VLS user guide

Cyril Deguet

Alexis de Lattre

VLS user guide

by Cyril Deguet and Alexis de Lattre

This document is the complete user guide of VLS.

Table of Contents

Chapter 1. Introduction

What is the VideoLAN project ?

Overview

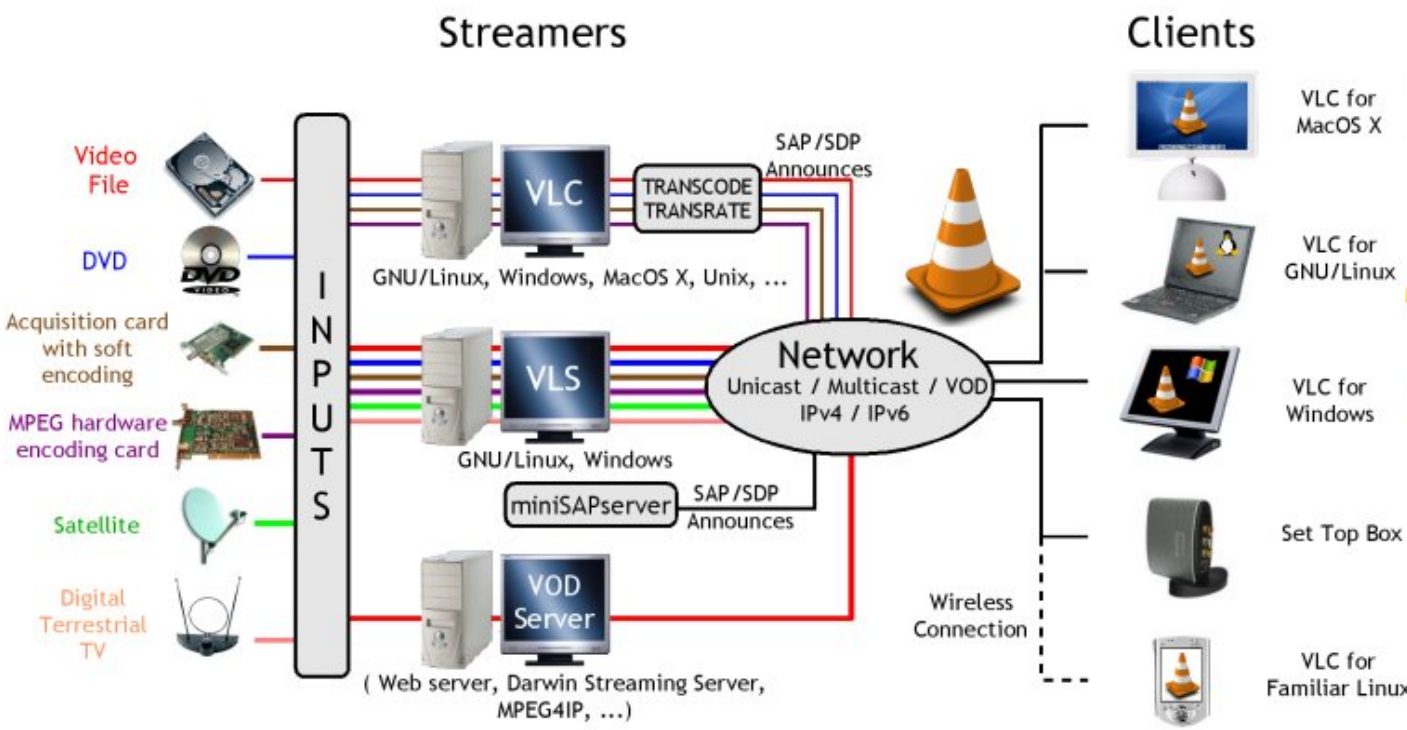
VideoLAN is a complete software solution for video streaming, developed by students of the Ecole Centrale Paris (http://www.ecp.fr) and developers from all over the world, under the GNU General Public License (http://www.gnu.org/copyleft/gpl.html) (GPL). VideoLAN is designed to stream MPEG videos on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

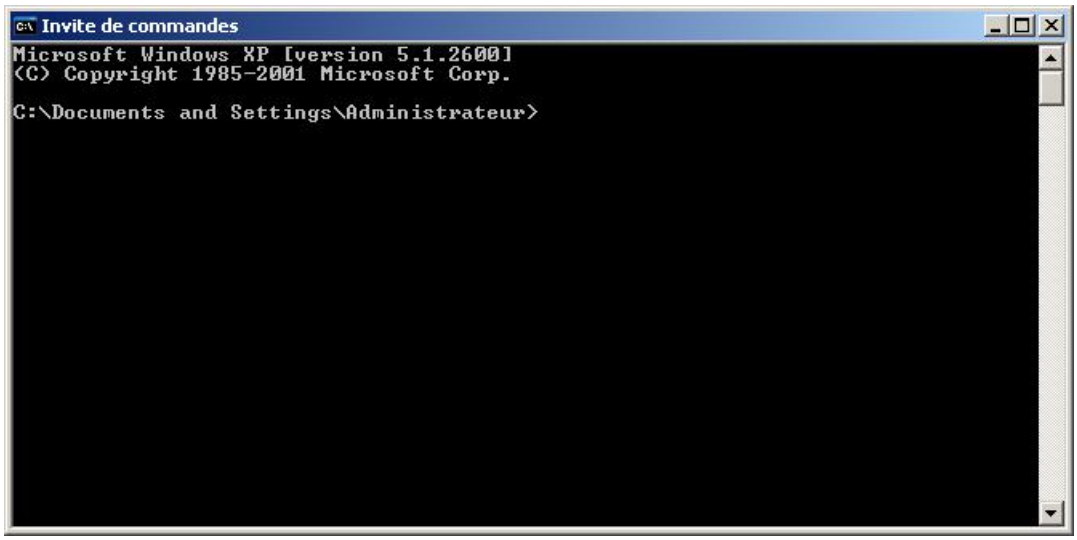
Windows

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
  ```

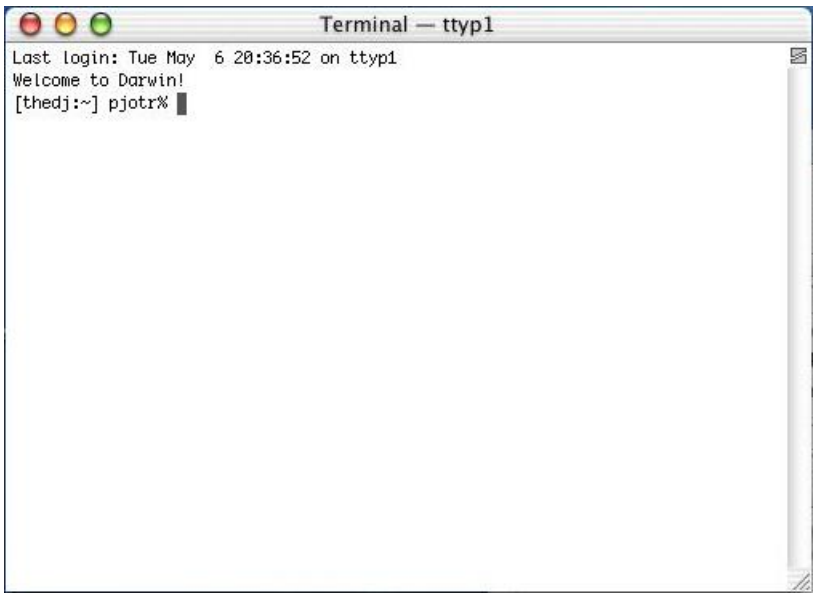
- commands that should be typed as a regular user have a **%** prompt:

  ```
  % 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

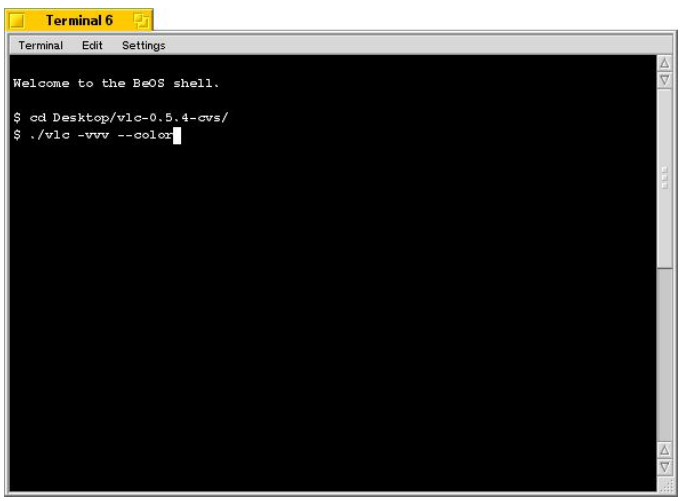


Note: Under Mac OS X, you need to be in the directory where the program is installed to run it, and start the command with **./**.

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. Installing VLS

Installing 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`.

GNULinux & 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 ,
- *libdvdread* 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 terrestial 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-libdvb=PATH_TO_LIBDVB
  ```

Then, compile and install:

```
% make
# make install
```

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

Uninstalling VLS

Windows

Go to the *Control Panel*, click on *Add and remove programs*, select *VLS* and click on *Modify/Remove* and follow the steps to uninstall the program.

If you compiled VLS from sources

Go to the directory containing VLS sources and run:

```
# make uninstall
```

Then you can remove the VLS sources.

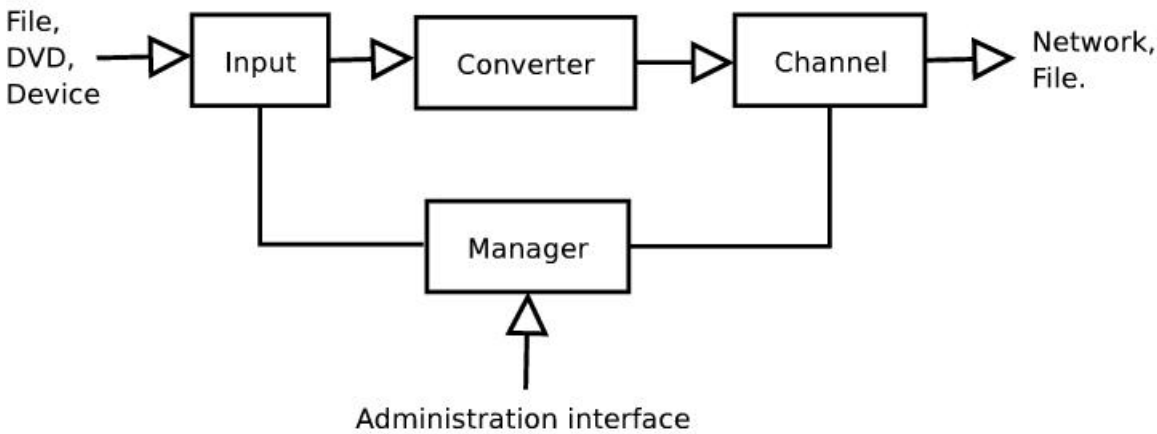
Chapter 3. Overview and basic concepts

VLS structure

From a user's point of view, VLS can be divided into four kinds of components:

- a *manager* ,
- *inputs* ,
- *converters* ,
- and *channels*.

Figure 3-1. VLS structure



Input

The role of an *input* is to read MPEG streams from a given source (file, DVD, DVB card, device, ...), and feed the right *converters* with these streams. An input may be able to read several streams, which are called *programs*. There are several kinds of inputs:

- the *local* input, which can read videos from files or DVDs ,
- the *video* input, which can read videos from MPEG encoding cards devices ,
- the *dvb* input, which can read videos from DVB cards ,
- the *v4l* input, which can read from acquisition cards supported by the Video4Linux drivers.

You can use several inputs and play several programs at the same time.

Converter

The role of a *converter* is to receive a stream from an input, and convert it into the MPEG-TS format. VLS is able to convert PS streams (from DVDs, for instance) into TS streams (ps2ts converter). Of course, it can also read TS streams, and fix them by handling stream discontinuities (ts2ts converter).

Channel

A *channel* receives a stream from a converter, and send it to a given destination (network, file, ...). If you want, you can call a "channel" an "output": it is the same thing !). Currently, two kinds of channels are supported: *network* and *file*. Note that, at the moment, VLS can support only one output per stream, so you cannot play a stream on the network and write it into a file at the same time. The network output is highly configurable: you can choose which network interface you want to use, and specify source and destination IP addresses .

Manager

The *manager* controls the way streams are sent. Through an *administration interface*, you can tell the manager to start, stop, suspend, resume, forward or rewind the different programs. You can also get a list of all programs available in the Program Table. The manager gets this table from the VLS configuration file (`vls.cfg`), so it cannot be changed once VLS has been started. At the moment, you cannot ask the manager whether a given stream is being broadcasted, but you will get an error message if you try to stop a stream that was not broadcasted .

Administration interface

There are currently two ways to launch the streaming:

- you can use the *command line* to give arguments at startup;
- or you can use the *telnet interface* to start/stop/pause the streaming whenever you want.

When using the telnet interface, you must authenticate before typing any command, because any user may not be allowed to execute any command (this can be configured in the `vls.cfg` configuration file).

Chapter 4. Configuration

VLS reads its configuration from the `vls.cfg` configuration file, which is supposed to be located in the current directory or in `SYSCONF_DIR/videolan/vls` (where *SYSCONF_DIR* is `/usr/local/etc` if you built and installed VLS by hand, or is `/etc` if you installed the debian binary package).

To write a `vls.cfg` file, use the one supplied with VLS as a start-point.

General structure

VLS configuration file `vls.cfg` is divided into sections, and each section may contain several variables:

```
BEGIN "FirstSection"
  Variable1 = "value1"
  Variable2 = "value2"
  [...]
END

BEGIN "SecondSection"
  Variable1 = "value1"
  Variable3 = "value3"
  [...]
END



[...]
```

All section names, variable names and values are not case-sensitive. There can be empty sections and subsections. Comments must follow a # character. Some variables have a default value; it means that you can ommit to declare these variables, and then they will be given their default value.

Writing a `vls.cfg`

Here is an explanation of all the sections you can find in a `vls.cfg`:

Section "VLS"

This section contains application wide settings.

`LogFile = "name"`

Name of VLS log file. If left empty "", then no logging to files is done. Default is "vls.log".

`SystemLog = "[disable|enable]"`

Logging to the SystemLog. Today, only the SystemLog using syslogd is implemented: compile with **./configure --enable-syslog**.

Caution

If VLS is started as **vlsd**, then the following configuration is mandatory:

```
BEGIN "Vls"
  LogFile   = ""
  SystemLog = "enabled"
  ScreenLog = "disabled"
END
```

```
ScreenLog = "[disable|enable]"
```

Logging to the console.

Example:

```
BEGIN "Vls"
  LogFile   = "vls.log"
  SystemLog = "disable"
  ScreenLog = "enable"
END
```

Section "Groups"

In this section, you can define some groups of users, and which commands these users are allowed to execute. For each group you want to define, you must add a line in the following format:

```
groupname = "command1|command2|..."
```

This adds a group "groupname", the users of which are allowed to execute command1, command2, and so on. At the moment, the available commands are: help, browse, start, suspend, resume, forward, rewind, stop, shutdown, logout.

Example:

```
BEGIN "Groups"
  monitor = "help|browse|logout"
  master  = "help|browse|start|resume|suspend|forward|rewind|stop|shutdown|logout"
END
```

Section "Users"

This section contains a list of users allowed to control VLS through an administration interface. For each user, add a line in the following format:

```
username = "password:groupname"
```

This adds a user "username", who belongs to the group "groupname" (defined in the "Groups" section) and can log in with the password "password".

- Under Unix/Linux, the password must be encrypted, with a tool such as **mkpasswd**, or with the UNIX function "crypt".
- Under Windows, the password must be in clear text.

Example for Unix/Linux:

```
BEGIN "Users"
```

```
  monitor = "3BcKWoiQn0vi6:monitor"       # password is 'monitor'
  admin   = "42BKiCguFAL/c:master"        # password is 'Vir4Gv5S'
END
```

Section "Telnet"

In this section, you can configure the telnet administration interface.

`LocalPort = "port"`

Defines which port will be used for the telnet server. Default port is "9999".

`Domain = "domain"`

Either "inet4" or "inet6" (default is "inet4"). If you want to use IPv4 addresses, put "inet4", and if you want to use IPv6, put "inet6".

`LocalAddress = "IP address"`

Defines on which IP address the telnet server will listen for requests. Default address is "0.0.0.0" (or "0::0" with IPv6).

Example:

```
BEGIN "Telnet"
  LocalPort = "9999"
END
```

Section "NativeAdmin"

Same syntax as "Telnet". Not used yet.

Section "Inputs"

In this section, you can define which inputs you want to use. For each input you need, add a line in the following format:

`InputName = "Type"`

This adds a input named "InputName", the type of which is "Type". As explained before, there are several types of input:

- "local" to play a stream from a file or a DVD ,
- "video" to play a stream from an MPEG encoding card ,
- "dvb" to play a stream from a DVB card ,
- "v4l" to play a stream from a Video4Linux device.

Each input must be configured in its own section (see next paragraph).

Example:

```
BEGIN "Inputs"
  local1 = "local"
  pvr    = "video"
  dvb1   = "dvb"
  tuner  = "v4l"
```

```
END
```

Inputs configuration

For each input declared in the "Inputs" section, excepted "local" inputs, you must add a section with the same name as the corresponding input. For instance, if you declared an input "pvr", there should be one section named "pvr" too. The syntax of such sections depends on the type of the corresponding input.

To configure a local input, you don’t have to do anything. Except when another trickplay strategy must be used:

```
BEGIN "Local1"
  ProgramCount = "1"
  TrickPlay    = "normal"
END
```

"Local1" is the name of the local input you want to configure. "ProgramCount" is the number of programs assigned to this input. "TrickPlay" is the trickplay strategy that is used by this input (default is "normal").

To configure a video input, add a section in the following format:

```
BEGIN "VideoInputName"
  Device = "device"
  Type   = "type"
END
```

"VideoInputName" is the name of the video input you want to configure. "Device" is the path of the MPEG encoding card you want to read from (default is "/dev/video"). "Type" is either "Mpeg2-PS" or "Mpeg2-TS", depending on your device configuration (default is "Mpeg2-PS").

Example for a Hauppauge WinTV-PVR-250 card:

```
BEGIN "pvr"
  Device = "/dev/video0"
  Type   = "Mpeg2-PS"
END
```

To configure a dvb input, add a section in the following format:

```
BEGIN "DvbInputName"
  DeviceNumber = "devicenumber"
  SendMethod   = "0"
END
```

"DvbInputName" is the name of the dvb input you want to configure. Set "SendMethod" to "0" if you to stream the complete DVB stream and set it to "1" if you only want to stream the MPEG audio and video streams (default is "0"). "DeviceNumber" is the number of the DVB device you want to read from (read from `/dev/ost/dvr<devicenumber>`, default is ""). The dvb configuration file is defined by the driver. You can find it in `$HOME/.dvbrc` for /dev/dvb/adapter0 or in `$HOME/.dvbrc.X` for `/dev/dvb/adapterX` .

Example:

```
BEGIN "dvb1"
  DeviceNumber = "0"
  TrickPlay = "normal"
END
```

Section "Channels"

In this section, you can define the channels (outputs) you want to use. For each channel, write a line in the following format:

```
ChannelName = "Type"
```

This adds a channel named "ChannelName", the type of which is "Type". "Type" must be either "network" or "file". Like inputs, channels must be configured in their own section.

Example:

```
BEGIN "Channels"
  localhost  = "network"
  client1    = "network"
  client2    = "network"
  multicast1 = "network"
  multicast2 = "network"
  localfile  = "file"
END
```

Channels configuration

For each channel declared in the "Channels" section, you must add a section with the same name as the corresponding channel. The syntax of such a section depends on the type of the corresponding channel.

To configure a network channel, add a section in the following format:

```
BEGIN "NetChannelName"
  Domain    = "Domain"
  Type      = "Type"
  SrcHost   = "SourceHost"
  SrcPort   = "SourcePort"
  DstHost   = "DestHost"
  DstPort   = "DestPort"
  TTL       = "ttl"
  Interface = "Interface"
END
```

- "NetChannelName" is the name of the network channel you want to configure.
- "Domain" is either "inet4" if you use IPv4 addresses, or "inet6" if you use IPv6 (default is "inet4").
- "Type" is either "unicast", "broadcast" or "multicast" (default is "unicast"), depending on what you want to do (and on your "DstHost" address).
- "SourceHost" is the IP address (or DNS name) from which VLS will send the stream.
- "SourcePort" is the UDP port from which the stream will be sent.
- "DestHost" is the IP address (or DNS name) to which the stream will be sent.
- "DestPort" is the UDP port to which the stream will be sent (default is "1234").
- "TTL" is an option useful only if "Type" is "multicast" (default value is "0"). You can use it to increase the TTL of your multicast packets if they have to cross several routers.
- "Interface" is an option only supported under GNU/Linux, to force the stream to be sent through a given network interface ("eth1" for instance) To use this option, you must have super-user permissions.

Note: "SrcHost" and "SrcPort" are optional (if you don't set them, VLS will not 'bind' the socket).

To configure a file channel, add a section in the following format:

```
BEGIN "FileChannelName"
  FileName = "file"
  Append   = "append"
END
```

"FileChannelName" is the name of the file channel you want to configure. "file" is the name of the file where the stream will be stored (default is "fileout.ts"). "append" is either "yes" or "no", and indicates whether VLS will append the stream at the end of the file, or rewrite it.

Example:

```
BEGIN "localhost"         # The client is on the same host as the server
  DstHost = "localhost"
  DstPort = "1234"
END

BEGIN "client1"           # unicast towards client1
  DstHost = "192.168.1.2"
  DstPort = "1234"
END

BEGIN "client2"           # unicast towards client2 in IPv6
  Domain  = "inet6"
  DstHost = "3ffe:ffff::2:12:42"
  DstPort = "1234"
END

BEGIN "multicast1"         # multicast streaming
  Type    = "multicast"
  DstHost = "239.2.12.42"
  DstPort = "1234"
  TTL     = "2"
END

BEGIN "multicast2"         # multicast streaming in IPv6
  Domain  = "inet6"
  Type    = "multicast"
  DstHost = "ff08::1"
  DstPort = "1234"
  TTL     = "12"
END

BEGIN "localfile"         # file output
  FileName = "stream.ts"
  Append   = "no"
END
```

Caution

If you use Windows, you should specify the "SrcHost" and "SrcPort" fields. For example:

```
BEGIN "client1"         # The client is on the same host as the server
  SrcHost = "192.168.1.1"  # IP of VLS
  SrcPort = "1242"         # Source port : the value is not important
  DstHost = "192.168.1.2"  # IP of the client
  DstPort = "1234"
END
```

Programs Configuration

As explained before, you must define the programs. Each one is a MPEG stream (a file, for example). To do this, you must add an "Input" section in your `vls.cfg` file. Each "Input" section must have the following syntax:

```
BEGIN "Input"
  FilesPath    = "path"
  ProgramCount = "count"
END
```

"path" is the path where your MPEG files are located (by default it is the current directory). "count" is the number of programs defined ("0" by default).

For each program you want to define, you must add a section with the following format:

```
BEGIN "number"
  Name     = "name"
  Type     = "type"
  FileName = "file"
  Device   = "device"
END
```

- "number" is the program number: the first program has number 1, the second number 2, and so on.
- "name" is the program name, by which you will tell VLS to start this program (see next chapter "Running VLS").
- "type" can be "Mpeg1-PS", "Mpeg2-PS", "Mpeg2-TS", or "DVD". If your stream is stored in a MPEG file (*.mpeg, *.mpg, *.vob, and so on...), it is probably in Mpeg1-PS or Mpeg2-PS format.
- if "type" is set to "Mpeg1-PS", "Mpeg2-PS", or "Mpeg2-TS", VLS will assume your stream is stored in the file "file", in the directory "path" ("path" being the variable defined in the "Input" section).
- if "type" is "DVD", the variable "Device" will be used instead of "FileName" (the variable "FilesPath" is not prepended to the device name !). The variable "Device" is the device of your DVD drive ("/dev/hdc" or "/dev/cdrom" for instance). You can also play a DVD copied on a hard disk: then "Device" is the directory where the .vob files are stored ("/mnt/data/VIDEO_TS" for instance).

  Note: VLS can stream MPEG files that meet two critera

  - 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).

Note: In order to play DVDs, you need to compile VLS with DVD support, which uses libdvdread and libdvdcss. You will need read *and* write access rights to the DVD device.

Full example:

```
BEGIN "Input"
  FilesPath = "/home/videolan/streams"
  ProgramCount = "4"
END

BEGIN "1"     # MPEG2 stream stored in /home/videolan/streams/Dolby.vob
  Name     = "dolby"
  FileName = "Dolby.vob"
  Type     = "Mpeg2-PS"
END

BEGIN "2"     # another file
  Name     = "canyon"
  FileName = "Dolby_Canyon.vob"
  Type     = "Mpeg2-PS"
END

BEGIN "3"     # DVD
  Name     = "dvd"
  Device   = "/dev/cdrom"
  Type     = "Dvd"
END

BEGIN "4"     # DVD stored on a hard disk
  Name     = "matrix"
  Device   = "/mnt/data/matrix/VIDEO_TS"
  Type     = "Dvd"
END
```

Chapter 5. Running VLS

Launching VLS

If you want to use the telnet interface, running VLS is very easy: just type **vls** in a shell console, and that's all. Running **vlsd** will start VLS as a daemon and will detach itself from the launching shell. Remember that VLS will try to load its configuration file (`vls.cfg`) from the current directory, and if there is no `vls.cfg` there, it will try to load it from `SYSCONF_DIR/etc/videolan` (see section *Configuration*).

> **Caution**
>
> If your log file is `vls.log` as in the example, VLS will need write access in the current directory, or you will see something like:
>
> ```
> *** Exception *** in copy constructor (0xbffffc98, copy of 0x80e30a8)
> Unable to open the log file "vls.log": Error: Could not open file 'vls.log':
> Permission denied
> ```
>
> Remember also that you must be root when using the "Interface" option in `vls.cfg`.

If everything is right, you will see something like:

```
VideoLAN Server v 0.5.3 (Jun  6 2003) - (c)1999-2003 VideoLAN
2002-03-09 17:24:51 [INFO/Vls]  Module "channel:file" registered
2002-03-09 17:24:51 [INFO/Vls]  Module "channel:network" registered
2002-03-09 17:24:51 [INFO/Vls]  Module "mpegreader:file" registered
2002-03-09 17:24:51 [INFO/Vls]  Module "mpegconverter:ts2ts" registered
[...]
```

What you can see on the screen (stderr) is exactly what goes in the log file `vls.log`.

When VLS has been successfully started, it doesn't take any command from its standard input, so you can put it into background (you can use the *screen* utility to do that).

On the other hand, if you want to use the command line interface, please see the VideoLAN HOWTO (http://www.videolan.org/doc/).

Using the telnet interface

After VLS has been launched, it opens a telnet server (on the port 9999 by default). You can connect to this server with the following command:

```
% telnet localhost 9999
```

You should see something like:

```
Trying 127.0.0.1...
Connected to vls.
Escape character is '^]'.
Videolan Server Administration System
Login:
```

Then you must authenticate with a login/password pair defined in `vls.cfg`. When you have been successfully authenticated, you should see a prompt like:

```
admin@vls>
>
```

Then you can type some commands, which are explained in the next paragraph. To log out, type **logout** after the telnet prompt.

Interface commands

help

Usage: help [command].

Called with no argument, "help" gives the list of all the commands (available or not). Called with one argument it gives details about how to use the specified command.

browse

Usage: browse [input].

Called without argument, "browse" gives all programs of inputs. Called with one argument it only gives the programs of the specified input. Each program is given with its status.

start

Usage: start <program> <channel> <input> [--loop] [--rtp]

"start" launches the specified program of the specified input and broadcasts it through the specified channel. The option "--loop" makes the program being repeated indefinitely. The option "--rtp" makes the TS packet to be send through the RTP protocol, as defined in RFC 1889 and RFC 2250.

stop

Usage: stop <channel>

"stop" ends the broadcast of the specified channel.

forward

Usage: forward <channel> <speed>

"forward" forwards the channel with the given speed. This does not work when reading directly from a device such as an MPEG encoding card, a DVB card or an acquisition card.

rewind

Usage: rewind <channel> <speed>

"rewind" rewinds the channel with the given speed. This does not work when reading directly from a device such as an MPEG encoding card, a DVB card or an acquisition card.

suspend

Usage: suspend <channel>

"suspend" suspends the streaming of the specified channel.

resume

Usage: resume <channel>

"resume" resumes the streaming of the specified channel.

logout

Usage: logout

"logout" closes the current administration session and the remote connection.

shutdown

Usage: shutdown

"shutdown" stops all the programs and shutdowns VLS.

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.