position*, *stream_time*, *stream_length*, *stream_state* : information on the currently playing stream
 - *volume* : current volume setting

Remark: The stacks, and local variables context is reseted before the page is executed.

The RPN evaluator

RPN means Reverse Polish Notation

Introduction

RPN could look strange but it's a fast and easy way to write expressions. It also avoids the use of (and).

Instead of writing (1 + 2) * 5 you just use 1 2 + 5 *.

The idea beyond it is: if we have a number or a string (using "), push it on the stack. If it is an operator (like +), pop the arguments from the stack, execute the operators and then push the result onto the stack. The result of the RPN sequence is the value on the top of the stack.

stack:	Word processed	
empty	1	1 is pushed on the stack
1	2	2 same things
1 2	+	+: remove 1 and 2 and write 3 on the stack
3	5	5 is pushed on the stack
3 5	*	*: remove 3 and 5 and write 15
15		<- result

Operators

Notation: ST(1) means the first stack element, ST(2) the second one ... and op the operator.

You have access to :

- Standard arithmetics operators: +, -, *, /, %: these ones push the result of ST(1) op ST(2) to the stack
- Binary operators: ! (push !ST(1)); ^, &, !: push the result ST(1) op ST(2)
- test: =, <, <=, >, >=: execute ST(1) op ST(2) and push -1 if true else 0
- string functions:
 - *strcat* : pushes the result of 'ST(1)ST(2)
 - *strcmp* : compares ST(1) and ST(2) (0 if equal)
 - *strncmp* : compares the first ST(1) characters of ST(2) and ST(3) (0 if equal)
 - *strsub* : extracts characters ST(1) to ST(2) of streaming ST(3)
 - *strlen* : pushes the length of ST(1)
 - *str_replace* : replaces string ST(2) with ST(1) in ST(3)
 - *url_encode* : encodes non-alphanumeric characters of ST(1) as %XX so that they can be safely passed as GET or POST variables
 - *url_extract* : performs the reverse operation of url_encode
 - *addslashes* : protects single quotes (') and double quotes (") of ST(1) with backslashes (\) so that they can be safely passed to a VLC playlist function
 - *stripslashes* : performs the reverse operation of addslashes
 - *htmlspecialchars* : encodes &, ", ', < and > of ST(1) as their &stuff; HTML counterpart, so that they don't interact with HTML tags
 - *realpath* : parses ST(1) as a filename path, and pushes an absolute path to that file, removing ~ and ../
- stack manipulation:
 - *dup* : pops ST(1) and pushes the same string twice
 - *drop* : pops ST(1) and drops it
 - *swap* : exchanges ST(1) and ST(2)
 - *flush* : empties the stack
- variables manipulation:
 - *store* : stores ST(2) in a local variable named ST(1)
 - *value* : pushes the value of the local variable named ST(1)
- player control:
 - *vlc_play* : plays the playlist item whose ID is ST(1), and pushes the return value of the play function (0 in case of success); see playlist functions below
 - *vlc_stop* : stops the playlist
 - *vlc_pause* : pauses the playlist
 - *vlc_next* : plays the next playlist item
 - *vlc_previous* : plays the previous playlist item

- *vlc_seek* : seeks the current input to a location defined in ST(1), for instance +3m (minutes), -20s, 45%, 1:12, 1h12m25s
- *vlc_var_type* : pushes the type of the variable ST(2) of object ST(1); the type is one of these strings *VLC_VAR_BOOL*, *VLC_VAR_INTEGER*, *VLC_VAR_HOTKEY*, *VLC_VAR_STRING*, *VLC_VAR_MODULE*, *VLC_VAR_FILE*, *VLC_VAR_DIRECTORY*, *VLC_VAR_VARIABLE*, *VLC_VAR_FLOAT*, *UNDEFINED* (no such variable) or *INVALID* (no input stream); the object is one of *VLC_OBJECT_ROOT*, *VLC_OBJECT_VLC*, *VLC_OBJECT_INTF*, *VLC_OBJECT_PLAYLIST*, *VLC_OBJECT_INPUT*, *VLC_OBJECT_VOUT*, *VLC_OBJECT_AOUT* or *VLC_OBJECT_SOUT*
- *vlc_var_set* : sets variable ST(2) of object ST(1) to ST(3)
- *vlc_var_get* : pushes the value of the variable ST(2) of object ST(1)
- *vlc_object_exists* : checks if object ST(1) exists
- *vlc_config_type* : pushes the type of the configuration variable ST(1); see *vlc_var_type* for a list of types
- *vlc_config_set* : sets configuration variable ST(1) to ST(2)
- *vlc_config_get* : pushes the value of the configuration variable ST(1)
- *vlc_config_save* : saves the modification made to the configuration variables of module ST(1) to the configuration file (ST(1) may be empty, in which case the whole configuration is saved) and pushes the return status (0 in case of success)
- *vlc_config_reset* : resets the configuration file to the default value (use with caution)
- playlist functions:
 - *playlist_add* : adds MRL ST(1) to the playlist, with name ST(2) and returns the playlist ID associated to this item; special characters must be escaped with addslashes first; it is very convenient to call 'toto.mpg' playlist_add vlc_play
 - *playlist_empty* : clears the playlist of all items
 - *playlist_move* : moves playlist item at position ST(2) to position ST(1)
 - *playlist_delete* : deletes playlist item ID ST(1)

The macros

The *control* macro

The use of the control macro is now deprecated in favour of the RPN functions above. The documentation is provided here for the maintenance of HTML pages still using this old API. The main problem with this API is that there is no way to retrieve the playlist ID of the last added item.

When asking for a page, you can pass arguments to it through the url. (e.g. using a <form>). Ex:

http://host:port/page.html?var=value&var2=value2&... The "control" macro tells a page to parse these arguments and to execute the ones that are allowed. param1 of this macro says which commands are allowed. If empty, all commands will be permitted.

Some commands require an argument that must be passed in the URL too.

Table B-1. URL commands

Name	Argument	Description
play	item (integer)	Play the specified playlist item

Name	Argument	Description
stop		Stop
pause		Pause
next		Go to next playlist item
previous		Go to previous playlist item
add	mrl (string)	Add a MRL (Media Resource Locator) to the playlist
delete	item (integer)	Delete the specified playlist item or list of playlist items
empty		Empty the playlist
close	id (hexa)	Close a specific connection
shutdown		Quit VLC

For example, you can restrict execution of the **shutdown** command to protected page (through a *.access* file), using the control macro in all unprotected pages.

The *include* macro

This macro is replaced by the contents of the file param1. If the file contains vlc macros, they are correctly parsed and replaced.

The *get* macro

This macro will be replaced by the value of the configuration variable which name is stored in param1 and which type is given by param2.

param1 must be the name of an existing configuration variable. param2 must be the right type of the variable. It can be one of *int*, *float*, or *string*.

Example: `<vlc id="get" param1="sout" param2="string" />` will be replaced in the output page by the value of sout.

The *set* macro

This macro allows to set the value of a configuration variable. The name is given by param1 and the type by param2 (like for get). The value is retrieved from the url using the name given in param1.

For example, if player.html contains `<vlc id="set" param1="sout" param2="string" />`, and if you browse at `http://host:ip/player.html?sout=sout_value`, the sout variable will be set to "sout_value". If the URL doesn't contain sout, nothing will be done.

The *rpn* macro

This macro allows you to interpret RPN commands. (See II).

The *if,else,end* macro

This macro allows you to control the parsing of the HTML page.

If param1 isn't empty, it is first executed with the RPN evaluator. If the first element from the stack is not 0, the test value is true, else false..

```
<vlc id="if" param1="1 2 =" />
  <!-- Never reached -->
<vlc id="else" />
  <p> Test succeed: 1 isn't equal to 2 </p>
<vlc id="end" />
```

You can also just use "if" and "end".

The *value* macro

If param1 isn't empty, it is first executed with the RPN evaluator. The macro is replaced with the value of the first element of the stack.

Note: If the element is the name of a local variable, its value will be displayed instead of its name.

The *foreach,end* macro

param1 is the name of the variable that will be used for the loop. param2 is the name of the set to be built:

- *integer*: take the first element from the stack to construct a set of integer. The stack element should be a string like: *first:last[:step][,first2:last2[:step2][,...]* (Ex: 1:5:2,6:8:1 will be expanded into 1,3,5,6,7,8)
- *directory*: take the first element of the stack as the base directory and construct a set of filename and directly in it. Each element has the following fields:
 - name: file/directory name
 - type: "directory" or "file" or "unknown"
 - size: size of the file
 - date
- *playlist*: set based on the playlist with fields: current is 1 if item is currently selected, 0 else. index is the index value, that can be used by the play or delete control command. name is the name of the item.
- *"information"*: Create information for the current playing stream. name is the name of the category, value is its value, info is a new set that can be parsed with a new foreach (subfields of info are name and value).
- *input variables* such as "program", "title", "chapter", "audio-es", "video-es" and "spu-es": Create lists for the current playing stream. Every list has the following fields:
 - name: item name (language for elementary streams, tracks, etc.) to display in places where a human-readable format is preferred
 - id: item ID to pass to the RPN function `vlc_var_set`, and returned by `vlc_var_get`
 - selected: 1 if the item is selected, 0 otherwise
- the name of a foreach variable if it's a set of set of value.

```
<vlc id="foreach" param1="cat" param2="informations" />
  <p> <vlc id="value" param1="cat.name" />
  <ul>
```

```
<vlc id="foreach" param1="info" param2="cat.info" />
  <li>
    <vlc id="value" param1="info.name" /> :
      <vlc id="value" param1="info.value" />
    </li>
  <vlc id="end" />
</ul>
<vlc id="end" />
```

For more details, have a look at the `share/http` directory of the VLC source tree...