

# **VLC user guide**

**Henri Fallon**

**Alexis de Lattre**

**Johan Bilien**

**Anil Daoud**

**Mathieu Gautier**

**Clément Stenac**

## **VLC user guide**

by Henri Fallon, Alexis de Lattre, Johan Bilién, Anil Daoud, Mathieu Gautier, and Clément Stenac

Copyright © 2002-2004 the VideoLAN project

This document is the complete user guide 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

<b>1. Introduction.....</b>	<b>1</b>
What is the VideoLAN project ?.....	1
What is a codec ? .....	3
How can I use VideoLAN ? .....	3
Command line usage .....	4
<b>2. Modules and options for VLC .....</b>	<b>8</b>
The modules .....	8
Access modules.....	8
Demuxers .....	10
Codec modules.....	12
Video outputs .....	14
Video filters modules .....	17
Audio outputs.....	19
Interface modules.....	21
OS support modules .....	24
Miscellaneous.....	24
Compilation Options .....	25
<b>3. Installing VLC.....</b>	<b>26</b>
Installing VLC.....	26
Uninstalling VLC.....	29
<b>4. The command line interface .....</b>	<b>31</b>
Introduction .....	31
Opening streams.....	31
Modules selection .....	32
Stream Output .....	32
Other Options .....	36
<b>5. The HTTP interface.....</b>	<b>39</b>
Introduction .....	39
The RPN evaluator .....	40
The macros .....	41
<b>6. Other interfaces .....</b>	<b>44</b>
Graphical user interfaces.....	44
Control Interfaces.....	44
<b>7. The Mozilla plugin.....</b>	<b>46</b>
Install the plugin.....	46
Build HTML pages that use the plugin.....	46
<b>A. GNU General Public License .....</b>	<b>48</b>
Preamble.....	48
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION .....	48
How to Apply These Terms to Your New Programs.....	52

# Chapter 1. Introduction

## What is the VideoLAN project ?

### Overview

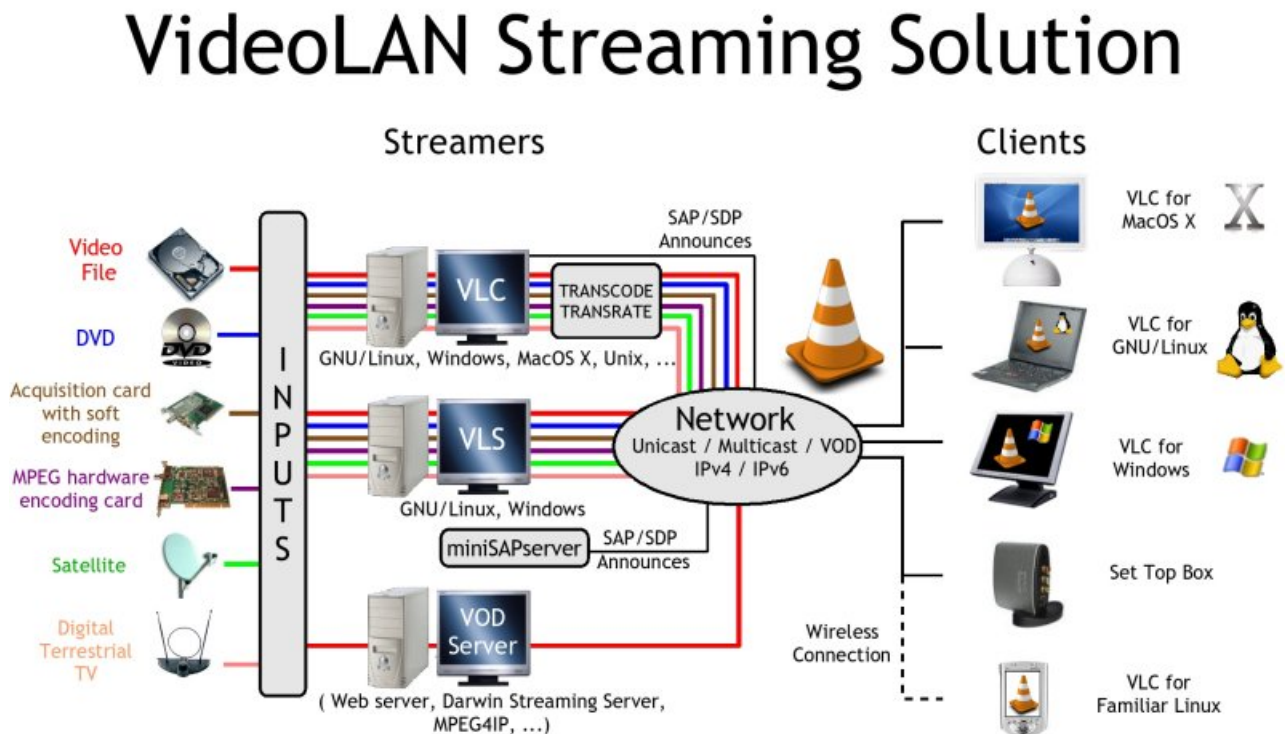
VideoLAN is a complete software solution for video streaming, developed by students of the Ecole Centrale Paris (<http://www.ecp.fr>) and developers from all over the world, under the GNU General Public License (<http://www.gnu.org/copyleft/gpl.html>) (GPL). VideoLAN is designed to stream MPEG videos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

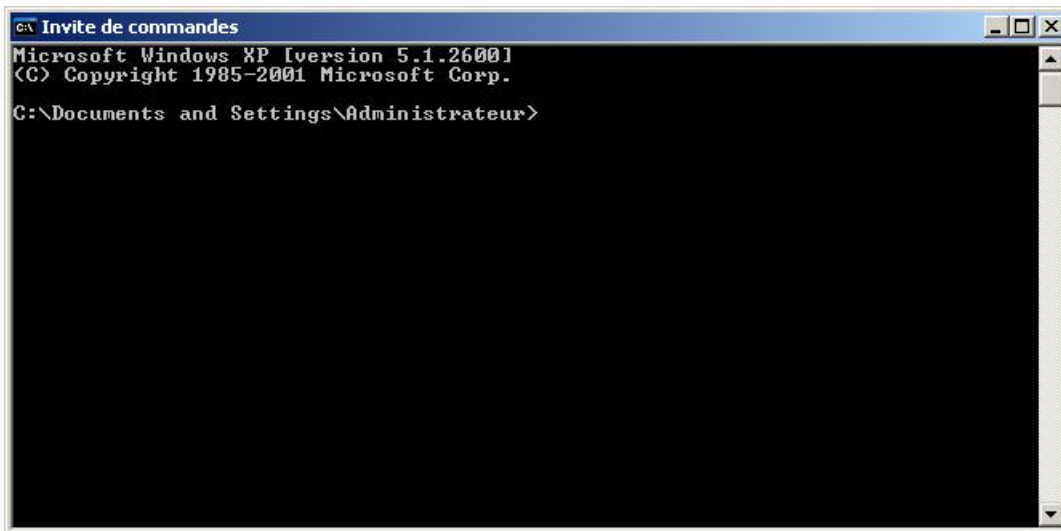
### Windows

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

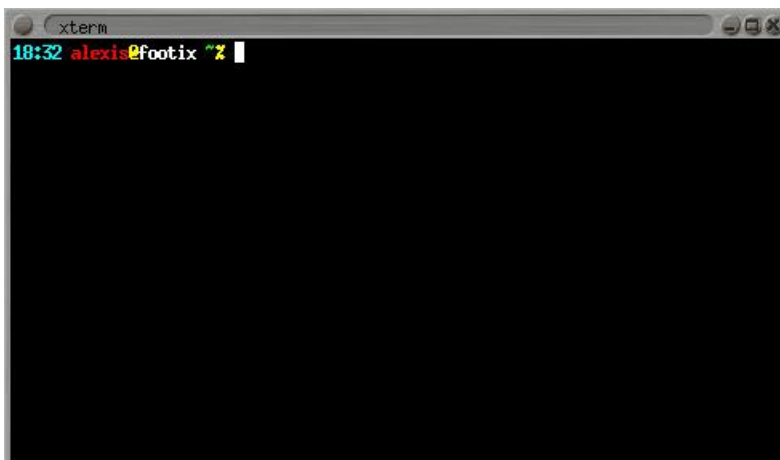


**Note:** Under Windows, you need to be in the directory where the program is installed to run it.

## Linux / Unix

Open a terminal :

Figure 1-3. Linux X terminal



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

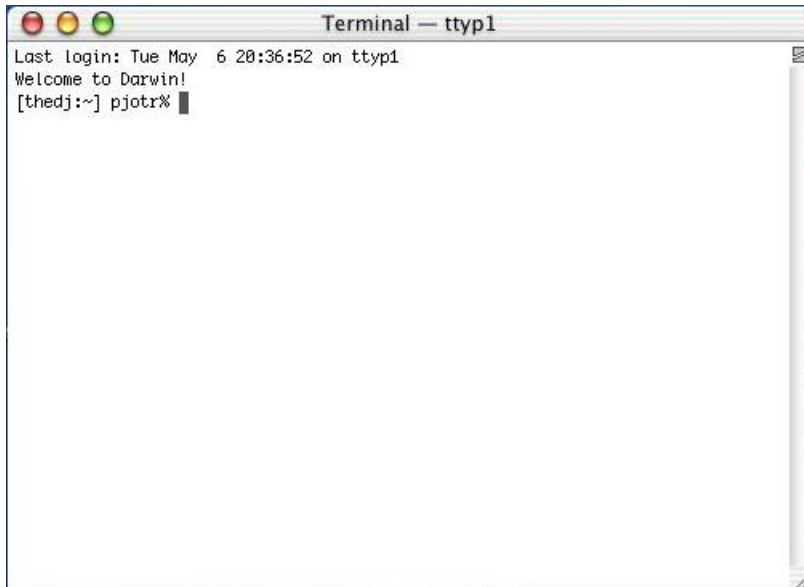
- commands that should be typed as *root* have a # prompt:  
# `command_to_be_typed_as_root`
- commands that should be typed as a regular user have a % prompt:  
% `command_to_be_typed_as_regular_user`



## Mac OS X

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

Figure 1-4. Mac OS X terminal

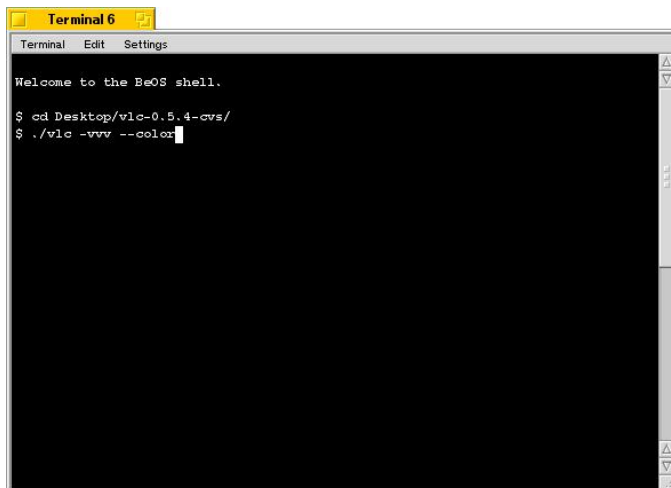


**Note:** Under Mac OS X, you need to be in the directory where the program is installed to run it, and start the command with `./`.

## 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. Modules and options for VLC

## The modules

VLC uses a modular system, which allows to add easily new fonctions and formats. Here is a description of nearly all the VLC modules. A few "internal" modules won't be explained here. For a complete list of all VLC modules, please have a look at the LIST file in the subdirectory "modules" of the VLC source tree.

If you installed VLC through a binary file, you will get the default modules. If, however, you want to customize VLC to your needs, you will have to compile VLC from sources.

Reading this part may be useful if you intend to compile VLC or if you want to know all the options available for modules you use.

The compilation itself is explained in the next chapter.

If you wish to compile a module which is stated *disabled by default*, you have to launch the configure script with:

```
% ./configure --enable-module_name
```

On the contrary, if you want to disable a module that is *default enabled*, you have to use:

```
% ./configure --disable-module_name
```

Each VLC module has its own help and options. To see every options associated with a module, use:

```
% vlc --advanced -p module_name
```

or use the "Preferences" Panel of your favorite graphical interface.

## Access modules

These modules allow VLC to read its streams from different sources. VLC tries to guess the most suitable module at runtime. If you want to force an input module launch VLC with: *--access modulename*

### **cdda**

*default: enabled*

This is the Audio CD input. At present, if you are have a new enough libvcdinfo (from vcdimager (<http://www.vcdimager.org/index.phtml>)) and a new enough libcdio (<http://www.gnu.org/software/libcdio>) you will get the newer navigation-capable plugin. Eventually when libcdio is more mature and handles more OSs the older plugin will disappear. The file `doc/intf-vcd.txt` in the vlc source gives detailed information about this plugin.

### **dvb**

*default: disabled*

*For GNU/Linux only*

This module allows to read from DVB-S, DVB-T, and DBC-C satellite, digital terrestrial, or cable cards. It uses the Video4Linux 2 API, that is only available in kernel 2.5.X and 2.6.X.

## **dvd**

*default: enabled*

This is the old DVD input module. It uses *libdvdcss* for DVD decryption (see the *libdvdcss* page (<http://developers.videolan.org/libdvdcss/>)).

## **dvdplay**

*default: enabled*

This is the regular DVD input module. It will need *libdvdcss* for DVD decryption (see the *libdvdcss* page (<http://developers.videolan.org/libdvdcss/>)) and *libdvdplay* for DVD navigation (see the *libdvdplay* (<http://developers.videolan.org/libdvdplay/>) page).

## **dvdread**

*default: disabled*

This is an alternative to the previous ones. It uses *libdvdread* for DVD reading (see the Ogle download page (<http://www.dtek.chalmers.se/groups/dvd/downloads.shtml>)) and *libdvdcss* for DVD decryption (see the *libdvdcss* page (<http://developers.videolan.org/libdvdcss/>)).

## **dshow**

*default: enabled*

*For Microsoft Windows only*

This module allows VLC to read from input cards that have a DirectShow driver. Most common webcams and acquisitions boards have such a driver.

## **http,ftp,udp,file,directory,mms**

*Always enabled*

These are standard input modules. The HTTP input can be used for Video On Demand.

## **pvr**

*default: disabled*

*For GNU/Linux only*

This module allows to read from Hauppauge PVR cards.

## **rstp/rtp/sdp (livedotcom)**

*default: enabled*

This module allows you to read VOD streams using the RTSP protocol. It uses the liveMedia library from <http://live.com/>

## satellite

*default: disabled*

This is an input module that allows to read directly from a Hauppauge WinTV Nova card under GNU/Linux. It requires drivers 0.9.4 available from linuxtv.org (<http://www.linuxtv.org/>).

## slp

*default: enabled*

This module allows to get the names and addresses for streams announced using the SLP protocol.

## v4l

*default: disabled*

*For GNU/Linux only*

This module allows to read from Video4Linux devices.

## vcd

*default: enabled*

This is the VideoCD input. At present, if you are have a new enough libvcdinfo (from vcdimager (<http://www.vcdimager.org/index.phtml>)) and a new enough libcdio (<http://www.gnu.org/software/libcdio>) you will get the newer navigation-capable plugin. Eventually when libcdio is more mature and handles more OSs the older plugin will disappear. The file `doc/intf-vcd.txt` in the vlc source gives detailed information about this plugin.

## Demuxers

In a video stream, the video signal and the audio one are always into "containers" formats. Demuxers extract the streams from it and pass it to the decoders.

For example, an AVI file can contain a MPEG-4 video, or an uncompressed video. AVI is only a storing format, not a compression format.

### avi

*Always enabled*

This module allows you to read `.avi` files.

Options:

- `--avi-interleaved`, `--no-avi-interleaved` : Enables or disables interlacing mode (default is disabled)
- `--avi-index`, `--no-avi-index` : Forces an index to be created. This can help for buggy streams (default is disabled)

## asf, aac, ogg, aiff, au, wav, mp4

*Always enabled*

This module allows you to read .asf, .aac, .ogg, .aiff, .au, .wav, .mp4 files.

## mkv

*default: enabled*

This module allows you to read files that use the Matroska format.

Options:

- *--mkv-seek-percent, --no-mkv-seek-percent* : When this option is enabled (this is the default behaviour), seek is based on percentage and not on time

## mod

*default: enabled*

This module allows you to read MOD files (extensions mod, s3m, xm, it, 669, amf, ams, dbm, dmf, dsm, far, mdl, med, mtm, okt, ptm, stm, ult, umx, mt2, psm). MOD files are audio files used on AMIGA computers.

Options:

- *--mod-noisereduction, --no-mod-noisereduction* : Default disabled. Add noise reduction to mod audio
- *--mod-reverb, --no-mod-reverb* : Default disabled. Add reverberation to mod audio
- *--mod-reverb-level <integer>* : Reverb level (0-100 defaults to 0)
- *--mod-reverb-delay <integer>* : Reverb delay in ms, 0 to 1000 ms, default to 40. (usually 40-200ms).
- *--mod-megabass, --no-mod-megabass* : Default disabled. Add megabass enhancements to mod audio.
- *--mod-megabass-level <integer>* : Mega bass level (0-100 defaults to 0).
- *--mod-megabass-range <integer>* : Mega bass cut-off (10-100 Hz defaults to 10).
- *--mod-surround, --no-mod-surround* : Default disabled. Add surround enhancements to mod audio.
- *--mod-surround-level <integer>* : Surround level (0-100 defaults to 0).
- *--mod-surround-level <integer>* : Surround delay (0-1000 ms defaults to 5, usually 5-40ms).

## nsv

*default: enabled*

This module allows you to read NSV (Nullsoft Streaming Video) streams. Most NSV streams use the VP6 codec which is currently unsupported by VLC (VLC only supports VP1 and VP3 for the moment).

## ps,ts

*Always enabled*

These modules allow you to read MPEG-2 Program Stream or Transport Stream files.

Options:

- *--vls-backward-compat, --no-vls-backward-compat* : Enable this option if you want to read VLS < 0.4 MPEG 2 TS streams.
- *--buggy-psi, --no-buggy-psi (ts module)* : Enable this option to help handling of files with buggy PSI.

## **pva**

This module allows you to read PVA files (.pva). PVA is a file format used by the software that comes with the TechnoTrend/Hauppage/Siemens/Galaxis DVB cards.

## **real**

*Always enabled*

This module allows you to read RealMedia (.rm) files. Note : Only very old audio codecs (rv10) are currently supported by VLC inside RealMedia files.

## **rawdv**

*Always enabled*

This module allows you to read DV files

## **dvbpsi**

*default: enabled*

This module allows to demux streams from a satellite card.

## **id3,m3u**

*Always enabled*

These modules allow you to read M3U, B4S, PLS, and ASX playlists, and ID3 tags.

# **Codec modules**

The following modules add codec (ie, compression formats) support.

## **a52**

*default: enabled*

This decoder uses liba52 (see the liba52 web site (<http://liba52.sourceforge.net/>) ).

## cinepak

*default: enabled*

This codec decodes the Cinepak format.

## faad

*default: disabled*

Faad is an MPEG-4 audio decoder.

## ffmpeg

*default: enabled*

This is a free MPEG-4/DivX/OpenDivX/Xvid codec (see the ffmpeg web site (<http://ffmpeg.sourceforge.net/>)).

Options:

- *--ffmpeg-dr, --no-ffmpeg-dr* : Enables or disables direct rendering (default enabled)
- *--ffmpeg-error-resilience* : Integer to set the error resilience factor, from -1 to 99 (-1 means no resilience at all)
- *--ffmpeg-workaround-bugs* : Try to fix some bugs. Add the values corresponding to the bugs you want to fix, and pass it as argument: 1 = autodetect, 2 = old msmpeg4, 4 = xvid interlaces, 8 = ump4, 16 = no padding, 32 = ac vlc, 64 = Qpel chroma
- *--ffmpeg-hurry-up, --no-ffmpeg-hurry-up* : Allow the decoder to decode only partially or even to skip frames, if it is late. This can be useful with low end computers, but produces weird pictures.
- *--ffmpeg-truncated* : Truncates the stream : -1=auto, 0=disabled, 1=enabled
- *--ffmpeg-pp-q* : Postprocessing quality (integer between 0 and 6). A higher level requires more CPU power and will produce better pictures.
- *--ffmpeg-pp-name* : Series of postprocessing filters

## libmpeg2

*default: enabled*

This codec allows to read MPEG2 files.

## mad

*default: enabled*

This codec is a very smart MP3 decoder, that only uses integers. This allows its use for CPU which don't handle floating point numbers (on PDA, for example).

## ogt

*default: enabled*



OGT handles the SVCD subtitles (Philips Overlay Graphics Text or OGT) and subtitles used by Chaoji VCDs (also known as CVD).

A description of the format for these kinds of subtitles can be found in the vlc source in `doc/subtitles/svcd-subtitles.txt` and `doc/subtitles/cvd-subtitles.txt` werden

Use the option `--spu-channel number` to select a subtitle track to show. The number should be -1, 0, 1, 2, or 3. The CVD and SVCD subtitle formats allow for 4 subtitles. The value -1 deselects any subtitles .

## **spudec**

*default: enabled*

spudec handles the DVD subtitles. At present full color subtitles and transparency are not handled.

For a description of the format for these kinds of subtitles see <http://sam.zoy.org/writings/dvd/subtitles>.

## **tarkin**

*default: disabled*

tarkin is a new codec (experimental) by the Ogg Project (see the Ogg Vorbis web site (<http://www.vorbis.com>)).

## **theora**

*default: disabled*

theora is a new codec (experimental) by the Ogg Project (see the Ogg Vorbis web site (<http://www.vorbis.com>)).

## **tremor**

*default: disabled*

This is an Ogg/Vorbis codec that only makes integer calculus, which allow its use on CPU which don't have floating point support (see the Ogg Vorbis web site (<http://www.vorbis.com>)).

## **vorbis**

*default: enabled*

This codec allows you to read the Vorbis (audio) encoded files.

## **xvid**

*default: disabled*

This codec allows you to read files encoded with Xvid via the xvidcore library (see Xvid web site (<http://www.xvid.org/>)).

## Video outputs

Video outputs modules allow VLC to display video on your screen. When launched, VLC tries to guess the most suitable video output module for your system. However, if you want to force a specific module, use the following command line switch: `--vout modulename`

### directx

*default enabled on win32*

*For Windows only*

This video output uses Microsoft Direct X libraries. It is recommended for the win32 port.

You may indicate the path to directX libraries and headers with the `--with-directx=PATH` switch, when running the configure script.

### x11

*default: enabled*

*For Unix with X11 servers only*

This is the basic X11 video output. It only requires a working X11 server. You will need xlibs headers to compile it (*xlibs-dev* package on Debian systems).

Options:

- `--x11-display <string>` X11 display name to use
- `--x11-altfullscreen`, `--no-x11-altfullscreen` : use alternate fullscreen mode to bypass window managers (default disabled)
- `--x11-shm`, `--no-x11-shm` : use x11 shared memory (default enabled)
- `--x11-xineramascreen <integer>` : choose the screen to be used for fullscreen mode, when working with xinerama

### xvideo

*default: enabled*

*For GNU/Linux only*

It requires an xvideo compliant graphic card (it is the case for nearly all modern cards). It uses hardware acceleration for YUV transformation and rescaling.

Options:

- `--xvideo-display <string>` : X11 display name to use
- `--xvideo-adaptor <integer>` : xvideo adaptor number
- `--xvideo-altfullscreen`, `--no-xvideo-altfullscreen` : use alternate fullscreen mode to bypass window managers (default disabled)
- `--xvideo-shm`, `--no-xvideo-shm` : use x11 shared memory (default enabled)
- `--xvideo-xineramascreen <integer>` : choose the screen to be used for fullscreen mode, when working with xinerama
- `--xvideo-chroma <string>` : XVimage chroma format

## sdl

*default: enabled*

This video output uses sdl libraries. You need at least version 1.1.6 of this libraries.

You may indicate the path to the *sdl-config* program with the **--with-sdl-config-path=PATH** switch, when running the configure script, if you intend to compile VLC.

## wingdi

*default: enabled on win32*

*For Windows only*

This video output uses GDI. It is designed for users who don't have Direct X, but the perfs are very low. If you have DirectX installed on your system, do not use the wingdi video output

## fb

*default: enabled on GNU/Linux*

*For GNU/Linux only*

This is the frame buffer video output. Your kernel has to be compiled with frame buffer support. sein.

Options:

- **--fbdev <string>** : framebuffer device (default is /dev/fb0)

## glide

*default: disabled*

This video output uses Glide libraries (hardware acceleration for 3Dfx cards).

You may indicate the path to the library with the **--with-glide=PATH** configure option, if you intend to compile VLC.

## mga

*default: disabled*

*For GNU/Linux only*

This module provides hardware acceleration for Matrox cards under GNU/Linux.

## ggi

*default: disabled*

## caca

*default: disabled*

This is the Color ASCII Art Video Output. This video output uses the libcaca library to display video through Color ASCII art. It requires libcaca development headers (*libcaca-dev* package under Debian GNU/Linux) to compile.

## aa

*default: disabled*

This is the ASCII Art Video Output. This video output uses the aalib library to display video through ASCII art. It requires aalib headers (*aalib1-dev* package under Debian GNU/Linux) to compile.

## svglib

*default: disabled*

*For GNU/Linux only*

This is a video output for the SVGAlib library.

## qte

*default: disabled*

*For iPaq only*

This is a video output for QT Embedded, an iPaq-specific graphical library. .

## Video filters modules

These modules allow you to perform modifications on the rendered image (deinterlacing, hue/contrast/saturation adjusting, cropping and so on). To enable them use the following VLC command-line switch: `--filter filter1,filter2,...`

### adjust

*Always enabled*

This filter allows you to alter image contrast, hue, saturation and brightness.

Options:

- `--contrast <float>` : set the image contrast, between 0 and 2. Defaults to 1
- `--brightness <float>` : set the image brightness, between 0 and 2. Defaults to 1
- `--hue <integer>` : Set the image hue, between 0 and 360. Defaults to 0
- `--saturation <float>` : Set the image saturation, between 0 and 3. Defaults to 1

## deinterlace

*Always enabled*

This filter deinterlaces video. It is useful with streams coming from a digital satellite channel or digital terrestrial television channels .

Options:

- `--deinterlace-mode {discard,blend,mean,bob,linear}` : deinterlacing mode (default: discard)

## crop

*Always enabled*

This filter allows you to crop parts of the image.

Options:

- `--crop-geometry <string>` : set the geometry of the zone to crop. This is set as width x height + left offset + top offset (in pixels).
- `--autocrop, --noautocrop` : automatic black border cropping (default disabled)

## transform

*Always enabled*

This filter allows to rotate the video in several ways .

Options:

- `--transform-type {90,180,270,hflip,vflip}` : transformation type, one of '90', '180', '270', 'hflip' and 'vflip'.

## distort

*Always enabled*

This filter adds a distortion effect to the video. Who said it was useless ? :-)

Options:

- `--distort-mode {wave,ripple}` : distortion mode, one of "wave" and "ripple"

## invert

*Always enabled*

This filter inverses colors.

## **motionblur**

*Always enabled*

This filter adds a "motion blur" effect to the image.

Options:

- `--blur-factor <integer>` : the degree of blurring from 1 to 127

## **wall**

*Always enabled*

This filter allows you to have the video cut in pieces in several windows, which you can order as you wish. It can be used to generate image walls with several sources.

Options:

- `--wall-cols <integer>` : Select the number of vertical videowindows in which to split the video
- `--wall-rows <integer>` : Select the number of horizontal videowindows in which to split the video
- `--wall-active <string>` : comma separated list of active windows, defaults to all

## **logo**

*Always enabled*

This filter allows you to insert a png logo inside the video, adjust its transparency with the mouse wheel and move it around using the mouse.

Options:

- `--logo-file <string>` : Logo filename (The file must be in PNG RGBA 8bits format).
- `--logo-x <integer>` : X coordinate of the logo, You can move the logo by left-clicking on it.
- `--logo-y <integer>` : Y coordinate of the logo, You can move the logo by left-clicking on it.
- `--logo-transparency <integer>` : Transparency of the logo (255-0). You can change it by middle-clicking and moving mouse left or right.

## **clone**

*Always enabled*

This filter allows you to clone the image.

- `--clone-count <integer>` : number of video windows in which to clone the video
- `--clone-vout-list <string>` : specific vout modules that you want to activate

## Audio outputs

These modules allow you to choose the way the sound will be output to your audio system. VLC tries to guess the most appropriate audio output module for your system at runtime. However if you want to force a specific output module, use the following command line switch: `--aout modulename`

### coreaudio

*default: enabled on Mac OS X*

*For Mac OS X only*

This audio output uses CoreAudio under Mac OS X

Options:

- `--coreaudio-dev <integer>` : force a specific audio device

### directx

*default: enabled on Windows*

*For Windows only*

This audio output uses directX under Windows

### OSS

*default: enabled on GNU/Linux*

*For GNU/Linux and Unix only*

This is the audio output for OSS (Open Sound System) output (`/dev/dsp`, for example, under Linux). It requires that your kernel was compiled with support for your sound card, or, if you use ALSA (Advanced Linux Sound System), the OSS emulation layer must be active.

Options:

- `--dspdev <string>` : set dsp OSS driver
- `--oss-buggy, --no-oss-buggy` Some buggy OSS drivers just don't like when their internal buffers are completely filled (the sound gets heavily hashed). If you have one of these drivers, then you need to enable this option (default disabled)

### alsa

*default: disabled*

*For GNU/Linux only*

This is the sound output for ALSA (Advanced Linux Sound Architecture). It only works under Linux, and it requires that you installed the ALSA drivers and libraries.

Options:

- `--alsadev <string>` : ALSA device name

## esd

*default: disabled*

*For GNU/Linux & Unix only*

This sound output has ESD (Enlightened Sound Daemon) support (usually used with Gnome). You must have the daemon and its libraries installed.

## arts

*default: disabled*

*For GNU/Linux & Unix only*

This sound output has aRts (KDE's sound server) support. You must have the daemon and its libraries installed.

## waveout

*default: enabled under Windows*

*For Windows only*

This is the Wave output, which is used by the windows port.

## sdl

*default: enabled*

This audio output uses SDL. Please refer to the video output.

## Interface modules

These modules allow you to choose the interface or interfaces you want to use (either graphical or control interfaces).

### dummy

*Always enabled*

Use this interface when you want *no* interface! That is you just want to give a self-contained command-line string.

For example:

```
% vlc --intf dummy vcd://
```

### gestures

*Always enabled*

This module allows you to control VLC via mouse gestures .

Options:



- `--gestures-threshold` : The integer argument is the threshold of move required for a mouse gesture to be recorded
- `--gestures-button {left,middle,right}` : The button to click before a gesture

## gnome

*default: disabled*

*For GNU/Linux only*

This is the Gnome interface. It needs gnome libraries (*libgnome32* package under Debian) and headers (*libgnome-dev* package under Debian) if you wish to compile it. Warning: this interface is no longer maintained, and outdated .

Options:

- `--gnome-tooltips`, `--no-gnome-tooltips` : Whether to show tooltips (default enabled)
- `--gnome-toolbartext`, `--no-gnome-toolbartext` : Whether to show text on buttons (default enabled)
- `--gnome-prefs-maxh` : Argument is an integer that gives the maximum height for configuration windows

## gtk

*default: enabled*

This is the GTK+ interface. It needs gtk libraries and headers files if you are compiling it. Note that it can also be used under Windows. Warning: this interface is no longer maintained, and outdated .

## http

*Always enabled*

This module allows you to remote control your VLC via a web browser. You can create custom web pages. .

Options:

- `--http-host` : The argument, a string, gives the host address. It must look like `address[:port]`, where `address` can be an IP address or a DNS name, and `port` is the port on which the HTTP interface will be available (default is 8080). Default address is `localhost`, which will be ok for almost everybody
- `--http-src` : Gives the path to the HTML pages to be served.

## joystick

*default: disabled*

*For GNU/Linux only*

This module allows you to control VLC via a joystick with many options. .

## kde

*default: disabled*

*For GNU/Linux only*

This is the KDE interface module. You will need the libraries (*kdelibs3* package on Debian) and headers (*kde-devel* package under Debian) if you wish to compile it. Warning: this interface is no longer maintained, and outdated .

Options:

- *--kde-uirc* : Path to the `ui.rc` file

## **lirc**

*default: disabled*

*For GNU/Linux only*

This interface module allows you to control VLC through a remote. A `lircrc` example is provided to help you configure it to your remote (see `doc/lirc/example.lircrc`).

## **ncurses**

*default: disabled*

*For GNU/Linux only*

This is a text interface, using `ncurses` library. You will need `ncurses` headers if you want to compile it (*libncurses5-dev* package on Debian) .

## **opie**

*default: disabled*

This is an interface plugin for the Qt Embedded library (iPaq graphical library) .

## **qt**

*default: disabled*

This is the QT interface module. You will need the libraries (*libqt2* package under Debian) and headers (*libqt-dev* package under Debian) if you wish to compile it. Warning: this interface is no longer maintained, and outdated .

## **rc**

*Always enabled*

This is the Remote Control interface module. It allows you to control VLC via commands, such as *play*, *stop*, etc... or via a script. This interface is text-based, so you should use it when you are in console mode .

Options:

- *--rc-show-pos*, *--no-rc-show-pos* : Indicates whether the position in the stream must be shown (outputs one timer line every seconds) (default disabled)
- *--fake-ty*, *--no-fake-ty* : Enables or disables TTY virtual terminal emulation (default disabled)

## skins

*default: enabled*

This skinnable interface module works under Win32 and X11. You can create your own skins very easily with XML files .

Options:

- `--skin_last` : This option allows you to specify the last used skin (the argument is a string).
- `--skin_config` : This allows you to set the skin configuration (the argument is a string).
- `--show_in_tray, --noshow_in_tray` : Indicates whether the application must be shown in traybar (default disabled).
- `--show_in_taskbar, --noshow_in_taskbar` : Indicates whether the application must be shown in the taskbar (default enabled).

## wxwindows

*default: enabled*

The wxWindows interface is a portable interface that is currently working under GNU/Linux and Windows. It is now the most up-to-date graphical interface available under both Windows and GNU/Linux .

## OS support modules

The following modules add support for different OSs.

### macosx

This is the MacOS X support module, including a native interface.

### qnx

This is the QNX RTOS support module.

## Miscellaneous

This section describes a few more modules that don't belong to any of the categories described before.

### sout

*default: enabled*

Stream Output is a new feature of VLC that allows it to stream an MPEG-1, MPEG-2 or MPEG-4/DivX file or a DVD.

For more details, please have a look at the *The command line interface* section.

## test-suite

*default: disabled*

This builds a special VLC, for testing purposes only.

## mozilla

*default: disabled*

This is not really a module. When enabled, a VLC-based Mozilla plugin is built.

## xosd

*default: disabled*

*For Unix only*

This module outputs the current stream to an "OSD" (On Screen Display).

# Compilation Options

There are a few options that you can set when running the configure script, which are not related to modules.

You can have a look at these options by typing:

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

You can for example control all the installation directories, the system for which you want to build VLC for (if not guessed correctly),...

You can also choose to enable or disable some optimizations.

## --disable-plugins

If you select this option, no plugins will be enabled. This is definitely not recommended, as you would get a very poor VLC, and should only be used for testing purposes.

# Chapter 3. Installing VLC

## Installing 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 libdvdisplay0 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 ,
- *libdvdisplay* 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.

## Uninstalling VLC

### Windows

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

### BeOS

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

### Mac OS X

Drag the VLC application to your trash can.

### Debian GNU/Linux

Remove the packages that you installed:

```
# apt-get remove --purge vlc-gnome vlc-mad libdvdcss2 libdvbpsi1
```

### GNU/Linux Redhat, Mandrake and SuSE

Uninstall the RPM packages that you installed:

```
# rpm -e vlc-version vlc-mad-version vlc-gnome-version
libdvdcss2-version libdvbpsi1-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 4. The command line interface

## Introduction

Many options are only available through command line. They are detailed here.

## Opening streams

The following commands start VLC and add the first element to the playlist.

### Opening a file

Start VLC with:

```
% vlc -vvv my_file.mpg
```

Although VLC should be able to recognize the file type, you may tell VLC what codec to use with the `--codec` option. For example to play `my_file.mpg` using `ffmpeg` audio/video decoder do:

```
% vlc -vvv --codec ffmpeg my_file.mpg
```

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:

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

or (VCD):

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

or (Audio CD):

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

where `device` is the complete path to your DVD or CD-ROM drive.

### Receiving a network stream

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

```
% vlc -vvv udp:[@:server_port]
```

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

```
% vlc -vvv udp:@multicast_address[:server_port]
```

To receive a HTTP stream, start VLC with :

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

To receive a RTSP stream, start VLC with:

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

## Modules selection

VLC tries 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 (for the complete list of modules, see the *Modules and options for VLC* modules section) :

- `--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.
- `--aout <module>` allows you to select the audio output module.
- `--vout <module>` allows you to select the video output module.
- `--filter <module>` allows you to add a video filter module.
- `--memcpy <module>` allows you to choose a memory copy module.

## Stream Output

### Description of the stream output

VLC's stream output allows VLC to be used as a streaming server instead of a client ! It has very extended capabilities:

- stream in unicast and multicast on an IPv4 or IPv6 network everything that VLC is able to read, via UDP, RTP or HTTP;
- save the input stream to a file in AVI, PS, TS or OGG format;
- transcode an input stream, and then, send it, to the network or to a file.

To know about the full possibilities of VLC's stream output, see the [streaming features page](http://www.videolan.org/streaming/features.html) (<http://www.videolan.org/streaming/features.html>).

### Architecture and syntax

the stream output has a powerful architecture that uses modules. Each module has capabilities, and you can *chain* the modules to enhance the possibilities.

Here is the list of the modules currently available:

- *standard* "sends" the stream via an *access output* module: for example, UDP, file, HTTP, ... You will probably want to use this module at the end of your chains.
- *transcode* allows you to transcode the audio and the video of the input stream "on the fly" (if your computer is powerful enough).

- *duplicate* allows you to create a second chain, where the stream will be handled in an independant way.
- *display* allows you to display the input stream, as VLC would normally do. Used with the *duplicate* module, this allows you to view the stream as you send it.
- *rtp* streams over RTP (one UDP port for each elementary stream).
- *es* allows you to make separate Elementary Streams (ES) out of an input stream.

Each of these modules may take options. Here is the syntax that you must use:

```
% vlc input_stream --sout '#module1{option1=...,option2=...}:module2{option1=...,option2=...}:...'
```

You may also use the following syntax :

```
% vlc input_stream --sout-module1-option1=... --sout-module1-option2=... --sout-module2-option1=... --sout-mo
```

For example, to transcode a stream and send it, use:

```
% vlc input_stream --sout '#transcode{options}:standard{options}'
```

## Description of the modules

### standard (alias std)

Sends a stream.

Options:

- *access*: how to send: *file*, *udp*, *rtp*, *http*.
- *mux*: which muxer (ie, which format) will be used. It can be one of *avi* (for AVI format) *ogg* (for OGG format) *ps* (for MPEG2-PS format) *ts* (for MPEG2-TS format).
- *url*: if you use the *file* access, it will be the location where to store the stream; if you use another access, it will be the unicast or multicast IP address where you want to stream.
- *sap*: if you use the *udp* or *rtp* accesses, use this option to announce your stream, using SAP/SDP.
- *name!* This option contains the name under which you want to announce the program.
- *slp*: like *sap*, but use the SLP protocol. You need to have libslp on your system.
- *sap\_ipv*: if you use the *sap* option, use this option to specify if you want to send the SAP announces in IPv4 or IPv6. The value of this option is *4* or *6*.

**Note:** If you are streaming in multicast, you may want to use the global option `--ttl 12` to set the TTL to a value superior to 1.

### display

Displays the stream.

Options:

- *noaudio*: Ignore audio.
- *novideo*: Ignore video.

## duplicate

Duplicates the stream to a new stream output chain.

Options:

- *dst*: A new stream output chain of modules, as described earlier.
- *select*: This option allows you to specify which programs or es you wish to stream for the previous *dst*, if the input stream has several ones.

Format is "program=*program\_id*" or es="*es\_id*". You can also use ranges, like "program=*start\_program-end\_program*". You can select only es of a given type, with :

- no-audio, noaudio, audio
- no-video, novideo, video
- no-spu, nospu, spu

You may use several selectors, comma-separated

## transcode

Changes the codec and/or bitrates for a stream.

Options:

- *acodec*: the new audio codec. It can be one of *mpga* (MPEG audio layer 2), *a52* or *ac3* (AC3 sound) or *vorb* (Vorbis).
- *ab*: audio bitrate in Kbps.
- *vcodec*: the new video codec. It can be one of *mp4v* (MPEG4), *mpgv* (MPEG1), *DIV1*, *DIV2*, *DIV3* (DivX 1,2,3), *H263* (H263), *I263* (H263I), *WMV1* or *WMV2* (Windows Media Video 1 or 2), *MJPG* (MJPEG), *MJPB* (MJPEGB).
- *width*: video width.
- *height*: video height.
- *vb*: video bitrate in kbps.
- *vt*: video bitrate tolerance in bps.
- *deinterlace*: deinterlace the stream.
- *croptop*: number of pixels removed from the top of the video.
- *cropbottom*: number of pixel removed from the bottom of the video.
- *cropleft*: number of pixels removed from the left of the video.
- *cropright*: number of pixels removed from the right of the video.
- *hq*: high quality transcoding (uses more CPU).
- *qmin*: minimum video quantiser scale (VBR)
- *qmax*: maximum video quantiser scale (VBR).

## rtp

Sends a stream over RTP

Options:

- *dst*: destination IP address (unicast or multicast).

- *port*: destination UDP port.
- *sdp*: how to send the sdp file: *http://server\_ip:server\_port/path* for http, or *rtsp://server\_ip:server\_port/path* for rtsp, where *server\_ip* is the address of one of the server's interface, and *path* where the sdp file will be accessible on the server. You can also use *sap://* to announce your stream by SAP.

**Note:** When using *sdp=rtsp*, it is possible not to give any destination address. VLC will then send the stream to any client that connects to it.

## es

Make separate Elementary Streams.

Options:

- *access\_audio*: how to send the audio track: *file*, *udp*, *rtp* or *http*.
- *access\_video*: how to send the vidéo track: *file*, *udp*, *rtp* or *http*.
- *access*: if you want the audio and the video tracks to use the same access, use this option instead of the two above.
- *mux\_audio*: which muxer (ie, which format) will be used for the audio track. It can be one of *avi* (for AVI format), *ogg* (for OGG format), *ps* (for MPEG2-PS format), *ts* (for MPEG2-TS format).
- *mux\_video*: which muxer (ie, which format) will be used for the video track. It can be one of *avi* (for AVI format), *ogg* (for OGG format), *ps* (for MPEG2-PS format), *ts* (for MPEG2-TS format).
- *mux*: if you want the audio and the video tracks to use the same muxer, use this option instead of the two above.
- *url\_audio*: if you use the *file* access, it will be the location where to store the audio track; if you use another access, it will be the unicast or multicast IP address where you want to stream.
- *url\_video*: if you use the *file* access, it will be the location where to store the vidéo track; if you use another access, it will be the unicast or multicast IP address where you want to stream.
- *url*: if you want the audio and the video tracks to use the same url, use this option instead of the two above.

**Note:** In the *url* options, if you use the *file* access, you can use the following macros in the filename:

- *%n* = stream number
- *%c* = FOURCC ;
- *%m* = muxer ;
- *%a* = access.

## Miscellaneous

Here are a few additional global options

- *--sout-all*, *--no-sout-all* Enable streaming of all ES (default disabled). By default VLC will only stream one audio ES and one video ES (the first ones). If you enable *sout-all*, all ES (audio, video and SPU) will be streamed.
- *--sout-keep*, *--no-sout-keep* Keep *sout* open (default disabled) : use the same *sout* instance across the various playlist items, if possible.

- `--no-sout-audio` disables audio stream output.
- `--no-sout-video` disables video stream output.

The stream output also offers a simplified syntax, with which you can only use the *standard* module main options:

```
% vlc input_stream --sout access/mux:url
```

where *access*, *mux* and *url* are as defined in the options of the standard module.

## Examples

To understand fully the complex syntax of VLC's stream output, please look at the use cases of the VideoLAN 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.
- `--aout-rate <integer>` sets the audio output frequency (Hz).
- `--desync <integer>` compensates desynchronization of audio (ms).
- `--headphone` activates headphone virtual spatialization effect.
- `--headphone-dim` sets headphone characteristic dimension.

### Video options

- `--novideo` disables video output.
- `--greyscale` turns video output into greyscale mode.
- `--fullscreen` sets fullscreen video.
- `--nooverlay` disables hardware acceleration for the video output.
- `--width, --height <integer>` sets the video window dimensions.
- `--zoom <float>` adds a zoom factor.
- `--aspect-ratio <mode>` forces source aspect ratio.
- `--spumargin <integer>` forces SPU subtitles position.

## Playlist options

- `--playlist` launches playlist on startup.
- `--random` plays files randomly forever.
- `--enqueue` enqueues items in playlist.
- `--loop` loops playlist on end.

## 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

- `--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.

## Miscellaneous options

- `--quiet` deactivates the console messages.
- `--color` displays color messages.
- `--search-path <string>` specifies interface default search path.
- `--plugin-path <string>` specifies plugin search path.
- `--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.



## Help options

- `--verbose <verbosity>` specifies verbosity level.
- `--help` gives you all available options.
- `--longhelp` gives you a detailed version of the available options.
- `--version` gives you information about the current VLC version.
- `--list` displays a list of available plugins.
- `--module <module>` displays help about specified module.

# Chapter 5. The HTTP interface

## Introduction

*This page is incomplete.*

## Presentation

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)
```

The HTTP interface will start listening at host:port (localhost:8080 if omitted), and will reproduce the structure of /directory at http://host:port/ ( vlc\_source\_path/share/http if omitted ).

VLC is shipped with a set of files that should be enough for generic needs. The rest of this chapter will cover the ways to customize these pages.

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.)
- 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)
- *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

Remark: The stacks, and local variables context is resetted 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: strcat pushes the result of 'ST(1)ST(2)', strcmp compares ST(1) and ST(2), pushes -1 if ST(1)<ST(2), 0 if equal, 1 else, strlen pushes the length of ST(1).
- stack manipulation: dup duplicates ST(1), drop removes ST(1), swap exchanges ST(1) and ST(2), and 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), url\_extract pushes the value of the ST(1) part of the url parameters.

## The macros

### The *control* macro

When asking for a page, you can pass arguments to it through the url. (eg 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 5-1. URL commands**

Name	Argument	Description
play	item (integer)	Play the specified playlist item
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 *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.
- "informations": Create informations 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).
- "hosts": Create the list of host we are listening. Contains the "id" (opaque id), host, ip and port fields.
- "urls": Create the list of urls currently available. Fields are id, stream (1 if we have a stream, 0 else), url, mime, protected (1 if protected, 0 else), used (is it currently used ?).
- "connections": Create the list of current connections. Fields are: id, an opaque id that can be used in the close command, ip, url, and status (HTTP error code).
- 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...

# Chapter 6. Other interfaces

To select an interface: if you want it to be your main interface, either start VLC from the command line with *--intf Myinterface*, or use the Preferences Panel to set the *Interface Module* option. For Graphical User Interfaces, you might also have shortcuts created by the install program. If you want the interface to be an additional one, use the preferences window of your GUI to set the *extra interface modules* option. Alternatively, you can use the *--extraintf* command line option.

## Graphical user interfaces

VLC includes several graphical user interfaces (GUI), according to your operating system

- *Windows*

There are two different graphical user interfaces for Windows. Shortcuts are created for both of them in the Start menu.

*wxWindows*

This interface works both for Linux and Windows. It is currently the most complete and fully-featured interface.

*Skins*

This interface also works both for Linux and Windows. It allows dramatic modifications of the look and feel of your VLC. This is still under development.

- *GNU/Linux*

Many graphical user interfaces are available for GNU/Linux (GTK+, Gnome, QT, KDE), but only two of them are currently maintained: the *wxWindows* and the *Skins* interface. You should probably use one of these.

- *Mac OS X*

Only one native GUI exists for this OS.

## Control Interfaces

Control interfaces are here to provide alternative ways to control your VLC.

### http

This interface was discussed in the previous chapter. It allows to remotely or locally control VLC through a webpage .

### gestures

This allows you to control your VLC by doing mouse gestures in the video output window. By default, it is used by right-clicking and doing the gestures. Gestures are : left (previous), right (next), up-right (fullscreen), down-right (stop).

### corba

This allows you to control VLC using CORBA.

## **dummy**

Use this interface when you want *no* interface! That is you just want to give a self-contained command-line string

For example :

```
% vlc --intf dummy vcd://
```

## **rc**

This allows you to control VLC by typing commands in its terminal window.

## **lirc**

Under GNU/Linux, this allows you to control VLC through an infrared remote control.

## **joystick**

Under GNU/Linux, this allows you to control VLC using a joystick.



# Chapter 7. The Mozilla plugin

VLC can also be embedded in a web browser ! For the moment, this function is only available with Mozilla (<http://www.mozilla.org/>) under GNU/Linux. An experimental plugin is also available for Mac OS X and Microsoft Windows.

## 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
```

### Compile the sources yourself

Install the Mozilla development package (`mozilla-dev` under Debian).

Install the required libraries like for a normal VLC install (from the sources, or from the packages with the development packages).

Download the sources of the latest release: get the file `vlc-version.tar.gz` from the VLC sources download page (<http://www.videolan.org/vlc/download-sources.html>). Uncompress-it, configure-it, compile and install:

```
% tar xvzf vlc-version.tar.gz
% cd vlc-version
% ./configure --enable-mozilla
% make
% su
Password: [Root Password]
# make install
```

## Build HTML pages that use the plugin

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

In this example, the plugin will read a multicast 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="udp:@239.255.12.42" />
<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>
```

# 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.