

VideoLAN HOWTO

Alexis de Lattre

Marc Ariberti

Anil Daoud

Clément Stenac

VideoLAN HOWTO

by Alexis de Lattre, Marc Ariberti, Anil Daoud, and Clément Stenac

Copyright © 2002-2004 the VideoLAN project

This document describes how to use the complete VideoLAN streaming solution.

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

Table of Contents

1. Introduction.....	1
What is the VideoLAN project ?.....	1
What is a codec ?	3
How can I use VideoLAN ?	3
Command line usage	4
2. Install the required software and hardware.....	8
Install VLC.....	8
Install VLS	11
3. Receive and save a stream	13
Receive a stream with VLC	13
Save a stream with VLC.....	13
Receive a stream with a set-top-box	14
4. Stream a file.....	15
Stream a file with VLC	15
Stream a file with VLS.....	15
5. Stream a DVD	17
Stream a DVD with VLC.....	17
Stream a DVD with VLS	17
6. Stream a DVB channel (satellite or digital terrestrial TV).....	19
Install the DVB drivers.....	19
Stream with VLS.....	19
Stream with VLC	19
7. Stream from an MPEG encoding card	21
Stream with the Hauppauge WinTV-PVR-250/350 card	21
Stream with the Visiontech Kfir card.....	23
8. Stream from an acquisition card or a webcam	26
Install the Video for Linux drivers	26
Stream with VLC	26
Stream with VLS.....	27
9. Advanced use of VLC's stream output (transcoding, multiple streaming, etc...)	30
The syntax	30
Examples.....	30
10. Stream in IPv6.....	33
Stream in IPv6.....	33
Receive in IPv6	34
11. Video On Demand.....	35
Overview	35
On the server side.....	35
On the client side.....	35
12. Add a channel information service.....	36
Send announces when streaming with VLC	36
Send announces for the programs streamed by VLS with the mini-SAP-server	36
Start VLC(s) on the client(s).....	37

A. GNU General Public License	38
Preamble.....	38
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION	38
How to Apply These Terms to Your New Programs.....	42

Chapter 1. Introduction

What is the VideoLAN project ?

Overview

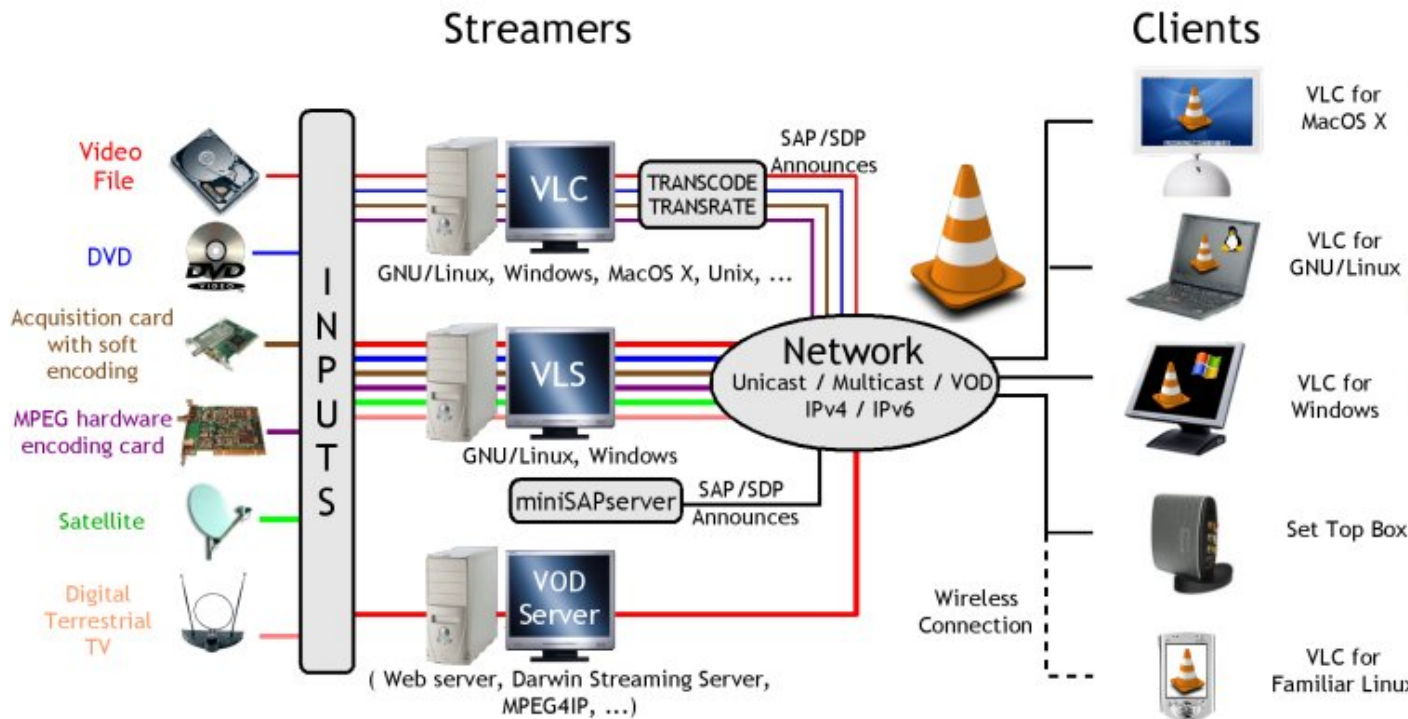
VideoLAN is a complete software solution for video streaming, developed by students of the Ecole Centrale Paris (<http://www.ecp.fr>) and developers from all over the world, under the GNU General Public License (<http://www.gnu.org/copyleft/gpl.html>) (GPL). VideoLAN is designed to stream MPEG videos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

The VideoLAN solution includes:

- VLS (VideoLAN Server), which can stream MPEG-1, MPEG-2 and MPEG-4 files, DVDs, digital satellite channels, digital terrestrial television channels and live videos on the network in unicast or multicast,
- VLC (initially VideoLAN Client), which can be used as a server to stream MPEG-1, MPEG-2 and MPEG-4 files, DVDs and live videos on the network in unicast or multicast ; or used as a client to receive, decode and display MPEG streams under multiple operating systems.

Here is an illustration of the complete VideoLAN solution:

Figure 1-1. Global VideoLAN solution



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

VideoLAN software

VLC

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

- MPEG-1, MPEG-2 and MPEG-4 / DivX files from a hard disk, a CD-ROM drive, ...
- DVDs and VCDs,
- from a satellite card (DVB-S),
- MPEG-1, MPEG-2 and MPEG-4 streams from the network sent by VLS or VLC's stream output.

VLC can also be used as a server to stream:

- MPEG-1, MPEG-2 and MPEG-4 / DivX files,
- DVDs,
- from an MPEG encoding card,

to:

- one machine (i.e. to one IP address): this is called *unicast*,
- a dynamic group of machines that the clients can join or leave (i.e. to a multicast IP address): this is called *multicast*, in IPv4 or IPv6 .

To get the complete list of VLC's possibilities on each platform supported, see the VLC features page (<http://www.videolan.org/vlc/features.html>).

Note: VLC doesn't work on Mac OS 9, and will probably never do.

VLS

VLS can stream:

- an MPEG-1, MPEG-2 or MPEG-4 files stored on a hard drive or on a CD,
- a DVD located in a local DVD drive or copied on a hard disk,
- a satellite card (DVB-S) or a digital terrestrial television card (DVB-T) ,
- an MPEG encoding card ;

to :

- one machine (i.e. to one IP address): this is called *unicast*,
- a dynamic group of machines that the clients can join or leave (i.e. to a multicast IP address): this is called *multicast*, in IPv4 or IPv6 .

A Pentium 100 MHz with 32 MB of memory should be enough to send one stream on the network. When streaming a lot of videos stored on a hard drive, the actual limitation is not the processor but the hard drive and the network connection.

VLS works under Linux and Windows. To get the complete list of VLS's possibilities on each platform supported, see the streaming features page (<http://www.videolan.org/streaming/features.html>).

Mini-SAP-server

You can add a channel information service based on the SAP/SDP standard to the VideoLAN solution. The mini-SAP-server sends announces about the multicast programs on the network in IPv4 or IPv6, and VLCs receive these announces and automatically add the programs announced to their playlist.

The mini-SAP-server works under Linux and Mac OS X.

What is a codec ?

To fully understand the VideoLAN solution, you must understand the difference between a *codec* and a *container format*

- A *codec* is a compression algorithm, used to reduce the size of a stream. There are audio codecs and video codecs. MPEG-1, MPEG-2, MPEG-4, Vorbis, DivX, ... are codecs
- A *container format* contains one or several streams already encoded by codecs. Very often, there is an audio stream and a video one. AVI, Ogg, MOV, ASF, ... are container formats. The streams contained can be encoded using different codecs. In a perfect world, you could put any codec in any container format. Unfortunately, there are some incompatibilities. You can find a matrix of possible codecs and container formats on the features page (<http://www.videolan.org/streaming/features.html>)

To decode a stream, VLC first *demuxes* it. This means that it reads the container format and separates audio, video, and subtitles, if any. Then, each of these are passed *decoders* that do the mathematical processing to decompress the streams .

There is a particular thing about MPEG:

- MPEG is a *codec*. There are several versions of it, called MPEG-1, MPEG-2, MPEG-4, ...
- MPEG is also a container format, sometimes referred to as *MPEG System*. There are several types of MPEG: ES, PS, and TS

When you play an MPEG video from a DVD, for instance, the MPEG stream is actually composed of several streams (called Elementary Streams, ES): there is one stream for video, one for audio, another for subtitles, and so on. These different streams are mixed together into a single Program Stream (PS). So, the .VOB files you can find in a DVD are actually MPEG-PS files. But this PS format is not adapted for streaming video through a network or by satellite, for instance. So, another format called Transport Stream (TS) was designed for streaming MPEG videos through such channels.

How can I use VideoLAN ?

Documentation

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

- the *VideoLAN HOWTO*. This document is the complete guide of the VideoLAN streaming solution. It will give you practical examples to set up your streaming solution.
- the *VLC user guide*. This document is the complete guide for VLC.
- the *VLS user guide*. This document is the complete guide for VLS.
- 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

If you have problems using VideoLAN, and if you don't find the answer to your problems in the documentation, please look at the online archive of the mailing-lists (<http://www.via.ecp.fr/via/ml/videolan-en.html>). There are two English-speaking mailing-lists for the users:

- vlc@videolan.org for the questions on VLC ,
- streaming@videolan.org for the questions on VLS, mini-SAP-server and the network .

If you want to subscribe or unsubscribe to the mailing-lists, please go to the mailing-list page (<http://www.videolan.org/support/lists.html>).

You can also talk with VideoLAN users and developers on IRC: server *irc.freenode.net*, channel *#videolan* .

If you find a bug, please follow the instructions on the bug reporting page (<http://www.videolan.org/support/bug-reporting.html>) .

Command line usage

- VLC has many different graphical interfaces, that are organized quite differently in order to be in harmony with the guidelines of each operating system supported. Documenting the use of each graphical interface is too long, and some features are only available via the command line interface. Therefore we decided to document only the command line interface, but in many cases it should be easy to guess how to use the graphical interface for the same use !
- VLS has a command line and a telnet interface, but no graphical interface !

All the commands that show up in this document should be typed inside a terminal. .

Open a terminal

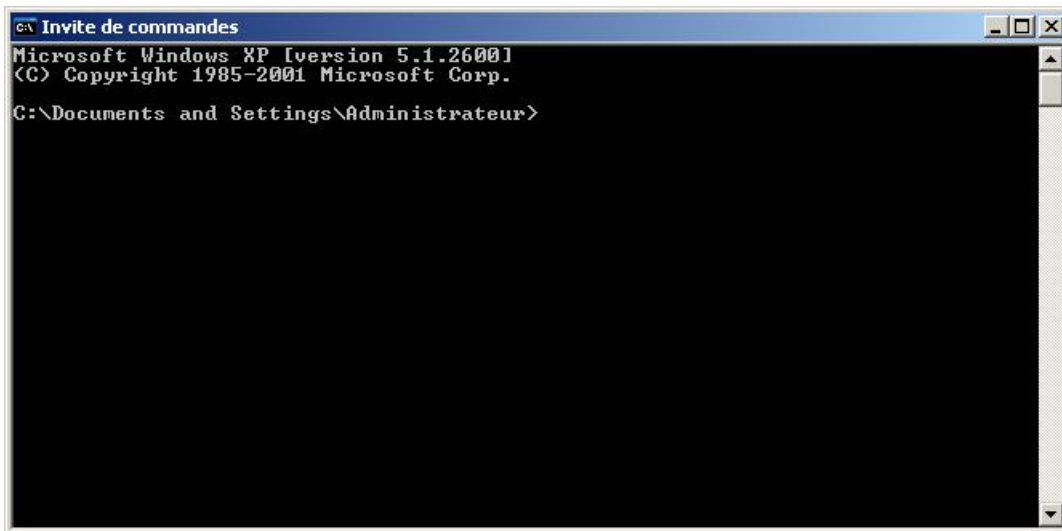
Windows

Click on *Start, Run* and type:

- **cmd Enter** (Windows 2000 / XP),
- **command Enter** (Windows 95 / 98 / ME).

The terminal appears

Figure 1-2. Windows terminal

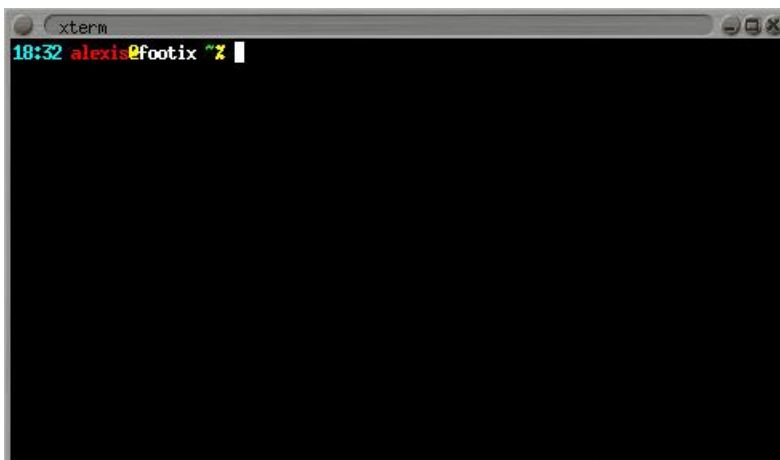


Note: Under Windows, you need to be in the directory where the program is installed to run it.

Linux / Unix

Open a terminal :

Figure 1-3. Linux X terminal



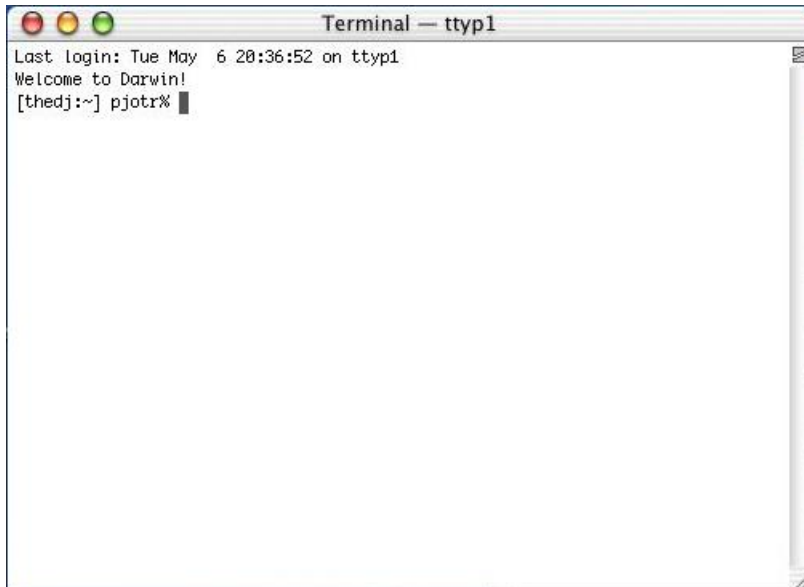
In the documentation, we adopt the following conventions for the Unix commands:

- commands that should be typed as *root* have a # prompt:
`command_to_be_typed_as_root`
- commands that should be typed as a regular user have a % prompt:
% `command_to_be_typed_as_regular_user`

Mac OS X

Go to *Applications*, open the folder *Utilities* and double-click on *Terminal* :

Figure 1-4. Mac OS X terminal

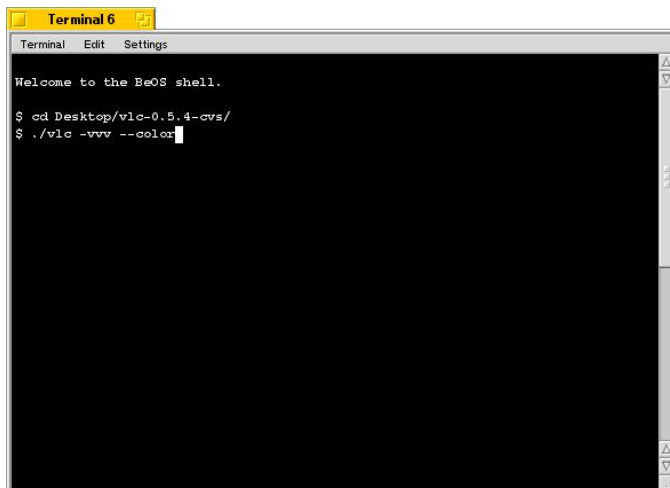


Note: Under Mac OS X, you need to be in the directory where the program is installed to run it, and start the command with *./*.

BeOS

In the deskbar, go to *Application* and then *Terminal*:

Figure 1-5. BeOS terminal



Note: Under BeOS, you need to be in the directory where the program is installed to run it, and start the command with `./`.

Chapter 2. Install the required software and hardware

Install VLC

There are VLC binaries available for the many OSes, but not for all supported OSes. 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 file from the VLC Windows download page (<http://www.videolan.org/vlc/download-windows.html>). Launch the `.exe` to install VLC.

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.

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`).

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 libdvdplay0 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 just 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)

The method below is for any Unix system supported by VLC, for which there is no packages available. It explains how to compile and install VLC and the needed libraries from their source code.

You can also compile VLC under Linux this way if you want to modify the default supported modules.

Install the libraries

Many libraries are needed :

- *libdvbpsi* (compulsory) ,
- *mpeg2dec* (compulsory) ,
- *libdvdcss* if you want to be able to read encrypted DVDs ,
- *libdvdplay* if you want to have DVD menu navigation ,
- *a52dec* if you want to be able to decode the AC3 (i.e. A52) sound format often used in DVDs ,

- *ffmpeg*, *libmad*, *faad2* if you want to read MPEG 4 / DivX files ,
- *libogg* & *libvorbis* if you want to read Ogg Vorbis files .

You can either download the libraries from the VLC sources download page (<http://www.videolan.org/vlc/download-sources.html>) and compile them as described in the next paragraph or get the libraries and the respective development headers from your favorite distributor (via *.deb* and *.rpm* packages for example). However only tarballs from our website are officially supported.

For each library :

- **uncompress :**

```
% tar xvzf library.tar.gz
```

or

```
% tar xvjf library.tar.bz2
```

- **configure :**

```
% cd library
```

```
% ./configure
```

Note: *ffmpeg* needs to be configured with *--enable-pp*

- **compile and install :**

```
% make
```

```
# make install
```

Check that the configuration file */etc/ld.so.conf* contains the following line :

```
/usr/local/lib
```

If the line is not present, add-it and then run (as root): *laufen*:

```
# ldconfig
```

Install VLC

Download the sources of the lastest release : get the file *vlc-version.tar.gz* from the VLC sources download page (<http://www.videolan.org/vlc/download-sources.html>). Uncompress-it :

```
% tar xvzf vlc-version.tar.gz
```

```
% cd vlc-version
```

If you use sources from SVN, you need to run :

```
% ./bootstrap
```

Tarballs are already bootstrapped

To get the list of configuration options, do :

```
% ./configure --help
```

Please note that all the modules are described in the *Modules* section of the VLC User Guide (<http://www.videolan.org/doc/vlc-user-guide/>) .

Examples of very simple configurations:

- if you want a basic VLC, do :

```
% ./configure
```

- if you want the Gnome interface instead of the GTK interface (you will need the development packages of Gnome) :

```
% ./configure --enable-gnome
```

Then, compile and install :

```
% make
```

```
% su
```

```
Password: [Root Password]
```

```
# make install
```

Please note that the installation (**make install** command) is not mandatory. You can execute VLC from where you compiled it.

Install VLS

Windows

Download the ZIP file from the VLS Windows download page (<http://www.videolan.org/streaming/download-vls-windows.html>), unzip-it and run `setup.exe`.

GNU/Linux & Mac OS X

Install the libraries

Many libraries are needed for particular uses :

- *libdvbpsi* (always needed)
- *libdvdcss* if you want to be able to access encrypted DVDs ,
- *libdvddread* if you want to be able to stream DVDs ,
- *libdvb* if you want to be able to stream from a DVB card (a satellite card or a digital terrestrial TV card).

Download the libraries from the VLS sources download page (<http://www.videolan.org/streaming/download-vls-sources.html>).

For each library, uncompress, configure (unless for *libdvb* which doesn't have a *./configure*), compile and install:

```
% tar xvzf library.tar.gz
```

```
% cd library
```

```
% ./configure
```

```
% make
```

```
# make install
```

On GNU/Linux, check that the configuration file `/etc/ld.so.conf` contains the following line:

```
/usr/local/lib
```

If the line is not present, add-it and then run:

```
# ldconfig
```

On Mac OS X, run :

```
# ranlib /usr/local/lib/*.a
```

Install VLS

Download the sources of the latest release : get the file `vls-version.tar.gz` from the VLS sources download page (<http://www.videolan.org/streaming/download-vls-sources.html>). Uncompress-it and generate `./configure`:

```
% tar xvzf vls-version.tar.gz
% cd vls-version
```

To get the list of configuration options, do :

```
% ./configure --help
```

Then configure VLS:

- if you want a basic VLS without DVD support, do möchten, machen Sie :

```
% ./configure --disable-dvd
```

- if you want a VLS with DVD support, do:

```
% ./configure
```

- if you want a VLS with DVB support, do:

```
% ./configure --enable-dvb --with-dvb=PATH_TO_DVB_DRIVERS --with-libdvd=PATH_TO_LIBDVB
```

Then, compile and install:

```
% make
# make install
```

You can also do a **make uninstall**, **make clean** or **make distclean** as needed.

Chapter 3. Receive and save a stream

Receive a stream with VLC

Receive an unicast stream

```
% vlc -vvv udp:
```

Receive a multicast stream

```
% vlc -vvv udp:@239.255.12.42
```

where *239.255.12.42* is the multicast IP address you want to join.

Receive an HTTP/FTP/MMS stream

Use one of the following command lines:

- ```
% vlc -vvv http://example/stream.xyz
```

where *http://example/stream.xyz* is the HTTP address of the stream;

- ```
% vlc -vvv ftp://example/stream.xyz
```

where *ftp://example/stream.xyz* is the FTP address of the stream;

- ```
% vlc -vvv mms://viptvr.yacast.fr/encoderfranceinfo
```

where *mms://viptvr.yacast.fr/encoderfranceinfo* is the MMS address of the stream.

### Receive a RTP stream available through RTSP

```
% vlc -vvv rtsp://www.hardradio.com/tonbeme.mov
```

where *rtsp://www.hardradio.com/tonbeme.mov* is the address of the stream.

### Receive a stream described by an SDP file

```
% vlc -vvv http://server.example.org/stream.sdp
```

## Save a stream with VLC

VLC can save the stream to the disk. In order to do this, use the Stream Output of VLC : you can do it via the graphical interface, or you can add to the command line the following argument:

```
--sout file/muxer:stream.xyz
```

where:

- *muxer* is one of the formats supported by VLC's stream output, i.e. :
  - *avi* for AVI format,
  - *ogg* for OGG format,
  - *ps* for MPEG2-PS format,
  - *ts* or *ts\_dvbpsi* for MPEG2-PS format (the latter uses *libdvbpsi* for the task).
- and *stream.xyz* is the name of the file you want to save the stream to, with the right extension.

## Receive a stream with a set-top-box

Some set-top-boxes with Ethernet cards can receive MPEG2-TS streams over UDP and support multicast. This is the case of the Pace (<http://www.pace.co.uk>) and Aminocom (<http://www.aminocom.com>) set top boxes. We have tested the *AmiNET 100* set-top-box from Aminocom: it is nice and small, and it works very well with VideoLAN.

# Chapter 4. Stream a file

## Stream a file with VLC

```
% vlc -vvv video1.xyz --sout udp:192.168.0.42 --ttl 12
```

where:

- `video1.xyz` is the file you want to stream,
- `192.168.0.42` is either:
  - the IP address of the machine you want to unicast to;
  - or the DNS name the machine you want to unicast to;
  - or a multicast IP address.
- `12` is the value of the TTL (Time To Live) of your IP packets (which means that the stream will be able to cross 11 routers).

If you want to stream the file continuously, add the `--loop` option.

## Stream a file with VLS

```
% vls -vv -d udp:192.168.0.42 file:video1.xyz --ttl 12
```

where:

- `video1.xyz` is the file you want to stream,
- `192.168.0.42` is either:
  - the IP address of the machine you want to unicast to;
  - or the DNS name the machine you want to unicast to;
  - or a multicast IP address.
- `12` is the value of the TTL (Time To Live) of your IP packets (which means that the stream will be able to cross 11 routers).

When you want to stop VLS, use the key combination **Ctrl-c**.

If you want to stream the file continuously, add the `--loop` option.

**Note:** VLS can stream MPEG files that meet two criteria

- the file must be MPEG *PS* (Program Stream) or MPEG *TS* (Transport Stream), that contain video and audio multiplexed. VLS cannot stream MPEG *ES* (Elementary Stream), i.e. a file with only audio or video.

In order to know if an MPEG file is MPEG *PS*, MPEG *TS* or MPEG *ES*, read the file with VLC and look at the messages (in the messageswindow or use the command line `vlc -vvv`).

- If you see a line:

```
[00000107] main module debug: using demux module "ts_dvbpsi"
```

it means the file is MPEG TS.

- If you see a line:

```
[00000109] main module debug: using demux module "ps"
```

it means the file is MPEG PS.

- If you see a line:

```
[00000109] main module debug: using demux module "es"
```

it means the file is MPEG ES, VLS can't stream it.

- the sequence header of the video must repeat itself regularly, which is often the case with MPEG-2, but very rare with MPEG-1. There is no easy way to know if the sequence header is repeated regularly. Files with a *.vob* extension are normally MPEG-2 files and files with *.mpg* or *.mpeg* extension are usually MPEG-1 files.

You can download this streamable MPEG-2 PS file for your tests : *presentation\_short.vob*  
([ftp://ftp.videolan.org/pub/videolan/streams/presentation/presentation\\_short.vob](ftp://ftp.videolan.org/pub/videolan/streams/presentation/presentation_short.vob)).

## Chapter 5. Stream a DVD

**Note:** Under Unix/Linux, you must have write access to the device corresponding to your DVD drive. For that, you should be in the *disk* or *cdrom* group (look at the permissions in */dev*). If you're not, add yourself to the group:

```
adduser your_login disk_or_cdrom
```

and then restart your session.

### Stream a DVD with VLC

```
% vlc -vvv dvdsimple:/dev/dvd --sout udp:192.168.0.12 --ttl 12
```

where:

- */dev/dvd* is the name of your DVD drive (put *D*: under Windows if *D* is the letter of your DVD drive) or the directory where you copied your DVD ,
- *192.168.0.42* is either:
  - the IP address of the machine you want to unicast to;
  - or the DNS name the machine you want to unicast to;
  - or a multicast IP address.
- *12* is the value of the TTL (Time To Live) of your IP packets (which means that the stream will be able to cross 11 routers).

If you want to stream the DVD continuously, add the *--loop* option.

### Stream a DVD with VLS

**Note:** This is possible under GNU/Linux only.

```
% vls -vv -d udp:192.168.0.42 dvd:/dev/dvd --ttl 12
```

where:

- */dev/dvd* is the name of your DVD drive or the directory where you copied your DVD,
- *192.168.0.42* is either:
  - the IP address of the machine you want to unicast to;
  - or the DNS name the machine you want to unicast to;
  - or a multicast IP address.
- *12* is the value of the TTL (Time To Live) of your IP packets (which means that the stream will be able to cross 11 routers).

If you want to stream the DVD continuously, add the `--loop` option.

# Chapter 6. Stream a DVB channel (satellite or digital terrestrial TV)

**Note:** This is possible under GNU/Linux only.

## Install the DVB drivers

If you want to be able to stream from a DVB card (a satellite card or a digital terrestrial TV card), you need to install the DVB drivers:

- if you use a Linux 2.6.x kernel, you just need to compile the right modules;
- if you are using a Linux 2.4.x kernel, you must download the latest release of the DVB drivers from the DVB drivers download page (<http://www.linuxtv.org/download/dvb/>) of the LinuxTV (<http://www.linuxtv.org/>) Project.

Uncompress the tarball and follow the instructions written in the `INSTALL` file to compile and install the drivers.

## Stream with VLS

Put a `.dvbrc` file containing the DVB channels (satellite or digital terrestrial TV channels) you want to stream in your home directory (some are provided in the `libdvb` tarball for the satellite channels).

Run VLS with the following command line :

```
% vls -vv -d udp:192.168.0.42 dvb:"EUROSPORT" --ttl 12
```

where:

- `"EUROSPORT"` is the channel you want to stream as written in your `~/ .dvbrc` file ,
- `192.168.0.42` is either :
  - the IP address of the machine you want to unicast to;
  - or the DNS name the machine you want to unicast to;
  - or a multicast IP address.
- `12` is the value of the TTL (Time To Live) of your IP packets (which means that the stream will be able to cross 11 routers).

## Stream with VLC

**Note:** For this, you must use the SVN version of VLC and compile it with `./configure --enable-dvb` This feature should still be considered as experimental.

```
% vlc -vvv --program 8211 dvb:12597000:0:3:27500000 --sout udp:192.168.0.12 --ttl 12
```

where:

- 8211 is the PID of the DVB channel you would like to stream, expressed as a decimal number;
- 12597000 is the frequency of the transponder you want to stream from in Hz;
- 0 is the polarization (0 for vertical, 1 for horizontal);
- 3 is the FEC (3 for 3/4, 9 for auto);
- 27500000 is the transponder symbolrate in Hz;
- *192.168.0.12* is either:
  - the IP address of the machine you want to unicast to;
  - or the DNS name the machine you want to unicast to;
  - or the multicast IP address.
- *12* is the value of the TTL (Time To Live) of your IP packets (which means that the stream will be able to cross 11 routers).

# Chapter 7. Stream from an MPEG encoding card

**Note:** This is possible under GNU/Linux only.

VideoLAN supports two kinds of MPEG-2 encoding cards:

- Hauppauge WinTV-PVR-250 and WinTV-PVR-350,
- Visiontech Kfir.

The Hauppauge WinTV-PVR-250/350 gives much better results and is cheaper than the Visiontech Kfir.

## Stream with the Hauppauge WinTV-PVR-250/350 card

### Install the drivers

First, you will have to patch your kernel to support the v4l2 API (Video 4 Linux version 2). The patch is available on the Video4Linux HQ (<http://bytesex.org/v4l/>).

Once your kernel is ready, install the CVS version of the Linux drivers for the Hauppauge WinTV-PVR-250/350. They are hosted on [ivtv.sourceforge.net](http://ivtv.sourceforge.net) (<http://ivtv.sourceforge.net/>). Then, you will have to create the device and load the modules; for this, please refer to the documentation shipped with the drivers.

### Stream with VLC

**Note:** You must add `--enable-pvr` to `./configure` to use this feature.

```
% vlc -vvv pvr:/dev/video0:norm=secam:size=720x576:frequency=576250:
bitrate=3000000:maxbitrate=4000000 --sout udp:192.168.0.42 --ttl 12 --cr-average 1000
```

where :

- `/dev/video0` is the device corresponding to the encoding card ,
- `norm=secam` is name of the standard of the analogic signal (possible values are pal, secam, and ntsc) ,
- `size=720x576` is the size of the video you want to stream ,
- `frequency=576250` is the frequency in kHz of the channel you want to stream ,
- `bitrate=3000000` is the average bitrate of the stream ,
- `maxbitrate=4000000` is the maximum bitrate of the stream ,
- `192.168.0.42` is either :
  - the IP address of the machine you want to unicast to ;
  - or the DNS name the machine you want to unicast to ;
  - or a multicast IP address.

- 12 is the value of the TTL (Time To Live) of your IP packets (which means that the stream will be able to cross 11 routers).
- 1000 is a secret value to work around a bug of the card.

## Stream with VLS

You must use the SVN version of VLS.

There is no command line interface for this input for the moment, so you will have to use the configuration file.

Here is a typical configuration file `vls.cfg` to run VLS with a Hauppauge WinTV-PVR-250/350 encoding card:

```
vls.cfg (VLS configuration file)
Example of the VideoLAN HOWTO for Hauppauge WinTV-PVR-250/350 encoding cards

begin "inputs"

 pvr0 = "video"

end

begin "input"

 trickplay = "normal"

end

begin "pvr0"

 # This card is a Hauppauge WinTV-PVR-250/350
 cardtype = "pvr"

 # Device name
 device = "/dev/video0"

 # Stream type
 type = "mpeg2-ps"

 # Norm : put "pal", "secam" or "ntsc"
 norm = "pal"

 # Framerate : put "30" for NTSC and "25" for PAL and SECAM
 framerate = "25"

 # Input number : put - "0" for Composite on S-video plug
 # - "4" for tuner
 # - "6" for S-video on S-video plug
 inputnumber = "4"

 # If you use the tuner, put the frequency
 frequency = "567250"

 # Image size : put "widthxheight" or "subqcif" (128x96) or "qsif" (160x120)
 # or "qcif" (176x144) or "sif" (320x240) or "cif" (352x288) or "vga" (640x480)
 size = "720x576"

 # Bitrate in Kbit/s
 bitrate = "7000"
```

```

Maximum bitrate in Kbit/s
maxbitrate = "9000"

Bitrate mode : put "cbr" (constant bitrate) or "vbr" (variable bitrate)
bitratemode = "vbr"

end

begin "channels"

 channel1 = "network"

end

begin "channel1"

 # Unicast or multicast IP address
 dsthost = "192.168.0.42"

 # Destination port
 dstport = "1234"

 # If it's a multicast IP address, uncomment the line below
 #type = "multicast"

 # If it's a multicast IP address, set the "Time To Live" below
 #ttl = "12"

end

begin "launchonstartup"

 command1 = "start video channel1 pvr0"

end

Once you have adapted the configuration file above, run VLS:

% vls -vv -f vls.cfg

```

## Stream with the Visiontech Kfir card

### Install the drivers

If you want to be able to stream from a Visiontech Kfir card, you need to install its Linux drivers. Download the latest release of the drivers from the drivers download page (<http://www.linuxtv.org/download/mpeg2/>) of the LinuxTV web site (<http://www.linuxtv.org/>).

Uncompress the tarball and follow the instructions written in the `INSTALL` file to compile and install the drivers.

**Note:** If you have a VIA chipset, you need to disable USB in the BIOS.

## Stream with VLC

```
% vlc -vvv kfir:/dev/video --sout udp:192.168.0.42 --ttl 12
```

where :

- `/dev/video` is the device corresponding to the Kfir card ,
- `192.168.0.42` is either :
  - the IP address of the machine you want to unicast to ;
  - or the DNS name the machine you want to unicast to ;
  - or a multicast IP address.
- `12` is the value of the TTL (Time To Live) of your IP packets (which means that the stream will be able to cross 11 routers).

## Stream with VLS

There is no command line interface for this input for the moment, so you will have to use the configuration file.

Here is a typical configuration file `vls.cfg` to run VLS with a VisionTech Kfir encoding card:

```
vls.cfg (VLS configuration file)
Example of the VideoLAN HOWTO for the Visiontech Kfir encoding card

BEGIN "Inputs"

 kfir = "video"

END

BEGIN "kfir"

 # Visiontech Kfir device
 Device = "/dev/video"

 # Stream type (default is "Mpeg2-PS")
 Type = "Mpeg2-PS"

END

BEGIN "Channels"

 channel1 = "network"

END

BEGIN "channel1"

 # Unicast or multicast IP address
 DstHost = "192.168.0.42"

 # Destination port
 DstPort = "1234"
```

```
If it's a multicast IP address, uncomment the line below
#Type = "multicast"

If it's a multicast IP address, set the "Time To Live" below
#TTL = "12"

END

BEGIN "LaunchOnStartUp"

 command1 = "start video channell kfir"

END
```

Once you have adapted the configuration file above, run VLS:

```
% vls -vv -f vls.cfg
```

# Chapter 8. Stream from an acquisition card or a webcam

**Note:** This is possible under GNU/Linux only.

## Install the Video for Linux drivers

If you want to stream from an acquisition card or a webcam, a video4linux driver must be available for it. You can find more information about video4linux and supported devices here (<http://www.exploits.org/v4l>).

Compile the right module for your device, and insert it into your kernel (Some video4linux modules are shipped with the 2.4.x Linux kernels). You can test your device by using any of the listed programs in the *Video: TV and PVR/DVR* section of this page (<http://www.exploits.org/v4l/>).

Note that v4l2 modules will also work with VLC.

## Stream with VLC

**Note:** You must add `--enable-v4l` to `.configure` to use this feature.

```
% vlc -vvv v4l:/dev/video:norm=secam:frequency=543250:size=640x480:channel=0:adev=/dev/dsp:audio=0
--sout '#transcode{vcodec=mp4v,acodec=mpga,vb=3000,ab=256,vt=800000,keyint=80,deinterlace}:std{access=udp,mux=rtmp}'
```

where:

- `/dev/video` is the device corresponding to your acquisition card or your webcam,
- `norm=secam` is name of the standard of the analogic signal (possible values are pal, secam, and ntsc),
- `frequency=543250` is the frequency of the channel in kHz (*Warning* : for VLC < 0.6.1, Frequency is channel frequency in MHz multiplied by 16),
- `size=640x480` is the size of the video you want (you can also put the standard size like *subqcif* (128x96), *qsif* (160x120), *qcif* (176x144), *sif* (320x240), *cif* (352x288) or *vga* (640x480)),
- `channel=0` is the number of the channel (usually 0 is for tuner, 1 for composite and 2 for svideo),
- `adev=/dev/dsp` is the audio device,
- `audio=1` is the number of the audio channel (usually 0 is for mono and 1 for stereo),
- `vcodec=mp4v` is the video format you want to encode in (*mp4v* is MPEG-4, *mpgv* is MPEG-1, and there is also *h263*, *DIV1*, *DIV2*, *DIV3*, *I420*, *I422*, *I444*, *RV24*, *YUY2*),
- `acodec=mpga` is the audio format you want to encode in (*mpga* is MPEG audio layer 2, *a52* is A52 i.e. AC3 sound),
- `vb=3000` is the video bitrate in Kbit/s es el bitrate de vídeo en Kbit/s ,
- `ab=256` is the audio bitrate in Kbit/s
- `vt=800000` is the video bitrate tolerance in bit/s,

- `keyint=80` is the maximum interval between two "I" images in number of images,
- `deinterlace` tells VLC to deinterlace the video on the fly,
- `192.168.0.42` is either:
  - the IP address of the machine you want to unicast to;
  - or the DNS name the machine you want to unicast to;
  - or a multicast IP address.
- `12` is the value of the TTL (Time To Live) of your IP packets (which means that the stream will be able to cross 11 routers).

## Stream with VLS

**Note:** You need the SVN version of VLS. When you compile it, you must do a `./bootstrap` and `./configure --enable-v4l --with-ffmpeg=PATH` (where `PATH` is the absolute path to the directory where you compiled ffmpeg, and not the relative path!).

There is no command line interface for this input for the moment, so you will have to use the configuration file.

Here is a typical configuration file `vls.cfg` to run VLS with a v4l device:

```
vls.cfg (VLS configuration file)
Example of the VideoLAN HOWTO for the Video for Linux input

BEGIN "Inputs"

 webcam = "v4l"

END

BEGIN "webcam"

 # V4L device
 Device = "/dev/video"

 # Channel Source : Webcam is often "0"
 # TV Card, may be Tuner(0), Composite(1), Svideo(2)
 Channel = "0"

 # Norm: PAL=0, NTSC=1, SECAM=2
 Norm = "0"

 # Frequency of the channel in Mhz, multiplied by 16 (tuner cards only)
 Frequency = "7668"

 # Size possibilities: can be either:
 # - empty string : default size
 # - width x height
 # - subQCIF(128x96), QCIF(160x120), QCIF(176x144), SIF(320x240),
 # CIF(352x288), VGA (640x480)
 Size = ""
```

```

DeInterlace: "0" = no ; "1" = yes
DeInterlace = "0"

Video compression format: "mpeg1" or "mpeg4"
Compression = "mpeg4"

Video bitrate (approx.) in kbps
Bitrate = "500"

Quality: "1.0" (good) to "31.0" (bad)
Quality = "1.0"

Sound device
AudioDevice = "/dev/dsp"

Audio compression format: "mp2", "mp3" or "ogg"
AudioCompression = "mp2"

Audio bitrate in kbps
AudioBitrate = "64"

Audio frequency ("16000", "22050", "24000", "32000", "44100" or "48000")
AudioFreq = "16000"

Audio channel: "1" = mono ; "2" = stereo
AudioChannel = "2"

END

BEGIN "Channels"

 channel1 = "network"

END

BEGIN "channel1"

 # Unicast or multicast IP address
 DstHost = "192.168.0.42"

 # Destination port
 DstPort = "1234"

 # If it's a multicast IP address, uncomment the line below
 #Type = "multicast"

 # If it's a multicast IP address, set the "Time To Live" below
 #TTL = "12"

END

BEGIN "LaunchOnStartUp"

 command1 = "start video channel1 webcam"

END

```

Once you have adapted the configuration file above, run VLS:

```
% vls -vv -f vls.cfg
```

# Chapter 9. Advanced use of VLC's stream output (transcoding, multiple streaming, etc...)

## The syntax

Please refer to the *command line interface* chapter of the VLC user guide to learn the syntax of VLC's stream output. You can find the VLC user guide on the documentation page (<http://www.videolan.org/doc/>).

## Examples

### Transcoding

Transcode the input stream and send it to a multicast IP address with the associated SAP announce:

```
% vlc -vvv input_stream --sout '#transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128,deinterlace}:standard{access=udp,mux=ts,url=239.255.12.42,sap,name="TestStream"}'
```

Display the input stream, transcode it and send it to a multicast IP address with the associated SAP announce:

```
% vlc -vvv input_stream --sout '#duplicate{dst=display,dst="transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128,deinterlace}:standard{access=udp,mux=ts,url=239.255.12.42,sap,name="TestStream"}"}'
```

Transcode the input stream, display the transcoded stream and send it to a multicast IP address with the associated SAP announce:

```
% vlc -vvv input_stream --sout '#transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128,deinterlace}:duplicate{dst=display,dst=standard{access=udp,mux=ts,url=239.255.12.42,sap,name="TestStream"}'}'
```

### Multiple streaming

Send a stream to a multicast IP address and a unicast IP address:

```
% vlc -vvv input_stream --sout '#duplicate{dst=standard{access=udp,mux=ts,url=239.255.12.42,sap,name="TestStream"},dst=standard{access=udp,mux=ts,url=192.168.1.2}}'
```

Display the stream and send it to two unicast IP addresses:

```
% vlc -vvv input_stream --sout '#duplicate{dst=display,dst=standard{access=udp,mux=ts,url=192.168.1.12},dst=standard{access=udp,mux=ts,url=192.168.1.42}}'
```

Send parts of a multiple program input stream:

```
% vlc -vvv multiple_program_input_stream --sout '#duplicate{dst=standard{access=udp,mux=ts,url=239.255.12.42},select="program=12345",dst=standard{access=udp,mux=ts,url=239.255.12.43},select="video,program=1234-2345"}'
```

This command sends the program of the input stream which id is 12345 to 239.255.12.42 and all video programs with id between 1234 and 2345 to 239.255.12.43.

## Transcoding and multiple streaming

Transcode the input stream, display the transcoded stream and send it to a multicast IP address with the associated SAP announce and an unicast IP address:

```
% vlc -vvv input_stream --sout '#transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128,deinterlace}:
duplicate{dst=display,dst=standard{access=udp,mux=ts,url=239.255.12.42,sap,name="TestStream"},
dst=standard{access=udp,mux=ts,url=192.168.1.2}}'
```

Display the input stream, transcode it and send it to two unicast IP addresses:

```
% vlc -vvv input_stream --sout '#duplicate{dst=display,
dst="transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128}:
duplicate{dst=standard{access=udp,mux=ts,url=192.168.1.2},
dst=standard{access=udp,mux=ts,url=192.168.1.12}"}'
```

Send the input stream to a multicast IP address and the transcoded stream to another multicast IP address with the associated SAP announces:

```
% vlc -vvv input_stream --sout '#duplicate{dst=
standard{access=udp,mux=ts,url=239.255.1.2,sap,name="OriginalStream"},
dst="transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128}:
standard{access=udp,mux=ts,url=239.255.1.3,sap,name="TranscodedStream}"}'
```

## HTTP streaming

Stream in HTTP:

- on the server, run:

```
% vlc -vvv input_stream --sout '#standard{access=http,mux=ogg,url=server.example.org:8080}'
```

- on the client(s), run:

```
% vlc http://server.example.org:8080
```

Transcode and stream in HTTP:

```
% vlc -vvv input_stream --sout '#transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128}:
standard{access=http,mux=ogg,url=server.example.org:8080}'
```

For example, if you want to stream an audio CD in Ogg/Vorbis over HTTP:

```
% vlc -vvv cdda:/dev/cdrom --sout '#transcode{acodec=vorb,ab=128}:
standard{access=http,mux=ogg,url=server.example.org:8080}'
```

## RTP streaming

Stream in RTP:

- on the server, run:

```
% vlc -vvv input_stream --sout '#rtp{dst=192.168.0.12,port=1234,sdp=http://server.example.org:8080/test.sdp}'
```

- on the client(s), run:

```
% vlc http://server.example.org:8080/test.sdp
```

## Use the es module

Separate audio and video in two PS files:

```
% vlc -vvv input_stream --sout
'#es{access=file,mux=ps,url_audio=audio-%c.%m,url_video=video-%c.%m}'
```

Extract the audio track of the input stream to a TS file:

```
% vlc -vvv input_stream --sout '#es{access_audio=file,mux_audio=ts,url_audio=audio-%c.%m}'
```

Stream in unicast the audio track on a port and the video track on another port:

- on the server side:

```
% vlc -vvv input_stream --sout
'#es{access=rtp,mux=ts,url_audio=192.168.1.2:1212,url_video=192.168.1.2:1213}'
```

- on the client side: to receive the audio:

```
% vlc udp:@:1212
```

to receive the video:

```
% vlc udp:@:1213
```

Stream in multicast the video and dump the audio in a file:

```
% vlc -vvv input_stream --sout '#es{access_video=udp,mux_video=ts,url_video=239.255.12.42,
access_audio=file,mux_audio=ps,url_audio=audio-%c.%m}'
```

**Note:** You can also combine the es module with the other modules to set-up even more complex solution.

# Chapter 10. Stream in IPv6

## Stream in IPv6

### Requirements

You will need an IPv6-aware operating system, like Linux 2.4 or 2.6 with the *ipv6* module loaded, Windows 2000 with the IPv6 stack, Windows XP Service Pack 1 or Mac OS X version 10.2 or higher. Please look at the features pages on the VideoLAN web site (<http://www.videolan.org>) to know about the status of IPv6 in VLC and VLS for each O.S..

**Note:** Under Windows 2000, you must add by hand a default multicast IPv6 route, with the following command:

```
ipv6 rtu ff::/8 4
```

where the last number (4 in this example) is the number of your true IPv6 interface. To have a list of your IPv6 interfaces, run **ipv6 if**.

### Warning

Under Windows XP, you may have problems with a hidden IPv6 firewall. To solve the problem, go to the list of Windows Services and stop the IPv6 firewalling service.

### Warning

If you are using VMware under Linux, you will have to stop VMware and unload the VMware kernel modules, because we noticed it prevented IPv6 streaming !

### Stream with VLC

```
% vlc -vvv video1.xyz --ipv6 --sout udp:[ff08::1] --ttl 12
```

where:

- `video1.xyz` is the file you want to stream (you can also put **dvdsimple:/dev/dvd** to stream a DVD or any other input configuration) ,
- `ff08::1` is either :
  - the IPv6 address of the machine you want to unicast to ;
  - or the multicast IPv6 address.
- `12` is the value of the TTL (Time To Live) of your IP packets (which means that the stream will be able to cross 11 routers).

**Note:** Under Unix/Linux, you may have to protect the square brackets around the IPv6 address:

```
% vlc -vvv video1.xyz --ipv6 --sout udp:[ff08::1] --ttl 12
```

**Note:** You may have to specify the output network interface:

```
% vlc -vvv video1.xyz --ipv6 --sout udp:[ff08::1%eth0] --ttl 12
```

where *eth0* is the name of the network interface (under Linux the network interfaces are named *ethX*, under Mac OS X it's *enX* and under Windows it's *X*, where *X* is the appropriate number).

## Stream with VLS

You will need to use the configuration file `vls.cfg`. Please, see the VLS user guide (<http://www.videolan.org/doc/>) to know how to write one for IPv6 streaming in unicast or multicast.

## Receive in IPv6

### Receive an unicast stream

```
% vlc -vvv --ipv6 udp:
```

### Receive a multicast stream

```
% vlc -vvv --ipv6 udp:@[ff08::1]
```

**Note:** Under Unix/Linux, you may have to protect the square brackets around the IPv6 address:

```
% vlc -vvv --ipv6 udp:@\[ff08::1\]
```

**Note:** You may have to specify the output network interface:

```
% vlc -vvv video1.xyz --ipv6 --sout udp:[ff08::1%eth0] --ttl 12
```

where *eth0* is the name of the network interface (under Linux the network interfaces are named *ethX*, under Mac OS X it's *enX* and under Windows it's *X*, where *X* is the appropriate number).

# Chapter 11. Video On Demand

## Overview

With Video On Demand (VOD), the user can start the video when he wants, make pauses, go forward and back in the video. It is of course the best in video streaming and the dream for every user.

VOD is a very big consumer of resources for the server and the network. VOD is unicast, not multicast : this means that the network and server resources needed are directly proportional to the number of clients.

The design of VideoLAN's VOD solution is very simple. The idea is to do HTTP streaming, i.e. stream an MPEG video encapsulated in HTTP. The regulation of the bitrate between the client and the server is done automatically by TCP. With HTTP version 1.1, there is the possibility to seek in a file downloaded, that's what we use to seek in the video.

## On the server side

On the VOD server, you need a running Web server. For example, you can use a Linux server running Apache. Other operating systems and other Web servers should work too, but we have never tested.

Make your MPEG-1, MPEG-2 or MPEG-4 / DivX files available to the clients on the Web server.

For example, we have a Web server whose DNS name is *localserver*. On this server, we put an MPEG file `video1.mpg` which will be available to the clients at the URL `http://localserver/video1.mpg`.

## On the client side

```
% vlc -vvv http://localserver/video1.mpg
```

VLC starts to read the stream nearly immediately and you can seek in the stream, make pauses, etc... as if the stream was a local file.

# Chapter 12. Add a channel information service

Typing multicast addresses is not very fun... that's why you need a channel information service ! VideoLAN has implemented a channel information service based on the SAP/SDP standard. The SAP announces about the multicast programs are sent on the network (via the multicast address *224.2.127.254* reserved for this purpose). On the client side, VLCs receive these announces and automatically add the programs announced to their playlist.

VLC can send the SAP announces associated with the programs it is streaming. VLS can't do that, but an independant program, the mini-SAP-server, can send the announces for the programs sent by VLS.

## Send announces when streaming with VLC

To send announces with VLC, you need to use the complex syntax of VLC's stream output, like this:

```
% vlc -vvv input_stream --sout '#standard{access=udp,url=239.255.12.42,sap,name="Test Stream"}' --ttl 12
```

where *video1.xyz* is the file you want to stream, *239.255.12.42* is the multicast IP address you want to stream on, *Test Stream* is the name that will be used for this program in the SAP announces and *12* is the value of the TTL (Time To Live) of the stream and of the SAP announces.

To do the SAP announces in IPv6, just add the *sap\_ipv=6* option:

```
% vlc -vvv input_stream --sout '#standard{access=udp,url=239.255.12.42,sap,name="Test Stream",sap_ipv=6}' --t
```

## Send announces for the programs streamed by VLS with the mini-SAP-server

**Note:** The mini-SAP-server is only available for Linux and Mac OSX.

### Install the mini-SAP-server

Download the latest version of the mini-SAP-server from the streaming download page (<http://www.videolan.org/streaming/>).

Install-it:

```
% tar xvzf miniSAPserver-version.tar.gz
% cd miniSAPserver-version
% ./configure
% make
```

### Configure the mini-SAP-server

Edit the configuration file *sap.cfg* shipped with the tarball. It should contain a global section with the Time To Live (TTL) and the IP version (IPv4 or IPv6) used for the SAP announces and a section per program announced. Use the comments to understand each parameter.

The Time To Live option indicates the maximum number of routers that the SAP announce packets can cross before being dropped.

## Run the mini-SAP-server

Start the mini-SAP-server:

```
% ./sapsrvr -c sap.cfg
```

If you want to run the mini-SAP-server in the background, use the **-d** command line switch. For instance:

```
% ./sapsrvr -d -c sap.cfg
```

You can also omit the **-c** command line switch if your config file is `/etc/sap.cfg`.

## Start VLC(s) on the client(s)

- If the announces are sent in IPv4 :

```
% vlc -vvv --extraintf sap
```

- If the announces are sent in IPv6 :

```
% vlc -vvv --extraintf sap --sap-ipv6
```

Then open the playlist: you should see the names of the programs announced in SAP. When you double-click on the name of a program, VLC will subscribe to the multicast address and start to play the stream !

# 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.