

Guía de usuario de VLS

Cyril Deguet

Alexis de Lattre

Guía de usuario de VLS

by Cyril Deguet and Alexis de Lattre

Copyright © 2002-2004 el proyecto VideoLAN

Este documento es la guía de usuario completa de VLS .

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

Table of Contents

1. Introducción	1
¿Qué es el proyecto VideoLAN?	1
¿Qué es un codec?	3
¿Cómo se usa VideoLAN?	3
Utilización de la línea de comandos	4
2. Instalación de VLS.....	8
Instalación de VLS	8
Desinstalación de VLS.....	9
3. Visión general y conceptos básicos	10
Estructura de VLS	10
Interfaz de administración.....	11
4. Configuración.....	12
Estructura general	12
Construyendo un vls.cfg	12
5. Uso de VLS	21
Ejecutando VLS	21
Uso del interfaz telnet	21
Comandos de la interfaz.....	22
A. GNU General Public License	24
Preamble.....	24
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION	24
How to Apply These Terms to Your New Programs.....	28

Chapter 1. Introducción

¿Qué es el proyecto VideoLAN?

Descripción general

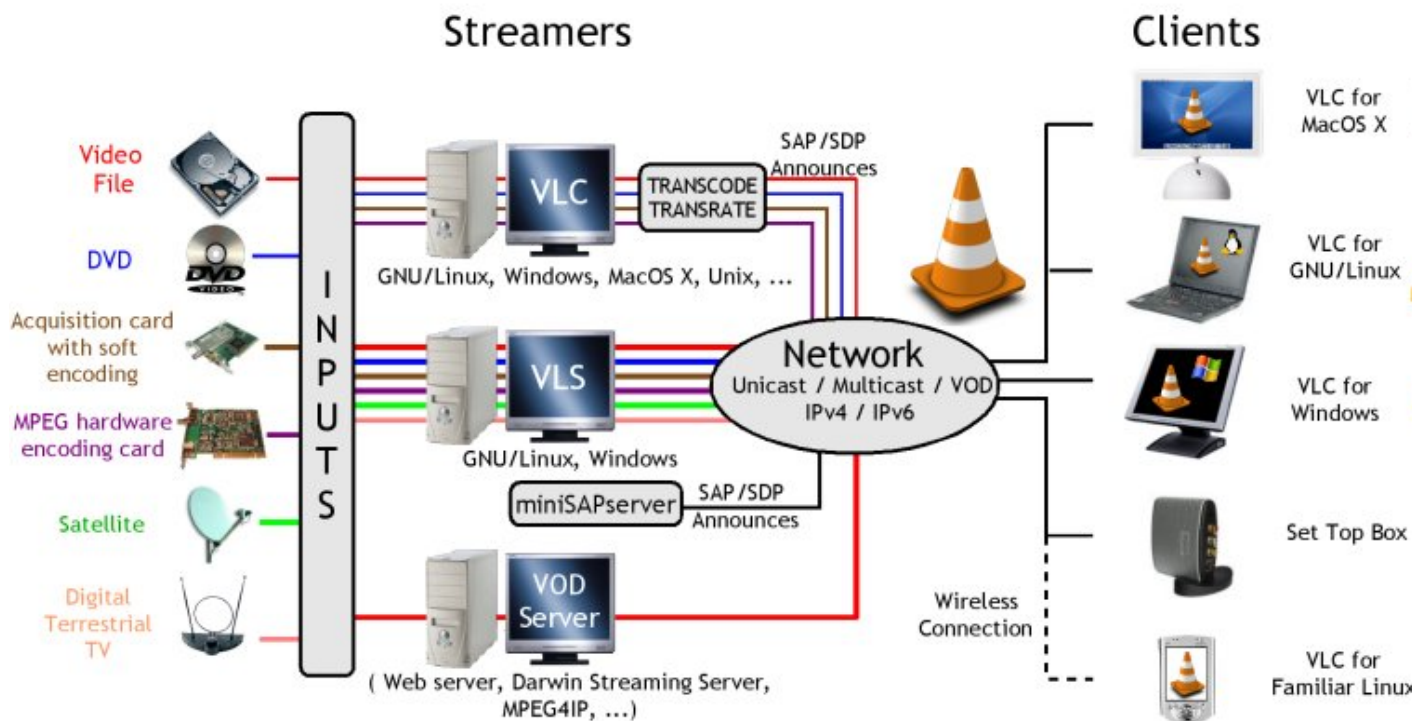
VideoLAN es una solución de software completa para transmisión de vídeo, desarrollada por estudiantes de Ecole Centrale Paris (<http://www.ecp.fr>) y desarrolladores de todo el mundo, dentro de GNU General Public License (<http://www.gnu.org/copyleft/gpl.html>) (GPL). VideoLAN está diseñado para transmitir vídeo MPEG en redes con gran capacidad de ancho de banda.

La solución VideoLAN incluye :

- VLS (Servidor VideoLAN), el cual puede transmitir archivos MPEG-1, MPEG-2 y MPEG-4, DVDs, canales digitales de satélite, canales digitales de televisión terrestre y vídeo en vivo sobre la red en unicast o multicast,
- VLC (inicialmente cliente VideoLAN), el cual puede ser usado como servidor para transmitir archivos MPEG-1, MPEG-2 y MPEG-4, DVDs y vídeo en vivo sobre la red en unicast o multicast; o usado como cliente para recibir, decodificar y visualizar flujos MPEG sobre varios sistemas operativos.

A continuación se muestra una ilustración de la solución VideoLAN completa :

Figure 1-1. La solución VideoLAN global



Más detalles sobre el proyecto se pueden encontrar en la web de VideoLAN <http://www.videolan.org/>.

Software VideoLAN

VLC

VLC trabaja sobre muchas plataformas : Linux, Windows, Mac OS X, BeOS, *BSD, Solaris, Familiar Linux, Yopy/Linupy y QNX. Puede leer :

- Archivos MPEG-1, MPEG-2 y MPEG-4 / DivX desde un disco duro, un CD-ROM, ...
- DVDs y VCDs,
- desde un tarjeta receptora de satélite (DVB-S),
- Flujos MPEG-1, MPEG-2 and MPEG-4 desde la red enviados por la salida de VLS o VLC's.

VLC también puede ser usado como servidor para transmitir :

- archivos MPEG-1, MPEG-2 y MPEG-4 / DivX ,
- DVDs,
- desde una tarjeta codificadora MPEG,

a :

- una máquina (p.e. a una dirección IP) : esto es lo que se llama *unicast*,
- un grupo dinámico de máquinas a las que el cliente puede conectarse o desconectarse (p.e. a una dirección IP multicast) : esto es lo que se llama *multicast*,

en IPv4 o IPv6 .

Para conseguir la lista completa de las funcionalidades de VLC en cada plataforma, vea la página de características de VLC <http://www.videolan.org/vlc/features.html>.

Note: VLC no trabaja en Mac OS 9, y probablemente no lo hará nunca.

VLS

VLS puede transmitir :

- un fichero MPEG-1, MPEG-2 o MPEG-4 almacenado en un disco duro, un CD, ...,
- un DVD insertado en un lector de DVD, o copiado en un disco duro,
- une carte satellite (DVB-S) ou une carte de télévision numérique terrestre (DVB-T) ,
- una tarjeta codificadora MPEG ;

a :

- una máquina (p.e. a una dirección IP) : esto es lo que se llama *unicast*,
- un grupo dinámico de máquinas a las que el cliente puede conectarse o desconectarse (p.e. a una dirección IP multicast) : esto es lo que se llama *multicast*,

en IPv4 o en IPv6 .

Un Pentium a 100 MHz con 32 MB de memoria debería ser suficiente para enviar un flujo a la red. Cuando se transmiten muchos vídeos almacenados en un disco duro, la limitación no es el procesador sino el disco duro y la conexión de red.

VLS trabaja bajo Linux y Windows. Para conseguir la lista completa de las funcionalidades de VLS en cada plataforma, vea la página de características de transmisión <http://www.videolan.org/streaming/features.html>.

Mini-SAP-server

Se puede añadir información de servicio a un canal basado en el estándar SAP/SDP para la solución VideoLAN. El mini-servidor-SAP envía anuncios acerca de los programas multicast en la red en IPv4 o IPv6, y VLC recibe estos anuncios y automáticamente añade los anuncios de programas a su lista de reproducción.

El mini-servidor-SAP trabaja bajo Linux y Mac OS X.

¿Qué es un codec?

Para entender bien la solución VideoLAN, se debe comprender la diferencia entre un *codec* y un *formato contenedor*

- Un *codec* es un algoritmo de compresión, utilizado para reducir el tamaño de un flujo. Existen codecs de audio y codecs de vídeo. MPEG-1, MPEG-2, MPEG-4, Vorbis, DivX, ... son codecs
- Un *formato contenedor* contiene uno o varios flujos ya codificados por codecs. A menudo, hay un flujo de audio y uno de vídeo. AVI, Ogg, MOV, ASF, ... son formatos contenedor. Los flujos que contengan pueden ser codificados utilizando diferentes codecs. En un mundo perfecto, se podría utilizar cualquier codec en cualquier formato contenedor. Desafortunadamente, existen algunas incompatibilidades. Se puede encontrar una tabla de posibles codecs y formatos contenedor en la página de funcionalidades (<http://www.videolan.org/streaming/features.html>)

Para decodificar un flujo, VLC primero lo *demultiplexa*. Esto significa que lee el formato contenedor y separa audio, vídeo, y, si los hay, subtítulos. Entonces, cada uno de estos flujos son enviados a sus *descodificadores* que realizan un procesamiento matemático para descomprimir los flujos .

Existe un caso particular en MPEG:

- MPEG es un *codec*. Existen varias versiones, llamadas MPEG-1, MPEG-2, MPEG-4, ...
- MPEG es también un formato contenedor, a veces se denomina como *MPEG Sistema*. Existen varios tipos de MPEG: ES, PS, and TS

Cuando se reproduce, por ejemplo, un vídeo MPEG de un DVD, el flujo MPEG está compuesto por varios flujos (llamados flujos elementales , ES): existe uno para el vídeo, uno para el audio, otro para subtítulos, y así sucesivamente. Estos flujos distintos se juntan para formar un único flujo de programa (PS). Así que, los ficheros .VOB que puedes encontrar en un DVD son realmente ficheros MPEG-PS. Pero este formato PS no está adaptado para la difusión de vídeo a través de una red o por satélite, por ejemplo. Así que, otro formato llamado Transport Stream (TS) fue diseñado para la difusión de vídeos MPEG a través de estos escenarios.

¿Cómo se usa VideoLAN?

Documentación

La documentación de usuario de VideoLAN se compone de 4 documentos :

- *VideoLAN HOWTO*. Este documento es la guía completa de la solución de streaming VideoLAN. Le mostrará ejemplos prácticos para la instalación de su solución de streaming.

- *VLC user guide*. Este documento es la guía completa para VLC.
- *VLS user guide*. Este documentos es la guía completa para VLS.
- *VideoLAN FAQ*. Este documento contiene preguntas frecuentes sobre VideoLAN.

La última versión de estos documentos se puede encontrar en la página de documentación <http://www.videolan.org/doc/>.

Soporte de usuario

Si tiene problemas usando VideoLAN, y no encuentra la respuesta a esos problemas en la documentación, por favor mire la documentación online de las listas de correo <http://www.via.ecp.fr/via/ml/videolan-en.html>. Hay dos listas de correo en inglés para los usuarios :

- vlc@videolan.org para las preguntas sobre VLC ,
- streaming@videolan.org para las preguntas sobre VLS, el mini-servidor-SAP y la red .

Si desea subscribirse o darse de baja en las listas de correo, por favor vaya a la página de listas de correo <http://www.videolan.org/support/lists.html>.

Puede hablar con usuarios y desarrolladores de VideoLAN en el IRC : servidor *irc.freenode.net*, canal *#videolan* .

Si encuentra un bug, por favor siga las instrucciones de la página de informe de bugs <http://www.videolan.org/support/bug-reporting.html> .

Utilización de la línea de comandos

- VLC tiene varias interfaces gráficas, que están organizadas de forma diferente con el fin de estar en armonía con las líneas generales de cada sistema operativo. Documentar el uso de cada interfaz gráfica es demasiado largo, y algunas características son sólo disponibles a través de la interfaz de línea de comandos. Por lo tanto decidimos documentar sólo la interfaz de línea de comandos, ¡pero en muchos casos debería ser fácil averiguar cómo usar la interfaz gráfica con el mismo propósito!
- VLS est doté d'une interface en ligne de commande, et d'une interface Telnet, mais n'a pas d'interface graphique !

Todos los comandos que se describen en este documento deben ser tecleados dentro de un terminal .

Abrir un terminal

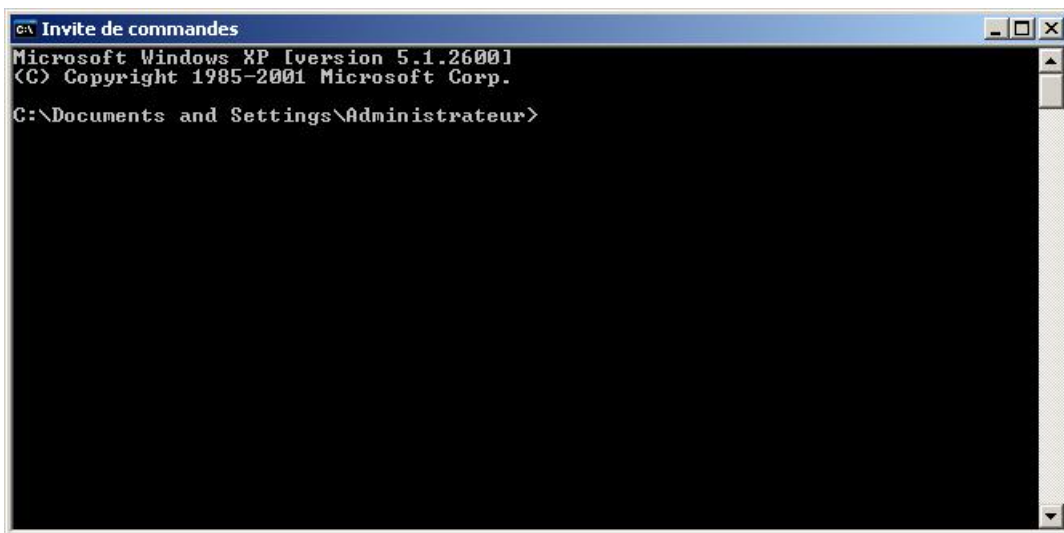
Windows

Pinchar en *Inicio*, *Ejecutar* y teclear :

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

En el terminal aparece

Figure 1-2. Windows terminal

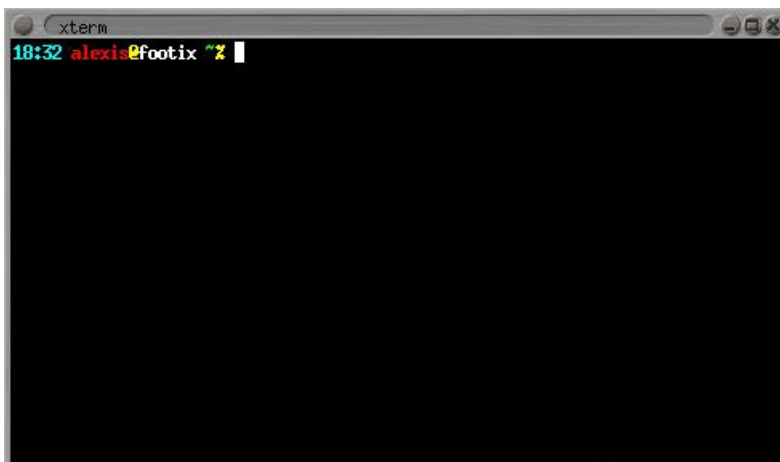


Note: Bajo Windows, se necesita estar en el directorio donde está instalado el programa para ejecutarlo.

Linux / Unix

Abrir un terminal :

Figure 1-3. Linux X terminal



En la documentación, adoptamos la siguiente convención para los comandos Unix :

- los comandos que deben ser tecleados como *root* tienen un # prompt :

comando_a_ser_tecleado_como_root
- los comandos que deben ser tecleados como un usuario normal tienen un % prompt :

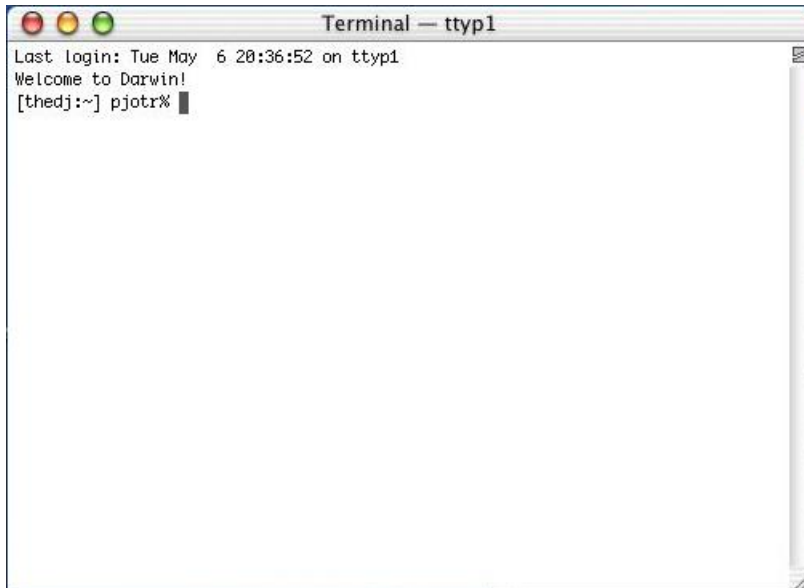
%

comandos_a_ser_tecleados_como_usuario_normal

Mac OS X

Ir a *Aplicaciones*, abrir la carpeta *Utilidades* y hacer doble-click en *Terminal* :

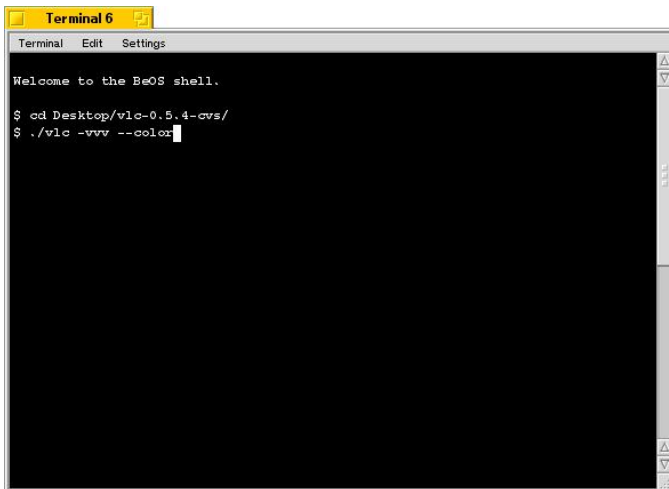
Figure 1-4. Mac OS X terminal



Note: Bajo Mac OS X, se necesita estar en el directorio donde el programa ha sido instalado para ejecutarlo, y empezar el comando con *./*.

BeOS

En la barra de herramientas, vaya a *Aplicaciones* y después a *Terminal* :

Figure 1-5. BeOS terminal

Note: Bajo BeOS, se necesita estar en el directorio donde el programa ha sido instalado para ejecutarlo, y empezar el comando con `./`.

Chapter 2. Instalación de VLS

Instalación de VLS

Windows

Descárgese el fichero ZIP de la página de decargas de VLS para Windows (<http://www.videolan.org/streaming/download-vls-windows.html>), descomprímalo y ejecute `setup.exe`.

GNULinux & Mac OS X

Instalar las librerías

Son necesarias varias librerías :

- *libdvbpsi* (obligatoria)
- *libdvdcss* para poder leer DVDs encriptados ,
- *libdvddread* para transmitir DVDs ,
- *libdvb* para poder transmitir desde una tarjeta DVB (satélite o digital terrestre) .

Descárguese las librerías de la página del código fuente de VLS (<http://www.videolan.org/streaming/download-vls-sources.html>) .

Para cada librería, descomprima, configure, (menos para *libdvb* que no tiene *./configure*), compile e instale :

```
% tar xvzf library.tar.gz
% cd library
% ./configure
% make
# make install
```

Verifique que el fichero de configuración `/etc/ld.so.conf` contiene la siguiente línea :

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

Si la línea no está presente, añádela y ejecute `starten Sie dann` :

```
# ldconfig
```

En Mac OS X, ejecute :

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

Instalar VLS

Descárguese el código fuente de la última versión : consiga el fichero `vls-version.tar.gz` de la página del código fuente de VLS (<http://www.videolan.org/streaming/download-vls-sources.html>). Descomprímalo y ejecute *./configure* :

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

Para conseguir la lista de opciones de configuración, haga folgendes :

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

Después configure VLS :

- para un VLS simple, sin soporte para DVD, haga :

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

- para un VLS con soporte para DVD, haga :

```
% ./configure
```

- para un VLS con soporte para DVB, haga :

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

Después, compile e instale :

```
% make
```

```
# make install
```

Puede ejecutar **make uninstall**, **make clean** o **make distclean** siempre que lo necesite .

Desinstalación de VLS

Windows

Vaya al *Panel de Control*, pulse en *Añadir o Eliminar programas*, seleccione *VLS* y pulse en *Modificar/Eliminar* y siga los pasos del programa de desinstalación .

Si ha compilado VLS de las fuentes

Vaya al directorio que contiene las fuentes de VLS y ejecute :

```
# make uninstall
```

Podrá borrar después las fuentes de VLS .

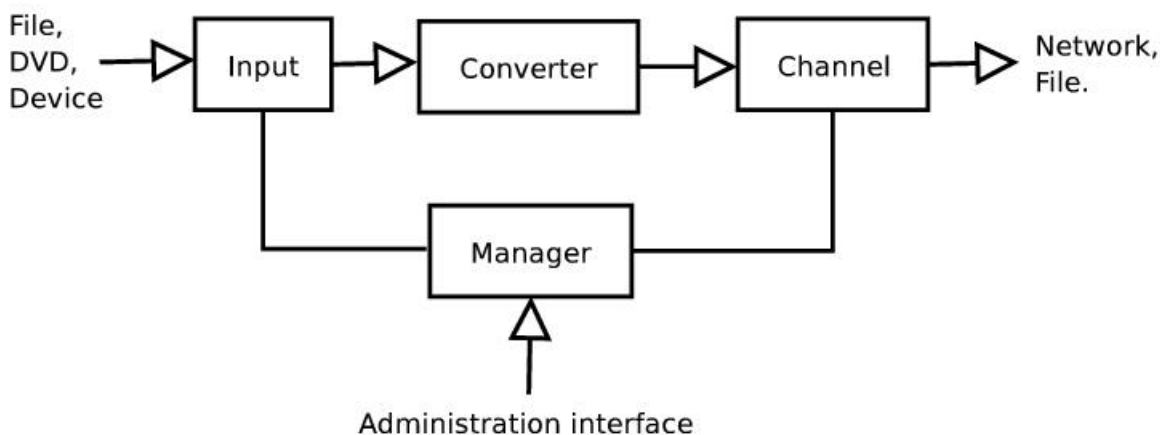
Chapter 3. Visión general y conceptos básicos

Estructura de VLS

Desde el punto de vista de un usuario, VLS puede ser dividido en cuatro tipos de componentes:

- un *gestor* ,
- *entradas* ,
- *convertidores* ,
- y *salidas* .

Figure 3-1. Estructura de VLS



Entradas

El papel de una *entrada* es el de leer flujos MPEG de una fuente dada (archivo, DVD, tarjetas DVB, dispositivos, ...), y alimentar a los *convertidores* correctos con estos flujos. Una entrada puede leer varios flujos, que son llamados *programas*. Hay varios tipos de entradas:

- la entrada *local*, que puede leer de archivos o DVDs,
- la entrada *video*, que puede leer de tarjetas codificadoras MPEG,
- la entrada *dvb*, que puede leer de tarjetas DVB,
- la entrada *v4l*, que puede leer de tarjetas capturadoras soportadas por los drivers Video4Linux.

Se pueden usar varias entradas y tener varios programas al mismo tiempo.

Los convertidores

El papel de un *convertidor* es el de recibir un flujo de una entrada, y convertirlo en formato MPEG-TS. VLS es capaz de convertir flujos PS (de DVDs, por ejemplo) a flujos TS (convertidor ps2ts). Por supuesto, también puede leer flujos TS, y repararlos manejando las discontinuidades en el flujo (convertidor ts2ts).

Salidas

Una salida recibe un flujo de un conversor, y lo manda a un destino dado (red, fichero, ...). Actualmente existen dos tipos de salidas soportadas: *network* y *file*. Nótese que, por el momento, VLS sólo soporta una salida por flujo, por lo que no se puede enviar al mismo tiempo a una red y escribir a un fichero. La salida de red es bastante configurable: se puede elegir qué interfaz de red se quiere utilizar, y especificar las direcciones IP de origen y destino.

El gestor

El *gestor* controla el modo en el que son enviados los flujos. A través de una *interfaz de administración*, se le puede decir al gestor que comience, pare, suspenda, o reinicie los diferentes programas. Se puede también conseguir la lista de todos los programas disponibles en la Tabla de Programas. El gestor consigue esta tabla del fichero de configuración de VLS (`vls.cfg`), por lo que no puede ser modificado una vez que VLS es arrancado. Por el momento, no se puede preguntar al gestor si un flujo está siendo enviado, pero devuelve un mensaje de error si se intenta parar un flujo que no está siendo transmitido.

Interfaz de administración

Actualmente existen dos modos de controlar VLS:

- se puede usar la *línea de comandos* para proporcionarle argumentos en el arranque;
- o se puede usar la *interfaz Telnet* para arrancar/parar/pausar el envío siempre que se quiera.

Cuando se usa el interfaz Telnet, hay que autenticarse antes de escribir comandos, para prevenir que usuarios no deseados sean libres de hacerlo (esto se debe configurar en el fichero de configuración `vls.cfg`).

Chapter 4. Configuración

VLS carga su configuración del fichero de configuración `vls.cfg`, que se supone que se encuentra en el directorio actual o en `SYSCONF_DIR/videolan/vls` (donde `SYSCONF_DIR` es `/usr/local/etc` si instalastes VLS a mano, o es `/etc` si lo instalastes como un paquete debian).

Para escribir un fichero `vls.cfg`, usa como punto de partida el proporcionado con VLS.

Estructura general

El fichero de configuración `vls.cfg` está dividido en secciones, y cada sección puede contener varias variables:

```
BEGIN "Seccion1"
    Variable1 = "valor1"
    Variable2 = "valor2"
    [...]
END

BEGIN "Seccion2"
    Variable1 = "valor1"
    Variable3 = "valor3"
    [...]
END

[...]
```

Todos los nombres de sección, nombres de variable y valores no son case-sensitive. Puede haber secciones y subsecciones vacías. Los comentarios deben comenzar con un carácter `#`. Algunas variables tienen un valor por defecto; por lo que se puede omitir declararlas, y tendrán su valor por defecto.

Construyendo un `vls.cfg`

A continuación se explican todas las secciones que se pueden encontrar en el fichero `vls.cfg`:

Section "VLS"

Esta sección contiene los parámetros globales.

`LogFile = "name"`

Nombre del fichero de logs de VLS. Si se deja vacío "", no se crea fichero de logs. Por defecto es `"vls.log"`.

`SystemLog = "[disable|enable]"`

Generando Logs en el SystemLog. De momento, sólo está implementado el SystemLog usando syslogd: compile con la opción `./configure --enable-syslog`.

Caution

Si VLS es ejecutado como **vlsd**, entonces la siguiente configuración es obligatoria:

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

ScreenLog = "[disable|enable]"

Generando mensajes de log en la consola.

Ejemplo:

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

Section "Groups"

En esta sección, se pueden definir los grupos de usuarios, y qué comandos son permitidos para estos usuarios. Para cada grupo que se quiera definir, se debe añadir una línea en el siguiente formato:

groupname = "command1|command2|..."

Esto añade un grupo "groupname", y sus usuarios tienen permitido ejecutar el comando1, comando2, y así sucesivamente. Por el momento, los comandos disponibles son: help, browse, start, suspend, resume, forward, rewind, stop, shutdown, logout.

Ejemplo:

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

Section "Users"

Esta sección contiene una lista de los usuarios con permisos para controlar VLS a través de una interfaz de administración. Por cada usuario, se debe añadir una línea con el siguiente formato:

username = "password:groupname"

Esto añade un usuario "username", que pertenece al grupo "groupname" (definido en la sección "Groups") y que puede autenticarse con la contraseña "password".

- Bajo Unix/Linux, la contraseña debe estar encriptada, con una herramienta como **mkpasswd**, o con la función UNIX "crypt".
- Bajo Windows, la contraseña debe estar en limpio.

Ejemplo para Unix/Linux:

```
BEGIN "Users"
    monitor = "3BcKWoiQn0vi6:monitor"      # password es 'monitor'
    admin   = "42BKicGuFAL/c:master"       # password es 'Vir4Gv5S'
END
```

Section "Telnet"

En esta sección, se puede configurar la interfaz de administración Telnet.

```
LocalPort = "port"
```

Define qué puerto será utilizado por el servidor Telnet. El puerto por defecto es "9999".

```
Domain = "domain"
```

Se puede utilizar "inet4" o "inet6" (por defecto "inet4"). Si se quiere utilizar direcciones IPv4, utilice "inet4", y si se quiere IPv6, utilice "inet6".

```
LocalAddress = "IP address"
```

Define en qué dirección IP estará escuchando el servidor Telnet para aceptar peticiones. Por defecto es "0.0.0.0" (ó "0::0" con IPv6).

Ejemplo:

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

Section "NativeAdmin"

Usa la misma sintaxis que "Telnet". No se utiliza todavía.

Section "Inputs"

En esta sección, se puede definir qué entradas se quieren usar. Por cada entrada que se necesite, se debe añadir una línea en el siguiente formato:

```
InputName = "Type"
```

Esto añade una entrada con el nombre "InputName", del tipo "Type". Como se explicó anteriormente, existen varios tipos de entradas:

- "local" para leer de un fichero o un DVD,
- "video" para leer de una tarjeta capturadora MPEG,
- "dVB" para leer de una tarjeta DVB,
- "v4l" para leer de un dispositivo Video4Linux.

Cada entrada debe estar configurada en su propia sección (ver el siguiente párrafo).

Ejemplo:

```
BEGIN "Inputs"
```

```

local1 = "local"
pvr     = "video"
dvb1    = "dvb"
tuner   = "v4l"
END

```

Configuración de entradas

Para cada entrada declarada en la sección "Inputs", excepto para las entradas "local", se debe añadir una sección con el mismo nombre que la entrada correspondiente. Por ejemplo, si se declara una entrada "pvr", debe haber también una sección llamada "pvr". La sintaxis de cada sección depende del tipo de la entrada correspondiente.

Para configurar una entrada local, no se debe hacer nada. Excepto cuando otra estrategia trickplay deba ser usada:

```

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

```

"Local1" es el nombre de la entrada local que se quiere configurar. "ProgramCount" es el número de programas asignados a esta entrada. "TrickPlay" es la estrategia trickplay que se usa en esta entrada (por defecto "normal").

Para configurar una entrada video, se debe añadir una entrada en el siguiente formato:

```

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

```

"VideoInputName" es el nombre de la entrada de vídeo que se quiere configurar. "Device" es la ruta de la tarjeta codificadora MPEG de la que se quiere leer (por defecto es "/dev/video0"). "Type" puede ser "Mpeg2-PS" o "Mpeg2-TS", dependiendo de la configuración de la tarjeta (por defecto es "Mpeg2-PS").

Ejemplo para una tarjeta Hauppauge WinTV-PVR-250:

```

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

```

Para configurar una entrada dvb, se debe añadir una sección en el siguiente formato:

```

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

```

"DvbInputName" es el nombre de la entrada dvb que se quiere configurar. Se debe poner "SendMethod" a "0" si se quiere transmitir el flujo DVB completo o se debe poner a "1" si sólo se quieren transmitir los flujos MPEG de audio y vídeo (por defecto es "0"). "DeviceNumber" es el número de la tarjeta DVB de la que se quiere leer (lee de /dev/ost/dvr<devicenumber>, por defecto es ""). El fichero de configuración dvb es definido por el driver. Se puede encontrar en \$HOME/.dvbrc para /dev/dvb/adapater0 ó en \$HOME/.dvbrc.X para /dev/dvb/adapaterX.

Ejemplo:

```

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

```

END

Section "Channels"

En esta sección, se pueden definir las salidas que se quieren utilizar. Para cada salida, se debe escribir una línea en el siguiente formato:

```
ChannelName = "Type"
```

Esto añade una salida llamada "ChannelName", del tipo "Type". "Type" debe ser "network" o "file". Como las entradas, las salidas deben ser configuradas en su propia sección.

Ejemplo:

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

Configuración de las salidas

Para cada salida declarada en la sección "Channels", se debe añadir una sección con el mismo nombre que su salida correspondiente. La sintaxis de cada sección depende del tipo de salida.

Para configurar una salida de red, se debe añadir una sección en el siguiente formato:

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

- "NetChannelName" es el nombre de la salida de red que se quiere configurar.
- "Domain" puede ser "inet4" si se usan direcciones IPv4, o "inet6" si se usa IPv6 (por defecto es "inet4") .
- "Type" puede ser "unicast", "broadcast" o "multicast" (por defecto es "unicast"), dependiendo de lo que se quiera hacer (en la dirección "DstHost").
- "SourceHost" es la dirección IP (o nombre DNS) desde la cual VLS enviará el flujo.
- "SourcePort" es el puerto UDP desde el que se enviará el flujo.
- "DestHost" es la dirección IP (o nombre DNS) a la cual se enviará el flujo.
- "DestPort" es el puerto UDP al que se enviará el flujo (por defecto es "1234").

- "TTL" es una opción válida solamente si "Type" es "multicast" (el valor por defecto es "0"). Se puede usar para incrementar el TTL de los paquetes multicast si estos tienen que cruzar varios routers.
- "Interface" es una opción sólo soportada bajo GNU/Linux, para forzar que el flujo sea enviado a través de una interfaz de red dada ("eth1" por ejemplo). Para usar esta opción, se deben tener permisos de superusuario.

Note: "SrcHost" y "SrcPort" son opcionales.

Para configurar una salida a fichero, se debe añadir una sección en el siguiente formato:

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

"FileChannelName" es el nombre de la salida a fichero que se quiere configurar. "file" es el nombre del fichero donde se guardará el flujo (por defecto "fileout.ts"). "append" puede ser "yes" o "no", e indica si VLS tiene que añadir el flujo al final del fichero file, o sobrescribirlo.

Ejemplo:

```
BEGIN "localhost"          # El cliente está en la misma máquina que el servidor
  DstHost = "localhost"
  DstPort = "1234"
END
```

```
BEGIN "client1"           # unicast hacia client1
  DstHost = "192.168.1.2"
  DstPort = "1234"
END
```

```
BEGIN "client2"           # unicast hacia client2 en 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 en IPv6
  Domain  = "inet6"
  Type    = "multicast"
  DstHost = "ff08::1"
  DstPort = "1234"
  TTL     = "12"
END
```

```
BEGIN "localfile"         # volcado a fichero
  FileName = "stream.ts"
  Append   = "no"
END
```

Caution

Si se usa, se deberían especificar los campos "SrcHost" y "SrcPort". Por ejemplo:

```
BEGIN "client1"          # El cliente está en la misma máquina que el servidor
  SrcHost = "192.168.1.1" # IP de VLS
  SrcPort = "1242"        # Puerto Origen : el valor no es importante
  DstHost = "192.168.1.2" # IP del cliente
  DstPort = "1234"
END
```

Configuración de los Programas

Como se ha explicado anteriormente, se deben definir programas. Cada uno es un flujo MPEG (un fichero, por ejemplo). Para hacer esto, se debe añadir una sección "Input" en el fichero `vls.cfg`. Cada sección "Input" debe tener la siguiente sintaxis:

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

"path" es la ruta donde se encuentran los ficheros MPEG (por defecto es el directorio actual). "count" es el número de programas definidos ("0" por defecto).

Para cada programa que se quiera definir, se debe añadir una sección con el siguiente formato:

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

- "number" es el número de programa: el primer programa tiene el número 1, el segundo 2, y así sucesivamente.
- "name" es el nombre del programa, por el cual se le dirá a VLS que arranque dicho programa (ver el siguiente capítulo "Ejecutando VLS").
- "type" puede ser "Mpeg1-PS", "Mpeg2-PS", "Mpeg2-TS", o "DVD". Si el flujo es guardado en un fichero MPEG (*.mpeg, *.mpg, *.vob, y demás...), es probable que se encuentre en formato Mpeg1-PS o Mpeg2-PS.
- si "type" se fija a "Mpeg1-PS", "Mpeg2-PS", o "Mpeg2-TS", VLS asumirá que el flujo es guardado en el fichero "file", en el directorio "path" (siendo "path" la variable definida en la sección "Input").
- si "type" es "DVD", la variable "Device" será utilizada en lugar de "FileName" (la variable "FilesPath" no se refiere al nombre del lector !). La variable "Device" es el lector DVD ("/dev/hdc" o "/dev/cdrom" por ejemplo). Se puede leer también de un DVD copiado en el disco duro: entonces "Device" es el directorio donde se han guardado los ficheros .vob ("/mnt/data/VIDEO_TS" por ejemplo).

Note: VLS puede transmitir ficheros MPEG que sigan dos criterios

- el fichero debe ser MPEG PS (Program Stream) o MPEG TS (Transport Stream), el cual contiene vídeo y audio multiplexado. VLS no puede transmitir MPEG ES (Elementary Stream), es decir, un fichero sólo con audio o con vídeo.

Para saber si un fichero MPEG es MPEG PS, MPEG TS o MPEG ES, hay que leer el fichero con VLC y mirar los mensajes (en la ventana de mensajes o usar el comando **vlc -vvv**) .

- Si usted ve la siguiente línea :

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

significa que el fichero es MPEG TS .

- Si usted ve la siguiente línea :

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

significa que el fichero es MPEG PS .

- Si usted ve la siguiente línea :

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

significa que el fichero es MPEG ES, y VLS no puede transmitirlo .

- la cabecera de la secuencia de vídeo debe repetirse regularmente lo cual es frecuente en el caso de MPEG-2, pero muy raramente en MPEG-1. No hay una manera fácil de saber si la cabecera de la secuencia se repite regularmente. Los ficheros con extensión *.vob* son normalmente ficheros MPEG-2 y los ficheros con extensión *.mpeg* o *.mpeg* son normalmente ficheros MPEG-1. .

Puede descargarse este fichero transmisible en MPEG-2 PS para sus pruebas: *presentation_short.vob* (ftp://ftp.videolan.org/pub/videolan/streams/presentation/presentation_short.vob) .

Note: Para leer DVDs, se necesita compilar VLS con soporte DVD, que usa *libdvdread* y *libdvdcss*. Se necesitan permisos de lectura y escritura en el lector de DVD.

Ejemplo completo:

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

BEGIN "1"      # Flujo MPEG2 guardado en /home/videolan/streams/Dolby.vob
  Name       = "dolby"
  FileName   = "Dolby.vob"
  Type       = "Mpeg2-PS"
END

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

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

BEGIN "4"      # DVD guardado en el disco duro
  Name       = "matrix"
  Device     = "/mnt/data/matrix/VIDEO_TS"
```

```
    Type      = "Dvd"  
END
```

Chapter 5. Uso de VLS

Ejecutando VLS

Si se quiere usar el interfaz telnet, ejecutar VLS es muy sencillo: simplemente escriba **vls** en una consola de comandos, y eso es todo. Ejecutando **vlsd** hará que VLS ejecute como un demonio y será independiente de la consola. Recuerda que VLS intentará cargar el fichero de configuración (**vls.cfg**) desde el directorio actual, y si no hay un fichero **vls.cfg** en él, intentará cargarlo desde **SYSCONF_DIR/etc/videolan** (vea la sección *Configuración*).

Caution

Si su fichero de logs es **vls.log**, como en el ejemplo, VLS requerirá permisos de escritura en el directorio actual, o verá algo como esto:

```
*** 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
```

Recuerde también que debe ser root cuando use la opción "Interface" en el fichero **vls.cfg**.

Si todo va bien, verá algo como esto:

```
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
[...]
```

Lo que se ve en la pantalla (stderr) es exactamente lo que se guardará en el fichero de logs **vls.log**.

Cuando VLS se ha ejecutado correctamente, no recoge ningún comando de su entrada estándar, por lo que se puede poner en segundo plano (background) (se puede usar la utilidad **screen** para ello).

Si se desea usar la interfaz en línea de comandos, por favor, refiérase a VideoLAN HOWTO (<http://www.videolan.org/doc/>).

Uso del interfaz telnet

Una vez que VLS se ha ejecutado, abre un servidor telnet (en el puerto 9999 por defecto). Se puede conectar a este servidor con el siguiente comando:

```
% telnet localhost 9999
```

Se debería ver algo como esto:

```
Trying 127.0.0.1...
Connected to vls.
Escape character is '^['.
```

```
VideoLAN Server Administration System
```

```
Login:
```

Debe autenticarse con un usuario/contraseña definidos en el fichero `vls.cfg`. Cuando se haya autenticado correctamente, debería verse un 'prompt' como este:

```
admin@vls>
>
```

Se puede escribir los comandos, que son explicados en el siguiente párrafo. Para salir, escriba **logout** después del prompt del telnet.

Comandos de la interfaz

help

Uso: help [comando].

Sin argumentos, "help" proporciona la lista de todos los comandos (disponibles o no). Con un argumentos, proporciona detalles acerca de cómo utilizar un comando específico.

browse

Uso: browse [entrada].

Sin argumentos, "browse" proporciona todos los programas de todas las entradas. Con un argumento, proporciona los programas de la entrada seleccionada. Cada programa se muestra con su estado.

start

Uso: start <programa> <salida> <entrada> [--loop] [--rtp]

"start" ejecuta el programa seleccionado de la entrada seleccionada y lo transmite por la salida especificada. La opción "--loop" hace que el programa se repita indefinidamente. La opción "--rtp" hace que el paquete TS sea enviado por el protocolo RTP, como se define en RFC 1889 y RFC 2250.

stop

Uso: stop <channel>

"stop" finaliza el envío del canal especificado.

forward

Uso: forward <channel> <speed>

"forward" produce el avance rápido del vídeo a la velocidad dada. No funciona cuando se lee directamente de un dispositivo como una tarjeta codificadora MPEG, una tarjeta DVB o una tarjeta capturadora.

rewind

Uso: `rewind <channel> <speed>`

"rewind" produce el retencido rápido del vídeo a la velocidad dada. No funciona cuando se lee directamente de un dispositivo como una tarjeta codificadora MPEG, una tarjeta DVB o una tarjeta capturadora.

suspend

Uso: `suspend <channel>`

"suspend" detiene la transmisión de la salida especificada.

resume

Uso: `resume <channel>`

"resume" continua la transmisión de la salida especificada.

logout

Uso: `logout`

"logout" cierra la actual sesión de administración y la conexión remota.

shutdown

Uso: `shutdown`

"shutdown" detiene todos los programas y termina la ejecución de 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.