

# **MPlayer – The Movie Player for LINUX**

# Table of Contents

<b><u>MPlayer – The Movie Player for LINUX</u></b> .....	<b>1</b>
<u><a href="http://www.mplayerhq.hu">http://www.mplayerhq.hu</a></u> .....	1
<u>How to read this documentation</u> .....	5
<u>Chapter 1. Introduction</u> .....	5
<u>Important</u> .....	6
<u>1.1. History</u> .....	7
<u>1.2. Installation</u> .....	8
<u>1.2.1. Software requirements</u> .....	9
<u>1.2.2. Codecs</u> .....	9
<u>Note</u> .....	10
<u>1.2.3. Video cards</u> .....	10
<u>1.2.4. Sound cards</u> .....	12
<u>1.2.5. Features</u> .....	12
<u>1.3. What about the GUI?</u> .....	13
<u>1.4. Subtitles and OSD</u> .....	13
<u>1.4.1. MPlayer's own subtitle format (MPsub)</u> .....	14
<u>1.4.2. Installing OSD and subtitles</u> .....	15
<u>1.4.3. OSD menu</u> .....	16
<u>Note</u> .....	16
<u>1.5. RTC</u> .....	16
<u>Note</u> .....	17
<u>Chapter 2. Features</u> .....	17
<u>2.1. Supported formats</u> .....	19
<u>2.1.1. Video formats</u> .....	19
<u>Note</u> .....	20
<u>Note</u> .....	21
<u>2.1.2. Audio formats</u> .....	23
<u>2.2. Supported codecs</u> .....	24
<u>2.2.1. Video codecs</u> .....	24
<u>Note</u> .....	25
<u>Note</u> .....	26
<u>Note</u> .....	26
<u>Note</u> .....	27
<u>Note</u> .....	29
<u>2.2.2. Audio codecs</u> .....	29
<u>2.2.3. Win32 codecs importing HOWTO</u> .....	31
<u>Note</u> .....	31
<u>Note</u> .....	32
<u>Note</u> .....	32
<u>2.3. Output devices</u> .....	32
<u>2.3.1. Video output devices</u> .....	32
<u>Note</u> .....	34
<u>Note</u> .....	35
<u>Security risk</u> .....	36
<u>Note</u> .....	39
<u>Note</u> .....	40
<u>Note</u> .....	41
<u>Warning</u> .....	41

# Table of Contents

## MPlayer – The Movie Player for LINUX

<u>Note</u> .....	43
<u>Important</u> .....	44
<u>Note</u> .....	48
<u>Important</u> .....	54
<u>2.3.2. Audio output devices</u> .....	56
<u>Note</u> .....	65
<u>Note</u> .....	66
<u>2.4. TV input</u> .....	67
<u>2.4.1. Compilation</u> .....	68
<u>2.4.2. Usage tips</u> .....	68
<u>2.4.3. Examples</u> .....	69
<u>2.5. Edit Decision Lists (EDL)</u> .....	69
<u>2.5.1. Using an EDL file</u> .....	70
<u>2.5.2. Making an EDL file</u> .....	70
<u>Chapter 3. Usage</u> .....	70
<u>3.1. Command line</u> .....	70
<u>3.2. Control</u> .....	71
<u>3.2.1. Controls configuration</u> .....	72
<u>3.2.2. Control from LIRC</u> .....	75
<u>3.2.3. Slave mode</u> .....	75
<u>3.3. Streaming from network or pipes</u> .....	75
<u>Note</u> .....	76
<u>3.4. Remote streams</u> .....	76
<u>3.4.1. Compiling the server</u> .....	76
<u>3.4.2. Using remote streams</u> .....	76
<u>Chapter 4. Frequently Asked Questions</u> .....	76
<u>4.1. Development</u> .....	79
<u>4.2. Compilation</u> .....	80
<u>4.3. General questions</u> .....	82
<u>4.4. Playback problems</u> .....	84
<u>4.5. Video/audio driver problems (vo/ao)</u> .....	85
<u>Warning</u> .....	86
<u>4.6. DVD playback</u> .....	86
<u>4.7. Feature requests</u> .....	87
<u>4.8. Encoding</u> .....	87
<u>Chapter 5. CD/DVD usage</u> .....	88
<u>5.1. CD/DVD drives</u> .....	88
<u>5.2. DVD playback</u> .....	89
<u>Note</u> .....	89
<u>5.3. VCD playback</u> .....	91
<u>Note</u> .....	91
<u>Chapter 6. Ports</u> .....	91
<u>6.1. Linux</u> .....	92
<u>6.1.1. Debian packaging</u> .....	92
<u>6.1.2. RPM packaging</u> .....	92
<u>6.1.3. ARM</u> .....	92
<u>6.2. *BSD</u> .....	93

# Table of Contents

## MPlayer – The Movie Player for LINUX

6.2.1. FreeBSD.....	93
6.2.2. OpenBSD.....	93
6.2.3. Darwin.....	93
6.3. Sun Solaris.....	93
6.4. Silicon Graphics Irix.....	94
6.5. QNX.....	95
6.6. Windows.....	95
6.6.1. Cygwin.....	96
6.6.2. MinGW.....	96
6.7. Mac OS.....	96
6.8. HP UX.....	97
6.9. Amiga/MorphOS (GeekGadgets).....	97
Chapter 7. Encoding with MEncoder.....	97
7.1. Encoding 2 or 3-pass MPEG-4 ("DivX").....	98
7.2. Encoding to MPEG format.....	99
7.3. Rescaling movies.....	99
7.4. Stream copying.....	99
7.5. Fixing AVIs with broken index or interleaving.....	99
7.5.1. Appending multiple AVI files.....	100
Note.....	100
7.6. Encoding with the libavcodec codec family.....	100
7.7. Encoding from multiple input image files (JPEGs,PNGs or TGAs).....	101
Note.....	101
7.8. Extracting DVD subtitles to Vobsub file.....	101
7.9. Preserving aspect ratio.....	102
7.10. Custom inter/intra matrices.....	102
7.11. Making a high quality MPEG4 ("DivX") rip of a DVD movie.....	103
7.11.1. Cropping.....	103
7.11.2. Quality level.....	104
7.11.3. Files over 2GB.....	104
7.11.4. Deinterlacing.....	104
7.11.5. Inverse telecine.....	104
7.11.6. Scaling and aspect ratio.....	105
7.11.7. Summing it up.....	105
Appendix A. Mailing lists.....	105
Note.....	106
Appendix B. How to report bugs.....	106
B.1. How to fix bugs.....	107
B.2. How to report bugs.....	107
B.3. Where to report bugs.....	107
B.4. What to report.....	108
B.4.1. System Information.....	108
B.4.2. Hardware and drivers.....	108
B.4.3. Configure problems.....	109
B.4.4. Compilation problems.....	109
B.4.5. Playback problems.....	109
B.4.6. Crashes.....	110

# Table of Contents

## MPlayer – The Movie Player for LINUX

<u>B.5. I know what I am doing.....</u>	110
<u>Appendix C. Known bugs.....</u>	111
<u>C.1. Special system/CPU-specific bugs/problems.....</u>	111
<u>C.2. Various A–V sync and other audio problems.....</u>	111
<u>C.2.1. General audio delay or jerky sound (exists with all or many files).....</u>	112
<u>C.2.2. Audio delay/de–sync specific to one or a few files.....</u>	112
<u>C.2.3. No sound at all.....</u>	112
<u>C.2.4. No picture at all (just plain grey/green window).....</u>	113
<u>C.2.5. Video–out problems.....</u>	113
<u>Appendix D. MPlayer skin format.....</u>	113
<u>D.1. Overview.....</u>	114
<u>D.1.1. Directories.....</u>	114
<u>D.1.2. Image formats.....</u>	114
<u>D.1.3. Skin components.....</u>	114
<u>Note.....</u>	115
<u>D.1.4. Files.....</u>	115
<u>D.2. The skin file.....</u>	115
<u>D.2.1. Main window and playbar.....</u>	117
<u>Note.....</u>	117
<u>Warning.....</u>	117
<u>Note.....</u>	118
<u>Note.....</u>	119
<u>D.2.2. Subwindow.....</u>	119
<u>D.2.3. Skin menu.....</u>	120
<u>D.3. Fonts.....</u>	120
<u>D.3.1. Symbols.....</u>	121
<u>D.4. GUI messages.....</u>	121
<u>Note.....</u>	121
<u>Appendix E. Developer cries.....</u>	123
<u>E.1. GCC 2.96.....</u>	123
<u>E.2. Binary distribution.....</u>	125
<u>E.3. nVidia.....</u>	125
<u>E.4. Joe Barr.....</u>	125
<u>Appendix F. How to send patches.....</u>	126

# MPlayer – The Movie Player for LINUX

<http://www.mplayerhq.hu>

Copyright © 2000–2003 Arpad Gereoffy (A'rpi/ESP-team)

---

## Table of Contents

### How to read this documentation

#### 1. Introduction

##### 1.1. History

##### 1.2. Installation

###### 1.2.1. Software requirements

###### 1.2.2. Codecs

###### 1.2.3. Video cards

###### 1.2.3.1. YUV cards

###### 1.2.3.2. Non-YUV cards

###### 1.2.3.3. Cirrus-Logic cards

###### 1.2.4. Sound cards

###### 1.2.5. Features

##### 1.3. What about the GUI?

##### 1.4. Subtitles and OSD

###### 1.4.1. MPlayer's own subtitle format (MPsub)

###### 1.4.2. Installing OSD and subtitles

###### 1.4.3. OSD menu

##### 1.5. RTC

#### 2. Features

##### 2.1. Supported formats

###### 2.1.1. Video formats

###### 2.1.1.1. MPEG files

###### 2.1.1.2. AVI files

###### 2.1.1.3. ASF/WMV files

###### 2.1.1.4. QuickTime/MOV files

###### 2.1.1.5. VIVO files

###### 2.1.1.6. FLI files

###### 2.1.1.7. RealMedia (RM) files

###### 2.1.1.8. NuppelVideo files

###### 2.1.1.9. yuv4mpeg files

###### 2.1.1.10. FILM files

###### 2.1.1.11. RoQ files

###### 2.1.1.12. OGG/OGM files

###### 2.1.1.13. SDP files

###### 2.1.1.14. PVA files

###### 2.1.1.15. GIF files

###### 2.1.2. Audio formats

###### 2.1.2.1. MP3 files

###### 2.1.2.2. WAV files

###### 2.1.2.3. OGG/OGM files (Vorbis)

###### 2.1.2.4. WMA/ASF files

## MPlayer – The Movie Player for LINUX

- 2.1.2.5. MP4 files
- 2.1.2.6. CD audio
- 2.1.2.7. XMMS
- 2.2. Supported codecs
  - 2.2.1. Video codecs
    - 2.2.1.1. DivX4/DivX5
    - 2.2.1.2. FFmpeg/libavcodec
    - 2.2.1.3. XAnim codecs
    - 2.2.1.4. VIVO video
    - 2.2.1.5. MPEG 1/2 video
    - 2.2.1.6. MS Video1
    - 2.2.1.7. Cinepak CVID
    - 2.2.1.8. RealVideo
    - 2.2.1.9. XviD
    - 2.2.1.10. Sorenson
  - 2.2.2. Audio codecs
    - 2.2.2.1. Software AC3 decoding
    - 2.2.2.2. Hardware AC3 decoding
    - 2.2.2.3. libmad support
    - 2.2.2.4. VIVO audio
    - 2.2.2.5. RealAudio
    - 2.2.2.6. QDesign codecs
    - 2.2.2.7. Qualcomm codecs
    - 2.2.2.8. AAC codec
  - 2.2.3. Win32 codecs importing HOWTO
    - 2.2.3.1. VFW codecs
    - 2.2.3.2. DirectShow codecs
- 2.3. Output devices
  - 2.3.1. Video output devices
    - 2.3.1.1. Setting up MTRR
    - 2.3.1.2. Video outputs for traditional video cards
      - 2.3.1.2.1. Xv
      - 2.3.1.2.2. DGA
      - 2.3.1.2.3. SDL
      - 2.3.1.2.4. SVGalib
      - 2.3.1.2.5. Framebuffer output (FBdev)
      - 2.3.1.2.6. Matrox framebuffer (mga\_vid)
      - 2.3.1.2.7. 3Dfx YUV support
      - 2.3.1.2.8. OpenGL output
      - 2.3.1.2.9. AAlib – text mode displaying
      - 2.3.1.2.10. VESA – output to VESA BIOS
      - 2.3.1.2.11. X11
      - 2.3.1.2.12. VIDIX
      - 2.3.1.2.13. DirectFB
      - 2.3.1.2.14. DirectFB/Matrox (dfbmga)
    - 2.3.1.3. MPEG decoders
      - 2.3.1.3.1. DVB output and input
      - 2.3.1.3.2. DXR2
      - 2.3.1.3.3. DXR3/Hollywood+
    - 2.3.1.4. Other visualization hardware
      - 2.3.1.4.1. Zr

## MPlayer – The Movie Player for LINUX

- 2.3.1.4.2. [Blinkenlights](#)
- 2.3.1.5. [TV-out support](#)
  - 2.3.1.5.1. [Matrox G400 cards](#)
  - 2.3.1.5.2. [Matrox G450/G550 cards](#)
  - 2.3.1.5.3. [ATI cards](#)
  - 2.3.1.5.4. [Voodoo 3](#)
  - 2.3.1.5.5. [nVidia](#)
  - 2.3.1.5.6. [Neomagic](#)
- 2.3.2. [Audio output devices](#)
  - 2.3.2.1. [Audio/Video synchronisation](#)
  - 2.3.2.2. [Soundcard experiences, recommendations](#)
  - 2.3.2.3. [Audio filters](#)
    - 2.3.2.3.1. [Up/Downsampling](#)
    - 2.3.2.3.2. [Changing the number of channels](#)
    - 2.3.2.3.3. [Sample format converter](#)
    - 2.3.2.3.4. [Delay](#)
    - 2.3.2.3.5. [Software volume control](#)
    - 2.3.2.3.6. [Equalizer](#)
    - 2.3.2.3.7. [Panning filter](#)
    - 2.3.2.3.8. [Sub-woofer](#)
    - 2.3.2.3.9. [Surround-sound decoder](#)
    - 2.3.2.3.10. [Audio Exporter](#)
  - 2.3.2.4. [Audio plugins \(deprecated\)](#)
    - 2.3.2.4.1. [Up/Downsampling](#)
    - 2.3.2.4.2. [Surround Sound decoding](#)
    - 2.3.2.4.3. [Sample format converter](#)
    - 2.3.2.4.4. [Delay](#)
    - 2.3.2.4.5. [Software volume control](#)
    - 2.3.2.4.6. [Extrastereo](#)
    - 2.3.2.4.7. [Volume normalizer](#)
- 2.4. [TV input](#)
  - 2.4.1. [Compilation](#)
  - 2.4.2. [Usage tips](#)
  - 2.4.3. [Examples](#)
- 2.5. [Edit Decision Lists \(EDL\)](#)
  - 2.5.1. [Using an EDL file](#)
  - 2.5.2. [Making an EDL file](#)
- 3. [Usage](#)
  - 3.1. [Command line](#)
  - 3.2. [Control](#)
    - 3.2.1. [Controls configuration](#)
      - 3.2.1.1. [Key names](#)
      - 3.2.1.2. [Commands](#)
    - 3.2.2. [Control from LIRC](#)
    - 3.2.3. [Slave mode](#)
  - 3.3. [Streaming from network or pipes](#)
  - 3.4. [Remote streams](#)
    - 3.4.1. [Compiling the server](#)
    - 3.4.2. [Using remote streams](#)
- 4. [Frequently Asked Questions](#)
- 5. [CD/DVD usage](#)



- 5.1. [CD/DVD drives](#)
- 5.2. [DVD playback](#)
- 5.3. [VCD playback](#)
- 6. [Ports](#)
  - 6.1. [Linux](#)
    - 6.1.1. [Debian packaging](#)
    - 6.1.2. [RPM packaging](#)
    - 6.1.3. [ARM](#)
  - 6.2. [\\*BSD](#)
    - 6.2.1. [FreeBSD](#)
    - 6.2.2. [OpenBSD](#)
    - 6.2.3. [Darwin](#)
  - 6.3. [Sun Solaris](#)
  - 6.4. [Silicon Graphics Irix](#)
  - 6.5. [QNX](#)
  - 6.6. [Windows](#)
    - 6.6.1. [Cygwin](#)
    - 6.6.2. [MinGW](#)
  - 6.7. [Mac OS](#)
  - 6.8. [HP UX](#)
  - 6.9. [Amiga/MorphOS \(GeekGadgets\)](#)
- 7. [Encoding with MEncoder](#)
  - 7.1. [Encoding 2 or 3-pass MPEG-4 \("DivX"\)](#)
  - 7.2. [Encoding to MPEG format](#)
  - 7.3. [Rescaling movies](#)
  - 7.4. [Stream copying](#)
  - 7.5. [Fixing AVIs with broken index or interleaving](#)
    - 7.5.1. [Appending multiple AVI files](#)
  - 7.6. [Encoding with the libavcodec codec family](#)
  - 7.7. [Encoding from multiple input image files \(JPEGs, PNGs or TGAs\)](#)
  - 7.8. [Extracting DVD subtitles to Vobsub file](#)
  - 7.9. [Preserving aspect ratio](#)
  - 7.10. [Custom inter/intra matrices](#)
  - 7.11. [Making a high quality MPEG4 \("DivX"\) rip of a DVD movie](#)
    - 7.11.1. [Cropping](#)
    - 7.11.2. [Quality level](#)
    - 7.11.3. [Files over 2GB](#)
    - 7.11.4. [Deinterlacing](#)
    - 7.11.5. [Inverse telecine](#)
    - 7.11.6. [Scaling and aspect ratio](#)
    - 7.11.7. [Summing it up](#)
- A. [Mailing lists](#)
- B. [How to report bugs](#)
  - B.1. [How to fix bugs](#)
  - B.2. [How to report bugs](#)
  - B.3. [Where to report bugs](#)
  - B.4. [What to report](#)
    - B.4.1. [System Information](#)
    - B.4.2. [Hardware and drivers](#)
    - B.4.3. [Configure problems](#)
    - B.4.4. [Compilation problems](#)

## MPlayer – The Movie Player for LINUX

*B.4.5. Playback problems*

*B.4.6. Crashes*

*B.4.6.1. How to conserve information about a reproducible crash*

*B.4.6.2. How to extract meaningful information from a core dump*

*B.5. I know what I am doing...*

*C. Known bugs*

*C.1. Special system/CPU-specific bugs/problems*

*C.2. Various A–V sync and other audio problems*

*C.2.1. General audio delay or jerky sound (exists with all or many files)*

*C.2.2. Audio delay/de–sync specific to one or a few files*

*C.2.3. No sound at all*

*C.2.4. No picture at all (just plain grey/green window)*

*C.2.5. Video–out problems*

*D. MPlayer skin format*

*D.1. Overview*

*D.1.1. Directories*

*D.1.2. Image formats*

*D.1.3. Skin components*

*D.1.4. Files*

*D.2. The skin file*

*D.2.1. Main window and playbar*

*D.2.2. Subwindow*

*D.2.3. Skin menu*

*D.3. Fonts*

*D.3.1. Symbols*

*D.4. GUI messages*

*E. Developer cries*

*E.1. GCC 2.96*

*E.2. Binary distribution*

*E.3. nVidia*

*E.4. Joe Barr*

*F. How to send patches*

## How to read this documentation

If you are a first–time installer: be sure to read everything from here to the end of the Installation section, and follow the links you will find. If you have any other questions, return to the [Table of Contents](#) and search for the topic, read the [FAQ](#), or try grepping through the files. Most questions should be answered somewhere here and the rest has probably already been asked on our [mailing lists](#). Check the [archives](#), there is a lot of valuable information to be found there.

## Chapter 1. Introduction

### Table of Contents

*1.1. History*

*1.2. Installation*

*1.2.1. Software requirements*

*1.2.2. Codecs*

*1.2.3. Video cards*

## MPlayer – The Movie Player for LINUX

### 1.2.3.1. YUV cards

### 1.2.3.2. Non-YUV cards

### 1.2.3.3. Cirrus-Logic cards

### 1.2.4. Sound cards

### 1.2.5. Features

## 1.3. What about the GUI?

### 1.4. Subtitles and OSD

#### 1.4.1. MPlayer's own subtitle format (MPsub)

#### 1.4.2. Installing OSD and subtitles

#### 1.4.3. OSD menu

## 1.5. RTC

MPlayer is a movie player for Linux (runs on many other Unices, and **non-x86** CPUs, see Ports). It plays most MPEG, VOB, AVI, OGG/OGM, VIVO, ASF/WMA/WMV, QT/MOV/MP4, FLI, RM, NuppelVideo, yuv4mpeg, FILM, RoQ, PVA, Matroska files, supported by many native, XAnim, RealPlayer, and Win32 DLL codecs. You can watch **VideoCD, SVCD, DVD, 3ivx, RealMedia, Sorenson, Theora, and DivX** movies too (and you don't need the avifile library at all!). Another big feature of mplayer is the wide range of supported output drivers. It works with X11, Xv, DGA, OpenGL, SVGAlib, fbdev, AALib, DirectFB, but you can use GGI and SDL (and this way all their drivers) and some lowlevel card-specific drivers (for Matrox, 3Dfx and Radeon, Mach64, Permedia3) too! Most of them supports software or hardware scaling, so you can enjoy movies in fullscreen. MPlayer supports displaying through some hardware MPEG decoder boards, such as the DVB and DXR3/Hollywood+. And what about the nice big antialiased shaded subtitles (**10 supported types**) with European/ISO 8859-1,2 (Hungarian, English, Czech, etc), Cyrillic, Korean fonts, and the onscreen display (OSD)?

The player is rock solid playing damaged MPEG files (useful for some VCDs), and it plays bad AVI files which are unplayable with the famous windows media player. Even AVI files without index chunk are playable, and you can temporarily rebuild their indexes with the `-idx` option, or permanently with MEncoder, thus enabling seeking! As you see, stability and quality are the most important things, but the speed is also amazing.

MEncoder (MPlayer's Movie Encoder) is a simple movie encoder, designed to encode MPlayer-playable movies (**AVI/ASF/OGG/DVD/VCD/VOB/MPG/MOV/VIV/FLI/RM/NUV/NET/PVA**) to other MPlayer-playable formats (see below). It can encode with various codecs, like **DivX4** (1 or 2 passes), libavcodec, **PCM/MP3/VBR MP3** audio. Also has powerful plugin system for video manipulation.

### **MEncoder features**

- Encoding from the wide range of fileformats and decoders of MPlayer
- Encoding to all the codecs of ffmpeg's libavcodec
- Video encoding from **V4L compatible TV tuners**
- Encoding/multiplexing to interleaved AVI files with proper index
- Creating files from external audio stream
- 1, 2 or 3 pass encoding
- **VBR MP3** audio

### **Important**

VBR MP3 audio doesn't always play nicely on windows players!

- PCM audio
- Stream copying

## MPlayer – The Movie Player for LINUX

- Input A/V synchronizing (PTS-based, can be disabled with `-mc 0` option)
- FPS correction with `-ofps` option (useful when encoding 29.97 fps VOB to 24 fps AVI)
- Using our very powerful plugin system (crop, expand, flip, postprocess, rotate, scale, rgb/yuv conversion)
- Can encode DVD/VOBsub **AND** text subtitles into the output file
- Can rip DVD subtitles to Vobsub format

### Planned features

- Even wider variety of available en/decoding formats/codecs (creating VOB files with DivX4/Indeo5/VIVO streams :).

MPlayer and MEncoder can be distributed under the terms of the GNU General Public License Version 2.

## 1.1. History

This began a year ago... I have tried lots of players under linux (mtv, xmps, dvdview, livid/oms, videolan, xine, xanim, avifile, xmpm) but they all have some problem. Mostly with special files or with audio/video sync. Most of them are unable to play both MPEG1, MPEG2 and AVI (DivX) files. Many players have image quality or speed problems too. So I've decided to write/modify one...

A'rpi, 2001

- **mpg12play v0.1–v0.3**: Sep 22–25, 2000

The first try, hacked together in a half hour! I've used libmpeg3 from <http://www.heroinewarrior.com> up to the version 0.3, but there were image quality and speed problems with it.

- **mpg12play v0.5–v0.87**: Sep 28–Oct 20, 2000

Mpeg codec replaced with DVDview by Dirk Farin, it was a great stuff, but it was slow and was written in C++ (A'rpi hates C++!!!)

- **mpg12play v0.9–v0.95pre5**: Oct 21–Nov 2, 2000

Mpeg codec was libmpeg2 (mpeg2dec) by Aaron Holtzman and Michel Lespinasse. It's great, optimized very fast C code with perfect image quality and 100% MPEG standard conformance.

- **MPlayer v0.3–v0.9**: Nov 18–Dec 4, 2000

It was a pack of two programs: mpg12play v0.95pre6 and my new simple AVI player 'avip' based on avifile's Win32 DLL loader.

- **MPlayer v0.10**: Jan 1, 2001

The MPEG and AVI player in a single binary!

- **MPlayer v0.11pre series**:

Some new developers joined and since 0.11 the MPlayer project is a team-work! Added ASF file support, and OpenDivX (see <http://www.projectmayo.com>) en/decoding.

- **MPlayer v0.17a "The IdegCounter"** Apr 27, 2001

## MPlayer – The Movie Player for LINUX

The release version of the 0.11pre after 4 months of heavy development! Try it, and be amazed! Thousands of new features added... and of course old code was improved too, bugs removed etc.

- **MPlayer 0.18 "The BugCounter"** Jul 9, 2001

2 months since 0.17 and here's a new release.. Completed ASF support, more subtitle formats, introduced libao (similar to libvo but to audio), even more stable than ever, and so on. It's a MUST!

- **MPlayer 0.50 "The Faszom(C)ounter"** Oct 8, 2001

Hmm. Release again. Tons of new features, beta GUI version, bugs fixed, new vo and ao drivers, ported to many systems, including opensource DivX codecs and much more. Try it!

- **MPlayer 0.60 "The RTFMCounter"** Jan 3, 2002

MOV/VIVO/RM/FLI/NUV fileformats support, native CRAM, Cinepak, ADPCM codecs, and support for XAnim's binary codecs; DVD subtitles support, first release of MEncoder, TV grabbing, cache, liba52, countless fixes.

- **MPlayer 0.90pre10 "The BirthdayCounter"** Nov 11, 2002

Although this is not a release, I am going to mention it because it came out 2 years after MPlayer v0.01. Happy birthday, MPlayer!

- **MPlayer 0.90rc1 "The CodecCounter"** Dec 7, 2002

Again not a release, but after adding Sorenson 3 (QuickTime) and Windows Media 9 support, MPlayer is the world's first movie player with support for all known video formats!

- **MPlayer 0.90 "The CounterCounter"** Apr 6, 2003

After more than 1 year, we finally concluded that the code was indeed stable again, and ready to be published as a release. Unfortunately we forgot even to increase the version number, and other annoying bugs went in, so get ready for...

- **MPlayer 0.91** Aug 13, 2003

The above mentioned and lot of other bugs have been fixed. This is the latest stable version.

- **MPlayer 1.0pre1 "Development on the beach"** Sep 1, 2003

Although this is not a stable release, I am going to mention it because it is the first pre version of the 1.0 series of MPlayer and it is intended to help the *big bug hunting party*. This is a huge step forward!

- **MPlayer 1.0** date yet unknown

## 1.2. Installation

A quick installation guide can be found in the README file. Please read it first and then come back here for the rest of the gory details.

In this section I'll try to guide you through the compiling and configuring process of MPlayer. It's not easy, but it won't necessarily be hard. If you experience a different behavior than what I explain, please search through this documentation and you'll find your answers. If you see links, please follow them and read carefully what they contain. It will take some time, but it IS worth it.

You need a fairly recent system. On Linux, 2.4.x kernels are recommended.

## 1.2.1. Software requirements

- **binutils** – suggested version is **2.11.x**. This program is responsible for generating MMX/ 3DNow!/etc instructions, thus very important.
- **gcc** – suggested versions are: **2.95.3** (maybe **2.95.4**) and **3.2+**. **Never** use 2.96 or 3.0.x! They generate faulty code for MPlayer. If you decide to change gcc from 2.96, then don't decide in favor of 3.x just because it's newer! Early releases of 3.x were even more buggy than 2.96. So downgrade to 2.95.x (downgrade `libstdc++` too, other programs may need it) or don't up/downgrade at all (but in this case, be prepared for runtime problems). If you vote for 3.x, try to use the latest version, early releases had various bugs, so be sure you use at least 3.1, it's tested and working. For detailed information about gcc 2.96's bugs (that are still NOT fixed, they have been WORKED AROUND in MPlayer!), see the [gcc 2.96](#) section and the [FAQ](#).
- **XFree86** – suggested version is **always the newest** (4.3). Normally, everyone wants this, as starting with XFree86 4.0.2, it contains the [XVideo](#) extension (somewhere referred to as **Xv**) which is needed to enable the hardware YUV acceleration (fast image display) on cards that support it. Make sure its **development package** is installed, too, otherwise it won't work. For some video cards you don't need XFree86. See list below.
- **make** – suggested version is **always the newest** (at least 3.79.x). This usually isn't important.
- **SDL** – it's not mandatory, but can help in some cases (bad audio, video cards that lag strangely with the xv driver). Always use the newest (beginning from 1.2.x).
- **libjpeg** – optional JPEG decoder, used by the `-mf` option and some QT MOV files. Useful for both MPlayer and MEncoder if you plan to work with jpeg files.
- **libpng** – recommended and default (M)PNG decoder. Required for GUI. Useful for both MPlayer and MEncoder.
- **lame** – recommended, needed for encoding MP3 audio with MEncoder, suggested version is *always the newest* (at least 3.90).
- **zlib** – recommended, necessary for compressed MOV header and PNG support.
- **libogg** – optional, needed for playing OGG file format.
- **libvorbis** – optional, needed for playing OGG Vorbis audio.
- **LIVE.COM Streaming Media** – optional, needed for playing RTSP/RTP streams.
- **directfb** – optional, from <http://www.directfb.org>
- **cdparanoia** – optional, for CDDA support
- **libfreetype** – optional, for TTF fonts support. At least 2.0.9 is required.
- **libxmms** – optional, for XMMS input plugin support. At least 1.2.7 is required.
- **libsmb** – optional, for Samba support.

## 1.2.2. Codecs

- **libavcodec**: This codec package is capable of decoding H263/MJPEG/RV10/DivX3/DivX4/DivX5/MP41/MP42/WMV1/WMV2/SVQ1/SVQ3 encoded video streams and WMA (Windows Media Audio) v1/v2 audio streams, on multiple platforms. It is also known to be the fastest for this task. See [FFmpeg](#) section for details. Features:
  - ◆ gain decoding of videos mentioned above, on non-x86 machines
  - ◆ encoding with most of the mentioned codecs
  - ◆ this codec is the **fastest codec available** for DivX/3/4/5 and other MPEG4 types.  
Recommended!
- **Win32 codecs**: If you plan to use MPlayer on x86 architecture, you will possibly need them. Download the Win32 codecs from our [codecs page](#) and install them to `/usr/local/lib/codecs` **BEFORE** compiling MPlayer, otherwise no Win32 support will be compiled!

## Note

The avifile project has similar codecs package, but it differs from ours, so if you want to use all supported codecs, then use our package (do not worry, avifile works with it without problems).

Features:

- ◆ you need this if you want to play or encode for example movies recorded with various hardware compressors, like tuner cards, digital cameras (example: DV, ATI VCR, MJPEG)
- ◆ needed if you want to play **WMV9/WMA9** movies.
- ◆ Not needed for old ASF's with MP41 or MP42 video (though VoxWare audio is frequent for these files – it's done by the Win32 codec), or WMV7. Also not needed for WMA (Windows Media Audio), libavcodec has opensource decoder for that.
- **QuickTime codecs:** on x86 platforms these codecs can be used to decode RPZA, and other QuickTime video, and QDesign audio streams. Installation instructions can be found in the Sorenson video codec section.
- **DivX4/DivX5:** information about this codec is available in the DivX4/DivX5 section. You possibly don't want this codec as **libavcodec** (see above) is much faster and has better quality than this, for both decoding and encoding. Features:
  - ◆ 1 pass or 2 pass encoding with MEncoder
  - ◆ can play old **DivX3** movies much faster than the Win32 DLL but slower than **libavcodec!**
  - ◆ it's closed-source, and only a x86 version is available.
- **XviD:** Open source encoding alternative to Divx4Linux. Features:
  - ◆ 1 pass or 2 pass encoding with MEncoder
  - ◆ it's open-source, so it's multiplatform.
  - ◆ it's about 2 times faster than divx4 when encoding – about the same quality
- The XAnim codecs are the best (full screen, hardware YUV zoom) for decoding **3ivx** and Indeo 3/4/5 movies, and some old formats. And they are multiplatform, so this is the only way to play Indeo on non-x86 platforms (well, apart from using XAnim:). But for example Cinepak movies are best played with MPlayer's own Cinepak decoder!
- For **Ogg Vorbis** audio decoding you need to install `libvorbis` properly. Use deb/rpm packages if available, or compile from source (this is a nightly updated tarball of Vorbis CVS).
- MPlayer can use the libraries of RealPlayer 8 or RealONE to play files with **RealVideo 2.0–4.0** video, and Sipro/Cook audio. See RealMedia file format section for installation instructions and more information.

## 1.2.3. Video cards

There are generally two kind of video cards. One kind (the newer cards) has **hardware scaling and YUV acceleration** support, the other cards don't.

### 1.2.3.1. YUV cards

They can display and scale (zoom) the picture to any size that fits in their memory, with **small CPU usage** (even when zooming), thus fullscreen is nice and very fast.

- **Matrox G200/G400/G450/G550 cards:** although a Vidix driver is provided, it is recommended to use the `mga_vid` module instead, for it works much better. Please see the mga\_vid section about its installation and usage. It is important to do these steps *before* compiling MPlayer, otherwise no `mga_vid` support will be built. Also check out the Matrox TV-out section. **If you don't use Linux,**

## MPlayer – The Movie Player for LINUX

your only possibility is the VIDIX driver: read the [VIDIX](#) section.

- **3Dfx Voodoo3/Banshee cards:** please see [tdfxfb](#) section in order to gain big speedup. It is important to do these steps **before** compiling MPlayer, otherwise no 3Dfx support will be built. Also see the [3dfx TV-out](#) section. If you use X, use at least **4.2.0**, as 3dfx Xv driver was broken in 4.1.0, and earlier releases.
- **ATI cards:** [VIDIX](#) driver is provided for the following cards: **Radeon, Rage128, Mach64** (Rage XL/Mobility, Xpert98). Also see the [ATI cards](#) section of the TV-out documentation, to know if you card's TV-out is supported under Linux/MPlayer.
- **S3 cards:** the Savage and Virge/DX chips have hardware acceleration. Use as new XFree86 version as possible, older drivers are buggy. Savage chips have problems with YV12 display, see [S3 Xv](#) section for details. Older, Trio cards have no, or slow hardware support.
- **nVidia cards:** may or may not be good choice for video playing. If you do not have a GeForce2 (or newer) card, it's not likely to work without bugs. **the built-in nVidia driver in XFree86 does not support hardware YUV acceleration on all nVidia cards.** You have to download nVidia's closed-source drivers from [nVidia.com](#). See the [nVidia Xv driver](#) section for details. Please also check the [nVidia TV-out](#) section if you wish to use a TV.
- **3DLabs GLINT R3 and Permedia3:** a VIDIX driver is provided (pm3\_vid). Please see the [VIDIX](#) section for details.
- **Other cards:** none of the above?
  - ◆ Try if the XFree86 driver (and your card) supports hardware acceleration. See the [Xv](#) section for details.
  - ◆ If it doesn't, then your card's video features aren't supported under your operating system :( If hardware scaling works under Windows, it doesn't mean it will work under Linux or other operating system, it depends on the drivers. Most manufacturers neither make Linux drivers nor release specifications of their chips – so you are unlucky if using their cards. See [Non-YUV cards](#).

### 1.2.3.2. Non-YUV cards

Fullscreen playing can be achieved by either enabling **software scaling** (use the `-zoom` or `-vf` option, but I warn you: this is slow), or switching to a smaller video mode, for example 352x288. If you don't have YUV acceleration, this latter method is recommended. Video mode switching can be enabled by using the `-vm` option and it works with the following drivers:

- **using XFree86:** see details in [DGA driver](#) and [X11 driver](#) sections. DGA is recommended! Also try DGA via SDL, sometimes it's better.
- **not using XFree86:** try the drivers in the following order: [vesa](#), [fbdev](#), [svgalib](#), [aalib](#).

### 1.2.3.3. Cirrus-Logic cards

- GD 7548: present on-board and tested in Compaq Armada 41xx notebook series.
  - ◆ XFree86 3: works in 8/16bpp modes. However, the driver is dramatically slow and buggy in 800x600@16bpp. **Recommended: 640x480@16bpp**
  - ◆ XFree86 4: the Xserver freezes soon after start unless acceleration is disabled, but then the whole thing gets slower than XFree86 3. No XVideo.
  - ◆ FBdev: framebuffer can be turned on with the `clgenfb` driver in the kernel, though for me it worked only in 8bpp, thus unusable. The `clgenfb` source had to be extended with the 7548 ID before compilation.
  - ◆ VESA: the card is only VBE 1.2 capable, so VESA output can't be used. Can't be workarounded with UniVBE.
  - ◆ SVGAlib: detects an older Cirrus chip. Usable but slow with `-bpp 8`.



## 1.2.4. Sound cards

- **Soundblaster Live!**: with this card you can use 4 or 6 (**5.1**) channels AC3 decoding instead of 2. Read the [Software AC3 decoding](#) section. For hardware AC3 passthrough you **must** use ALSA 0.9 with OSS emulation!
- **C-Media with SP/DIF out**: hardware AC3 passthrough is possible with these cards, see [Hardware AC3 decoding](#) section.
- Features of **other cards** aren't supported by MPlayer. **It's very recommended to read the [sound card](#) section!**

## 1.2.5. Features

- Decide if you need GUI. If you do, see the [GUI](#) section before compiling.
- If you want to install MEncoder (our great all-purpose encoder), see the [MEncoder](#) section.
- If you have a V4L compatible **TV tuner** card, and wish to watch/grab and encode movies with MPlayer, read the [TV input](#) section.
- There is a neat **OSD Menu** support ready to be used. Check the [OSD menu](#) section.

Then build MPlayer:

```
./configure
make
make install
```

At this point, MPlayer is ready to use. The directory `$PREFIX/share/mplayer` contains the `codecs.conf` file, which is used to tell the program all the codecs and their capabilities. This file is needed only when you want to change its properties, as the main binary contains an internal copy of it. Check if you have `codecs.conf` in your home directory (`~/mplayer/codecs.conf`) left from old MPlayer versions, and remove it.

Note that if you have a `codecs.conf` in `~/mplayer/`, the builtin and system `codecs.conf` will be ignored completely. Do not do this unless you want to fiddle with MPlayer internals as this can cause many problems. If you want to change the codecs search order, use the `-vc`, `-ac`, `-vfm`, or `-afm` options either on the command line or in your config file (see the manual page).

Debian users can build a `.deb` package for themselves, it's very simple. Just exec

```
fakeroot debian/rules binary
```

in MPlayer's root directory. See [Debian packaging](#) for detailed instructions.

**Always browse the output of `./configure`**, and the `configure.log` file, they contain information about what will be built, and what will not. You may also want to view `config.h` and `config.mak` files. If you have some libraries installed, but not detected by `./configure`, then check if you also have the proper header files (usually the `-dev` packages) and their version matches. The `configure.log` file usually tells you what is missing.

Though not mandatory, the fonts should be installed in order to gain OSD, and subtitle functionality. The recommended method is installing a TTF font file and telling MPlayer to use it. See the [Subtitles and OSD](#) section for details.

## 1.3. What about the GUI?

The GUI needs GTK 1.2.x (it isn't fully GTK, but the panels are). The skins are stored in PNG format, so GTK, `libpng` (and their devel stuff, usually called `gtk-dev` and `libpng-dev`) has to be installed. You can build it by specifying `--enable-gui` during `./configure`. Then, to turn on GUI mode, you have to execute the **gmplayer** binary.

Currently you can't use the `-gui` option on the command line, due to technical reasons.

As MPlayer doesn't have a skin included, you have to download them if you want to use the GUI. See the [download page](#). They should be extracted to the usual system-wide directory (`$PREFIX/share/mplayer/Skin`), or to `$HOME/.mplayer/Skin`. MPlayer by default looks in these directories for a directory named `default`, but you can use the `-skin newskin` option, or the `skin=newskin` config file directive to use the skin in `*/Skin/newskin` directory.

## 1.4. Subtitles and OSD

MPlayer can display subtitles along with movie files. Currently the following formats are supported:

- VobSub
- OGM
- CC (closed caption)
- MicroDVD
- SubRip
- SubViewer
- Sami
- VPlayer
- RT
- SSA
- Unknown
- MPsub
- AQTitle
- JACOsub

MPlayer can dump the previously listed subtitle formats (**except the three first**) into the following destination formats, with the given options:

- MPsub: `-dumpmpsub`
- SubRip: `-dumpsrtsub`
- MicroDVD: `-dumpmicrodvds`
- JACOsub: `-dumpjac`
- Sami: `-dumpsami`

MEncoder can dump DVD subtitles into VobSub format.

The command line options differ slightly for the different formats:

**VobSub subtitles.** VobSub subtitles consist of a big (some megabytes) `.SUB` file, and optional `.IDX` and/or `.IFO` files. If you have files like `sample.sub`, `sample.ifo` (optional), `sample.idx` – you have to pass MPlayer the `-vobsub sample [-vobsubid id]` options (full path optional). The `-vobsubid`

## MPlayer – The Movie Player for LINUX

option is like `-sid` for DVDs, you can choose between subtitle tracks (languages) with it. In case that `-vobsubid` is omitted, MPlayer will try to use the languages given by the `-slang` option and fall back to the `langidx` in the `.IDX` file to set the subtitle language. If it fails, there will be no subtitles.

**Other subtitles.** The other formats consist of a single text file containing timing, placement and text information. Usage: If you have a file like `sample.txt`, you have to pass the option `-sub sample.txt` (full path optional).

### Adjusting subtitle timing and placement:

`-subdelay sec`  
Delays subtitles by `sec` seconds. Can be negative.

`-subfps RATE`  
Specify frame/sec rate of subtitle file (float number).

`-subpos 0-100`  
Specify the position of subtitles.

If you experience a growing delay between the movie and the subtitles when using a MicroDVD subtitle file, most likely the frame rate of the movie and the subtitle file are different. Please note that the MicroDVD subtitle format uses absolute frame numbers for its timing, and therefore the `-subfps` option cannot be used with this format. As MPlayer has no way to guess the frame rate of the subtitle file, you have to manually convert the frame rate. There is a little perl script in the `contrib` directory of the MPlayer FTP site to do this conversion for you.

About DVD subtitles, read the [DVD](#) section.

### 1.4.1. MPlayer's own subtitle format (MPsub)

MPlayer introduces a new subtitle format called **MPsub**. It was designed by Gabucino. Basically its main feature is being *dynamically* time-based (although it has frame-based mode too). Example (from [DOCS/tech/mpsub.sub](#)):

```
FORMAT=TIME
# first number : wait this much after previous subtitle disappeared
# second number : display the current subtitle for this many seconds

15 3
A long long, time ago...

0 3
in a galaxy far away...

0 3
Naboo was under an attack.
```

So you see, the main goal was to **make subtitle editing/timing/joining/cutting easy**. And, if you – say – get an SSA subtitle but it's badly timed/delayed to your version of the movie, you simply do a

```
mplayer dummy.avi -sub source.ssa -dumpmpsub
```

A `dump.mpsub` file will be created in the current directory, which will contain the source subtitle's text, but in **MPsub** format. Then you can freely add/subtract seconds to/from the subtitle.

Subtitles are displayed with a technique called '**OSD**', **On Screen Display**. OSD is used to display current time, volume bar, seek bar etc.

### 1.4.2. Installing OSD and subtitles

You need an MPlayer font package to be able to use OSD/SUB feature. There are many ways to get it:

- Use the font generator tool at `TOOLS/subfont-c`. It's a complete tool to convert from TTF/Type1/etc font to mplayer font pkg (read `TOOLS/subfont-c/README` for details).
- Use the font generator GIMP plugin at `TOOLS/subfont-GIMP` (note: you must have HSI RAW plugin too, see <http://realtime.ssu.ac.kr/~lethean/mplayer/>).
- using a TrueType (TTF) font, by the means of the `freetype` library. Version 2.0.9 or greater is mandatory! Then you have two methods:
  - ◆ use the `-font /path/to/sample_font.ttf` option to specify a TrueType font file on every occasion
  - ◆ create a symlink:

```
ln -s /path/to/sample_font.ttf ~/.mplayer/subfont.ttf
```

If MPlayer was compiled with `fontconfig` support, the above methods won't work, instead the `-font` expects a `fontconfig` font name and defaults to the sans-serif font. To get a list of fonts known to `fontconfig`, use `fc-list`. Example: `-font 'Bitstream Vera Sans'`

- Download ready-to-use font packages from MPlayer site. Note: currently available fonts are limited for ISO 8859-1/2 support, but there are some other (including Korean, Russian, ISO 8859-8 etc) fonts at `contrib/font` section of FTP, made by users.

Font should have appropriate `font.desc` file which maps unicode font positions to the actual code page of the subtitles text. Other solution is to have subtitles encoded in UTF8 encoding and use `-utf8` option or just name the subtitles file `<video_name>.utf` and have it in the same dir as the video file. Recoding from different codepages to UTF8 could be done by using `konwert` or `iconv` programs.

**Table 1.1. Some URLs**

URL	Comment
<a href="ftp://ftp.mplayerhq.hu/MPlayer/releases/fonts/">ftp://ftp.mplayerhq.hu/MPlayer/releases/fonts/</a>	ISO fonts
<a href="ftp://ftp.mplayerhq.hu/MPlayer/contrib/fonts/">ftp://ftp.mplayerhq.hu/MPlayer/contrib/fonts/</a>	various fonts by users
<a href="http://realtime.ssu.ac.kr/~lethean/mplayer/">http://realtime.ssu.ac.kr/~lethean/mplayer/</a>	Korean fonts and RAW plugin

If you chose non-TTF fonts, UNZIP the file you downloaded to `~/.mplayer` or `$PREFIX/share/mplayer`. Then rename or symlink one of the extracted directories to `font`, for example:

```
ln -s ~/.mplayer/arial-24 ~/.mplayer/font
```

Now you have to see a timer at the upper left corner of the movie (switch it off with the `o` key).

(subtitles are *always enabled*, for disabling them please read the man page)

OSD has 4 states (switch with `o`):

## MPlayer – The Movie Player for LINUX

1. volume bar + seek bar (default)
2. volume bar + seek bar + timer + file position percentage on seeking
3. volume bar + seek bar + timer + total duration of the media
4. subtitles only

You can change default behaviour by setting `osdlevel` variable in config file, or the `-osdlevel` command line option.

### 1.4.3. OSD menu

MPlayer has a completely user definiable OSD Menu interface.

#### Note

the Preferences menu is currently UNIMPLEMENTED!

#### Installation

1. compile MPlayer by passing the `--enable-menu` to `./configure`
2. make sure you have an OSD font installed
3. copy `etc/menu.conf` to your `.mplayer` directory
4. copy `etc/input.conf` to your `.mplayer` directory, or to the system-wide MPlayer config dir (default: `/usr/local/etc/mplayer`)
5. check and edit `input.conf` to enable menu movement keys (it is described there).
6. start MPlayer by the following example:

```
$ mplayer -menu file.avi
```

7. push any menu key you defined

## 1.5. RTC

There are three timing methods in MPlayer.

- **To use the old method**, you don't have to do anything. It uses `usleep()` to tune A/V sync, with +/- 10ms accuracy. However sometimes the sync has to be tuned even finer.
- **The new timer** code uses PC's RTC (Real Time Clock) for this task, because it has precise 1ms timers. It is automagically enabled when available, but requires root privileges, a `setuid root` MPlayer binary or a properly set up kernel. If you are running kernel 2.4.19pre8 or later you can adjust the maximum RTC frequency for normal users through the `/proc` filesystem. Use this command to enable RTC for normal users:

```
echo 1024 > /proc/sys/dev/rtc/max-user-freq
```

If you do not have such a new kernel, you can also change one line in `drivers/char/rtc.c` and recompile your kernel. Find the section that reads

```
* We don't really want Joe User enabling more
* than 64Hz of interrupts on a multi-user machine.
*/
if ((rtc_freq > 64) && (!capable(CAP_SYS_RESOURCE)))
```

and change the 64 to 1024. You should really know what you are doing, though. You can see the new timer's efficiency in the status line. The power management functions of some notebook BIOSes with speedstep CPUs interact badly with RTC. Audio and video may get out of sync. Plugging the external power connector in before you power up your notebook seems to help. You can always turn off RTC support with the `-nortc` option. In some hardware combinations (confirmed during usage of non-DMA DVD drive on an ALi1541 board) usage of the RTC timer causes skippy playback. It's recommended to use the third method in these cases.

- **The third timer code** is turned on with the `-softsleep` option. It has the efficiency of the RTC, but it doesn't use RTC. On the other hand, it requires more CPU.

### Note

**NEVER install a setuid root MPlayer binary on a multiuser system!** It's a clear way for everyone to become root.

## Chapter 2. Features

### Table of Contents

#### 2.1. Supported formats

##### 2.1.1. Video formats

- 2.1.1.1. MPEG files
- 2.1.1.2. AVI files
- 2.1.1.3. ASF/WMV files
- 2.1.1.4. QuickTime/MOV files
- 2.1.1.5. VIVO files
- 2.1.1.6. FLI files
- 2.1.1.7. RealMedia (RM) files
- 2.1.1.8. NuppelVideo files
- 2.1.1.9. yuv4mpeg files
- 2.1.1.10. FILM files
- 2.1.1.11. RoQ files
- 2.1.1.12. OGG/OGM files
- 2.1.1.13. SDP files
- 2.1.1.14. PVA files
- 2.1.1.15. GIF files

##### 2.1.2. Audio formats

- 2.1.2.1. MP3 files
- 2.1.2.2. WAV files
- 2.1.2.3. OGG/OGM files (Vorbis)
- 2.1.2.4. WMA/ASF files
- 2.1.2.5. MP4 files
- 2.1.2.6. CD audio
- 2.1.2.7. XMMS

#### 2.2. Supported codecs

##### 2.2.1. Video codecs

- 2.2.1.1. DivX4/DivX5
- 2.2.1.2. FFmpeg/libavcodec
- 2.2.1.3. XAnim codecs
- 2.2.1.4. VIVO video

## MPlayer – The Movie Player for LINUX

- 2.2.1.5. MPEG 1/2 video
- 2.2.1.6. MS Video1
- 2.2.1.7. Cinepak CVID
- 2.2.1.8. RealVideo
- 2.2.1.9. XviD
- 2.2.1.10. Sorenson
- 2.2.2. Audio codecs
  - 2.2.2.1. Software AC3 decoding
  - 2.2.2.2. Hardware AC3 decoding
  - 2.2.2.3. libmad support
  - 2.2.2.4. VIVO audio
  - 2.2.2.5. RealAudio
  - 2.2.2.6. QDesign codecs
  - 2.2.2.7. Qualcomm codecs
  - 2.2.2.8. AAC codec
- 2.2.3. Win32 codecs importing HOWTO
  - 2.2.3.1. VFW codecs
  - 2.2.3.2. DirectShow codecs
- 2.3. Output devices
  - 2.3.1. Video output devices
    - 2.3.1.1. Setting up MTRR
    - 2.3.1.2. Video outputs for traditional video cards
      - 2.3.1.2.1. Xv
      - 2.3.1.2.2. DGA
      - 2.3.1.2.3. SDL
      - 2.3.1.2.4. SVGAlib
      - 2.3.1.2.5. Framebuffer output (FBdev)
      - 2.3.1.2.6. Matrox framebuffer (mga\_vid)
      - 2.3.1.2.7. 3Dfx YUV support
      - 2.3.1.2.8. OpenGL output
      - 2.3.1.2.9. AAlib – text mode displaying
      - 2.3.1.2.10. VESA – output to VESA BIOS
      - 2.3.1.2.11. X11
      - 2.3.1.2.12. VIDIX
      - 2.3.1.2.13. DirectFB
      - 2.3.1.2.14. DirectFB/Matrox (dfbmga)
    - 2.3.1.3. MPEG decoders
      - 2.3.1.3.1. DVB output and input
      - 2.3.1.3.2. DXR2
      - 2.3.1.3.3. DXR3/Hollywood+
    - 2.3.1.4. Other visualization hardware
      - 2.3.1.4.1. Zr
      - 2.3.1.4.2. Blinkenlights
    - 2.3.1.5. TV-out support
      - 2.3.1.5.1. Matrox G400 cards
      - 2.3.1.5.2. Matrox G450/G550 cards
      - 2.3.1.5.3. ATI cards
      - 2.3.1.5.4. Voodoo 3
      - 2.3.1.5.5. nVidia
      - 2.3.1.5.6. Neomagic
  - 2.3.2. Audio output devices

- 2.3.2.1. Audio/Video synchronisation
- 2.3.2.2. Soundcard experiences, recommendations
- 2.3.2.3. Audio filters
  - 2.3.2.3.1. Up/Downsampling
  - 2.3.2.3.2. Changing the number of channels
  - 2.3.2.3.3. Sample format converter
  - 2.3.2.3.4. Delay
  - 2.3.2.3.5. Software volume control
  - 2.3.2.3.6. Equalizer
  - 2.3.2.3.7. Panning filter
  - 2.3.2.3.8. Sub-woofer
  - 2.3.2.3.9. Surround-sound decoder
  - 2.3.2.3.10. Audio Exporter
- 2.3.2.4. Audio plugins (deprecated)
  - 2.3.2.4.1. Up/Downsampling
  - 2.3.2.4.2. Surround Sound decoding
  - 2.3.2.4.3. Sample format converter
  - 2.3.2.4.4. Delay
  - 2.3.2.4.5. Software volume control
  - 2.3.2.4.6. Extrastereo
  - 2.3.2.4.7. Volume normalizer

## 2.4. TV input

2.4.1. Compilation

2.4.2. Usage tips

2.4.3. Examples

## 2.5. Edit Decision Lists (EDL)

2.5.1. Using an EDL file

2.5.2. Making an EDL file

# 2.1. Supported formats

It is important to clarify a common mistake. When people see a file with a .AVI extension, they immediately conclude that it is not an MPEG file. That is not true. At least not entirely. Contrary to popular belief such a file *can* contain MPEG1 video.

You see, a **codec** is not the same as a **file format**. Examples of video **codecs** are: MPEG1, MPEG2, DivX, Indeo5, 3ivx. Examples of file **formats** are: MPG, AVI, ASF.

In theory, you can put an OpenDivX video and MP3 audio into an **MPG format file**. However, most players will not play it, since they expect MPEG1 video and MP2 audio (unlike **AVI**, **MPG** does not have the necessary fields to describe its video and audio streams). Or you might put MPEG1 video into an AVI file. FFmpeg and MEncoder can create these files.

## 2.1.1. Video formats

### 2.1.1.1. MPEG files

MPEG files come in different guises:



## MPlayer – The Movie Player for LINUX

- **MPG:** This is the most **basic** form of the MPEG file formats. It contains MPEG1 video, and MP2 (MPEG–1 layer 2) or rarely MP1 audio.
- **DAT:** This is the very same format as MPG with a different extension. It is used on **Video CDs**. Due to the way VCDs are created and Linux is designed, the DAT files cannot be played nor copied from VCDs as regular files. You have to use `vcd://` to play a Video CD.
- **VOB:** This is the MPEG file format on **DVDs**. It is the same as MPG, plus the capability to contain subtitles or non–MPEG (AC3) audio. It contains encoded MPEG2 video and usually AC3 audio, but DTS, MP2 and uncompressed LPCM are allowed, too. **Read the DVD section!**

Series of frames form independent groups in MPEG files. This means that you can cut/join an MPEG file with standard file tools (like **dd**, **cut**), and it remains completely functional.

One important feature of MPGs is that they have a field to describe the aspect ratio of the video stream within. For example SVCDs have 480x480 resolution video, and in the header that field is set to 4:3, so that it is played at 640x480. AVI files do not have this field, so they have to be rescaled during encoding or played with the `-aspect` option.

### 2.1.1.2. AVI files

Designed by Microsoft, **AVI (Audio Video Interleaved)** is a widespread multipurpose format currently used mostly for DivX and DivX4 video. It has many known drawbacks and shortcomings (for example in streaming). It supports one video stream and 0 to 99 audio streams and can be as big as 2GB, but there exists an extension allowing bigger files called **OpenDML**. Microsoft currently strongly discourages its use and encourages ASF/WMV. Not that anybody cares.

There is a hack that allows AVI files to contain an Ogg Vorbis audio stream, but makes them incompatible with standard AVI. MPlayer supports playing these files. Seeking is also implemented but severely hampered by badly encoded files with confusing headers. Unfortunately the only encoder currently capable of creating these files, NanDub, has this problem.

### Note

DV cameras create raw DV streams that DV grabbing utilities convert to two different types of AVI files. The AVI will then contain either separate audio and video streams that MPlayer can play or the raw DV stream for which support is under development.

There are two kinds of AVI files:

- **Interleaved:** Audio and video content is interleaved. This is the standard usage. Recommended and mostly used. Some tools create interleaved AVIs with bad sync. MPlayer detects these as interleaved, and this climaxes in loss of A/V sync, probably at seeking. These files should be played as non–interleaved (with the `-ni` option).
- **Non–interleaved:** First comes the whole video stream, then the whole audio stream. It thus needs a lot of seeking, making playing from network or CD–ROM difficult.

MPlayer supports two kinds of timings for AVI files:

- **bps–based:** It is based on the bitrate/samplerate of the video/audio stream. This method is used by most players, including `avifile` and Windows Media Player. Files with broken headers, and files created with VBR audio but not VBR–compliant encoder will result in A/V desync with this method

## MPlayer – The Movie Player for LINUX

(mostly at seeking).

- **interleaving-based:** It does not use the bitrate value of the header, instead it uses the relative position of interleaved audio and video chunks, making badly encoded files with VBR audio playable.

Any audio and video codec is allowed, but note that VBR audio is not well supported by most players. The file format makes it possible to use VBR audio, but most players expect CBR audio, thus they fail with VBR. VBR is uncommon and Microsoft's AVI specs only describe CBR audio. I also noticed that most AVI encoders/multiplexers create bad files when using VBR audio. There are only two known exceptions: NanDub and MEncoder.

### 2.1.1.3. ASF/WMV files

ASF (Active Streaming Format) comes from Microsoft. They developed two variants of ASF, v1.0 and v2.0. v1.0 is used by their media tools (Windows Media Player and Windows Media Encoder) and is very secret. v2.0 is published and patented :). Of course they differ, there is no compatibility at all (it is just another legal game). MPlayer supports only v1.0, as nobody has ever seen v2.0 files :). Note that ASF files nowadays come with the extension `.WMA` or `.WMV`.

### 2.1.1.4. QuickTime/MOV files

These formats were designed by Apple and can contain any codec, CBR or VBR. They usually have a `.QT` or `.MOV` extension. Note that since the MPEG4 group chose QuickTime as the recommended file format for MPEG4, their MOV files come with a `.MPG` or `.MP4` extension (Interestingly the video and audio streams in these files are real MPG and AAC files. You can even extract them with the `-dumpvideo` and `-dumpaudio` options.).

## Note

Most new QuickTime files use **Sorenson** video and QDesign Music audio. See our Sorenson codec section.

### 2.1.1.5. VIVO files

MPlayer happily demuxes VIVO file formats. The biggest disadvantage of the format is that it has no index block, nor a fixed packet size or sync bytes and most files lack even keyframes, so forget seeking!

The video codec of VIVO/1.0 files is standard **h.263**. The video codec of VIVO/2.0 files is a modified, nonstandard **h.263v2**. The audio is the same, it may be **g.723 (standard)**, or **Vivo Siren**.

See the VIVO video codec and VIVO audio codec sections for installation instructions.

### 2.1.1.6. FLI files

**FLI** is a very old file format used by Autodesk Animator, but it is a common file format for short animations on the net. MPlayer demuxes and decodes FLI movies and is even able to seek within them (useful when looping with the `-loop` option). FLI files do not have keyframes, so the picture will be messy for a short time after seeking.

### 2.1.1.7. RealMedia (RM) files

Yes, MPlayer can read (demux) RealMedia (.rm) files. Seeking works, but you have to explicitly specify the `-forceidx` option (the format supports keyframes). Here are the lists of the supported RealVideo and RealAudio codecs.

### 2.1.1.8. NuppelVideo files

NuppelVideo is a TV grabber tool (AFAIK:). MPlayer can read its .NUV files (only NuppelVideo 5.0). Those files can contain uncompressed YV12, YV12+RTJpeg compressed, YV12 RTJpeg+lzo compressed, and YV12+lzo compressed frames. MPlayer decodes (and also **encodes** them with MEncoder to DivX/etc!) them all. Seeking works.

### 2.1.1.9. yuv4mpeg files

yuv4mpeg / yuv4mpeg2 is a file format used by the mjpegtools programs. You can grab, produce, filter or encode video in this format using these tools. The file format is really a sequence of uncompressed YUV 4:2:0 images.

### 2.1.1.10. FILM files

This format is used on old Sega Saturn CD-ROM games.

### 2.1.1.11. RoQ files

RoQ files are multimedia files used in some ID games such as Quake III and Return to Castle Wolfenstein.

### 2.1.1.12. OGG/OGM files

This is a new fileformat from Xiphophorus. It can contain any video or audio codec, CBR or VBR. You'll need `libogg` and `libvorbis` installed before compiling MPlayer to be able to play it.

### 2.1.1.13. SDP files

SDP is an IETF standard format for describing video and/or audio RTP streams. (The "LIVE.COM Streaming Media" are required.)

### 2.1.1.14. PVA files

PVA is an MPEG-like format used by DVB TV boards' software (e.g.: MultiDec, WinTV under Windows).

The PVA specifications can be downloaded from the following address:

[http://www.technotrend.de/download/av\\_format\\_v1.pdf](http://www.technotrend.de/download/av_format_v1.pdf)

### 2.1.1.15. GIF files

The **GIF** format is a common format for web graphics. There are two versions of the GIF spec, GIF87a and GIF89a. The main difference is that GIF89a allows for animation. MPlayer supports both formats through use of `libungif` or another libgif-compatible library. Non-animated GIFs will be displayed as single frame videos. (Use the `-loop` and `-fixed-vo` options to display these longer.)

## MPlayer – The Movie Player for LINUX

MPlayer currently does not support seeking in GIF files. GIF files do not necessarily have a fixed frame size, nor a fixed framerate. Rather, each frame is of independent size and is supposed to be positioned in a certain place on a field of fixed-size. The framerate is controlled by an optional block before each frame that specifies the next frame's delay in centiseconds.

Standard GIF files contain 24-bit RGB frames with at most an 8-bit indexed palette. These frames are usually LZW-compressed, although some GIF encoders produce uncompressed frames to avoid patent issues with LZW compression.

If your distribution does not come with `libungif`, download a copy from the [libungif homepage](#). For detailed technical information, have a look at the [GIF89a specification](#).

### 2.1.2. Audio formats

MPlayer is a **movie** and not a **media** player, although it can play some audio file formats (they are listed in the sections below). This is not a recommended usage of MPlayer, you better use [XMMS](#).

#### 2.1.2.1. MP3 files

You may have problems playing certain MP3 files that MPlayer will misdetect as MPEGs and play incorrectly or not at all. This cannot be fixed without dropping support for certain broken MPEG files and thus will remain like this for the foreseeable future. The `-demuxer` flag described in the man page may help you in these cases.

#### 2.1.2.2. WAV files

#### 2.1.2.3. OGG/OGM files (Vorbis)

Requires properly installed `libogg` and `libvorbis`.

#### 2.1.2.4. WMA/ASF files

#### 2.1.2.5. MP4 files

#### 2.1.2.6. CD audio

MPlayer can use `cdparanoia` to play CDDA (Audio CD). The scope of this section does not contain enumerating `cdparanoia`'s features.

See the man page's `-cdda` option which can be used to pass options to `cdparanoia`.

#### 2.1.2.7. XMMS

MPlayer can use XMMS input plugins to play many file formats. There are plugins for SNES game tunes, SID tunes (from Commodore 64), many Amiga formats, `.xm`, `.it`, VQF, musepack, Bonk, shorten and many others. You can find them at the [XMMS input plugin page](#).

For this feature you need to have XMMS and compile MPlayer with `./configure --enable-xmms`. If that does not work, you might need to set the XMMS plugin and library path explicitly by way of the `--with-xmmsplugindir` and `--with-xmmslibdir` options.

## 2.2. Supported codecs

### 2.2.1. Video codecs

See the [codec status table](#) for the complete, daily generated list. Quite a few codecs are available for download from our homepage. Grab them from our [codecs page](#).

The most important ones above all:

- **MPEG1 (VCD)** and **MPEG2 (DVD)** video
- native decoders for **DivX ;-)**, **OpenDivX (DivX4)**, **DivX 5.01**, **3ivX**, **M\$ MPEG4** v1, v2 and other MPEG4 variants
- native decoder for **Windows Media Video 7/8 (WMV1/WMV2)**, and Win32 DLL decoder for **Windows Media Video 9 (WMV3)**, both used in .wmv files
- native **Sorenson 1 (SVQ1)** decoder
- native **Sorenson 3 (SVQ3)** decoder
- **3ivx** v1, v2 decoder
- Cinepak and **Intel Indeo** codecs (3.1,3.2,4.1,5.0)
- **MJPEG**, **AVID**, **VCR2**, **ASV2** and other hardware formats
- **VIVO** 1.0, 2.0, **I263** and other **h263(+)** variants
- **FLI/FLC**
- **RealVideo 1.0 & 2.0** from libavcodec, and **RealVideo 3.0 & 4.0** codecs using RealPlayer libraries
- native decoder for HuffYUV
- Various old simple RLE-like formats

If you have a Win32 codec not listed here which is not supported yet, please read the [codec importing HOWTO](#) and help us add support for it.

#### 2.2.1.1. DivX4/DivX5

This section contains information about the DivX4 and DivX5 codecs of [Project Mayo](#). Their first available alpha version was OpenDivX 4.0 alpha 47 and 48. Support for this was included in MPlayer in the past, and built by default. We also used its postprocessing code to optionally enhance visual quality of MPEG1/2 movies. Now we use our own, for all file types.

The new generation of this codec is called DivX4 and can even decode movies made with the infamous DivX codec! In addition it is much faster than the native Win32 DivX DLLs but slower than libavcodec. Hence its usage as a decoder is **DISCOURAGED**. However, it is useful for encoding. One disadvantage of this codec is that it is not available under an Open Source license.

DivX4 works in two modes:

`-vc odivx`

Uses the codec in OpenDivX fashion. In this case it produces YV12 images in its own buffer, and MPlayer does colorspace conversion via libvo. (**Fast, recommended!**)

`-vc divx4`

Uses the colorspace conversion of the codec. In this mode you can use YUY2/UYYV, too. (**SLOW**)

The `-vc odivx` method is usually faster, due to the fact that it transfers image data in YV12 (planar YUV 4:2:0) format, thus requiring much less bandwidth on the bus. For packed YUV modes (YUY2, UYYV) use

## MPlayer – The Movie Player for LINUX

the `-vc divx4` method. For RGB modes the speed is the same, differing at best according to your current color depth.

### Note

If your `-vo` driver supports direct rendering, then `-vc divx4` may be faster, or even the fastest solution.

The Divx4/5 binary codec library can be downloaded from [avifile](#) or [divx.com](#) Unpack it, run `./install.sh` as root and do not forget adding `/usr/local/lib` to your `/etc/ld.so.conf` and running `ldconfig`.

Get the CVS version of the OLD OpenDivx core library like this:

1.

```
cvs -d:pserver:anonymous@cvs.projectmajo.com:/cvsroot login
```

•

```
cvs -d:pserver:anonymous@cvs.projectmajo.com:/cvsroot co divxcore
```

- This core library is split into a `dec` and `enc` library that have to be compiled separately. For the `dec` library, simply type

```
cd divxcore/dec/build/linux
make
cp libdivxdec.so /usr/local/lib
ln -s libdivxdec.so /usr/local/lib/libdivxdec.so.0
cp ../../src/dec.h /usr/local/include
```

- Alas, for the `enc` library there is no Linux Makefile available, and the MMX optimized code only works on Windows. You can still compile it, though, by using this [Makefile](#).

```
cd ../../../../enc/build
mkdir linux
cd linux
cp path/Makefile .
make
cp libdivxenc.so /usr/local/lib
ln -s libdivxenc.so /usr/local/lib/libdivxenc.so.0
cp ../../src/enc.h /usr/local/include
```

MPlayer autodetects DivX4/DivX5 if it is properly installed, just compile as usual. If it does not detect it, you did not install or configure it correctly.

### 2.2.1.2. FFmpeg/libavcodec

[FFmpeg](#) contains an **open source** codec package, which is capable of decoding streams with various audio and video codecs. It also offers an impressive encoding facility and higher speed than the Win32 codecs or the DivX.com DivX4/5 libraries!

It contains a lot of nice codecs, especially important are the MPEG4 variants: DivX3, DivX4, DivX5, Windows Media Video 7/8 (WMV1/WMV2). Also a very interesting one is the WMA decoder.

The most recent codec deserving credit is the **Sorenson 3** (SVQ3) codec. This is the first, completely opensource implementation. It's even faster than the original. Be sure to prefer this instead of the binary codec!

## MPlayer – The Movie Player for LINUX

For a complete list of supported codecs please visit the FFmpeg homepage. Supported video and audio codecs.

If you use an MPlayer release you have libavcodec right in the package, just build as usual. If you use MPlayer from CVS you have to extract libavcodec from the FFmpeg CVS tree as FFmpeg releases are very rare. The CVS is mostly stable and offers the most features. In order to achieve this do:

1. 

```
cvs -d:pserver:anonymous@mplayerhq.hu:/cvsroot/ffmpeg login
```
2. 

```
cvs -d:pserver:anonymous@mplayerhq.hu:/cvsroot/ffmpeg co ffmpeg
```
3. Move the `libavcodec` directory from the FFmpeg sources to the root of the MPlayer CVS tree. It should look like this: `main/libavcodec`  
  
Symlinking is **not** enough, you have to copy/move it!
4. If you prefer having libavcodec updated at the same time you update MPlayer CVS, put the following line into the `main/CVS/Entries` file:  
  

```
D/libavcodec////
```
5. Compile. `configure` should detect problems before compilation.

### Note

MPlayer from CVS does contain a `libavcodec` subdirectory, but it does **not** contain the source for libavcodec! You must follow the steps above to obtain the source for this library.

With FFmpeg and my Matrox G400, I can view even the highest resolution DivX movies without dropped frames on my K6-2 500.

### 2.2.1.3. XAnim codecs

### Note

Be advised that the XAnim binary codecs are packaged with a piece of text claiming to be a legally binding software license which, besides other restrictions, forbids the user to use the codecs in conjunction with any program other than XAnim. However, the XAnim author has yet to bring legal action against anyone for codec-related issues.

**INSTALLATION AND USAGE.** MPlayer is capable of employing the XAnim codecs for decoding. Follow the instructions to enable them:

1. Download the codecs you wish to use from the XAnim site. The **3ivx** codec is not there, but at the 3ivx site.
2. **OR** download the codecs pack from our codecs page
3. Use the `--with-xanimlibdir` option to tell `configure` where to find the XAnim codecs. By default, it looks for them at `/usr/local/lib/codecs`, `/usr/local/lib/xanim/mods`, `/usr/lib/xanim/mods` and `/usr/lib/xanim`. Alternatively you can set the environment variable `XANIM_MOD_DIR` to the directory of the XAnim codecs.
4. Rename/symlink the files, cutting out the architecture stuff, so they will have filenames like these:  
`vid_cvid.xa`, `vid_h263.xa`, `vid_iv50.xa`

## MPlayer – The Movie Player for LINUX

XAnim is video codec family xanim, so you may want to use the `-vfm xanim` option to tell MPlayer to use them if possible.

Tested codecs include: **Indeo 3.2, 4.1, 5.0, CVID, 3ivX, h263.**

### 2.2.1.4. VIVO video

MPlayer can play Vivo (1.0 and 2.0) videos. The most suitable codec for 1.0 files is FFmpeg's H263 decoder, you can use it with the `-vc ffh263` option. For 2.0 files, use the Win32 DLL through the `-vc vivo` option. If you do not supply command line options MPlayer selects the best codec automatically.

### 2.2.1.5. MPEG 1/2 video

MPEG1 and MPEG2 are decoded by the multiplatform native `libmpeg2` library, whose source code is included in MPlayer. We handle buggy MPEG 1/2 video files by catching `Signal 11 (segmentation fault)`, and quickly reinitializing the codec, continuing exactly from where the failure occurred. This recovery technique has no measurable speed penalty.

### 2.2.1.6. MS Video1

This is a very old and very bad codec from Microsoft. In the past it was decoded with the `msvidc32.dll` Win32 codec, now we have our own open source implementation (by [Mike Melanson](#)).

### 2.2.1.7. Cinepak CVID

MPlayer uses its own open source, multiplatform Cinepak decoder (by [Dr. Tim Ferguson](#)) by default. It supports YUV outputs, so that hardware scaling is possible if the video output driver permits it.

### 2.2.1.8. RealVideo

MPlayer supports decoding all versions of RealVideo:

- RealVideo 1.0, 2.0 (fourcc RV10, RV20) – en/decoding supported by `libavcodec`
- RealVideo 3.0, 4.0 (fourcc RV30, RV40) – decoding supported by **RealPlayer libraries**

It is recommended to download and install RealPlayer8 or RealONE, because MPlayer can use their libraries to decode files with RealVideo 2.0 – 4.0 video. The MPlayer `configure` script should detect the RealPlayer libraries in the standard locations of a full installation. If it does not, tell `configure` where to look with the `--with-reallibdir` option.

As a second chance you can download the codecs from <http://www.mplayerhq.hu/MPlayer/releases/codecs/>. After downloading the Real codecs package, extract it to the `/usr/local/lib/codecs` directory, or you can specify a unique path with the `--with-reallibdir` option.

## Note

RealPlayer libraries currently **only work with Linux, FreeBSD, NetBSD and Cygwin on the x86, Alpha and PowerPC (Linux/Alpha and Linux/PowerPC have been tested) platforms.**



### 2.2.1.9. XviD

XviD is a forked development of the OpenDivX codec. It happened when ProjectMayo changed OpenDivX to closed source DivX4 (now DivX5), and the non-ProjectMayo people working on OpenDivX got angry, then started XviD. So both projects have the same origin.

#### ADVANTAGES

- open source
- its API is compatible with DivX4 so adding support for it is easy
- 2-pass encoding support
- nice encoding quality, higher speed than DivX4 (you can optimize it for your box while compiling)

#### DISADVANTAGES

- currently it does not properly **decode** all DivX/DivX4 files (no problem as libavcodec can play them)
- under development

#### INSTALLING XVID CVS

It is currently available only from CVS. Here are download and installation instructions (you need at least autoconf 2.50, automake and libtool):

1. 

```
cvs -z3 -d:pserver:anonymous@cvs.xvid.org:/xvid login
```
2. 

```
cvs -z3 -d:pserver:anonymous@cvs.xvid.org:/xvid co xvidcore
```
3. 

```
cd xvidcore/build/generic
```
4. 

```
./bootstrap.sh
```
5. 

```
./configure
```

You may have to add some options (examine the output of

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

6. 

```
)  
make && make install
```
7. If you specified `--enable-divxcompat`, copy `../src/divx4.h` to `/usr/local/include/`.
8. Recompile MPlayer with `--with-xvidcore=/path/to/libxvidcore.a`.

### 2.2.1.10. Sorenson

Sorenson is a video codec developed by Sorenson Media and licensed to Apple who distribute it with their QuickTime Player. We are currently able to decode all version of Sorenson video files with the following decoders:

- Sorenson Video v1 (fourcc *SVQ1*) – decoding supported by **native codec** (libavcodec)

## MPlayer – The Movie Player for LINUX

- Sorenson Video v3 (fourcc *SVQ3*) – decoding supported by **native codec** ([libavcodec](#))

### COMPILING MPLAYER WITH QUICKTIME LIBRARIES SUPPORT

#### Note

currently only 32bit Intel platforms are supported.

1. download MPlayer CVS
2. download QuickTime DLL pack from <http://www.mplayerhq.hu/MPlayer/releases/codecs/>
3. extract QuickTime DLL pack to your Win32 codecs directory (default:  
/usr/local/lib/codecs/)
4. compile MPlayer

### 2.2.2. Audio codecs

#### The most important audio codecs above all:

- MPEG layer 1/2/3 (MP1/2/3) audio (**native code**, with MMX/SSE/3DNow! optimization)
- Windows Media Audio 7 and 8 (aka WMAv1 and WMAv2) (**native code**, with [libavcodec](#))
- Windows Media Audio 9 (WMAv3) (using DMO DLL)
- AC3 Dolby audio (**native code**, with MMX/SSE/3DNow! optimization)
- AC3 passing through soundcard hardware
- AAC
- Ogg Vorbis audio codec (**native library**)
- RealAudio: DNET (low bitrate AC3), Cook, Sipro and ATRAC3
- QuickTime: Qualcomm and QDesign audio codecs
- VIVO audio (g723, Vivo Siren)
- Voxware audio (using DirectShow DLL)
- alaw and ulaw, various gsm, adpcm and pcm formats and other simple old audio codecs

#### 2.2.2.1. Software AC3 decoding

This is the default decoder used for files with AC3 audio.

The AC3 decoder can create audio output mixes for 2, 4, or 6 speakers. When configured for 6 speakers, this decoder provides separate output of all the AC3 channels to the sound driver, allowing for full "surround sound" experience without the external AC3 decoder required to use the hwac3 codec.

Use the `-channels` option to select the number of output channels. Use `-channels 2` for a stereo downmix. For a 4 channel downmix (Left Front, Right Front, Left Surround and Right Surround outputs), use `-channels 4`. In this case, any center channel will be mixed equally to the front channels. `-channels 6` will output all the AC3 channels as they are encoded – in the order Left, Right, Left Surround, Right Surround, Center and LFE.

The default number of output channels is 2.

To use more than 2 output channels, you will need to use OSS, and have a sound card that supports the appropriate number of output channels via the `SNDCCTL_DSP_CHANNELS` ioctl. An example of a suitable driver is `emu10k1` (used by Soundblaster Live! cards) from August 2001 or newer (ALSA CVS is also

supposed to work).

### 2.2.2.2. Hardware AC3 decoding

You need an AC3 capable sound card, with digital out (SP/DIF). The card's driver must properly support the AFMT\_AC3 format (C-Media does). Connect your AC3 decoder to the SP/DIF output, and use the `-ac hwac3` option. It is experimental but known to work with C-Media cards and Soundblaster Live! + ALSA (but not OSS) drivers and DXR3/Hollywood+ MPEG decoder cards.

### 2.2.2.3. libmad support

[libmad](#) is a multiplatform, integer (internally 24bit PCM) only MPEG audio decoding library. It does not handle broken files well, and it sometimes has problems with seeking, but it may perform better on FPU-less (such as [ARM](#)) platform than mp3lib.

To enable support, compile with the `--enable-mad` configure option.

### 2.2.2.4. VIVO audio

The audio codec used in VIVO files depends on whether it is a VIVO/1.0 or VIVO/2.0 file. VIVO/1.0 files have **g.723** audio, and VIVO/2.0 files have **Vivo Siren** audio. Both are supported.

### 2.2.2.5. RealAudio

MPlayer supports decoding nearly all versions of RealAudio:

- RealAudio DNET – decoding supported by `liba52`
- RealAudio Cook/Sipro/ATRAC3 – decoding supported by **RealPlayer libraries**

On how to install RealPlayer libraries, see the [RealMedia file format](#) section.

### 2.2.2.6. QDesign codecs

QDesign audio streams (fourcc:*QDMC*, *QDM2*) are found in MOV/QT files. Both versions of this codec can be decoded with QuickTime libraries. For installation instructions please see the [Sorenson video codec](#) section.

### 2.2.2.7. Qualcomm codecs

Qualcomm audio streams (fourcc:*Qclp*) is found in MOV/QT files. It can be decoded with QuickTime libraries. For installation instructions please see the [Sorenson video codec](#) section.

### 2.2.2.8. AAC codec

AAC (Advanced Audio Coding) is an audio codec sometimes found in MOV and MP4 files. An open source decoder called FAAD is available from <http://www.audiocoding.com>. MPlayer includes libfaad 2.0RC1, so you do not need to get it separately.

If you use gcc 3.2 which fails to compile our internal FAAD or want to use the external library for some other reason, download the library from the [download page](#) and pass `--with-externalfaad` to configure. You do not need all of faad2 to decode AAC files, libfaad is enough. Build it like this:

## MPlayer – The Movie Player for LINUX

```
cd faad2/  
chmod +x bootstrap  
./bootstrap  
./configure  
cd libfaad  
make  
make install
```

Binaries are not available from audiocoding.com, but you can (apt-)get Debian packages from [Christian Marillat's homepage](#), Mandrake RPMs from the [P.L.F](#) and RedHat RPMs from [Dominik Mierzejewski's homepage](#).

### 2.2.3. Win32 codecs importing HOWTO

#### 2.2.3.1. VFW codecs

VFW (Video for Windows) is the old Video API for Windows. Its codecs have the .DLL or (rarely) .DRV extension. If MPlayer fails at playing your AVI with this kind of message:

```
UNKNOWN video codec: HFYU (0x55594648)
```

It means your AVI is encoded with a codec which has the HFYU fourcc (HFYU = HuffYUV codec, DIV3 = DivX Low Motion, etc.). Now that you know this, you have to find out which DLL Windows loads in order to play this file. In our case, the `system.ini` contains this information in a line that reads:

```
VIDC.HFYU=huffyuv.dll
```

So you need the `huffyuv.dll` file. Note that the audio codecs are specified by the MSACM prefix:

```
msacm.l3acm=L3codeca.acm
```

This is the MP3 codec. Now that you have all the necessary information (fourcc, codec file, sample AVI), submit your codec support request by mail, and upload these files to the FTP site:

```
ftp://ftp.mplayerhq.hu/MPlayer/incoming/[codecname]/
```

### Note

On Windows NT/2000/XP search for this info in the registry, e.g. search for "VIDC.HFYU". To find out how to do this, look at the old DirectShow method below.

#### 2.2.3.2. DirectShow codecs

DirectShow is the newer Video API, which is even worse than its predecessor. Things are harder with DirectShow, since

- `system.ini` does not contain the needed information, instead it is stored in the registry and
- we need the GUID of the codec.

#### New Method:

Using Microsoft GraphEdit (fast)

1. Get GraphEdit from either DirectX SDK or [doom9](#)

2. Start **graphedit.exe**.
3. From the menu select Graph -> Insert Filters.
4. Expand item `DirectShow Filters`
5. Select the right codec name and expand item.
6. In the entry `DisplayName` look at the text in winged brackets after the backslash and write it down (five dash-delimited blocks, the GUID).
7. The codec binary is the file specified in the `Filename` entry.

## Note

If there is no `Filename` and `DisplayName` contains something like `device:dmo`, then it is a DMO-Codec.

### Old Method:

Take a deep breath and start searching the registry...

1. Start **regedit**.
2. Press **Ctrl+F**, disable the first two checkboxes, and enable the third. Type in the fourcc of the codec (e.g. **TM20**).
3. You should see a field which contains the path and the filename (e.g. `C:\WINDOWS\SYSTEM\TM20DEC.AX`).
4. Now that you have the file, we need the GUID. Try searching again, but now search for the codec's name, not the fourcc. Its name can be acquired when Media Player is playing the file, by checking File -> Properties -> Advanced. If not, you are out of luck. Try guessing (e.g. search for TrueMotion).
5. If the GUID is found you should see a `FriendlyName` and a `CLSID` field. Write down the 16 byte CLSID, this is the GUID we need.

## Note

If searching fails, try enabling all the checkboxes. You may have false hits, but you may get lucky...

Now that you have all the necessary information (fourcc, GUID, codec file, sample AVI), submit your codec support request by mail, and upload these files to the FTP site:

```
ftp://ftp.mplayerhq.hu/MPlayer/incoming/[codecname]/
```

## 2.3. Output devices

### 2.3.1. Video output devices

#### 2.3.1.1. Setting up MTRR

It is VERY recommended to check if the MTRR registers are set up properly, because they can give a big performance boost.

Do a **cat /proc/mtrr**:

```
--($:~)-- cat /proc/mtrr  
reg00: base=0xe4000000 (3648MB), size= 16MB: write-combining, count=9
```

Note

## MPlayer – The Movie Player for LINUX

```
reg01: base=0xd8000000 (3456MB), size= 128MB: write-combining, count=1
```

It's right, shows my Matrox G400 with 16MB memory. I did this from XFree 4.x.x , which sets up MTRR registers automatically.

If nothing worked, you have to do it manually. First, you have to find the base address. You have 3 ways to find it:

1. from X11 startup messages, for example:

```
(--) SVGA: PCI: Matrox MGA G400 AGP rev 4, Memory @ 0xd8000000, 0xd4000000
(--) SVGA: Linear framebuffer at 0xD8000000
```

- from `/proc/pci` (use **lspci -v** command):

```
01:00.0 VGA compatible controller: Matrox Graphics, Inc.: Unknown device 0525
Memory at d8000000 (32-bit, prefetchable)
```

- from `mga_vid` kernel driver messages (use **dmesg**):

```
mga_mem_base = d8000000
```

Then let's find the memory size. This is very easy, just convert video RAM size to hexadecimal, or use this table:

1 MB	0x100000
2 MB	0x200000
4 MB	0x400000
8 MB	0x800000
16 MB	0x1000000
32 MB	0x2000000

You know base address and memory size, let's setup MTRR registers! For example, for the Matrox card above (base=0xd8000000) with 32MB ram (size=0x2000000) just execute:

```
echo "base=0xd8000000 size=0x2000000 type=write-combining" >| /proc/mtrr
```

Not all CPUs support MTRRs. For example older K6-2's (around 266MHz, stepping 0) doesn't support MTRR, but stepping 12's do (**cat /proc/cpuinfo** to check it).

### 2.3.1.2. Video outputs for traditional video cards

#### 2.3.1.2.1. Xv

Under XFree86 4.0.2 or newer, you can use your card's hardware YUV routines using the XVideo extension. This is what the option `'-vo xv'` uses. Also, this driver supports adjusting brightness/contrast/hue/etc (unless you use the old, slow DirectShow DivX codec, which supports it everywhere), see the man page.

In order to make this work, be sure to check the following:

1. You have to use XFree86 4.0.2 or newer (former versions don't have XVideo)
2. Your card actually supports hardware acceleration (modern cards do)
3. X loads the XVideo extension, it's something like this:

## MPlayer – The Movie Player for LINUX

(II) Loading extension XVideo

in /var/log/XFree86.0.log

### Note

This loads only the XFree86's extension. In a good install, this is always loaded, and doesn't mean that the **card's** XVideo support is loaded!

- Your card has Xv support under Linux. To check, try **xvinfo**, it is the part of the XFree86 distribution. It should display a long text, similar to this:

```
X-Video Extension version 2.2
screen #0
  Adaptor #0: "Savage Streams Engine"
    number of ports: 1
    port base: 43
    operations supported: PutImage
    supported visuals:
      depth 16, visualID 0x22
      depth 16, visualID 0x23
    number of attributes: 5
  (...)
  Number of image formats: 7
    id: 0x32595559 (YUY2)
      guid: 59555932-0000-0010-8000-00aa00389b71
      bits per pixel: 16
      number of planes: 1
      type: YUV (packed)
    id: 0x32315659 (YV12)
      guid: 59563132-0000-0010-8000-00aa00389b71
      bits per pixel: 12
      number of planes: 3
      type: YUV (planar)
  (...etc...)
```

It must support YUY2 packed, and YV12 planar pixel formats to be usable with MPlayer.

- And finally, check if MPlayer was compiled with 'xv' support. Do a **mplayer -vo help | grep xv** . If 'xv' support was built a line similar to this should appear:

```
      xv      X11/Xv
2.3.1.2.1.1. 3dfx cards
```

Older 3dfx drivers were known to have problems with XVideo acceleration, it didn't support either YUY2 or YV12, and so. Verify that you have XFree86 version 4.2.0 or greater, it works OK with YV12 and YUY2. Previous versions, including 4.1.0, **crashes with YV12**. If you experience strange effects using `-vo xv`, try SDL (it has XVideo, too) and see if it helps. Check the [SDL](#) section for details.

**OR**, try the NEW `-vo tdfxfb` driver! See the [tdfxfb](#) section.

#### 2.3.1.2.1.2. S3 cards

S3 Savage3D's should work fine, but for Savage4, use XFree86 version 4.0.3 or greater (in case of image problems, try 16bpp). As for S3 Virge: there is xv support, but the card itself is very slow, so you better sell it.

## Note

It's currently unclear which Savage models lack YV12 support, and convert by driver (slow). If you suspect your card, get a newer driver, or ask politely on the mplayer-users mailing list for an MMX/3DNow enabled driver.

### 2.3.1.2.1.3. nVidia cards

nVidia isn't always a very good choice under Linux (according to nVidia, this is not true)... XFree86's open-source driver supports most of these cards, but for some cases, you'll have to use the binary closed-source nVidia driver, available at [nVidia's web site](#). You'll always need this driver if you want 3D acceleration, too.

Rival128 cards don't have XVideo support with XFree86's nVidia driver :( Complain to nVidia.

However, MPlayer contains a VIDIX driver for most nVidia cards. Currently it is in beta stage, and has some drawbacks. For more information, see [nVidia VIDIX](#) section.

### 2.3.1.2.1.4. ATI cards

The GATOS driver (which you should use, unless you have Rage128 or Radeon) has VSYNC enabled by default. It means that decoding speed (!) is synced to the monitor's refresh rate. If playing seems to be slow, try disabling VSYNC somehow, or set refresh rate to a n\*(fps of the movie) Hz.

Radeon VE – if you need X, use XFree86 4.2.0 or greater for this card. No TV out support. Of course with MPlayer you can happily get **accelerated** display, with or without **TV output**, and no libraries or X are needed. Read the VIDIX section.

### 2.3.1.2.1.5. NeoMagic cards

These cards can be found in many laptops. You must use XFree86 4.3.0 or above, or else use Stefan Seyfried's Xv-capable drivers. Just choose the one that applies to your version of XFree86.

XFree86 4.3.0 includes Xv support, yet Bohdan Horst sent a small patch against the XFree86 sources that speeds up framebuffer operations (so XVideo) up to four times. The patch has been included in XFree86 CVS and should be in the next release after 4.3.0.

To allow playback of DVD sized content change your XF86Config like this:

```
Section "Device"
    [...]
    Driver "neomagic"
    Option "OverlayMem" "829440"
    [...]
EndSection
```

### 2.3.1.2.1.6. Trident cards

If you want to use xv with a trident card, provided that it doesn't work with 4.1.0, install XFree 4.2.0. 4.2.0 adds support for fullscreen Xv support with the Cyberblade XP card.

Alternatively, MPlayer contains a VIDIX driver for the Cyberblade/i1 card.



## MPlayer – The Movie Player for LINUX

### 2.3.1.2.1.7. Kyro/PowerVR cards

If you want to use Xv with a Kyro based card (for example Hercules Prophet 4000XT), you should download the drivers from the [PowerVR site](#)

### 2.3.1.2.2. DGA

**PREAMBLE.** This document tries to explain in some words what DGA is in general and what the DGA video output driver for MPlayer can do (and what it can't).

**WHAT IS DGA.** DGA is short for *Direct Graphics Access* and is a means for a program to bypass the X server and directly modifying the framebuffer memory. Technically spoken this happens by mapping the framebuffer memory into the memory range of your process. This is allowed by the kernel only if you have superuser privileges. You can get these either by logging in as `root` or by setting the SUID bit on the MPlayer executable (**not recommended**).

There are two versions of DGA: DGA1 is used by XFree 3.x.x and DGA2 was introduced with XFree 4.0.1.

DGA1 provides only direct framebuffer access as described above. For switching the resolution of the video signal you have to rely on the XVidMode extension.

DGA2 incorporates the features of XVidMode extension and also allows switching the depth of the display. So you may, although basically running a 32 bit depth X server, switch to a depth of 15 bits and vice versa.

However DGA has some drawbacks. It seems it is somewhat dependent on the graphics chip you use and on the implementation of the X server's video driver that controls this chip. So it does not work on every system...

**INSTALLING DGA SUPPORT FOR MPLAYER.** First make sure X loads the DGA extension, see in `/var/log/XFree86.0.log`:

```
(II) Loading extension XFree86-DGA
```

See, XFree86 4.0.x or greater is **highly recommended!** MPlayer's DGA driver is autodetected by `./configure`, or you can force it with `--enable-dga`.

If the driver couldn't switch to a smaller resolution, experiment with options `-vm` (only with X 3.3.x), `-fs`, `-bpp`, `-zoom` to find a video mode that the movie fits in. There is no converter right now :(

Become `root`. DGA needs root access to be able to write directly video memory. If you want to run it as user, then install MPlayer SUID root:

```
chown root /usr/local/bin/mplayer
chmod 750 /usr/local/bin/mplayer
chmod +s /usr/local/bin/mplayer
```

Now it works as a simple user, too.

## Security risk

This is a **big** security risk! **Never** do this on a server or on a computer that can be accessed by other people because they can gain root privileges through SUID root MPlayer.

## MPlayer – The Movie Player for LINUX

Now use `-vo dga` option, and there you go! (hope so:) You should also try if the `-vo sdl:dga` option works for you! It's much faster!

**RESOLUTION SWITCHING.** The DGA driver allows for switching the resolution of the output signal. This avoids the need for doing (slow) software scaling and at the same time provides a fullscreen image. Ideally it would switch to the exact resolution (except for honoring aspect ratio) of the video data, but the X server only allows switching to resolutions predefined in `/etc/X11/XF86Config` (`/etc/X11/XF86Config-4` for XFree 4.X.X respectively). Those are defined by so-called modelines and depend on the capabilities of your video hardware. The X server scans this config file on startup and disables the modelines not suitable for your hardware. You can find out which modes survive with the X11 log file. It can be found at: `/var/log/XFree86.0.log`.

These entries are known to work fine with a Riva128 chip, using the `nv.o` X server driver module.

```
Section "Modes"
  Identifier "Modes[0]"
  Modeline "800x600" 40      800 840 968 1056   600 601 605 628
  Modeline "712x600" 35.0    712 740 850 900    400 410 412 425
  Modeline "640x480" 25.175  640 664 760 800    480 491 493 525
  Modeline "400x300" 20      400 416 480 528    300 301 303 314 Doublescan
  Modeline "352x288" 25.10   352 368 416 432    288 296 290 310
  Modeline "352x240" 15.750  352 368 416 432    240 244 246 262 Doublescan
  Modeline "320x240" 12.588  320 336 384 400    240 245 246 262 Doublescan
EndSection
```

**DGA & MPLAYER.** DGA is used in two places with MPlayer: The SDL driver can be made to make use of it (`-vo sdl:dga`) and within the DGA driver (`-vo dga`). The above said is true for both; in the following sections I'll explain how the DGA driver for MPlayer works.

**FEATURES.** The DGA driver is invoked by specifying `-vo dga` at the command line. The default behavior is to switch to a resolution matching the original resolution of the video as close as possible. It deliberately ignores the `-vm` and `-fs` options (enabling of video mode switching and fullscreen) – it always tries to cover as much area of your screen as possible by switching the video mode, thus refraining to use a single additional cycle of your CPU to scale the image. If you don't like the mode it chooses you may force it to choose the mode matching closest the resolution you specify by `-x` and `-y`. By providing the `-v` option, the DGA driver will print, among a lot of other things, a list of all resolutions supported by your current `XF86Config` file. Having DGA2 you may also force it to use a certain depth by using the `-bpp` option. Valid depths are 15, 16, 24 and 32. It depends on your hardware whether these depths are natively supported or if a (possibly slow) conversion has to be done.

If you should be lucky enough to have enough offscreen memory left to put a whole image there, the DGA driver will use doublebuffering, which results in much smoother movie replaying. It will tell you whether double-buffering is enabled or not.

Doublebuffering means that the next frame of your video is being drawn in some offscreen memory while the current frame is being displayed. When the next frame is ready, the graphics chip is just told the location in memory of the new frame and simply fetches the data to be displayed from there. In the meantime the other buffer in memory will be filled again with new video data.

Doublebuffering may be switched on by using the option `-double` and may be disabled with `-nodouble`. Current default option is to disable doublebuffering. When using the DGA driver, onscreen display (OSD) only works with doublebuffering enabled. However, enabling doublebuffering may result in a big speed penalty (on my K6-II+ 525 it used an additional 20% of CPU time!) depending on the implementation of

## MPlayer – The Movie Player for LINUX

DGA for your hardware.

**SPEED ISSUES.** Generally spoken, DGA framebuffer access should be at least as fast as using the X11 driver with the additional benefit of getting a fullscreen image. The percentage speed values printed by MPlayer have to be interpreted with some care, as for example, with the X11 driver they do not include the time used by the X server needed for the actual drawing. Hook a terminal to a serial line of your box and start **top** to see what is really going on in your box.

Generally spoken, the speedup done by using DGA against 'normal' use of X11 highly depends on your graphics card and how well the X server module for it is optimized.

If you have a slow system, better use 15 or 16 bit depth since they require only half the memory bandwidth of a 32 bit display.

Using a depth of 24 bit is even a good idea if your card natively just supports 32 bit depth since it transfers 25% less data compared to the 32/32 mode.

I've seen some AVI files already be replayed on a Pentium MMX 266. AMD K6-2 CPUs might work at 400 MHZ and above.

**KNOWN BUGS.** Well, according to some developers of XFree, DGA is quite a beast. They tell you better not to use it. Its implementation is not always flawless with every chipset driver for XFree out there.

- With XFree 4.0.3 and `nv.o` there is a bug resulting in strange colors.
- ATI driver requires to switch mode back more than once after finishing using of DGA.
- Some drivers simply fail to switch back to normal resolution (use **Ctrl+Alt+Keypad +** and **Ctrl+Alt+Keypad -** to switch back manually).
- Some drivers simply display strange colors.
- Some drivers lie about the amount of memory they map into the process's address space, thus `vo_dga` won't use doublebuffering (SIS?).
- Some drivers seem to fail to report even a single valid mode. In this case the DGA driver will crash telling you about a nonsense mode of 100000x100000 or something like that.
- OSD only works with doublebuffering enabled (else it flickers).

### 2.3.1.2.3. SDL

SDL (Simple Directmedia Layer) is basically a unified video/audio interface. Programs that use it know only about SDL, and not about what video or audio driver does SDL actually use. For example a Doom port using SDL can run on `svgalib`, `aalib`, `X`, `fbdev`, and others, you only have to specify the (for example) video driver to use with the `SDL_VIDEODRIVER` environment variable. Well, in theory.

With MPlayer, we used its X11 driver's software scaler ability for cards/drivers that doesn't support XVideo, until we made our own (faster, nicer) software scaler. Also we used its `aalib` output, but now we have ours which is more comfortable. Its DGA mode was better than ours, until recently. Get it now? :)

It also helps with some buggy drivers/cards if the video is jerky (not slow system problem), or audio is lagging.

SDL video output supports displaying subtitles under the movie, on the (if present) black bar.

**There are several command line switches for SDL:**

## MPlayer – The Movie Player for LINUX

- `-vo sdl:name`  
specifies SDL video driver to use (i.e. aalib, dga, x11)
- `-ao sdl:name`  
specifies SDL audio driver to use (i.e. dsp, esd, arts)
- `-noxv`  
disables XVideo hardware acceleration
- `-forcexv`  
tries to force XVideo acceleration

**Table 2.1. SDL only keys**

Key	Action
<b>c</b>	cycles available fullscreen modes
<b>n</b>	changes back to normal mode

### Known bugs:

- Keys pressed under sdl:aalib console driver repeat forever. (use `-vo aa!`) It's bug in SDL, I can't change it (tested with SDL 1.2.1).
- DO NOT USE SDL with GUI! It won't work as it should.

### 2.3.1.2.4. SVGAlib

**INSTALLATION.** You'll have to install `svgalib` and its development package in order for MPlayer build its SVGAlib driver (autodetected, but can be forced), and don't forget to edit `/etc/vga/libvga.config` to suit your card and monitor.

### Note

Be sure not to use the `-fs` switch, since it toggles the usage of the software scaler, and it's slow. If you really need it, use the `-sws 4` option which will produce bad quality, but is somewhat faster.

**EGA (4BPP) SUPPORT.** SVGAlib incorporates EGAlib, and MPlayer has the possibility to display any movie in 16 colors, thus usable in the following sets:

- EGA card with EGA monitor: 320x200x4bpp, 640x200x4bpp, 640x350x4bpp
- EGA card with CGA monitor: 320x200x4bpp, 640x200x4bpp

The bpp (bits per pixel) value must be set to 4 by hand: `-bpp 4`

The movie probably must be scaled down to fit in EGA mode:

```
-vf scale=640:350
```

or

```
-vf scale=320:200
```

For that we need fast but bad quality scaling routine:

```
-sws 4
```

## MPlayer – The Movie Player for LINUX

Maybe automatic aspect correction has to be shut off:

```
-noaspect
```

### Note

According to my experience the best image quality on EGA screens can be achieved by decreasing the brightness a bit: `-vf eq=-20:0`. I also needed to lower the audio samplerate on my box, because the sound was broken on 44kHz: `-srate 22050`.

You can turn on OSD and subtitles only with the `expand` filter, see the man page for exact parameters.

#### 2.3.1.2.5. Framebuffer output (FBdev)

Whether to build the FBdev target is autodetected during `./configure`. Read the framebuffer documentation in the kernel sources (`Documentation/fb/*`) for more information.

If your card doesn't support VBE 2.0 standard (older ISA/PCI cards, such as S3 Trio64), only VBE 1.2 (or older?): Well, VESAfb is still available, but you'll have to load SciTech Display Doctor (formerly UniVBE) before booting Linux. Use a DOS boot disk or whatever. And don't forget to register your UniVBE ;))

The FBdev output takes some additional parameters above the others:

```
-fb
    specify the framebuffer device to use (/dev/fb0)
-fbmode
    mode name to use (according to /etc/fb.modes)
-fbmodeconfig
    config file of modes (default /etc/fb.modes)
-monitor-hfreq, -monitor-vfreq, -monitor-dotclock
    important values, see example.conf
```

If you want to change to a specific mode, then use

```
mplayer -vm -fbmode name_of_mode filename
```

- `-vm` alone will choose the most suitable mode from `/etc/fb.modes`. Can be used together with `-x` and `-y` options too. The `-flip` option is supported only if the movie's pixel format matches the video mode's pixel format. Pay attention to the `bpp` value, fbdev driver tries to use the current, or if you specify the `-bpp` option, then that.
- `-zoom` option isn't supported (use `-vf scale`). You can't use 8bpp (or less) modes.
- You possibly want to turn the cursor off:

```
echo -e '\033[?25l'
```

or

```
setterm -cursor off
```

and the screen saver:

```
setterm -blank 0
```

Note

## MPlayer – The Movie Player for LINUX

To turn the cursor back on:

```
echo -e '\033[?25h'
```

or

```
setterm -cursor on
```

### Note

FBdev video mode changing *does not work* with the VESA framebuffer, and don't ask for it, since it's not an MPlayer limitation.

#### 2.3.1.2.6. Matrox framebuffer (mga\_vid)

This section is about the Matrox G200/G400/G450/G550 BES (Back-End Scaler) support, the mga\_vid kernel driver. It's actively developed by A'rpi, and it has hardware VSYNC support with triple buffering. It works on both framebuffer console and under X.

### Warning

This is Linux only! On non-Linux (tested on FreeBSD) systems, you can use [VIDIX](#) instead!

#### Installation:

1. To use it, you first have to compile mga\_vid.o:

```
cd drivers
make
```

2. Then create /dev/mga\_vid device:

```
mknod /dev/mga_vid c 178 0
```

and load the driver with

```
insmod mga_vid.o
```

3. You should verify the memory size detection using the **dmesg** command. If it's bad, use the mga\_ram\_size option (**rmmod mga\_vid** first), specify card's memory size in MB:

```
insmod mga_vid.o mga_ram_size=16
```

4. To make it load/unload automatically when needed, first insert the following line at the end of /etc/modules.conf:

```
alias char-major-178 mga_vid
```

Then copy the mga\_vid.o module to the appropriate place under /lib/modules/kernel version/somewhere.

Then run

```
depmod -a
```

## MPlayer – The Movie Player for LINUX

5. Now you have to (re)compile MPlayer, `./configure` will detect `/dev/mga_vid` and build the 'mga' driver. Using it from MPlayer goes by `-vo mga` if you have matroxfb console, or `-vo xmga` under XFree86 3.x.x or 4.x.x.

The `mga_vid` driver cooperates with Xv.

The `/dev/mga_vid` device file can be read for some info, for example by

```
cat /dev/mga_vid
```

and can be written for brightness change:

```
echo "brightness=120" > /dev/mga_vid
```

### 2.3.1.2.7. 3Dfx YUV support

This driver uses the kernel's `tdfx` framebuffer driver to play movies with YUV acceleration. You'll need a kernel with `tdfxfb` support, and recompile with

```
./configure --enable-tdfxfb
```

### 2.3.1.2.8. OpenGL output

MPlayer supports displaying movies using OpenGL, but if your platform/driver supports `xv` as should be the case on a PC with Linux, use `xv` instead, OpenGL performance is considerably worse. If you have an X11 implementation without `xv` support, OpenGL is a viable alternative.

Unfortunately not all drivers support this feature. The Utah-GLX drivers (for XFree86 3.3.6) support it for all cards. See <http://utah-glx.sourceforge.net> for details about how to install it.

XFree86(DRI) 4.0.3 or later supports OpenGL with Matrox and Radeon cards, 4.2.0 or later supports Rage128. See <http://dri.sourceforge.net> for download and installation instructions.

A hint from one of our users: the GL video output can be used to get vsynced TV output. You'll have to set an environment variable (at least on nVidia):

```
export $__GL_SYNC_TO_VBLANK=1
```

### 2.3.1.2.9. AAlib – text mode displaying

AAlib is a library for displaying graphics in text mode, using powerful ASCII renderer. There are *lots* of programs already supporting it, like Doom, Quake, etc. MPlayer contains a very usable driver for it. If `./configure` detects aalib installed, the `aalib libvo` driver will be built.

You can use some keys in the AA Window to change rendering options:

Key	Action
1	decrease contrast
2	increase contrast
3	decrease brightness
4	increase brightness

## MPlayer – The Movie Player for LINUX

<b>5</b>	switch fast rendering on/off
<b>6</b>	set dithering mode (none, error distribution, Floyd Steinberg)
<b>7</b>	invert image
<b>8</b>	toggles between aa and MPlayer control

The following command line options can be used:

```
-aaosdcolor=V
    change OSD color
-aasubcolor=V
    change subtitle color
```

where *V* can be: 0 (normal), 1 (dark), 2 (bold), 3 (bold font), 4 (reverse), 5 (special).

**AAlib itself provides a large sum of options. Here are some important:**

```
-aadriver
    set recommended aa driver (X11, curses, Linux)
-aaextended
    use all 256 characters
-aaeight
    use eight bit ASCII
-aahelp
    prints out all aalib options
```

### Note

The rendering is very CPU intensive, especially when using AA-on-X (using aalib on X), and it's least CPU intensive on standard, non-framebuffer console. Use SVGATextMode to set up a big textmode, then enjoy! (secondary head Hercules cards rock :) (but imho you can use `-vf 1bpp` option to get graphics on hga/fb:)

Use the `-framedrop` option if your computer isn't fast enough to render all frames!

Playing on terminal you'll get better speed and quality using the Linux driver, not curses (`-aadriver linux`). But therefore you need write access on `/dev/vcsa<terminal>`! That isn't autodetected by aalib, but `vo_aa` tries to find the best mode. See <http://aa-project.sourceforge.net/tune> for further tuning issues.

#### 2.3.1.2.10. VESA – output to VESA BIOS

This driver was designed and introduced as a **generic driver** for any video card which has VESA VBE 2.0 compatible BIOS. Another advantage of this driver is that it tries to force TV output on. *VESA BIOS EXTENSION (VBE) Version 3.0 Date: September 16, 1998* (Page 70) says:

**Dual-Controller Designs.** VBE 3.0 supports the dual-controller design by assuming that since both controllers are typically provided by the same OEM, under control of a single BIOS ROM on the same graphics card, it is possible to hide the fact that two controllers are indeed present from the application. This has the limitation of preventing simultaneous use of the independent controllers, but allows applications released before VBE 3.0 to operate normally. The VBE Function 00h (Return Controller Information) returns the combined information of both controllers, including the combined list of available modes. When the



## MPlayer – The Movie Player for LINUX

application selects a mode, the appropriate controller is activated. Each of the remaining VBE functions then operates on the active controller.

So you have chances to get working TV-out by using this driver. (I guess that TV-out frequently is standalone head or standalone output at least.)

### ADVANTAGES

- You have chances to watch movies **if Linux even doesn't know** your video hardware.
- You don't need to have installed any graphics' related things on your Linux (like X11 (aka XFree86), fbdev and so on). This driver can be run from **text-mode**.
- You have chances to get **working TV-out**. (It's known at least for ATI's cards).
- This driver calls `int 10h` handler thus it's not an emulator – it calls **real** things of *real* BIOS in *real-mode* (actually in `vm86` mode).
- You can use VIDIX with it, thus getting accelerated video display **and** TV output at the same time! (Recommended for ATI cards.)
- If you have VESA VBE 3.0+, and you had specified `monitor-hfreq`, `monitor-vfreq`, `monitor-dotclock` somewhere (config file, or commandline) you will get the highest possible refresh rate. (Using General Timing Formula). To enable this feature you have to specify **all** your monitor options.

### DISADVANTAGES

- It works only on **x86 systems**.
- It can be used only by `root`.
- Currently it's available only for **Linux**.

### Important

Don't use this driver with **GCC 2.96!** It won't work!

### COMMAND LINE OPTIONS AVAILABLE FOR VESA

`-vo vesa:opts`

currently recognized: `dga` to force `dga` mode and `nodga` to disable `dga` mode. In `dga` mode you can enable double buffering via the `-double` option. Note: you may omit these parameters to enable **autodetection** of `dga` mode.

### KNOWN PROBLEMS AND WORKAROUNDS

- If you have installed **NLS** font on your Linux box and run VESA driver from `text-mode` then after terminating MPlayer you will have **ROM font** loaded instead of national. You can load national font again by using **setsysfont** utility from the Mandrake distribution for example. (**Hint:** The same utility is used for localization of `fbdev`).
- Some **Linux graphics drivers** don't update active **BIOS mode** in DOS memory. So if you have such problem – always use VESA driver only from **text-mode**. Otherwise `text-mode (#03)` will be activated anyway and you will need restart your computer.
- Often after terminating VESA driver you get **black** screen. To return your screen to original state – simply switch to other console (by pressing **Alt+F<x>**) then switch to your previous console by the same way.

## MPlayer – The Movie Player for LINUX

- To get **working TV-out** you need have plugged TV-connector in before booting your PC since video BIOS initializes itself only once during POST procedure.

### 2.3.1.2.11. X11

Avoid if possible. Outputs to X11 (uses shared memory extension), with no hardware acceleration at all. Supports (MMX/3DNow/SSE accelerated, but still slow) software scaling, use the options `-fs -zoom`. Most cards have hardware scaling support, use the `-vo xv` output for them, or `-vo xmg` for Matroxes.

The problem is that most cards' driver doesn't support hardware acceleration on the second head/TV. In those cases, you see green/blue colored window instead of the movie. This is where this driver comes in handy, but you need powerful CPU to use software scaling. Don't use the SDL driver's software output+scaler, it has worse image quality!

Software scaling is very slow, you better try changing video modes instead. It's very simple. See the [DGA section's modelines](#), and insert them into your `XF86Config`.

- If you have XFree86 4.x.x: use the `-vm` option. It will change to a resolution your movie fits in. If it doesn't:
- With XFree86 3.x.x: you have to cycle through available resolutions with the **Ctrl+Alt+plus** and **Ctrl+Alt+minus** keys.

If you can't find the modes you inserted, browse XFree86's output. Some drivers can't use low pixelclocks that are needed for low resolution video modes.

### 2.3.1.2.12. VIDIX

**PREAMBLE.** VIDIX is the abbreviation for **VIDeo Interface for \*niX**. VIDIX was designed and introduced as an interface for fast user-space drivers providing such video performance as `mga_vid` does for Matrox cards. It's also very portable.

This interface was designed as an attempt to fit existing video acceleration interfaces (known as `mga_vid`, `rage128_vid`, `radeon_vid`, `pm3_vid`) into a fixed scheme. It provides highlevel interface to chips which are known as BES (BackEnd scalers) or OV (Video Overlays). It doesn't provide lowlevel interface to things which are known as graphics servers. (I don't want to compete with X11 team in graphics mode switching). I.e. main goal of this interface is to maximize the speed of video playback.

### USAGE

- You can use standalone video output driver: `-vo xvidix`. This driver was developed as X11's front end to VIDIX technology. It requires X server and can work only under X server. Note that, as it directly accesses the hardware and circumvents the X driver, pixmaps cached in the graphics card's memory may be corrupted. You can prevent this by limiting the amount of video memory used by X with the `XF86Config` option "VideoRam" in the device section. You should set this to the amount of memory installed on your card minus 4MB. If you have less than 8MB of video ram, you can use the option "XaaNoPixmapCache" in the screen section instead.
- There is a console VIDIX driver: `-vo cvidix`. This requires a working and initialized framebuffer for most cards (or else you'll just mess up the screen), and you'll have a similar effect as with `-vo mga` or `-vo fbdev`. nVidia cards however are able to output truly graphical video on a real text console. See the [nvidia\\_vid](#) section for more information.

## MPlayer – The Movie Player for LINUX

- You can use VIDIX subdevice which was applied to several video output drivers, such as: `-vo vesa:vidix` (**Linux only**) and `-vo fbdev:vidix`.

Indeed it doesn't matter which video output driver is used with **VIDIX**.

### REQUIREMENTS

- Video card should be in graphics mode (except nVidia cards with the `-vo cvidix` output driver).
- MPlayer's video output driver should know active video mode and be able to tell to VIDIX subdevice some video characteristics of server.

**USAGE METHODS.** When VIDIX is used as **subdevice** (`-vo vesa:vidix`) then video mode configuration is performed by video output device (**vo\_server** in short). Therefore you can pass into command line of MPlayer the same keys as for `vo_server`. In addition it understands `-double` key as globally visible parameter. (I recommend using this key with VIDIX at least for ATI's card). As for `-vo xvidix`, currently it recognizes the following options: `-fs -zoom -x -y -double`.

Also you can specify VIDIX's driver directly as third subargument in command line:

```
mplayer -vo xvidix:mga_vid.so -fs -zoom -double file.avi
```

or

```
mplayer -vo vesa:vidix:radeon_vid.so -fs -zoom -double -bpp 32 file.avi
```

But it's dangerous, and you shouldn't do that. In this case given driver will be forced and result is unpredictable (it may **freeze** your computer). You should do that **ONLY** if you are absolutely sure it will work, and MPlayer doesn't do it automatically. Please tell about it to the developers. The right way is to use VIDIX without arguments to enable driver autodetection.

VIDIX is new technology and it's extremely possible that on your system it won't work. In this case only solution for you it's port it (mainly libdha). But there is hope that it will work on those systems where X11 does.

Since VIDIX requires direct hardware access you can either run it as root or set the SUID bit on the MPlayer binary (**Warning: This is a security risk!**). Alternatively, you can use a special kernel module, like this:

1. Download the development version of ssvgalib (for example 1.9.17), **OR** download a version made by Alex especially for usage with MPlayer (it doesn't need the ssvgalib source to compile) from here.
2. Compile the module in the `ssvgalib_helper` directory (it can be found inside the `ssvgalib-1.9.17/kernel/` directory if you've downloaded the source from the ssvgalib site) and `insmod` it.
3. To create the necessary devices in the `/dev` directory, do a

```
make device
```

in the `ssvgalib_helper` dir, as root.

4. Move the `ssvgalib_helper` directory to `mplayer/main/libdha/ssvgalib_helper`.
5. Required if you download the source from the ssvgalib site: Remove the comment before the CFLAGS line containing "ssvgalib\_helper" string from the `libdha/Makefile`.
6. Recompile and install libdha.

## MPlayer – The Movie Player for LINUX

### 2.3.1.2.12.1. ATI cards

Currently most ATI cards are supported natively, from Mach64 to the newest Radeons.

There are two compiled binaries: `radeon_vid` for Radeon and `rage128_vid` for Rage 128 cards. You may force one or let the VIDIX system autoprobe all available drivers.

### 2.3.1.2.12.2. Matrox cards

Matrox G200, G400, G450 and G550 have been reported to work.

The driver supports video equalizers and should be nearly as fast as the Matrox framebuffer

### 2.3.1.2.12.3. Trident cards

There is a driver available for the Trident Cyberblade/i1 chipset, which can be found on VIA Epia motherboards.

The driver was written and is maintained by Alastair M. Robinson

### 2.3.1.2.12.4. 3DLabs cards

Although there is a driver for the 3DLabs GLINT R3 and Permedia3 chips, no one has tested it, so reports are welcome.

### 2.3.1.2.12.5. nVidia cards

There's a relatively new nVidia driver out there, it's known to work on Riva 128, TNT and GeForce2 chipsets, also others have been reported working.

## LIMITATIONS

- It's recommended to use the binary nVidia drivers for X before using this VIDIX driver, because some of the registers which need to be initialized haven't been discovered yet, so it will probably fail with the Open Source XFree86 `nv.o` driver.
- Currently only codecs capable of UYVY colorspace output can work in conjunction with this driver. Unfortunately, this excludes every single decoder from the libavcodec family. This leaves us with the following usable popular codecs: `cvid`, `divxds`, `xvid`, `divx4`, `wmv7`, `wmv8` and some others. Please note that this is only a temporal inconvenience. The usage syntax is as follows:

```
mplayer -vf format=uyvy -vc divxds divx3file.avi
```

An unique feature of the `nvidia_vid` driver is its ability to display video on **plain, pure, text-only console** – with no framebuffer or X magic whatsoever. For this purpose, we'll have to use the `cvidix` video output, as the following example shows:

```
mplayer -vf format=uyvy -vc divxds -vo cvidix example.avi
```

Reports awaited!

## MPlayer – The Movie Player for LINUX

### 2.3.1.2.12.6. SiS cards

This is very experimental code, just like `nvidia_vid`.

It's been tested on SiS 650/651/740 (the most common chipsets used in the SiS versions of the "Shuttle XPC" barebones boxes out there)

Reports awaited!

### 2.3.1.2.13. DirectFB

"DirectFB is a graphics library which was designed with embedded systems in mind. It offers maximum hardware accelerated performance at a minimum of resource usage and overhead."

– quoted from <http://www.directfb.org>

I'll exclude DirectFB features from this section.

Though MPlayer is not supported as a "video provider" in DirectFB, this output driver will enable video playback through DirectFB. It will – of course – be accelerated, on my Matrox G400 DirectFB's speed was the same as XVideo.

Always try to use the newest version of DirectFB. You can use DirectFB options on the command line, using the `-dfbopts` option. Layer selection can be done by the subdevice method, e.g.: `-vo directfb:2` (layer -1 is default: autodetect)

### 2.3.1.2.14. DirectFB/Matrox (dfbmga)

Please read the [main DirectFB](#) section or general informations.

This video output driver will enable CRTC2 (on the second head) on the Matrox G400/G450/G550 card, displaying video **independently** of the first head.

Instructions on how to make it work can be found in the [tech](#) section or directly on Ville Syrjala's [home page](#).

## Note

the first DirectFB version with which we could kick this working was 0.9.17 (it's buggy, needs that `surfacemanager` patch from the URL above). Anyway, porting of the CRTC2 code to `mga_vid` is underway.

### 2.3.1.3. MPEG decoders

#### 2.3.1.3.1. DVB output and input

MPlayer supports cards with the Siemens DVB chipset from vendors like Siemens, Technotrend, Galaxis or Hauppauge. The latest DVB drivers are available from the [Linux TV site](#). If you want to do software transcoding you should have at least a 1GHz CPU.

Configure should detect your DVB card. If it did not, force detection with

```
./configure --enable-dvb
```

## MPlayer – The Movie Player for LINUX

If you have ost headers at a non-standard path, set the path with

```
./configure --with-extraincdir=DVB source directory/ost/include
```

Then compile and install as usual.

**USAGE.** Hardware decoding (playing standard MPEG1/2 files) can be done with this command:

```
mplayer -ao mpegpes -vo mpegpes file.mpg/vob
```

Software decoding or transcoding different formats to MPEG1 can be achieved using a command like this:

```
mplayer -ao mpegpes -vo mpegpes yourfile.ext  
mplayer -ao mpegpes -vo mpegpes -vf expand yourfile.ext
```

Note that DVB cards only support heights 288 and 576 for PAL or 240 and 480 for NTSC. You **must** rescale for other heights by adding `scale=width:height` with the width and height you want to the `-vf` option. DVB cards accept various widths, like 720, 704, 640, 512, 480, 352 etc and do hardware scaling in horizontal direction, so you do not need to scale horizontally in most cases. For a 512x384 (aspect 4:3) DivX try:

```
mplayer -ao mpegpes -vo mpegpes -vf scale=512:576
```

If you have a widescreen movie and you do not want to scale it to full height, you can use the `expand=w:h` filter to add black bands. To view a 640x384 DivX, try:

```
mplayer -ao mpegpes -vo mpegpes -vf expand=640:576 file.avi
```

If your CPU is too slow for a full size 720x576 DivX, try downscaling:

```
mplayer -ao mpegpes -vo mpegpes -vf scale=352:576 file.avi
```

If speed does not improve, try vertical downscaling, too:

```
mplayer -ao mpegpes -vo mpegpes -vf scale=352:288 file.avi
```

For OSD and subtitles use the OSD feature of the expand filter. So, instead of `expand=w:h` or `expand=w:h:x:y`, use `expand=w:h:x:y:1` (the 5th parameter `:1` at the end will enable OSD rendering). You may want to move the image up a bit to get a bigger black zone for subtitles. You may also want to move subtitles up, if they are outside your TV screen, use the `-subpos <0-100>` option to adjust this (`-subpos 80` is a good choice).

In order to play non-25fps movies on a PAL TV or with a slow CPU, add the `-framedrop` option.

To keep the aspect ratio of DivX files and get the optimal scaling parameters (hardware horizontal scaling and software vertical scaling while keeping the right aspect ratio), use the new `dvbscale` filter:

```
for a 4:3 TV: -vf dvbscale,scale=-1:0,expand=-1:576:-1:-1:1  
for a 16:9 TV: -vf dvbscale=1024,scale=-1:0,expand=-1:576:-1:-1:1
```

**Using your DVB card for watching Digital TV (DVB input module).** First you need to pass the `dvb_shutdown_timeout=0` parameter to the kernel module `dvb-core`, or MPlayer will die after 10 seconds.

## MPlayer – The Movie Player for LINUX

You should also have the programs **scan** and **szap/tzap/czap** installed; they are all included in the drivers package.

Verify that your drivers are working properly with a program such as **dvbstream** (that is the base of the DVB input module).

Now you should compile a `~/ .mplayer/channels.conf` file, with the syntax accepted by **szap/tzap/czap**, or have **scan** compile it for you.

Make sure that you have *only* Free to Air channels in your `channels.conf` file, or MPlayer will hang on the others.

To show the first of the channels present in your list, run

```
mplayer dvb://
```

If you want to watch a specific channel, such as R1, run

```
mplayer dvb://R1
```

To change channels press the **h** (next) and **k** (previous) keys, or use the OSD menu (requires a working OSD subsystem).

If your `~/ .mplayer/menu.conf` contains a `<dvbsel>` entry, such as the one in the example file `etc/dvb-menu.conf` (that you can use to overwrite `~/ .mplayer/menu.conf`), the main menu will show a sub-menu entry that will permit you to choose one of the channels present in your `channels.conf`.

If you want to save a program to disk you can use

```
mplayer -dumpfile r1.ts -dumpstream dvb://R1
```

If you want to record it in a different format (re-encoding it) instead you can run a command such as

```
mencoder -o r1.avi -ovc xvid -xvidencopts bitrate=800 -oac mp3lame -lameopts cbr:br=128 -pp=ci
```

Read the man page for a list of options that you can pass to the DVB input module.

**FUTURE.** If you have questions or want to hear feature announcements and take part in discussions on this subject, join our MPlayer-DVB mailing list. Please remember that the list language is English.

In the future you may expect the ability to display OSD and subtitles using the native OSD feature of DVB cards, as well as more fluent playback of non-25fps movies and realtime transcoding between MPEG2 and MPEG4 (partial decompression).

### 2.3.1.3.2. DXR2

MPlayer supports hardware accelerated playback with the Creative DXR2 card.

First of all you will need properly installed DXR2 drivers. You can find the drivers and installation instructions at the DXR2 Resource Center site.

## MPlayer – The Movie Player for LINUX

### USAGE

```
-vo dxr2
    enable TV output
-vo dxr2:x11 or -vo dxr2:xv
    enable Overlay output in X11
-dxr2 <option1:option2:...>
    This option is used to control the DXR2 driver.
```

The overlay chipset used on the DXR2 is of pretty bad quality but the default settings should work for everybody. The OSD may be usable with the overlay (not on TV) by drawing it in the colorkey. With the default colorkey settings you may get variable results, usually you will see the colorkey around the characters or some other funny effect. But if you properly adjust the colorkey settings you should be able to get acceptable results.

Please see the man page for available options.

#### 2.3.1.3.3. DXR3/Hollywood+

MPlayer supports hardware accelerated playback with the Creative DXR3 and Sigma Designs Hollywood Plus cards. These cards both use the em8300 MPEG decoder chip from Sigma Designs.

First of all you will need properly installed DXR3/H+ drivers, version 0.12.0 or later. You can find the drivers and installation instructions at the [DXR3 & Hollywood Plus for Linux](#) site. `configure` should detect your card automatically, compilation should go without problems.

### USAGE

```
-vo dxr3:prebuf:sync:norm=x:device
    overlay activates the overlay instead of TVOut. It requires that you have a properly configured
    overlay setup to work right. The easiest way to configure the overlay is to first run autocal. Then run
    mplayer with dxr3 output and without overlay turned on, run dxr3view. In dxr3view you can tweak
    the overlay settings and see the effects in realtime, perhaps this feature will be supported by the
    MPlayer GUI in the future. When overlay is properly set up you will no longer need to use dxr3view.
    prebuf turns on prebuffering. Prebuffering is a feature of the em8300 chip that enables it to hold
    more than one frame of video at a time. This means that when you are running with prebuffering
    MPlayer will try to keep the video buffer filled with data at all times. If you are on a slow machine
    MPlayer will probably use close to, or precisely 100% of CPU. This is especially common if you play
    pure MPEG streams (like DVDs, SVCDs a.s.o.) since MPlayer will not have to reencode it to MPEG
    it will fill the buffer very fast. With prebuffering video playback is much less sensitive to other
    programs hogging the CPU, it will not drop frames unless applications hog the CPU for a long time.
    When running without prebuffering the em8300 is much more sensitive to CPU load, so it is highly
    suggested that you turn on MPlayer's -framedrop option to avoid further loss of sync. sync will
    turn on the new sync-engine. This is currently an experimental feature. With the sync feature turned
    on the em8300's internal clock will be monitored at all times, if it starts to deviate from MPlayer's
    clock it will be reset causing the em8300 to drop any frames that are lagging behind. norm=x will set
    the TV norm of the DXR3 card without the need for external tools like em8300setup. Valid norms are
    5 = NTSC, 4 = PAL-60, 3 = PAL. Special norms are 2 (auto-adjust using PAL/PAL-60) and 1
    (auto-adjust using PAL/NTSC) because they decide which norm to use by looking at the frame rate
    of the movie. norm = 0 (default) does not change the current norm. device = device number to use
    if you have more than one em8300 card. Any of these options may be left out. :prebuf:sync
```



## MPlayer – The Movie Player for LINUX

seems to work great when playing DivX movies. People have reported problems using the `prebuf` option when playing MPEG1/2 files. You might want to try running without any options first, if you have sync problems, or DVD subtitle problems, give `:sync` a try.

`-ao oss:/dev/em8300_ma-X`

For audio output, where *X* is the device number (0 if one card).

`-aop list=resample:fout=xxxxx`

The em8300 cannot play back samplerates lower than 44100Hz. If the sample rate is below 44100Hz select either 44100Hz or 48000Hz depending on which one matches closest. I.e. if the movie uses 22050Hz use 44100Hz as  $44100 / 2 = 22050$ , if it is 24000Hz use 48000Hz as  $48000 / 2 = 24000$  and so on. This does not work with digital audio output (`-ac hwac3`).

`-vf lavc/fame`

To watch non-MPEG content on the em8300 (i.e. DivX or RealVideo) you have to specify an MPEG1 video filter such as `libavcodec` (`lavc`) or `libfame` (`fame`). At the moment `lavc` is both faster and gives better image quality, it is suggested that you use that unless you have problems with it. See the man page for further info about `-vf lavc/fame`. Using `lavc` is highly recommended. Currently there is no way of setting the fps of the em8300 which means that it is fixed to 29.97fps. Because of this it is highly recommended that you use `-vf lavc=quality:25` especially if you are using prebuffering. Then why 25 and not 29.97? Well, the thing is that when you use 29.97 the picture becomes a bit jumpy. The reason for this is unknown to us. If you set it to somewhere between 25 and 27 the picture becomes stable. For now all we can do is accept this for a fact.

`-vf expand=-1:-1:-1:-1:1`

Although the DXR3 driver can put some OSD onto the MPEG1/2/4 video, it has much lower quality than MPlayer's traditional OSD, and has several refresh problems as well. The command line above will firstly convert the input video to MPEG4 (this is mandatory, sorry), then apply an expand filter which won't expand anything (`-1`: default), but apply the normal OSD onto the picture (that's what the "1" at the end does).

`-ac hwac3`

The em8300 supports playing back AC3 audio (surround sound) through the digital audio output of the card. See the `-ao oss` option above, it must be used to specify the DXR3's output instead of a soundcard.

### 2.3.1.4. Other visualization hardware

#### 2.3.1.4.1. Zr

This is a display-driver (`-vo zr`) for a number of MJPEG capture/playback cards (tested for DC10+ and Buz, and it should work for the LML33, the DC10). The driver works by encoding the frame to JPEG and then sending it to the card. For the JPEG encoding `libavcodec` is used, and required. With the special *cinerama* mode, you can watch movies in true wide screen provided that you have two beamers and two MJPEG cards. Depending on resolution and quality settings, this driver may require a lot of CPU power, remember to specify `-framedrop` if your machine is too slow. Note: My AMD K6-2 350MHz is (with `-framedrop`) quite adequate for watching VCD sized material and downscaled movies.

This driver talks to the kernel driver available at <http://mjpeg.sourceforge.net>, so you must get it working first. The presence of an MJPEG card is autodetected by the `configure` script, if autodetection fails, force detection with

```
./configure --enable-zr
```

The output can be controlled by several options, a long description of the options can be found in the man page, a short list of options can be viewed by running

## MPlayer – The Movie Player for LINUX

```
mplayer -zrhhelp
```

Things like scaling and the OSD (on screen display) are not handled by this driver but can be done using the video filters. For example, suppose that you have a movie with a resolution of 512x272 and you want to view it fullscreen on your DC10+. There are three main possibilities, you may scale the movie to a width of 768, 384 or 192. For performance and quality reasons, I would choose to scale the movie to 384x204 using the fast bilinear software scaler. The commandline is

```
mplayer -vo zr -sws 0 -vf scale=384:204 movie.avi
```

Cropping can be done by the `crop` filter and by this driver itself. Suppose that a movie is too wide for display on your Buz and that you want to use `-zrcrop` to make the movie less wide, then you would issue the following command

```
mplayer -vo zr -zrcrop 720x320+80+0 benhur.avi
```

if you want to use the `crop` filter, you would do

```
mplayer -vo zr -vf crop=720:320:80:0 benhur.avi
```

Extra occurrences of `-zrcrop` invoke *cinerama* mode, i.e. you can distribute the movie over several TV's or beamers to create a larger screen. Suppose you have two beamers. The left one is connected to your Buz at `/dev/video1` and the right one is connected to your DC10+ at `/dev/video0`. The movie has a resolution of 704x288. Suppose also that you want the right beamer in black and white and that the left beamer should have JPEG frames at quality 10, then you would issue the following command

```
mplayer -vo zr -zrdev /dev/video0 -zrcrop 352x288+352+0 -zrxdooff 0 -zrbw \  
-zrcrop 352x288+0+0 -zrdev /dev/video1 -zrquality 10 \  
movie.avi
```

You see that the options appearing before the second `-zrcrop` only apply to the DC10+ and that the options after the second `-zrcrop` apply to the Buz. The maximum number of MJPEG cards participating in *cinerama* is four, so you can build a 2x2 vidiwall.

Finally an important remark: Do not start or stop XawTV on the playback device during playback, it will crash your computer. It is, however, fine to **FIRST** start XawTV, **THEN** start MPlayer, wait for MPlayer to finish and **THEN** stop XawTV.

### 2.3.1.4.2. Blinkenlights

This driver is capable of playback using the Blinkenlights UDP protocol. If you don't know what Blinkenlights is, you don't need this driver.

### 2.3.1.5. TV-out support

#### 2.3.1.5.1. Matrox G400 cards

Under Linux you have two methods to get G400 TV out working:

## Important

for Matrox G450/G550 TV-out instructions, please see the next section!

### *XFree86*

Using the driver and the HAL module, available from [Matrox's site](#). This will give you X on the TV.

**This method doesn't give you accelerated playback** as under Windows! The second head has only YUV framebuffer, the *BES* (Back End Scaler, the YUV scaler on G200/G400/G450/G550 cards) doesn't work on it! The windows driver somehow workarounds this, probably by using the 3D engine to zoom, and the YUV framebuffer to display the zoomed image. If you really want to use X, use the `-vo x11 -fs -zoom` options, but it will be **SLOW**, and has **Macrovision** copy protection enabled (you can "workaround" Macrovision using this [perl script](#)).

### *Framebuffer*

Using the **matroxfb modules** in the 2.4 kernels. 2.2 kernels don't have the TVout feature in them, thus unusable for this. You have to enable ALL matroxfb-specific feature during compilation (except MultiHead), and compile them into **modules**! You'll also need I2C enabled.

1. Enter TVout and type `./compile.sh`. Install TVout/matroxset/matroxset somewhere into your PATH.
2. If you don't have **fbset** installed, put TVout/fbset/fbset somewhere into your PATH.
3. If you don't have **con2fb** installed, put TVout/con2fb/con2fb somewhere into your PATH.
4. Then enter into the TVout/ directory in the MPlayer source, and execute `./modules` as root. Your text-mode console will enter into framebuffer mode (no way back!).
5. Next, EDIT and run the `./matroxtv` script. This will present you to a very simple menu. Press **2** and **Enter**. Now you should have the same picture on your monitor, and TV. If the TV (PAL by default) picture has some weird stripes on it, the script wasn't able to set the resolution correctly (to 640x512 by default). Try other resolutions from the menu and/or experiment with fbset.
6. Yoh. Next task is to make the cursor on tty1 (or whatever) to disappear, and turn off screen blanking. Execute the following commands:

```
echo -e '\033[?25l'
setterm -blank 0
```

or

```
setterm -cursor off
setterm -blank 0
```

You possibly want to put the above into a script, and also clear the screen. To turn the cursor back:

```
echo -e '\033[?25h'
```

or

```
setterm -cursor on
```

7. Yeah kewl. Start movie playing with

```
mplayer -vo mga -fs -screenw 640 -screenh 512 filename
```

## MPlayer – The Movie Player for LINUX

(If you use X, now change to matroxfb with for example **Ctrl+Alt+F1**.) Change 640 and 512 if you set the resolution to other...

### 8. Enjoy the ultra-fast ultra-featured Matrox TV output (better than Xv)!

**Building a Matrox TV-out cable.** No one takes any responsibility, nor guarantee for any damage caused by this documentation.

**Cable for G400.** The CRTC2 connector's fourth pin is the composite video signal. The ground are the sixth, seventh and eighth pins. (info contributed from Balázs Rácz)

**Cable for G450.** The CRTC2 connector's first pin is the composite video signal. The ground are the fifth, sixth, seventh, and fifteenth (5, 6, 7, 15) pins. (info contributed from Balázs Kerekes)

#### 2.3.1.5.2. Matrox G450/G550 cards

TV output support for these cards has only been recently introduced, and is not yet in the mainstream kernel. Currently the **mga\_vid** module can't be used AFAIK, because the G450/G550 driver works only in one configuration: the first CRTC chip (with much more features) on the first display (on monitor), and the second CRTC (no **BES** – for explanation on BES, please see the G400 section above) on TV. So you can only use MPlayer's *fbdev* output driver at the present.

The first CRTC can't be routed to the second head currently. The author of the kernel matroxfb driver – Petr Vandrovec – will maybe make support for this, by displaying the first CRTC's output onto both of the heads at once, as currently recommended for G400, see the section above.

The necessary kernel patch and the detailed howto is downloadable from [http://www.bglug.ca/matrox\\_tvout/](http://www.bglug.ca/matrox_tvout/)

#### 2.3.1.5.3. ATI cards

**PREAMBLE.** Currently ATI doesn't want to support any of its TV-out chips under Linux, because of their licensed Macrovision technology.

#### ATI CARDS TV-OUT STATUS ON LINUX

- **ATI Mach64:** supported by [gatos](#).
- **ASIC Radeon VIVO:** supported by [gatos](#).
- **Radeon and Rage128:** supported by MPlayer! Check [VESA driver](#) and [VIDIX](#) sections.
- **Rage Mobility P/M, Radeon, Rage 128, Mobility M3/M4:** supported by [atitvout](#).

On other cards, just use the [VESA](#) driver, without VIDIX. Powerful CPU is needed, though.

Only thing you need to do – **Have the TV connector plugged in before booting your PC** since video BIOS initializes itself only once during POST procedure.

#### 2.3.1.5.4. Voodoo 3

Check [this URL](#).

## MPlayer – The Movie Player for LINUX

### 2.3.1.5.5. nVidia

First, you MUST download the closed–source drivers from <http://nvidia.com>. I will not describe the installation and configuration process because it does not cover the scope of this documentation.

After XFree86, XVideo, and 3D acceleration is properly working, edit your card's Device section in the XF86Config file, according to the following example (adapt for your card/TV):

```
Section "Device"
    Identifier      "GeForce"
    VendorName     "ASUS"
    BoardName      "nVidia GeForce2/MX 400"
    Driver         "nvidia"
    #Option        "NvAGP" "1"
    Option         "NoLogo"
    Option         "CursorShadow" "on"

    Option         "TwinView"
    Option         "TwinViewOrientation" "Clone"
    Option         "MetaModes" "1024x768,640x480"
    Option         "ConnectedMonitor" "CRT, TV"
    Option         "TVStandard" "PAL-B"
    Option         "TVOutFormat" "Composite"

EndSection
```

Of course the important thing is the TwinView part.

### 2.3.1.5.6. Neomagic

Tested on a Toshiba Tecra 8000. Its TV output chip is a miserable crap. Avoid if possible.

You must use `-vo vesa`. The tested chip was capable of 1.333333 aspect ratio only, so be sure to use the `-x`, `-y` options and/or the `-vf scale,crop,expand` filters if the box doesn't let you enable TV output. Maximum resolution was 720\*576 at 16bpp.

Known issues: VESA–only, 1.33333 limitation, image isn't always centered, movie becomes 4bpp in every 10 minutes, and stays that way. Frequent hard freezes, LCD display problems.

## 2.3.2. Audio output devices

### 2.3.2.1. Audio/Video synchronisation

MPlayer's audio interface is called *libao2*. It currently contains these drivers:

Driver	Comment
oss	OSS (ioctl) driver (supports hardware AC3 passthrough)
sdl	SDL driver (supports sound daemons like <b>ESD</b> and <b>ARTS</b> )
nas	NAS (Network Audio System) driver
alsa5	native ALSA 0.5 driver
alsa9	native ALSA 0.9 driver (supports hardware AC3 passthrough)

## MPlayer – The Movie Player for LINUX

sun	SUN audio driver (/dev/audio) for BSD and Solaris8 users
macosx	native MacOSX driver
win32	native Win32 driver
arts	native ARTS driver (mostly for KDE users)
esd	native ESD driver (mostly for GNOME users)

Linux sound card drivers have compatibility problems. This is because MPlayer relies on an in-built feature of *properly* coded sound drivers that enable them to maintain correct audio/video sync. Regrettably, some driver authors don't take the care to code this feature since it is not needed for playing MP3s or sound effects.

Other media players like [aviplay](#) or [xine](#) possibly work out-of-the-box with these drivers because they use "simple" methods with internal timing. Measuring showed that their methods are not as efficient as MPlayer's.

Using MPlayer with a properly written audio driver will never result in A/V desyncs related to the audio, except only with very badly created files (check the man page for workarounds).

If you happen to have a bad audio driver, try the `-autosync` option, it should sort out your problems. See the man page for detailed information.

### Some notes:

- If you have an OSS driver, first try `-ao oss` (this is the default). If you experience glitches, halts or anything out of the ordinary, try `-ao sdl` (NOTE: you need to have SDL libraries and header files installed). The SDL audio driver helps in a lot of cases and also supports ESD (GNOME) and ARTS (KDE).
- If you have ALSA version 0.5, then you almost always have to use `-ao alsa5`, since ALSA 0.5 has buggy OSS emulation code, and will **crash MPlayer** with a message like this:

```
DEMUXER: Too many (945 in 8390980 bytes) video packets in the buffer!
```

- On Solaris, use the SUN audio driver with the `-ao sun` option, otherwise neither video nor audio will work.
- If the sound clicks when playing from CD-ROM, turn on IRQ unmasking, e.g. **hdparm -u1 /dev/cdrom** (**man hdparm**). This is generally beneficial and described in more detail in the [CD-ROM](#) section.

### 2.3.2.2. Soundcard experiences, recommendations

On Linux, a 2.4.x kernel is highly recommended. Kernel 2.2 is not tested.

Linux sound drivers are primarily provided by the free version of OSS. These drivers have been superseded by [ALSA](#) (Advanced Linux Sound Architecture) in the 2.5 development series. If your distribution does not already use ALSA you may wish to try their drivers if you experience sound problems. ALSA drivers are generally superior to OSS in compatibility, performance and features. But some sound cards are only supported by the commercial OSS drivers from [4Front Technologies](#). They also support several non-Linux systems.

SOUND CARD	DRIVER				Max kHz	Max Channels	Max Opens [a]
	OSS/Free	ALSA	OSS/Pro	other			
		<a href="#">via82cxxx audio</a>	<a href="#">snd-via82xx</a>				

## MPlayer – The Movie Player for LINUX

	VIA onboard (686/A/B, 8233, 8235)						4–48 kHz or 48 kHz only, depending on the chipset
Aureal Vortex 2	none	none	OK	<u>Linux Aural Drivers</u> buffer size increased to 32k	48	4.1	5+
SB Live!	Analog OK, SP/DIF not working	Both OK	Both OK	<u>Creative's OSS driver (SP/DIF support)</u>	192	4.0/5.1	32
SB 128 PCI (es1371)	OK	?			48	stereo	2
SB AWE 64	max 44kHz	48kHz sounds bad			48		
GUS PnP	none	OK	OK		48		
Gravis UltraSound ACE							
Gravis UltraSound MAX	OK	OK (?)			48		
ESS 688	OK	OK (?)			48		
C–Media cards (CMI8338/8738)	OK	OK SP/DIF is supported with ALSA 0.9.x	?		44	stereo	1
Yamaha cards (*ymf*)	not OK (?) (maybe -ao sdl)	OK only with ALSA 0.5 with OSS emulation <b>AND</b> -ao sdl (!) (?)					
Cards with envy24 chips (like Terratec EWS88MT)	?	?	OK		?		
PC Speaker or DAC	OK	none		<u>Linux PC speaker OSS driver</u>	The driver emulates 44.1, maybe more.	mono	1

<sup>[a]</sup> the number of applications that are able to use the device *at the same time*.

Feedback to this document is welcome. Please tell us how MPlayer and your sound card(s) worked together.

### 2.3.2.3. Audio filters

The old audio plugins have been superseded by a new audio filter layer. Audio filters are used for changing the properties of the audio data before the sound reaches the sound card. The activation and deactivation of the filters is normally automated but can be overridden. The filters are activated when the properties of the audio data differ from those required by the sound card and deactivated if unnecessary. The `-af filter1,filter2,...` option is used to override the automatic activation of filters or to insert filters that are not automatically inserted. The filters will be executed as they appear in the comma separated list.

Example:

```
mplayer -af resample,pan movie.avi
```

would run the sound through the resampling filter followed by the pan filter. Observe that the list must not contain any spaces, else it will fail.

The filters often have options that change their behavior. These options are explained in detail in the sections below. A filter will execute using default settings if its options are omitted. Here is an example of how to use filters in combination with filter specific options:

```
mplayer -af resample=11025,pan=1:0.5:0.5 -channels 1 -srate 11025 media.avi
```

would set the output frequency of the resample filter to 11025Hz and downmix the audio to 1 channel using the pan filter.

The overall execution of the filter layer is controlled using the `-af-adv` option. This option has two suboptions:

`force` is a bit field that controls how the filters are inserted and what speed/accuracy optimizations they use:

- 0 Use automatic insertion of filters and optimize according to CPU speed.
- 1 Use automatic insertion of filters and optimize for the highest speed. *Warning:* Some features in the audio filters may silently fail, and the sound quality may drop.
- 2 Use automatic insertion of filters and optimize for quality.
- 3 Use no automatic insertion of filters and no optimization. *Warning:* It may be possible to crash MPlayer using this setting.
- 4 Use automatic insertion of filters according to 0 above, but use floating point processing when possible.
- 5 Use automatic insertion of filters according to 1 above, but use floating point processing when possible.
- 6 Use automatic insertion of filters according to 2 above, but use floating point processing when possible.
- 7



## MPlayer – The Movie Player for LINUX

Use no automatic insertion of filters according to 3 above, and use floating point processing when possible.

`list` is an alias for the `-af` option.

The filter layer is also affected by the following generic options:

`-v`

Increases the verbosity level and makes most filters print out extra status messages.

`-channels`

This option sets the number of output channels you would like your sound card to use. It also affects the number of channels that are being decoded from the media. If the media contains less channels than requested the channels filter (see below) will automatically be inserted. The routing will be the default routing for the channels filter.

`-srate`

This option selects the sample rate you would like your sound card to use (of course the cards have limits on this). If the sample frequency of your sound card is different from that of the current media, the resample filter (see below) will be inserted into the audio filter layer to compensate for the difference.

`-format`

This option sets the sample format between the audio filter layer and the sound card. If the requested sample format of your sound card is different from that of the current media, a format filter (see below) will be inserted to rectify the difference.

### 2.3.2.3.1. Up/Downsampling

MPlayer fully supports sound up/down-sampling through the `resample` filter. It can be used if you have a fixed frequency sound card or if you are stuck with an old sound card that is only capable of max 44.1kHz. This filter is automatically enabled if it is necessary, but it can also be explicitly enabled on the command line. It has three options:

`srate <8000-192000>`

is an integer used for setting the output sample frequency in Hz. The valid range for this parameter is 8kHz to 192kHz. If the input and output sample frequency are the same or if this parameter is omitted the filter is automatically unloaded. A high sample frequency normally improves the audio quality, especially when used in combination with other filters.

`sloppy`

is an optional binary parameter that allows the output frequency to differ slightly from the frequency given by `srate`. This option can be used if the startup of the playback is extremely slow. It is enabled by default.

`type <0-2>`

is an optional integer between 0 and 2 that selects which resampling method to use. Here 0 represents linear interpolation as resampling method, 1 represents resampling using a poly-phase filter-bank and integer processing and 2 represents resampling using a poly-phase filter-bank and floating point processing. Linear interpolation is extremely fast, but suffers from poor sound quality especially when used for up-sampling. The best quality is given by 2 but this method also suffers from the highest CPU load.

Example:

```
mplayer -af resample=44100:0:0
```

## MPlayer – The Movie Player for LINUX

would set the output frequency of the resample filter to 44100Hz using exact output frequency scaling and linear interpolation.

### 2.3.2.3.2. Changing the number of channels

The `channels` filter can be used for adding and removing channels, it can also be used for routing or copying channels. It is automatically enabled when the output from the audio filter layer differs from the input layer or when it is requested by another filter. This filter unloads itself if not needed. The number of options is dynamic:

`nch` *<1-6>*

is an integer between 1 and 6 that is used for setting the number of output channels. This option is required, leaving it empty results in a runtime error.

`nr` *<1-6>*

is an integer between 1 and 6 that is used for specifying the number of routes. This parameter is optional. If it is omitted the default routing is used.

`from1:to1:from2:to2:from3:to3...`

are pairs of numbers between 0 and 5 that define where each channel should be routed.

If only `nch` is given the default routing is used, it works as follows: If the number of output channels is bigger than the number of input channels empty channels are inserted (except mixing from mono to stereo, then the mono channel is repeated in both of the output channels). If the number of output channels is smaller than the number of input channels the exceeding channels are truncated.

Example 1:

```
mplayer -af channels=4:4:0:1:1:0:2:2:3:3 media.avi
```

would change the number of channels to 4 and set up 4 routes that swap channel 0 and channel 1 and leave channel 2 and 3 intact. Observe that if media containing two channels was played back, channels 2 and 3 would contain silence but 0 and 1 would still be swapped.

Example 2:

```
mplayer -af channels=6:4:0:0:0:1:0:2:0:3 media.avi
```

would change the number of channels to 6 and set up 4 routes that copy channel 0 to channels 0 to 3. Channel 4 and 5 will contain silence.

### 2.3.2.3.3. Sample format converter

The `format` filter converts between different sample formats. It is automatically enabled when needed by the sound card or another filter.

`bps` *<number>*

can be 1, 2 or 4 and denotes the number of bytes per sample. This option is required, leaving it empty results in a runtime error.

`f` *<format>*

is a text string describing the sample format. The string is a concatenated mix of: `alaw`, `mulaw` or `imaadpcm`, `float` or `int`, `unsigned` or `signed`, `le` or `be` (little or big endian). This option is required, leaving it empty results in a runtime error.

Example:

## MPlayer – The Movie Player for LINUX

```
mplayer -af format=4:float media.avi
```

would set the output format to 4 bytes per sample floating point data.

### 2.3.2.3.4. Delay

The `delay` filter delays the sound to the loudspeakers such that the sound from the different channels arrives at the listening position simultaneously. It is only useful if you have more than 2 loudspeakers. This filter has a variable number of parameters:

`d1:d2:d3...`

are floating point numbers representing the delays in ms that should be imposed on the different channels. The minimum delay is 0ms and the maximum is 1000ms.

To calculate the required delay for the different channels do as follows:

1. Measure the distance to the loudspeakers in meters in relation to your listening position, giving you the distances  $s_1$  to  $s_5$  (for a 5.1 system). There is no point in compensating for the sub-woofer (you will not hear the difference anyway).
2. Subtract the distances  $s_1$  to  $s_5$  from the maximum distance i.e.  $s[i] = \max(s) - s[i]$ ;  $i = 1...5$
3. Calculate the required delays in ms as  $d[i] = 1000*s[i]/342$ ;  $i = 1...5$

Example:

```
mplayer -af delay=10.5:10.5:0:0:7:0 media.avi
```

would delay front left and right by 10.5ms, the two rear channels and the sub by 0ms and the center channel by 7ms.

### 2.3.2.3.5. Software volume control

Software volume control is implemented by the `volume` audio filter. Use this filter with caution since it can reduce the signal to noise ratio of the sound. In most cases it is best to set the level for the PCM sound to max, leave this filter out and control the output level to your speakers with the master volume control of the mixer. In case your sound card has a digital PCM mixer instead of an analog one, and you hear distortion, use the MASTER mixer instead. If there is an external amplifier connected to the computer (this is almost always the case), the noise level can be minimized by adjusting the master level and the volume knob on the amplifier until the hissing noise in the background is gone. This filter has two options:

`v <-200 - +60>`

is a floating point number between -200 and +60 which represents the volume level in dB. The default level is 0dB.

`c`

is a binary control that turns soft clipping on and off. Soft-clipping can make the sound more smooth if very high volume levels are used. Enable this option if the dynamic range of the loudspeakers is very low. Be aware that this feature creates distortion and should be considered a last resort.

Example:

```
mplayer -af volume=10.1:0 media.avi
```

would amplify the sound by 10.1dB and hard-clip if the sound level is too high.

## MPlayer – The Movie Player for LINUX

This filter has a second feature: It measures the overall maximum sound level and prints out that level when MPlayer exits. This volume estimate can be used for setting the sound level in MEncoder such that the maximum dynamic range is utilized.

### 2.3.2.3.6. Equalizer

The `equalizer` filter represents a 10 octave band graphic equalizer, implemented using 10 IIR band pass filters. This means that it works regardless of what type of audio is being played back. The center frequencies for the 10 bands are:

Band No.	Center frequency
0	31.25 Hz
1	62.50 Hz
2	125.0 Hz
3	250.0 Hz
4	500.0 Hz
5	1.000 kHz
6	2.000 kHz
7	4.000 kHz
8	8.000 kHz
9	16.00 kHz

If the sample rate of the sound being played back is lower than the center frequency for a frequency band, then that band will be disabled. A known bug with this filter is that the characteristics for the uppermost band are not completely symmetric if the sample rate is close to the center frequency of that band. This problem can be worked around by up-sampling the sound using the `resample` filter before it reaches this filter.

This filter has 10 parameters:

`g1:g2:g3...g10`

are floating point numbers between -12 and +12 representing the gain in dB for each frequency band.

Example:

```
mplayer -af equalizer=11:11:10:5:0:-12:0:5:12:12 media.avi
```

would amplify the sound in the upper and lower frequency region while canceling it almost completely around 1kHz.

### 2.3.2.3.7. Panning filter

Use the `pan` filter to mix channels arbitrarily. It is basically a combination of the volume control and the channels filter. There are two major uses for this filter:

1. Down-mixing many channels to only a few, stereo to mono for example.
2. Varying the "width" of the center speaker in a surround sound system.

This filter is hard to use, and will require some tinkering before the desired result is obtained. The number of options for this filter depends on the number of output channels:

## MPlayer – The Movie Player for LINUX

*nch* <1-6>

is an integer between 1 and 6 and is used for setting the number of output channels. This option is required, leaving it empty results in a runtime error.

*100:101:102:...110:111:112:...ln0:ln1:ln2:...*

are floating point values between 0 and 1.  $l[i][j]$  determines how much of input channel  $j$  is mixed into output channel  $i$ .

Example 1:

```
mplayer -af pan=1:0.5:0.5 -channels 1 media.avi
```

would down-mix from stereo to mono.

Example 2:

```
mplayer -af pan=3:1:0:1:0.5:0.5 -channels 3 media.avi
```

would give 3 channel output leaving channels 0 and 1 intact, and mix channels 0 and 1 into output channel 2 (which could be sent to a sub-woofer for example).

### 2.3.2.3.8. Sub-woofer

The `sub` filter adds a sub woofer channel to the audio stream. The audio data used for creating the sub-woofer channel is an average of the sound in channel 0 and channel 1. The resulting sound is then low-pass filtered by a 4th order Butterworth filter with a default cutoff frequency of 60Hz and added to a separate channel in the audio stream. Warning: Disable this filter when you are playing DVDs with Dolby Digital 5.1 sound, otherwise this filter will disrupt the sound to the sub-woofer. This filter has two parameters:

*fc* <20-300>

is an optional floating point number used for setting the cutoff frequency for the filter in Hz. The valid range is 20Hz to 300Hz. For the best result try setting the cutoff frequency as low as possible. This will improve the stereo or surround sound experience. The default cutoff frequency is 60Hz.

*ch* <0-5>

is an optional integer between 0 and 5 which determines the channel number in which to insert the sub-channel audio. The default is channel number 5. Observe that the number of channels will automatically be increased to *ch* if necessary.

Example:

```
mplayer -af sub=100:4 -channels 5 media.avi
```

would add a sub-woofer channel with a cutoff frequency of 100Hz to output channel 4.

### 2.3.2.3.9. Surround-sound decoder

Matrix encoded surround sound can be decoded by the `surround` filter. Dolby Surround is an example of a matrix encoded format. Many files with 2 channel audio actually contain matrixed surround sound. To use this feature you need a sound card supporting at least 4 channels. This filter has one parameter:

*d* <0-1000>

is an optional floating point number between 0 and 1000 used for setting the delay time in ms for the rear speakers. This delay should be set as follows: if  $d1$  is the distance from the listening position to the front speakers and  $d2$  is the distance from the listening position to the rear speakers, then the delay

## MPlayer – The Movie Player for LINUX

$d$  should be set to 15ms if  $d1 \leq d2$  and to  $15 + 5*(d1-d2)$  if  $d1 > d2$ . The default value for  $d$  is 20ms.

Example:

```
mplayer -af surround=15 -channels 4 media.avi
```

would add surround sound decoding with 15ms delay for the sound to the rear speakers.

### 2.3.2.3.10. Audio Exporter

This audio filter exports the incoming signal to other processes using memory mapping (`mmap()`). Memory mapped areas contain a header:

```
int nch                /*number of channels*/
int size              /*buffer size*/
unsigned long long counter /*Used to keep sync, it's updated
                        every time new data is exported.*/
```

The rest is payload (non-interleaved) 16bit data.

*mmapped\_file*

The file you want this filter to export to. The default is to map to  
~/.mplayer/mplayer-af\_export.

*nsamples*

Number of samples per channel. The default is 512 samples.

Example:

```
mplayer -af export=/tmp/mplayer-af_export:1024 media.avi
```

would export 1024 samples per channel to /tmp/mplayer-af\_export.

### 2.3.2.4. Audio plugins (deprecated)

## Note

**Audio plugins have been deprecated by audio filters and will be removed soon.**

MPlayer has support for audio plugins. Audio plugins can be used for changing the properties of the audio data before the sound reaches the sound card. They are enabled using the `-aop` switch which takes a `list=plugin1,plugin2,...` argument. The `list` argument is required and determines which plugins should be used and in which order they should be executed. Example:

```
mplayer media.avi -aop list=resample,format
```

would run the sound through the resampling plugin followed by the format plugin.

The plugins can also have switches that change their behavior. These switches are explained in detail in the sections below. A plugin will execute using default settings if its switches are omitted. Here is an example of how to use plugins in combination with plugin specific switches:

```
mplayer media.avi -aop list=resample,format:fout=44100:format=0x8
```

## MPlayer – The Movie Player for LINUX

would set the output frequency of the resample plugin to 44100 Hz and the output format of the format plugin to AFMT\_U8.

Currently audio plugins can not be used in MEncoder.

### 2.3.2.4.1. Up/Downsampling

MPlayer fully supports up/downsampling of the sound. This plugin can be used if you have a fixed frequency sound card or if you are stuck with an old sound card that is only capable of max 44.1 kHz. Limitations in your hardware are not auto detected, so you have to specify the sample frequency explicitly. This plugin has one switch: `fout` which is used for setting the desired output sample frequency. It defaults to 48 kHz, and is given in Hz.

Usage:

```
mplayer media.avi -aop list=resample:fout=freq
```

where `freq` is the frequency in Hz, like 44100.

## Note

The output frequency should not be scaled up from the default value. Scaling up will cause the audio and video streams to be played in slow motion in addition to audio distortion.

### 2.3.2.4.2. Surround Sound decoding

MPlayer has an audio plugin that can decode matrix encoded surround sound. Dolby Surround is an example of a matrix encoded format. Many files with 2 channel audio actually contain matrixed surround sound. To use this feature you need a sound card supporting at least 4 channels.

Usage:

```
mplayer media.avi -aop list=surround
```

### 2.3.2.4.3. Sample format converter

If your sound card driver does not support signed 16-bit int data type, this plugin can be used to change the format to one which your sound card can understand. It has one switch, `format`, which can be set to one of the numbers found in `libao2/afmt.h`. This plugin is hardly ever needed and is intended for advanced users. Keep in mind that this plugin only changes the sample format and not the sample frequency or the number of channels.

Usage:

```
mplayer media.avi -aop list=format:format=outfmt
```

where `outfmt` is the required output format.

### 2.3.2.4.4. Delay

This plugin delays the sound and is intended as an example of how to develop new plugins. It can not be used for anything useful from a users perspective and is mentioned here for the sake of completeness only. Do not use this plugin unless you are a developer.

Note

## MPlayer – The Movie Player for LINUX

### 2.3.2.4.5. Software volume control

This plugin is a software replacement for the volume control, and can be used on machines with a broken mixer device. It can also be used if one wants to change the output volume of MPlayer without changing the PCM volume setting in the mixer. It has one switch `volume` that is used for setting the initial sound level. The initial sound level can be set to values between 0 and 255 and defaults to 101 which equals 0dB amplification. Use this plugin with caution since it can reduce the signal to noise ratio of the sound. In most cases it is best to set the level for the PCM sound to max, leave this plugin out and control the output level to your speakers with the master volume control of the mixer. If there is an external amplifier connected to the computer (this is almost always the case), the noise level can be minimized by adjusting the master level and the volume knob on the amplifier until the hissing noise in the background is gone.

Usage:

```
mplayer media.avi -aop list=volume:volume=0-255
```

This plugin also has compressor or "soft-clipping" capabilities. Compression can be used if the dynamic range of the sound is very high or if the dynamic range of the loudspeakers is very low. Be aware that this feature creates distortion and should be considered a last resort.

Usage:

```
mplayer media.avi -aop list=volume:softclip
```

### 2.3.2.4.6. Extrastereo

This plugin (linearly) increases the difference between left and right channels (like the XMMS extrastereo plugin) which gives some sort of "live" effect to playback.

Usage:

```
mplayer media.avi -aop list=extrastereo  
mplayer media.avi -aop list=extrastereo:mul=3.45
```

The default coefficient (`mul`) is a float number that defaults to 2.5. If you set it to 0.0, you will have mono sound (average of both channels). If you set it to 1.0, sound will be unchanged, if you set it to -1.0, left and right channels will be swapped.

### 2.3.2.4.7. Volume normalizer

This plugin maximizes the volume without distorting the sound.

Usage:

```
mplayer media.avi -aop list=volnorm
```

## 2.4. TV input

This section is about how to enable **watching/grabbing from V4L compatible TV tuner**. See the man page for a description of TV options and keyboard controls.



## 2.4.1. Compilation

1. First, you have to recompile. `./configure` will autodetect kernel headers of v4l stuff and the existence of `/dev/video*` entries. If they exist, TV support will be built (see the output of `./configure`).
2. Make sure your tuner works with another TV software in Linux, for example XawTV.

## 2.4.2. Usage tips

The full listing of the options is available on the manual page. Here are just a few tips:

- Use the `channels` option. An example:

```
-tv channels=26-MTV1,23-TV2
```

Explanation: using this option, only the 26 and 23 channels will be usable, and there will be a nice OSD text upon channel switching, displaying the channel's name. Spaces in the channel name must be replaced by the "\_" character.

- Choose some sane image dimensions. The dimensions of the resulting image should be divisible by 16.
- If you capture the video with the vertical resolution higher than half of the full resolution (i.e. 288 for PAL or 240 for NTSC), make sure you turned deinterlacing on. Otherwise you'll get a movie which is distorted during fast-motion scenes and the bitrate controller will be probably even unable to retain the specified bitrate as the interlacing artifacts produce high amount of detail and thus consume lot of bandwidth. You can enable deinterlacing with `-vf pp=DEINT_TYPE`. Usually `pp=1b` does a good job, but it can be matter of personal preference. See other deinterlacing algorithms in the manual and give it a try.
- Crop out the dead space. When you capture the video, the areas at the edges are usually black or contain some noise. These again consume lots of unnecessary bandwidth. More precisely it's not the black areas themselves but the sharp transitions between the black and the brighter video image which do but that's not important for now. Before you start capturing, adjust the arguments of the `crop` option so that all the crap at the margins is cropped out. Again, don't forget to keep the resulting dimensions sane.
- Watch out for CPU load. It shouldn't cross the 90% boundary for most of the time. If you have a large capture buffer, MEncoder can survive an overload for few seconds but nothing more. It's better to turn off the 3D OpenGL screensavers and similar stuff.
- Don't mess with the system clock. MEncoder uses the system clock for doing A/V sync. If you adjust the system clock (especially backwards in time), MEncoder gets confused and you will lose frames. This is an important issue if you are hooked to a network and run some time synchronization software like NTP. You have to turn NTP off during the capture process if you want to capture reliably.
- Don't change the `outfmt` unless you know what you are doing or your card/driver really doesn't support the default (YV12 colorspace). In the older versions of MPlayer/ MEncoder it was necessary to specify the output format. This issue should be fixed in the current releases and `outfmt` isn't required anymore, and the default suits the most purposes. For example, if you are capturing into DivX using libavcodec and specify `outfmt=RGB24` in order to increase the quality of the captured images, the captured image will be actually later converted back into YV12 so the only thing you achieve is a massive waste of CPU power.
- To specify the I420 colorspace (`outfmt=i420`), you have to add an option `-vc rawi420` due to a fourcc conflict with an Intel Indeo video codec.
- There are several ways of capturing audio. You can grab the sound either using your soundcard via an

## MPlayer – The Movie Player for LINUX

external cable connection between video card and line-in, or using the built-in ADC in the bt878 chip. In the latter case, you have to load the **btaudio** driver. Read the `linux/Documentation/sound/btaudio` file (in the kernel tree, not MPlayer's) for some instructions on using this driver.

- If MEncoder cannot open the audio device, make sure that it is really available. There can be some trouble with the sound servers like arts (KDE) or esd (GNOME). If you have a full duplex soundcard (almost any decent card supports it today), and you are using KDE, try to check the "full duplex" option in the sound server preference menu.

### 2.4.3. Examples

Dummy output, to AALib :)

```
mplayer -tv driver=dummy:width=640:height=480 -vo aa tv://
```

Input from standard V4L:

```
mplayer -tv driver=v4l:width=640:height=480:outfmt=i420 -vc rawi420 -vo xv tv://
```

A more sophisticated example. This makes MEncoder capture the full PAL image, crop the margins, and deinterlace the picture using a linear blend algorithm. Audio is compressed with a constant bitrate of 64kbps, using LAME codec. This setup is suitable for capturing movies.

```
mencoder -tv driver=v4l:width=768:height=576 \  
-ovc lavc -lavcopts vcodec=mpeg4:vbitrate=900 \  
-oac mp3lame -lameopts cbr:br=64 \  
-vf crop=720:544:24:16,pp=lb -o output.avi tv://
```

This will additionally rescale the image to 384x288 and compresses the video with the bitrate of 350kbps in high quality mode. The `vqmax` option looses the quantizer and allows the video compressor to actually reach so low bitrate even at the expense of the quality. This can be used for capturing long TV series, where the video quality isn't so important.

```
mencoder -tv driver=v4l:width=768:height=576 \  
-ovc lavc -lavcopts vcodec=mpeg4:vbitrate=350:vhq:vqmax=31:keyint=300 \  
-oac mp3lame -lameopts cbr:br=48 \  
-vf crop=720:540:24:18,pp=tn/lb,scale=384:288 -sws 1 -o output.avi tv://
```

It's also possible to specify smaller image dimensions in the `-tv` option and omit the software scaling but this approach uses the maximum available information and is a little more resistant to noise. The bt8x8 chips can do the pixel averaging only in the horizontal direction due to a hardware limitation.

## 2.5. Edit Decision Lists (EDL)

The edit decision list (EDL) system allows you to automatically skip or mute sections of videos during playback, based on a movie specific EDL configuration file.

This is useful for those who may want to watch a film in "family-friendly" mode. You can cut out any violence, profanity, Jar-Jar Binks .. from a movie according to your own personal preferences. Aside from this, there are other uses, like automatically skipping over commercials in video files you watch.

The EDL file format is pretty bare-bones. Once the EDL system has reached a certain level of maturity, an XML-based file format will probably be implemented (keeping backwards compatibility with previous EDL

formats).

The maximum number of EDL entries for the current incarnation of EDL is 1000. If you happen to need more, change the `#define MAX_EDL_ENTRIES` in the `edl.h` file.

### 2.5.1. Using an EDL file

Include the `-edl <filename>` flag when you run MPlayer, with the name of the EDL file you want applied to the video.

### 2.5.2. Making an EDL file

The current EDL file format is:

```
[begin second] [end second] [action]
```

Where the seconds are floating-point numbers and the action is either 0 for skip or 1 for mute. Example:

```
5.3    7.1    0
15     16.7   1
420    422    0
```

This will skip from second 5.3 to second 7.1 of the video, then mute at 15 seconds, unmute at 16.7 seconds and skip from second 420 to second 422 of the video. These actions will be performed when the playback timer reaches the times given in the file.

To create an EDL file to work from, use the `-edlout <filename>` flag. During playback, when you want to mark the previous two seconds to skip over, hit **i**. A corresponding entry will be written to the file for that time. You can then go back and fine-tune the generated EDL file.

## Chapter 3. Usage

### Table of Contents

#### 3.1. Command line

#### 3.2. Control

##### 3.2.1. Controls configuration

###### 3.2.1.1. Key names

###### 3.2.1.2. Commands

##### 3.2.2. Control from LIRC

##### 3.2.3. Slave mode

#### 3.3. Streaming from network or pipes

#### 3.4. Remote streams

##### 3.4.1. Compiling the server

##### 3.4.2. Using remote streams

### 3.1. Command line

MPlayer utilizes a complex playtree. It consists of global options written as first, for example

## MPlayer – The Movie Player for LINUX

```
mplayer -vfm 5
```

and options written after filenames, that apply only to the given filename/URL/whatever, for example:

```
mplayer -vfm 5 movie1.avi movie2.avi -vfm 4
```

You can group filenames/URLs together using { and }. It's useful with option `-loop`:

```
mplayer { 1.avi -loop 2 2.avi } -loop 3
```

The above command will play files in this order: 1, 1, 2, 1, 1, 2, 1, 1, 2.

Playing a file:

```
mplayer [options] [path/]filename
```

Playing more files:

```
mplayer [default options] [path/]filename1 [options for filename1] filename2 [options for filename2]
```

Playing VCD:

```
mplayer [options] vcd://trackno [-cdrom-device /dev/cdrom]
```

Playing DVD:

```
mplayer [options] dvd://titleno [-dvd-device /dev/dvd]
```

Playing from the WWW:

```
mplayer [options] http://site.com/file.asf
```

(playlists can be used, too)

Playing from RTSP:

```
mplayer [options] rtsp://server.example.com/streamName
```

Examples:

```
mplayer -vo x11 /mnt/Films/Contact/contact2.mpg
mplayer vcd://2 -cd-rom-device /dev/hdc
mplayer -afm 3 /mnt/DVDtrailers/alien4.vob
mplayer dvd://1 -dvd-device /dev/hdc
mplayer -abs 65536 -delay -0.4 -nobps ~/movies/test.avi
```

## 3.2. Control

MPlayer has a fully configurable, command driven, control layer which lets you control MPlayer with keyboard, mouse, joystick or remote control (using LIRC). See the man page for the complete list of keyboard controls.

### 3.2.1. Controls configuration

MPlayer allows you bind any key/button to any MPlayer command using a simple config file. The syntax consist of a key name followed by a command. The default config file location is `$HOME/.mplayer/input.conf` but it can be overridden using the `-input conf` option (relative path are relative to `$HOME/.mplayer`).

#### Example 3.1. A simple input control file

```
##
## MPlayer input control file
##

RIGHT seek +10
LEFT seek -10
- audio_delay 0.100
+ audio_delay -0.100
q quit
> pt_step 1
< pt_step -1
ENTER pt_step 1 1
```

#### 3.2.1.1. Key names

You can have a full list by running `mplayer -input keylist`.

#### Keyboard

- Any printable character
- SPACE
- ENTER
- TAB
- CTRL
- BS
- DEL
- INS
- HOME
- END
- PGUP
- PGDWN
- ESC
- RIGHT
- LEFT
- UP
- DOWN

#### Mouse (only supported under X)

- MOUSE\_BTN0 (Left button)
- MOUSE\_BTN1 (Right button)
- MOUSE\_BTN2 (Middle button)
- MOUSE\_BTN3 (Wheel)
- MOUSE\_BTN4 (Wheel)

- ...
- MOUSE\_BTN9

### Joystick (support must be enabled at compile time)

- JOY\_RIGHT or JOY\_AXIS0\_PLUS
- JOY\_LEFT or JOY\_AXIS0\_MINUS
- JOY\_UP or JOY\_AXIS1\_MINUS
- JOY\_DOWN or JOY\_AXIS1\_PLUS
- JOY\_AXIS2\_PLUS
- JOY\_AXIS2\_MINUS
- ...
- JOY\_AXIS9\_PLUS
- JOY\_AXIS9\_MINUS

### 3.2.1.2. Commands

You can have a full list of known commands by running **mplayer -input cmdlist**.

- seek (int) val [(int) type=0]  
  
Seek to some place in the movie. Type 0 is a relative seek of +/- val seconds. Type 1 seek to val % in the movie.
- audio\_delay (float) val  
  
Adjust the audio delay of val seconds
- quit  
  
Quit MPlayer
- pause  
  
Pause/unpause the playback
- grap\_frames  
  
Somebody know ?
- pt\_step (int) val [(int) force=0]  
  
Go to next/previous entry in playtree. Val sign tell the direction. If no other entry is available in the given direction it won't do anything unless force is non 0.
- pt\_up\_step (int) val [(int) force=0]  
  
Like pt\_step but it jump to next/previous in the parent list. It's useful to break inner loop in the playtree.
- alt\_src\_step (int) val  
  
When more than one source is available it select the next/previous one (only supported by asx playlist).
- sub\_delay (float) val [(int) abs=0]  
  
Adjust the subtitles delay of +/- val seconds or set it to val seconds when abs is non zero.
- osd [(int) level=-1]

## MPlayer – The Movie Player for LINUX

Toggle osd mode or set it to level when level > 0.

- volume (int) dir

Increase/decrease volume

- contrast (int) val [(int) abs=0]
- brightness (int) val [(int) abs=0]
- hue (int) val [(int) abs=0]
- saturation (int) val [(int) abs=0]

Set/Adjust video parameters. Val range from -100 to 100.

- frame\_drop [(int) type=-1]

Toggle/Set frame dropping mode.

- sub\_visibility

Adjust subtitles visibility.

- sub\_pos (int) val

Adjust subtitles position.

- vobsub\_lang

Change the language of VobSub subtitles.

- vo\_fullscreen

Switch fullscreen mode.

- vo\_ontop

Toggle stay-on-top. Supported by drivers which use X11, except SDL, as well as directx and gl2 under Windows.

- tv\_step\_channel (int) dir

Select next/previous tv channel.

- tv\_step\_norm

Change TV norm.

- tv\_step\_chanlist

Change channel list.

- gui\_loadfile
- gui\_loadsubtitle
- gui\_about
- gui\_play
- gui\_stop
- gui\_playlist
- gui\_preferences
- gui\_skinbrowser

GUI actions

### 3.2.2. Control from LIRC

Linux Infrared Remote Control – use an easy to build home-brewn IR-receiver, an (almost) arbitrary remote control and control your linux box with it! More about it at [www.lirc.org](http://www.lirc.org).

If you have installed the lirc-package, configure will autodetect it. If everything went fine, MPlayer will print a message like "Setting up lirc support..." on startup. If an error occurs it will tell you. If it doesn't tell you anything about LIRC there's no support compiled in. That's it :-)

The application name for MPlayer is – oh wonder – `mplayer`. You can use any `mplayer` commands and even pass more than one command by separating them with `\n`. Don't forget to enable the repeat flag in `.lircrc` when it make sense (seek, volume, etc). Here's an excerpt from my `.lircrc`:

```
begin
    button = VOLUME_PLUS
    prog = mplayer
    config = volume 1
    repeat = 1
end

begin
    button = VOLUME_MINUS
    prog = mplayer
    config = volume -1
    repeat = 1
end

begin
    button = CD_PLAY
    prog = mplayer
    config = pause
end

begin
    button = CD_STOP
    prog = mplayer
    config = seek 0 1\npause
end
```

If you don't like the standard location for the lirc-config file (`~/.lircrc`) use the `-lircconf filename` switch to specify another file.

### 3.2.3. Slave mode

The slave mode allow you to build simple frontend to MPlayer. When enabled (with the `-slave` option) MPlayer will read commands separated by new line (`\n`) from `stdin`.

## 3.3. Streaming from network or pipes

MPlayer can play files from network, using the HTTP, FTP, MMS or RTSP/RTP protocol.

Playing goes by simply using adding the URL to the command line. MPlayer also honors the `http_proxy` environment variable, and uses proxy if available. Proxy usage can also be forced:



## MPlayer – The Movie Player for LINUX

```
mplayer http_proxy://proxy.micorsops.com:3128/http://micorsops.com:80/stream.asf
```

MPlayer can read from stdin (*not* named pipes). This can be for example used to play from FTP:

```
wget ftp://micorsops.com/something.avi -O - | mplayer -
```

### Note

Note: it's also recommended to enable `-cache` when playback from network:

```
wget ftp://micorsops.com/something.avi -O - | mplayer -cache 8192 -
```

## 3.4. Remote streams

Remote streams allow you to access most MPlayer stream type from a remote host. The main purpose of this feature is to make it possible to directly use the CD or DVD drive of another computer on the network (provided you have the required bandwidth). On the downside some stream type (currently TV and MF) are not usable remotely because they are implemented at the demuxer level. It's sad for MF but TV stream would anyway require an insane amount of bandwidth.

### 3.4.1. Compiling the server

After having compiled MPlayer go to the `TOOLS/netstream` directory and enter `make` to build the server binary. You can then copy the netstream binary to the right place on your system (usually `/usr/local/bin` on Linux).

### 3.4.2. Using remote streams

First you have to start the server on the computer you intend to remotely access. Currently the server is very basic and doesn't have any commands line arguments so just enter `netstream`. Now you can for example play the second track of a VCD on the server with :

```
mplayer -cache 5000 mpst://servername/vcd://2
```

You can also access files on this server :

```
mplayer -cache 5000 mpst://servername//usr/local/movies/lol.avi
```

Note that paths which aren't starting with a `/` will be relative to the directory where the server is running. The `-cache` option is not needed but highly recommended.

Be aware that currently the server is not secure at all. So don't complain about the numerous exploits which are possible through this. Instead send some (good) patch to make it better or start writing your own server.

## Chapter 4. Frequently Asked Questions

### 4.1. *Development*

*Q: How do I create a proper patch for MPlayer?*

*Q: How can I support MPlayer development?*

*Q: How can I become an MPlayer developer?*

## MPlayer – The Movie Player for LINUX

Q: Why don't you use autoconf/automake?

### 4.2. Compilation

Q: Are there binary (RPM/deb) packages of MPlayer?

Q: Compilation stops with an error message similar to this one: cfft.c: In function `passf2': cfft.c:556: unable to find a register to spill in class `FLOAT\_REGS' cfft.c:556: this is the insn: (insn 235 233 246 (set (subreg:SF (reg/v:DI 29 rmm0 [110]) 0) (minus:SF (mem:SF (plus:SI (mult:SI (reg:SI 1 edx [112]) (const int 8 [0x8])) (reg/v:f:SI 3 ebx [62])) [4 S4 A32]) (reg:SF 8 st(0) [132]))) 533 {\*fop\_sf 1 nosse} (insn list 232 (nil)) (expr list:REG DEAD (reg:SF 8 st(0) [132]) (nil)))  
cfft.c:556: confused by earlier errors, bailing out

Q: Compilation stops with an error message similar to this one: In file included from mplayer.c:34: mw.h: In function `mplMainDraw': mw.h:209: Internal compiler error in print rtl and abort, at flow.c:6458 Please submit a full bug report, with preprocessed source if appropriate.

Q: Configure ends with this text, and MPlayer won't compile! Your gcc does not support even i386 for '-march' and '-mcpu'

Q: What's the problem with gcc 2.96?

Q: Great, I have gcc 3.0.1 from Red Hat/Mandrake, then I'm fine!

Q: I tried to compile MPlayer, but I got this output: In file included from /usr/include/g++-v3/bits/std\_cwchar.h:42, from /usr/include/g++-v3/bits/fpos.h:40, from /usr/include/g++-v3/bits/char\_traits.h:40, from /usr/include/g++-v3/bits/std\_string.h:41, from /usr/include/g++-v3/string:31, from libwin32.h:36, from DS\_AudioDecoder.h:4, from DS\_AudioDec.cpp:5: /usr/include/wchar.h: In function Long long int wcstoq(const wchar\_t\*, wchar\_t\*\*, int): /usr/include/wchar.h:514: cannot convert `const wchar\_t\* restrict' to `const  
Q: .. gcc 2.96 ... (Yes, some people are STILL flaming about gcc 2.96!)

Q: SDL output doesn't work or compile. The problem is ...

Q: I am still having trouble compiling with SDL support. gcc says something about undefined reference to SDL\_EnableKeyRepeat. What now?

Q: It doesn't compile, and it misses uint64\_t, inttypes.h and similar things ...

Q: I have Linux running on a Pentium III but ./configure doesn't detect SSE ...

Q: I have a Matrox G200/G400/G450/G550, how do I compile/use the mga\_vid driver?

Q: Hmm, strange. When loading the mga\_vid.o kernel module, I found this in the logs: Warning: loading mga\_vid.o will taint the kernel: no license

Q: During 'make', MPlayer complains about X11 libraries. I don't understand, I DO have X installed!?

Q: I can't compile SVGAlib. I'm using kernel 2.3/2.4 ...

Q: I compiled MPlayer with libdvdcss/libdivxdecord support, but when I try to start it, it says: error while loading shared libraries: lib\*.so.0: cannot load shared object file: No such file or directory I checked up on the file and it IS there in /usr/local/lib ...

Q: When compiling MEncoder, it segfaults at linking!

Q: MPlayer dies with segmentation fault upon pthread check!

Q: I'd like to compile MPlayer on Minix!

### 4.3. General questions

Q: Are there any mailing lists on MPlayer?

Q: I've found a nasty bug when I tried to play my favorite video! Who should I inform?

Q: I have problems playing files with the ... codec. Can I use them?

Q: When I start playing, I get this message but everything seems fine: Linux RTC init: ioctl (rtc pie on): Permission denied

Q: There is a timer in the upper left corner. How can I get rid of it?

Q: The -xy or -fs option doesn't work with the x11 driver (-vo x11) ...

Q: What is the meaning of the numbers on the status line?

Q: What if I don't want them to appear?

Q: Why is video\_out cpu usage zero (0%) for some files?

## MPlayer – The Movie Player for LINUX

Q: There are error messages about file not found /usr/local/lib/codecs/...

Q: Umm, what is "IdegCounter"?

Q: And what is "Faszom(C)ounter"?

Q: LIRC doesn't work, because ...

Q: Subtitles are very nice, the most beautiful I've ever seen, but they slow down playing! I know it's unlikely ...

Q: The onscreen display (OSD) is flickering!

Q: What exactly is this libavcodec thing?

Q: But configure tells me Checking for libavcodec ... no

Q: Icewm's taskbar keeps covering the movie in fullscreen mode!

Q: I can't access the GUI menu. I press right click, but I can't access any menu items!

Q: How can I run MPlayer in the background?

### 4.4. Playback problems

Q: I cannot pinpoint the cause of some strange playback problem.

Q: ... works with xine/avifile/... but doesn't with MPlayer.

Q: Audio goes out of sync playing an AVI file.

Q: MPlayer exits with some error when using l3codeca.acm.

Q: My computer plays MS DivX AVIs with resolutions ~ 640x300 and stereo mp3 sound too slow. When I use -nosound option, everything is OK (but quiet).

Q: MPlayer dies with MPlayer interrupted by signal 4 in module: decode\_video

Q: I have problems with [your window manager] and fullscreen xv/xmga/sdl/x11 modes ...

Q: I got this playing MPEG files: Can't find codec for video format 0x10000001!

Q: When starting MPlayer under KDE I just get a black screen and nothing happens. After about one minute the video starts playing.

Q: I have an AVI that produces a gray screen when played with -vc odivx and a green one with -vc divx4.

Q: When I play this movie I get video-audio desync and/or MPlayer crashes with the following message: DEMUXER: Too many (945 in 8390980 bytes) video packets in the buffer!

Q: I have an MJPEG file which works with other players but displays only a black image in MPlayer

Q: When I try to grab from my tuner, it works, but colors are strange. It's OK with other applications.

Q: I have A/V sync problems. Some of my AVIs play fine, but some play with double speed!

Q: All the WMV (or other..) files I play create a green/gray window and there is only sound! MPlayer prints: Detected video codec: [null] drv:0 (NULL codec (no decoding))

Q: I get very strange percentage values (way too big) while playing files on my notebook.

Q: The audio/video gets totally out of sync when I run MPlayer as root on my notebook. It works normal when i run it as a user.

Q: While playing a movie it suddenly gets jerky and I get the following message: Badly interleaved AVI file detected – switching to -ni mode...

Q: How can I play MPEG Layer 2 (mp2) audio files?

### 4.5. Video/audio driver problems (vo/ao)

Q: I have no sound when playing a video and get error messages similar to this one: AO: [oss] 44100Hz 2ch Signed 16-bit (Little-Endian) audio setup: Can't open audio device /dev/dsp: Device or resource busy couldn't open/init audio device -> NOSOUND Audio: no sound!!! Start playing...

Q: What about the DGA driver? I can't find it!

Q: OK, -vo help shows DGA driver, but it complains about permissions. Help me!

Q: When using Xvideo, my Voodoo 3/Banshee says: X Error of failed request: BadAccess (attempt to access private resource denied) Major opcode of failed request: 147 (MIT-SHM) Minor opcode of failed request: 1 (X ShmAttach) Serial number of failed request: 26 Current serial number in output stream: 27

Q: OpenGL (-vo gl) output doesn't work (hang/black window/X11 errors/...).

## MPlayer – The Movie Player for LINUX

Q: I have an nVidia TNT/TNT2 card, and I have a band with strange colors, right under the movie! Whose fault is this?

Q: I have an nVidia XYZ card, and when I click on the GUI's display window to toggle displaying the GUI panel, a black square appears where I clicked. I have the newest driver.

Q: Oh the world is cruel ...! SDL has only x11 target, but not xv!

### 4.6. DVD playback

Q: What about DVD navigation?

Q: While playing a DVD, I encountered this error: mplayer: ifo\_read.c:1143: ifoRead\_C ADT internal: Assertion nfo\_length / sizeof(cell adr t) >= c adt->nr of vobs' failed.

Q: Can I compile libdvread and libdvdcss on my sweet SPARC under Solaris?

Q: What about subtitles? Can MPlayer display them?

Q: How can I set the region code of my DVD-drive? I don't have Windows!

Q: Do I need to be (setuid) root/setuid fibmap mplayer to be able to play a DVD?

Q: Where can I get libdvread and libdvdcss packages?

Q: Is it possible to play/encode only selected chapters?

Q: My DVD playback is sluggish!

### 4.7. Feature requests

Q: If MPlayer is paused and I try to seek or press any key at all, MPlayer ceases to be paused. I would like to be able to seek in the paused movie.

Q: I'd like to seek +/- 1 frames instead of 10 seconds.

Q: How can I make MPlayer remember the options I use for this particular file?

### 4.8. Encoding

Q: How can I encode?

Q: How can I create VCDs?

Q: How can I join two video files?

Q: My tuner works, I can hear the sound and watch the video with MPlayer, but MEncoder doesn't encode audio!

Q: I can't encode DVD subtitles into the AVI!

Q: MEncoder segfaults on startup!

Q: How can I encode only selected chapters from a DVD?

Q: I'm trying to work with 2GB+ files on a VFAT file system. Does it work?

Q: Why is the recommended bitrate printed by MEncoder negative?

Q: I can't encode ASF files to AVI/DivX because it uses 1000 fps?

Q: How can I put subtitles in the output file?

## 4.1. Development

**Q:** How do I create a proper patch for MPlayer?

**A:** We made a [short document](#) describing all the necessary details. Please follow the instructions.

**Q:** How can I support MPlayer development?

**A:** We are more than happy to accept your hardware and software [donations](#). They help us in continuously improving

**Q:** How can I become an MPlayer developer?

**A:** We always welcome coders and documenters. Read the [technical documentation](#) to get a first grasp. Then you should join the [mplayer-dev-eng](#) mailing list and start coding.

**Q:** Why don't you use autoconf/automake?

**A:** We have a modular, handwritten build system. It does a reasonably good job, so why change? Besides, we dislike [people](#).

## 4.2. Compilation

**Q:** Are there binary (RPM/deb) packages of MPlayer?

**A:** See the [Debian](#) and [RPM](#) section for details.

**Q:** Compilation stops with an error message similar to this one:

```
cfft.c: In function `passf2':
cfft.c:556: unable to find a register to spill in class `FLOAT_REGS'
cfft.c:556: this is the insn:
(insn 235 233 246 (set (subreg:SF (reg/v:DI 29 rmm0 [110]) 0)
      (minus:SF (mem:SF (plus:SI (mult:SI (reg:SI 1 edx [112])
      (const_int 8 [0x8])))
      (reg/v:f:SI 3 ebx [62])) [4 S4 A32]
      (reg:SF 8 st(0) [132]))) 533 {*fop_sf_l_nosse} (insn_list
232 (nil)) (expr_list:REG_DEAD (reg:SF 8 st(0) [132])
      (nil)))
cfft.c:556: confused by earlier errors, bailing out
```

**A:** This is a known problem of gcc 3.2, upgrade to 3.3 to solve the problem. How to install gcc is described in the [gcc](#) section. You can use an external FAAD library as described in the [AAC](#) section.

**Q:** Compilation stops with an error message similar to this one:

```
In file included from mplayer.c:34:
mw.h: In function `mplMainDraw':
mw.h:209: Internal compiler error in print_rtl_and_abort, at flow.c:6458
Please submit a full bug report,
with preprocessed source if appropriate.
```

**A:** This is a known problem of gcc 3.0.4, upgrade to 3.1 to solve the problem. How to install gcc is described in the [gcc](#) section.

**Q:** Configure ends with this text, and MPlayer won't compile!

```
Your gcc does not support even i386 for '-march' and '-mcpu'
```

**A:** Your gcc isn't installed correctly, check the `config.log` file for details.

**Q:** What's the problem with gcc 2.96?

**A:** **We strongly discourage the use of gcc 2.96!** Read [this](#) document for details about why Red Hat released gcc 2.96 about. If you still really really want to use it, be sure to get the latest release and give the `--disable-gcc-check` option. Remember that you are on your own from this point. Do **not** report bugs, do **not** ask for help on the mailing lists. Write your own patches in case you run into problems.

**Q:** Great, I have gcc 3.0.1 from Red Hat/Mandrake, then I'm fine!

**A:** No, since there have been/are issues with these compilers as well. To check the status of current compilers' MPlayer section.

**Q:** I tried to compile MPlayer, but I got this output:

```
In file included from /usr/include/g++-v3/bits/std_cwchar.h:42,
      from /usr/include/g++-v3/bits/fpos.h:40,
      from /usr/include/g++-v3/bits/char_traits.h:40,
      from /usr/include/g++-v3/bits/std_string.h:41,
      from /usr/include/g++-v3/string:31,
      from libwin32.h:36,
      from DS_AudioDecoder.h:4,
      from DS_AudioDec.cpp:5:
/usr/include/wchar.h: In function Long long int wcstoq(const wchar_t*,
      wchar_t**, int)':
/usr/include/wchar.h:514: cannot convert `const wchar_t* __restrict' to
`const
```

## MPlayer – The Movie Player for LINUX

**A:** Upgrade your glibc to the latest release. On Mandrake, use 2.2.4–8mdk.

**Q:** .. gcc 2.96 ... (Yes, some people are STILL flaming about gcc 2.96!)

**A:** Quoted from a [mail](#) A'rpi sent to the [mplayer-users](#) list (the word 'ideg' is described below):

And we have idegs. And our idegcounter overflowed again and again.

Unfortunately MPlayer is out of our control. It's used by lamers, Linux users who can't even use Windows, compile a kernel. They installed (with default options) Mandrake or Red Hat or SuSE, and without RTFM! saying 'it doesn't work! help me! please! i'm new to Linux! help! oh! help me!'. We can't stop them, but at least to RTFM and to read the messages of ./configure and MPlayer.

And you clever guys come and flame us with gcc 2.96 and binary packages. Instead of helping users or making solve problems.

Half of our spare/free time is spent by answering silly mails here and making newer tricks and checks to cope with mails.

And there is a balance. On the one side are you, clever guys, saying we are very bad because we don't like to help, on the other side there are the 'new to Linux' guys who are showing us gcc 2.96 is buggy.

Conclusion: We can't be good. Half the people will always say we are bad.

Maybe we should close the project, make it closed source, commercial, and provide install support for it. This is the current work, so development could go faster, and we could earn lots of money with it and buy a big house. Do you want it? It seems.

**Q:** SDL output doesn't work or compile. The problem is ...

**A:** It was tested to work with SDL 1.2.x and may run on SDL 1.1.7+. It does **not** work with any previous version. So if you use an older version, you are on your own.

**Q:** I am still having trouble compiling with SDL support. gcc says something about

```
undefined reference to SDL_EnableKeyRepeat
```

. What now?

**A:** Where did you install the SDL library? If you installed in /usr/local (the default) then edit the top level configure file to -L/usr/local/lib after X\_LIBS=. Now type **make**. You're done!

**Q:** It doesn't compile, and it misses uint64\_t, inttypes.h and similar things ...

**A:** Copy [this inttypes.h](#) to /usr/local/include/ or an equivalent place and try again ...

**Q:** I have Linux running on a Pentium III but ./configure doesn't detect SSE ...

**A:** Only kernel versions 2.4.x support SSE (or try 2.2.19 or newer, but be prepared for problems).

**Q:** I have a Matrox G200/G400/G450/G550, how do I compile/use the mga\_vid driver?

**A:** Read the [mga\\_vid](#) section.

**Q:** Hmm, strange. When loading the mga\_vid.o kernel module, I found this in the logs:

```
Warning: loading mga_vid.o will taint the kernel: no license
```

**A:** The latest kernel modutils require a flag indicating the license (mainly to avoid kernel hackers debugging closed source kernel, modutils and MPlayer).

**Q:** During 'make', MPlayer complains about X11 libraries. I don't understand, I DO have X installed!?

## MPlayer – The Movie Player for LINUX

**A:** ... but you don't have the X development package installed. Or not correctly. It's called `XFree86-devel*` under Debian. Also check if the `/usr/X11` and `/usr/include/X11` symlinks exist (this can be a problem on `Redhat` created with these commands:

```
# ln -sf /usr/X11R6 /usr/X11
# ln -sf /usr/X11R6/include/X11 /usr/include/X11
```

Your distribution may differ from the [Filesystem Hierarchy Standard](#).

**Q:** I can't compile SVGAlib. I'm using kernel 2.3/2.4 ...

**A:** You have to edit SVGAlib's `Makefile.cfg` and comment `BACKGROUND = y` out.

**Q:** I compiled MPlayer with `libvcdcss/libdivxdecore` support, but when I try to start it, it says:

```
error while loading shared libraries: lib*.so.0: cannot load shared object file: No such file
```

I checked up on the file and it IS there in `/usr/local/lib` ...

**A:** Add `/usr/local/lib` to `/etc/ld.so.conf` and run **ldconfig**.

**Q:** When compiling MEncoder, it segfaults at linking!

**A:** This is a linker problem. Upgrading `binutils` should help (2.11.92.\* or newer should be good). Since it is not our fault.

**Q:** MPlayer dies with segmentation fault upon pthread check!

**A:** **chmod 644 /usr/lib/libc.so**

**Q:** I'd like to compile MPlayer on Minix!

**A:** Me too. :)

### 4.3. General questions

**Q:** Are there any mailing lists on MPlayer?

**A:** Yes. See the [mailing lists](#) section.

**Q:** I've found a nasty bug when I tried to play my favorite video! Who should I inform?

**A:** Please read the [bug reporting guidelines](#) and follow the instructions.

**Q:** I have problems playing files with the ... codec. Can I use them?

**A:** Check the [codec status](#), if it doesn't contain your codec, read the [codec documentation](#), especially the [codec import](#)

**Q:** When I start playing, I get this message but everything seems fine:

```
Linux RTC init: ioctl (rtc_pie_on): Permission denied
```

**A:** You need root privileges or a specially set up kernel to use the new timing code. For details see the [RTC](#) section of

**Q:** There is a timer in the upper left corner. How can I get rid of it?

**A:** Press **o** and try the `-osdlevel` option.

**Q:** The `-xy` or `-fs` option doesn't work with the `x11` driver (`-vo x11`) ...

**A:** It does, but you have to explicitly specify software scaling (very slow) with the `-zoom` option. You better use `XF86` specify the `-vm` and the `-fs` option, and you're done. Make sure you have the right modelines in your `XF86Conf` [DGA driver](#) and [SDL's DGA driver](#) work for you. It's much faster. If `SDL's DGA` works, use that, it'll be even faster.

**Q:** What is the meaning of the numbers on the status line?

**A:** Example:

```
A: 2.1 V: 2.2 A-V: -0.167 ct: 0.042 57/57 41% 0% 2.6% 0 4 49%
```

- A: audio position in seconds

## MPlayer – The Movie Player for LINUX

- V: video position in seconds
- A-V: audio-video difference in seconds (delay)
- ct: total A-V sync correction done
- frames played (counting from last seek)
- frames decoded (counting from last seek)
- video codec cpu usage in percent (for slices and DR this includes video\_out)
- video\_out cpu usage
- audio codec cpu usage in percent
- frames needed to drop to maintain A-V sync
- current level of image postprocessing (when using `-autoq`)
- current cache size used (around 50% is normal)

Most of them are for debug purposes and will be removed at some point.

**Q:** What if I don't want them to appear?

**A:** Use the `-quiet` option and read the man page.

**Q:** Why is video\_out cpu usage zero (0%) for some files?

**A:** It's not zero, but it's called from the codec and thus cannot be measured separately. You should try to play the file u... and check the difference to see the video\_out speed.

**A:** You are using Direct Rendering, where the codec renders to the video memory itself. In this case, the decoding per... percentage, too.

**Q:** There are error messages about file not found `/usr/local/lib/codecs/ ...`

**A:** Download the Win32 codecs from our [codecs page](#) (avifile's codec package has a different DLL set) and install it.

**Q:** Umm, what is "IdegCounter"?

**A:** A combination of a Hungarian and an English word. "Ideg" in Hungarian means the same as "nerve" in English, an... like "ydaegh". It was first used to measure the nervousness of A'rpi, after some (umm) "mysterious" disappearance

**Q:** And what is "Faszom(C)ounter"?

**A:** "Fasz" is a Hungarian word you don't want to know, the others are connected to the perverted minds of the MPlayer

**Q:** LIRC doesn't work, because ...

**A:** Are you sure you are using **mplayer** instead of **mplayer\_lirc**? Note that it was **mplayer\_lirc** for a long time, inclu... recently changed back to **mplayer**.

**Q:** Subtitles are very nice, the most beautiful I've ever seen, but they slow down playing! I know it's unlikely ...

**A:** After running `./configure`, edit `config.h` and replace `#undef FAST_OSD` with `#define FAST_OSD`.

**Q:** The onscreen display (OSD) is flickering!

**A:** You use a vo driver with single buffering (x11,xv). With xv, use the `-double` option. Also try `-vf expand`.

**Q:** What exactly is this libavcodec thing?

**A:** See the [libavcodec](#) section.

**Q:** But configure tells me

```
Checking for libavcodec ... no
```

**A:** You need to get libavcodec from FFmpeg's CVS. Read the instructions in the [libavcodec](#) section.

**Q:** Icewm's taskbar keeps covering the movie in fullscreen mode!

**A:** This shouldn't happen anymore, if it still does use the `-fstype layer` option and report it to the [mplayer-users](#) ma...

**Q:** I can't access the GUI menu. I press right click, but I can't access any menu items!

**A:** Are you using FVWM? Try the following:



## MPlayer – The Movie Player for LINUX

1. Start → Settings → Configuration → Base Configuration
2. Set Use Applications position hints to Yes

**Q:** How can I run MPlayer in the background?

**A:** Use:

```
mplayer options filename < /dev/null &
```

### 4.4. Playback problems

**Q:** I cannot pinpoint the cause of some strange playback problem.

**A:** Do you have a stray `codecs.conf` file in `~/mplayer/`, `/etc/`, `/usr/local/etc/` or a similar location? `codecs.conf` files can cause obscure problems. MPlayer will use its builtin one instead.

**Q:** ... works with `xine/avifile/...` but doesn't with MPlayer.

**A:** MPlayer is not `xine/avifile/...` Although these players have some code in common, the codecs (DLL) set, synchronisation is different and should not be compared. If you have a file MPlayer fails to play correctly but works in another player, read [guidelines](#) and upload the file to our FTP server.

**Q:** Audio goes out of sync playing an AVI file.

**A:** Try the `-bps` or `-nobps` option. If it does not improve, read [this](#) and upload the file to FTP.

**Q:** MPlayer exits with some error when using `l3codeca.acm`.

**A:** Check `ldd /usr/local/bin/mplayer` output. If it contains

```
libc.so.6 => /lib/libc.so.6 (0x4???????)
```

where "?" is any number then it's OK, the error is not here. If it is:

```
libc.so.6 => /lib/libc.so.6 (0x00???????)
```

then there is a problem with your kernel/libc. Maybe you are using some security patches (for example Solar Designer's `libseccomp` forces loading libraries to very low addresses. Because `l3codeca.acm` is a non-relocatable DLL, it must be loaded at a fixed address. You should change this. You should use a non-patched kernel, or use MPlayer's `-afm 1` option to disable using `l3codeca.acm`.

**Q:** My computer plays MS DivX AVIs with resolutions ~ 640x300 and stereo mp3 sound too slow. When I use `-nosound` (but quiet).

**A:** Your machine is too slow or your soundcard driver is broken. Consult the documentation to see if you can improve performance.

**Q:** MPlayer dies with

```
MPlayer interrupted by signal 4 in module: decode_video
```

**A:** Try running MPlayer on the machine you compiled on. Or recompile with runtime CPU detection (`./configure --enable-cpu-detection`). Don't use MPlayer on a CPU different from the one it was compiled on, without using the feature mentioned just now.

**Q:** I have problems with [your window manager] and fullscreen `xv/xmga/sdl/x11` modes ...

**A:** Read the [bug reporting guidelines](#) and send us a proper bug report.

**Q:** I got this playing MPEG files: Can't find codec for video format 0x10000001!

**A:** You have an old version of `codecs.conf` in `~/mplayer/`, `/etc/`, `/usr/local/etc/` or similar. Remove it. If you have the `vc=` option or something similar in your config file(s).

**Q:** When starting MPlayer under KDE I just get a black screen and nothing happens. After about one minute the video starts.

**A:** The KDE arts sound daemon is blocking the sound device. Either wait until the video starts or disable the arts-daemon. If you want to use arts sound, specify audio output via our native arts audio driver (`-ao arts`). If it fails or isn't compiled, you can use `artsdsp`. make sure your SDL can handle arts sound. Yet another option is to start MPlayer with `artsdsp`.

## MPlayer – The Movie Player for LINUX

**Q:** I have an AVI that produces a gray screen when played with `-vc odivx` and a green one with `-vc divx4`.

**A:** It's not a DivX file, but an MS MPEG4v3. If you have an old version of `codecs.conf` in `~/.mplayer/`, `/etc` or similar, remove it.

**Q:** When I play this movie I get video–audio desync and/or MPlayer crashes with the following message:

```
DEMUXER: Too many (945 in 8390980 bytes) video packets in the buffer!
```

**A:** This can have multiple reasons.

- Your CPU **and/or** video card **and/or** bus is too slow. MPlayer displays a message if this is the case (and then you can turn it up fast).
- If it is an AVI, maybe it has bad interleaving. Try the `-ni` option.
- Your sound driver is buggy, or you use ALSA 0.5 with `-ao oss`. See the [sound card section](#).
- The AVI has a bad header, try the `-nobps` option, and/or `-mc 0`.

**Q:** I have an MJPEG file which works with other players but displays only a black image in MPlayer

**A:** Use another codec to play the file, try `-vc ffmjpeg`.

**Q:** When I try to grab from my tuner, it works, but colors are strange. It's OK with other applications.

**A:** Your card probably misreports its colorspace capacity. Try with YUY2 instead of default YV12 (see the [TV section](#))

**Q:** I have A/V sync problems. Some of my AVIs play fine, but some play with double speed!

**A:** You have a buggy sound card/driver. Most likely it's fixed at 44100Hz, and you try to play a file which has 22050Hz. Try the `ao=oss` plugin.

**Q:** All the WMV (or other..) files I play create a green/gray window and there is only sound! MPlayer prints:

```
Detected video codec: [null] drv:0 (NULL codec (no decoding))
```

**A:** If you have an old version of `codecs.conf` in `~/.mplayer/`, `/etc/`, `/usr/local/etc/` or similar, remove it.

**Q:** I get very strange percentage values (way too big) while playing files on my notebook.

**A:** It's an effect of the power management / power saving system of your notebook (BIOS, not kernel). Plug the external power on your notebook. You can also try whether `cpufreq` (a SpeedStep interface for Linux) helps you.

**Q:** The audio/video gets totally out of sync when I run MPlayer as root on my notebook. It works normal when i run it as user.

**A:** This is again a power management effect (see above). Plug the external power connector in **before** you power on your notebook. Try the `ao=oss` option.

**Q:** While playing a movie it suddenly gets jerky and I get the following message:

```
Badly interleaved AVI file detected - switching to -ni mode...
```

**A:** Badly interleaved files and `-cache` don't work well together. Try `-nocache`.

**Q:** How can I play MPEG Layer 2 (mp2) audio files?

**A:** You have to use `-rawaudio on:format=0x50`.

### 4.5. Video/audio driver problems (vo/ao)

**Q:** I have no sound when playing a video and get error messages similar to this one:

```
AO: [oss] 44100Hz 2ch Signed 16-bit (Little-Endian)
audio_setup: Can't open audio device /dev/dsp: Device or resource busy
couldn't open/init audio device -> NOSOUND
Audio: no sound!!!
Start playing...
```

**A:** Are you running KDE or GNOME with the ARTS or ESD sound daemon? Try disabling the sound daemon or use the `ao=oss` option to make MPlayer use ARTS or ESD.

## MPlayer – The Movie Player for LINUX

**Q:** What about the DGA driver? I can't find it!

**A:** `./configure` autodetects your DGA driver. If `-vo help` doesn't show DGA, then there's a problem with your `--enable-dga` and read the [DGA](#) section. Alternatively, try SDL's DGA driver with the `-vo sdl:dga` option.

**Q:** OK, `-vo help` shows DGA driver, but it complains about permissions. Help me!

**A:** It works only if running as root! It's a DGA limitation. You should become root (`su -`), and try again. Another solution is to run as root, but it's not recommended!

```
chown root /usr/local/bin/mplayer
chmod 755 /usr/local/bin/mplayer
chmod +s /usr/local/bin/mplayer
```

### Warning

This is a **big** security risk! **Never** do this on a server or on a computer that you do not control completely because of the privileges through SUID root MPlayer. **You have been warned.**

**Q:** When using Xvideo, my Voodoo 3/Banshee says:

```
X Error of failed request: BadAccess (attempt to access private resource denied)
Major opcode of failed request: 147 (MIT-SHM)
Minor opcode of failed request: 1 (X_ShmAttach)
Serial number of failed request: 26
Current serial number in output stream:27
```

**A:** The `tdfx` driver in XFree86 4.0.2/4.0.3 had this bug. This was solved by [bugfix #621 of the XFree86 4.1.0 CVS](#) 1.0 or later. Alternatively, either download (at least) DRI version 0.6 from the [DRI homepage](#), or use CVS DRI.

**Q:** OpenGL (`-vo gl`) output doesn't work (hang/black window/X11 errors/...).

**A:** Your OpenGL driver doesn't support dynamic texture changes (`glTexSubImage`). It's known not to work with nVidia cards. It works with Utah-GLX/DRI and Matrox G400 cards. Also with DRI and Radeon cards. It won't work with DRI and 3DFX cards because of the 256x256 texture size limit.

**Q:** I have an nVidia TNT/TNT2 card, and I have a band with strange colors, right under the movie! Whose fault is this?

**A:** This is a bug of nVidia's binary X driver. These bugs appear **ONLY** with the TNT/TNT2 cards, and we can't do anything about it. Upgrade to the latest nVidia binary driver version. If still bad, complain to nVidia!

**Q:** I have an nVidia XYZ card, and when I click on the GUI's display window to toggle displaying the GUI panel, a black band appears. I have the newest driver.

**A:** Yes, nVidia corrected a previous bug (above), and introduced a new one. Let's congratulate them. UPDATE: According to the latest news, the bug has been fixed.

**Q:** Oh the world is cruel ...! SDL has only `x11` target, but not `xv`!

**A:** Try that `x11` target again. Now try `-vo x11 -fs -zoom`. See the difference? No?! OK, here comes the enlightenment: When `xv` is available, you don't have to worry about it ... Note: With SDL you can force/disable Xv using `-forcexv`.

### 4.6. DVD playback

**Q:** What about DVD navigation?

**A:** Support for `dvdnv` in MPlayer is currently broken, normal playback does work, though. If you want to have fancy navigation, use another player like Xine or Ogle. If you care about DVD navigation, send a [patch](#).

**Q:** While playing a DVD, I encountered this error:

```
mplayer: ifo_read.c:1143: ifoRead_C_ADT_internal: Assertion nfo_length / sizeof(cell_addr_t) >=
```

**A:** This is a known `libdvdread` 0.9.1/0.9.2 bug. Use [libmpdvdkit2](#), which is present in MPlayer source, and used by default.

**Q:** Can I compile libdvdrread and libdvdcss on my sweet SPARC under Solaris?

**A:** Who knows ... It's said to work, so please test it and send feedback. Refer to the documentation of libdvdrread and libdvdcss from the authors of libdvdrread. Use **libmpdvdkit2**, which is present in MPlayer source, and used by default.

**Q:** What about subtitles? Can MPlayer display them?

**A:** Yes. See the [DVD chapter](#).

**Q:** How can I set the region code of my DVD-drive? I don't have Windows!

**A:** Use the [regionset tool](#).

**Q:** Do I need to be (setuid) root/setuid fibmap\_mplayer to be able to play a DVD?

**A:** No. However you must have the proper rights on the DVD device entry (in /dev/).

**Q:** Where can I get libdvdrread and libdvdcss packages?

**A:** You don't need to. Use **libmpdvdkit2**, which is present in the MPlayer source, and used by default. You can get the [Ogle site](#).

**Q:** Is it possible to play/encode only selected chapters?

**A:** Yes, try the `-chapter` option.

**Q:** My DVD playback is sluggish!

**A:** Use the `-cache` option (described in the man page) and try enabling DMA for the DVD drive with the **hdparm** tool.

## 4.7. Feature requests

**Q:** If MPlayer is paused and I try to seek or press any key at all, MPlayer ceases to be paused. I would like to be able to seek while paused.

**A:** This is very tricky to implement without losing A/V synchronization. All attempts have failed so far, but patches are welcome.

**Q:** I'd like to seek +/- 1 frames instead of 10 seconds.

**A:** This won't be done. It was, but then it messed up A/V sync. Feel free to implement it, and send a patch. Don't ask for it to be done.

**Q:** How can I make MPlayer remember the options I use for this particular file?

**A:** Create a file named `movie.avi.conf` with the file-specific options in it and put it in `~/mplayer` or in the same directory as the file.

## 4.8. Encoding

**Q:** How can I encode?

**A:** Read the [MEncoder](#) section.

**Q:** How can I create VCDs?

**A:** Try the `menvcvd` script from the `TOOLS` subdirectory. With it you can encode DVDs or other movies to VCD or SVCD, or burn them directly to CD.

**Q:** How can I join two video files?

**A:** This has been discussed to no end on `mplayer-users`. Go search the [archives](#) for a complete answer. This is a complex task and may vary a lot depending on the kind of files you want to merge. MPEGs can be concatenated into a single file with tools, [avidemux](#) and `avimerge` (part of the [transcode](#) tool set), available that might do the job. You can also try `MEncoder` if you're sharing the same dimensions and codec. Try

```
cat file1 file2 > file3
mencoder -ovc copy -oac copy -o out.avi -forceidx file3.avi
```

**Q:** My tuner works, I can hear the sound and watch the video with MPlayer, but MEncoder doesn't encode audio!

**A:** TV audio encoding for Linux is currently unimplemented, we're working on it. At the moment it works only on BSD.

**Q:** I can't encode DVD subtitles into the AVI!

**A:** You have to specify the `-sid` option correctly!

**Q:** MEncoder segfaults on startup!

**A:** Upgrade DivX4Linux.

**Q:** How can I encode only selected chapters from a DVD?

**A:** Use the `-chapter` option correctly, like: `-chapter 5-7`

**Q:** I'm trying to work with 2GB+ files on a VFAT file system. Does it work?

**A:** No, VFAT doesn't support 2GB+ files.

**Q:** Why is the recommended bitrate printed by MEncoder negative?

**A:** Because the bitrate you encoded the audio with is too large to fit the movie on any CD. Check if you have libmp3lame.

**Q:** I can't encode ASF files to AVI/DivX because it uses 1000 fps?

**A:** Because ASF uses variable frame rate but AVI uses a fixed one, you have to set it by hand using `-ofps`.

**Q:** How can I put subtitles in the output file?

**A:** Just pass the `-sub <filename>` (or `-sid`, `-vobsub`, respectively) option to MEncoder.

## Chapter 5. CD/DVD usage

### Table of Contents

[5.1. \*CD/DVD drives\*](#)

[5.2. \*DVD playback\*](#)

[5.3. \*VCD playback\*](#)

### 5.1. CD/DVD drives

Linux documentation excerpt:

Modern CD-ROM drives can attain very high head speeds, yet some CD-ROM drives are capable of running at reduced speeds. There are several reasons that might make you consider changing the speed of a CD-ROM drive:

- There have been reports of read errors at high speeds, especially with badly pressed CD-ROMs. Reducing the speed can prevent data loss under these circumstances.
- Many CD-ROM drives are annoyingly loud, a lower speed may reduce the noise.

You can reduce the speed of IDE CD-ROM drives with **hdparm** or a program called **setcd**. It works like this:

```
hdparm -E [speed] [cdrom device]
```

```
setcd -x [speed] [cdrom device]
```

If you have root privileges the following command may also help:

```
echo file_readahead:2000000 > /proc/ide/[cdrom device]/settings
```

This sets prefetched file reading to 2MB, which helps with scratched CD-ROMs. If you set it to too high, the drive will continuously spin up and down, and will dramatically decrease the performance. It is recommended that you also tune your CD-ROM drive with **hdparm**:

```
hdparm -d1 -a8 -u1 cdrom device
```

## MPlayer – The Movie Player for LINUX

This enables DMA access, read-ahead, and IRQ unmasking (read the **hdparm** man page for a detailed explanation).

Please refer to `"/proc/ide/cdrom device/settings"` for fine-tuning your CD-ROM.

SCSI drives do not have a uniform way of setting these parameters (Do you know one? Tell us!) There is a tool that works for Plextor SCSI drives.

FreeBSD:

Speed: `cdcontrol [-f device] speed speed`

DMA: `sysctl hw.ata.atapi_dma=1`

## 5.2. DVD playback

For the complete list of available options, please read the man page. The Syntax for a standard Digital Versatile Disc (DVD) is as follows:

```
mplayer dvd://<track> [-dvd-device <device>]
```

Example:

```
mplayer dvd://1 -dvd-device /dev/hdc
```

The default DVD device is `/dev/dvd`. If your setup differs, make a symlink or specify the correct device on the command line with the `-dvd-device` option.

**New-style DVD support (mpdvdkit2).** MPlayer uses `libdvdread` and `libdvdcss` for DVD decryption and playback. These two libraries are contained in the `libmpdvdkit2/` subdirectory of the MPlayer source tree, you do not have to install them separately. We opted for this solution because we had to fix a `libdvdread` bug and apply a patch which adds **cracked CSS keys caching support** to `libdvdcss`. This results in a large speed increase because the keys do not have to be cracked every time before playing.

MPlayer can also use system-wide `libdvdread` and `libdvdcss` libraries, but this solution is **not** recommended, as it can result in bugs, library incompatibilities and slower speed.

### Note

In case of DVD decoding problems, try disabling supermount, or any other such facilities.

**DVD structure.** DVD disks have 2048 bytes per sector with ECC/CRC. They usually have an UDF filesystem on a single track, containing various files (small `.IFO` and `.BUK` files and big (1GB) `.VOB` files). They are real files and can be copied/played from the mounted filesystem of an unencrypted DVD.

The `.IFO` files contain the movie navigation information (chapter/title/angle map, language table, etc) and are needed to read and interpret the `.VOB` content (movie). The `.BUK` files are backups of them. They use **sectors** everywhere, so you need to use raw addressing of sectors of the disc to implement DVD navigation or decrypt the content.

## MPlayer – The Movie Player for LINUX

DVD support needs raw sector-based access to the device. Unfortunately you must (under Linux) be root to get the sector address of a file. That's why we don't use the kernel's filesystem driver at all, instead we reimplement it in userspace. `libdvdread 0.9.x` and `libmpdvdkit` do this. The kernel UDF filesystem driver is not needed as they already have their own builtin UDF filesystem driver. Also the DVD does not have to be mounted as only the raw sector-based access is used.

Sometimes `/dev/dvd` cannot be read by users, so the `libdvdread` authors implemented an emulation layer which transfers sector addresses to filenames+offsets, to emulate raw access on top of a mounted filesystem or even on a hard disk.

`libdvdread` even accepts the mountpoint instead of the device name for raw access and checks `/proc/mounts` to get the device name. It was developed for Solaris, where device names are dynamically allocated.

The default DVD device is `/dev/dvd`. If your setup differs, make a symlink, or specify the correct device on the command line with the `-dvd-device` option.

**DVD authentication.** The authentication and decryption method of the new-style DVD support is done using a patched `libdvdcss` (see above). The method can be specified through the environment variable `DVDCSS_METHOD`, which can be set to `key`, `disk` or `title`.

If nothing is specified it tries the following methods (default: `key`, `title request`):

1. **bus key:** This key is negotiated during authentication (a long mix of `ioctl`s and various key exchanges, crypto stuff) and is used to encrypt the title and disk keys before sending them over the unprotected bus (to prevent eavesdropping). The bus key is needed to get and predecrypt the crypted disk key.
2. **cached key:** MPlayer looks for already cracked title keys which are stored in the `~/ .mplayer/DVDKeys` directory (fast ;).
3. **key:** If no cached key is available, MPlayer tries to decrypt the disk key with a set of included player keys.
4. **disk:** If the key method fails (e.g. no included player keys), MPlayer will crack the disk key using a brute force algorithm. This process is CPU intensive and requires 64 MB of memory (16M 32Bit entries hash table) to store temporary data. This method should always work (slow).
5. **title request:** With the disk key MPlayer requests the crypted title keys, which are inside *hidden sectors* using `ioctl()`. The region protection of RPC-2 drives is performed in this step and may fail on such drives. If it succeeds, the title keys will be decrypted with the bus and disk key.
6. **title:** This method is used if the title request failed and does not rely on any key exchange with the DVD drive. It uses a crypto attack to guess the title key directly (by finding a repeating pattern in the decrypted VOB content and guessing that the plain text corresponding to the first encrypted bytes is a continuation of that pattern). The method is also known as "known plaintext attack" or "DeCSSPlus". In rare cases this may fail because there is not enough encrypted data on the disk to perform a statistical attack or because the key changes in the middle of a title. This method is the only way to decrypt a DVD stored on a hard disk or a DVD with the wrong region on an RPC2 drive (slow).

RPC-1 DVD drives only protect region settings through software. RPC-2 drives have a hardware protection that allows 5 changes only. It might be needed/recommended to upgrade the firmware to RPC-1 if you have a RPC-2 DVD drive. Firmware upgrades can be found on this [firmware page](#). If there is no firmware upgrade available for your device, use the [regionset tool](#) to set the region code of your DVD drive (under Linux).

**Warning:** You can only set the region 5 times.

## 5.3. VCD playback

For the complete list of available options, please read the man page. The Syntax for a standard Video CD (VCD) is as follows:

```
mplayer vcd://<track> [-cdrom-device <device>]
```

Example:

```
mplayer vcd://2 -cdrom-device /dev/hdc
```

The default VCD device is `/dev/cdrom`. If your setup differs, make a symlink or specify the correct device on the command line with the `-cdrom-device` option.

### Note

At least Plextor and some Toshiba SCSI CD-ROM drives have horrible performance reading VCDs. This is because the the `CDROMREADRAW ioctl` is not fully implemented for these drives. If you have some knowledge of SCSI programming, please [help us](#) implement generic SCSI support for VCDs.

In the meantime you can extract data from VCDs with [readvcd](#) and play the resulting file with MPlayer.

**VCD structure.** VCD disks consist of one or more tracks:

- The first track is a small 2048 bytes/sector data track with an iso9660 filesystem, usually containing Windows VCD player programs and maybe other information (images, text, etc).
- The second and other tracks are raw 2324 bytes/sector MPEG (movie) tracks, containing one MPEG PS data packet per sector instead of a filesystem. Similar to audio CD tracks, these tracks **cannot be mounted** (Did you ever mount an audio CD to play it?). As most movies are inside this track, you should try `vcd://2` first.
- There exist VCD disks without the first track (single track and no filesystem at all). They are still playable, but cannot be mounted.

**About .DAT files.** The ~600 MB file visible on the first track of the mounted VCD is not a real file! It is a so called ISO gateway, created to allow Windows to handle such tracks (Windows does not allow raw device access to applications at all). Under Linux you cannot copy or play such files (they contain garbage). Under Windows it is possible as its iso9660 driver emulates the raw reading of tracks in this file. To play a .DAT file you need the kernel driver which can be found in the Linux version of PowerDVD. It has a modified iso9660 filesystem (`vcdfs/iso9660-2.4.X.o`) driver, which is able to emulate the raw tracks through this shadow .DAT file. If you mount the disc using their driver, you can copy and even play .DAT files with MPlayer. But it will not work with the standard iso9660 driver of the Linux kernel! Use `vcd://` instead. Alternatives for VCD copying are the new [cdf](#) kernel driver (not part of the official kernel) that shows CD sessions as image files and [cdrdao](#), a bit-by-bit CD grabbing/copying application.

## Chapter 6. Ports

### Table of Contents

#### 6.1. [Linux](#)

##### 6.1.1. [Debian packaging](#)

#### 5.3. VCD playback



- 6.1.2. RPM packaging
- 6.1.3. ARM
- 6.2. \*BSD
  - 6.2.1. FreeBSD
  - 6.2.2. OpenBSD
  - 6.2.3. Darwin
- 6.3. Sun Solaris
- 6.4. Silicon Graphics Irix
- 6.5. QNX
- 6.6. Windows
  - 6.6.1. Cygwin
  - 6.6.2. MinGW
- 6.7. Mac OS
- 6.8. HP UX
- 6.9. Amiga/MorphOS (GeekGadgets)

## 6.1. Linux

The main development platform is Linux on x86, although MPlayer works on many other Linux ports. Binary packages of MPlayer are available from several sources. However, **none of these packages are supported**. Report problems to the authors, not to us.

### 6.1.1. Debian packaging

To build a Debian package, run the following command in the MPlayer source directory:

```
fakeroot debian/rules binary
```

As root you can then install the .deb package as usual:

```
dpkg -i ../mplayer_version.deb
```

Christian Marillat has been making unofficial Debian MPlayer, MEncoder and font packages for a while, you can (apt-)get them from his [homepage](#).

### 6.1.2. RPM packaging

Dominik Mierzejewski created and maintains official Red Hat RPM packages of MPlayer. They are available from his [homepage](#).

Mandrake RPM packages are available from the [P.L.F.](#) SuSE used to include a crippled version of MPlayer in their distribution. They have removed it in their latest releases. You can get working RPMs from [links2linux.de](#).

### 6.1.3. ARM

MPlayer works on Linux PDAs with ARM CPU e.g. Sharp Zaurus, Compaq Ipaq. The easiest way to obtain MPlayer is to get it from one of the [OpenZaurus](#) package feeds. If you want to compile it yourself, you should look at the [MPlayer](#) and the [libavcodec](#) directory in the OpenZaurus distribution buildroot. These always have the latest Makefile and patches used for building a CVS MPlayer with libavcodec. If you need a GUI

frontend, you can use `xmms–embedded`.

### 6.2. \*BSD

MPlayer runs on FreeBSD, OpenBSD, NetBSD, BSD/OS and Darwin. There are ports/pkgsrc/fink/etc versions of MPlayer available that are probably easier to use than our raw sources.

To build MPlayer you will need GNU make (`gmake` – native BSD make will not work) and a recent version of `binutils`.

If MPlayer complains about not finding `/dev/cdrom` or `/dev/dvd`, create an appropriate symbolic link:

```
ln -s /dev/your_cdrom_device /dev/cdrom
```

To use Win32 DLLs with MPlayer you will need to re-compile the kernel with "`option USER_LDT`" (unless you run FreeBSD-CURRENT, where this is the default).

#### 6.2.1. FreeBSD

If your CPU has SSE, recompile your kernel with "`options CPU_ENABLE_SSE`" (FreeBSD-STABLE or kernel patches required).

#### 6.2.2. OpenBSD

Due to limitations in different versions of `gas` (relocation vs MMX), you will need to compile in two steps: First make sure that the non-native `as` is first in your `$PATH` and do a **`gmake -k`**, then make sure that the native version is used and do **`gmake`**.

#### 6.2.3. Darwin

See the [Mac OS](#) section.

### 6.3. Sun Solaris

MPlayer should work on Solaris 2.6 or newer.

On **UltraSPARCs**, MPlayer takes advantage of their **VIS** extensions (equivalent to MMX), currently only in `libmpeg2`, `libvo` and `libavcodec`, but not in `mp3lib`. You can watch a VOB file on a 400MHz CPU. You'll need [mLib](#) installed.

To build the package you will need GNU make (`gmake`, `/opt/sfw/gmake`), native Solaris make will not work. Typical error you get when building with Solaris' make instead of GNU make:

```
% /usr/ccs/bin/make
make: Fatal error in reader: Makefile, line 25: Unexpected end of line seen
```

On Solaris SPARC, you need the GNU C/C++ Compiler; it does not matter if GNU C/C++ compiler is configured with or without the GNU assembler.

## MPlayer – The Movie Player for LINUX

On Solaris x86, you need the GNU assembler and the GNU C/C++ compiler, configured to use the GNU assembler! The mplayer code on the x86 platform makes heavy use of MMX, SSE and 3DNOW! instructions that cannot be compiled using Sun's assembler `/usr/ccs/bin/as`.

The configure script tries to find out, which assembler program is used by your "gcc" command (in case the autodetection fails, use the `--as=/wherever/you/have/installed/gnu-as` option to tell the configure script where it can find GNU "as" on your system).

Error message from configure on a Solaris x86 system using GCC without GNU assembler:

```
% configure
...
Checking assembler (/usr/ccs/bin/as) ... , failed
Please upgrade(downgrade) binutils to 2.10.1...
```

(Solution: Install and use a gcc configured with `--with-as=gas`)

Typical error you get when building with a GNU C compiler that does not use GNU as:

```
% gmake
...
gcc -c -Iloader -Ilibvo -O4 -march=i686 -mcpu=i686 -pipe -ffast-math
      -fomit-frame-pointer -I/usr/local/include -o mplayer.o mplayer.c
Assembler: mplayer.c
"(stdin)", line 3567 : Illegal mnemonic
"(stdin)", line 3567 : Syntax error
... more "Illegal mnemonic" and "Syntax error" errors ...
```

Due to bugs in Solaris 8, you may not be able to play DVD discs larger than 4 GB:

- The sd(7D) driver on Solaris 8 x86 has a bug when accessing a disk block >4GB on a device using a logical blocksize != DEV\_BSIZE (i.e. CD-ROM and DVD media). Due to a 32Bit int overflow, a disk address modulo 4GB is accessed (<http://groups.yahoo.com/group/solarisonintel/message/22516>). This problem does not exist in the SPARC version of Solaris 8.
- A similar bug is present in the hsf(7FS) filesystem code (aka ISO9660), hsf may not support partitions/disks larger than 4GB, all data is accessed modulo 4GB (<http://groups.yahoo.com/group/solarisonintel/message/22592>). The hsf problem can be fixed by installing patch 109764-04 (sparc) / 109765-04 (x86).

On Solaris with an UltraSPARC CPU, you can get some extra speed by using the CPU's VIS instructions for certain time consuming operations. VIS acceleration can be used in MPlayer by calling functions in Sun's [mediaLib](#).

VIS accelerated operations from mediaLib are used for mpeg2 video decoding and for color space conversion in the video output drivers.

## 6.4. Silicon Graphics Irix

You can either try to install the GNU install program, and (if you did not put it in your global path) then point to the location with:

```
./configure --with-install=/path/and/name/of/install
```

## MPlayer – The Movie Player for LINUX

Or you can use the default install delivered with IRIX 6.5 in which case you will have to edit the `Makefile` by hand a little bit. Change the following two lines:

```
$(INSTALL) -c -m 644 DOCS/mplayer.1 $(MANDIR)/man1/mplayer.1
$(INSTALL) -c -m 644 etc/codecs.conf $(CONFFDIR)/codecs.conf
```

to:

```
$(INSTALL) -m 644 mplayer.1 $(MANDIR)/man1/
$(INSTALL) -m 644 codecs.conf $(CONFFDIR)/
```

And then do (from within the MPlayer source dir):

```
cp DOCS/mplayer.1 . ; cp etc/codecs.conf .
```

and then go on with building and installing.

### 6.5. QNX

Works. You'll need to download SDL for QNX, and install it. Then run MPlayer with `-vo sdl:photon` and `-ao sdl:nto` options, and it should be fast.

The `-vo x11` output will be even slower than on Linux, since QNX has only X *emulation* which is VERY slow. Use SDL.

### 6.6. Windows

Yes, MPlayer runs on Windows under [Cygwin](#) and [MinGW](#). It does not have a GUI yet, but the command line version is almost completely functional. [Patches](#) are always welcome. You should check out the [mplayer-cygwin](#) mailing list for help and latest information.

Best results are achieved with the native DirectX video output driver (`-vo directx`) and the native Windows waveout audio driver (`-ao win32`) as OpenGL does not work and SDL is known to distort sound and image or crash on some systems. If the image is distorted, try turning off hardware acceleration with `-vo directx:noaccel`. Download [DirectX 7 header files](#) to compile the DirectX video output driver.

You can use Win32 codecs and Real Win32 codecs (not Real Linux codecs) if you want to. Put the codecs somewhere in your path or pass `--with-codecsdir=c:/path/to/your/codecs` (alternatively `--with-codecsdir=/path/to/your/codecs` only on Cygwin) to configure. We have had some reports that Real DLLs need to be writable by the user running MPlayer, but only on some systems. Try making them writable if you have problems. QuickTime DLLs also work, but you will have to put them in your Windows system directory (`C:\Windows\system\` or similar).

The Cygwin/MinGW console is rather slow. Redirecting output or using the `-quiet` option has been reported to improve performance on some systems. Direct rendering (`-dxr`) may also help. You can prevent OSD flicker through double buffering with the `-double` option. If playback is jerky, try `-autosync 100`. If some of these options help you, you may want to put them in your config file.

Sascha Sommer releases official Windows binaries from time to time, Joey Parrish makes unofficial Windows packages complete with installer. Look for these in the Windows section of [our projects page](#).

## 6.6.1. Cygwin

Cygwin versions prior to 1.5.0 do not include `inttypes.h`. Put this [inttypes.h](#) in `/usr/include/` in order to make MPlayer compile.

DirectX header files need to be extracted to `/usr/include/` or `/usr/local/include/`.

Instructions and files for making SDL run under Cygwin can be found on the [libsdl site](#).

You can play VCDs by playing the `.DAT` or `.MPG` files that Windows exposes on VCDs. It works like this (adjust for the drive letter of your CD-ROM):

```
mplayer d:/mpegav/avseq01.dat
```

```
mplayer /cygdrive/d/MPEG2/AVSEQ01.MPG
```

DVDs also work, adjust `-dvd-device` for the drive letter of your DVD-ROM:

```
mplayer dvd://<title> -dvd-device '\\.\d:'
```

## 6.6.2. MinGW

Installing a version of MinGW that could compile MPlayer used to be quite tricky, but it works out of the box now. Just install MinGW 3.1.0 or later and MSYS 1.0.9 or later and tell the MSYS postinstall that MinGW is installed.

If you use a version of MinGW before 3.1.0, you need to replace `/mingw/include/sys/types.h` with this [types.h](#).

Extract DirectX header files to `/mingw/include/`.

VCDs and DVDs work almost like Cygwin (adjust for the drive letter of your CD-ROM/DVD-ROM):

```
mplayer d:/mpegav/avseq01.dat
```

```
mplayer /d/MPEG2/AVSEQ01.MPG
```

```
mplayer dvd://<title> -dvd-device /d/
```

## 6.7. Mac OS

Only Mac OS X 10.2 and up is supported by the "raw" MPlayer source. Feel free to make support for older Mac OS versions and send patches!

Apple's modified GCC 3.x is preferred for compiling MPlayer especially when using libavcodec as Apple's modified GCC 2.95.x doesn't support C99 syntax well.

One can get an Aqua GUI for MPlayer together with compiled MPlayer binaries for Mac OS X from the [MPlayerOSX](#) project.

## 6.8. HP UX

Martin Gansser maintains a valuable [HOWTO](#) about building MPlayer on HP-UX. It even has a FAQ section!

Anyway, our "raw" MPlayer source is used to compile on HP-UX without flaws.

## 6.9. Amiga/MorphOS (GeekGadgets)

Nicholas Det at Genesi has done a big and powerful port of MPlayer for MorphOS. Sadly it's based on the 0.90 series.

Get it from [MorphZone](#):

- [MPlayer 0.91 binary](#)
- [MPlayer 0.91 source](#)
- [MEncoder 0.91 binary](#)

# Chapter 7. Encoding with MEncoder

## Table of Contents

- 7.1. [Encoding 2 or 3-pass MPEG-4 \("DivX"\)](#)
- 7.2. [Encoding to MPEG format](#)
- 7.3. [Rescaling movies](#)
- 7.4. [Stream copying](#)
- 7.5. [Fixing AVIs with broken index or interleaving](#)
  - 7.5.1. [Appending multiple AVI files](#)
- 7.6. [Encoding with the libavcodec codec family](#)
- 7.7. [Encoding from multiple input image files \(JPEGs, PNGs or TGAs\)](#)
- 7.8. [Extracting DVD subtitles to Vobsub file](#)
- 7.9. [Preserving aspect ratio](#)
- 7.10. [Custom inter/intra matrices](#)
- 7.11. [Making a high quality MPEG4 \("DivX"\) rip of a DVD movie](#)
  - 7.11.1. [Cropping](#)
  - 7.11.2. [Quality level](#)
  - 7.11.3. [Files over 2GB](#)
  - 7.11.4. [Deinterlacing](#)
  - 7.11.5. [Inverse telecine](#)
  - 7.11.6. [Scaling and aspect ratio](#)
  - 7.11.7. [Summing it up](#)

For the complete list of available MEncoder options and examples, please see the man page. For a series of hands-on examples and detailed guides on using several encoding parameters, read the [encoding-tips](#) that were collected from several mailing list threads on mplayer-users. Search the [archives](#) for a wealth of discussions about all aspects of and problems related to encoding with MEncoder.

## 7.1. Encoding 2 or 3-pass MPEG-4 ("DivX")

**2-pass encoding.** The name comes from the fact that this method encodes the file *twice*. The first encoding (dubbed pass) creates some temporary files (`*.log`) with a size of few megabytes, do not delete them yet (you can delete the AVI). In the second pass, the 2-pass output file is created, using the bitrate data from the temporary files. The resulting file will have much better image quality. If this is the first time you heard about this, you should consult some guides available on the Net.

This example shows how to encode a DVD to a 2-pass MPEG-4 ("DivX") AVI. Just two commands are needed:

```
rm frameno.avi
```

remove this file, which can come from a previous 3-pass encoding (it interferes with current one)

```
mencoder dvd://2 -ovc lavc -lavcopts vcodec=mpeg4:vpass=1 -oac copy -o movie.avi
mencoder dvd://2 -ovc lavc -lavcopts vcodec=mpeg4:vpass=2 -oac copy -o movie.avi
```

**3-pass encoding.** This is an extension of 2-pass encoding, where the audio encoding takes place in a separate pass. This method enables estimation of recommended video bitrate in order to fit on a CD. Also, the audio is encoded only once, unlike in 2-pass mode. The schematics:

1. Remove conflicting temporary file:

```
rm frameno.avi
```

2. *First pass:*

```
mencoder file/DVD -ovc frameno -oac mp3lame -lameopts vbr=3 -o frameno.avi
```

An audio-only avi file will be created, containing **only** the requested audio stream. Don't forget `-lameopts`, if you need to set it. If you were encoding a long movie, MEncoder prints the recommended bitrate values for 650Mb, 700Mb, and 800Mb destination sizes, after this pass finishes.

3. *Second pass:*

```
mencoder file/DVD -oac copy -ovc lavc -lavcopts vcodec=mpeg4:vpass=1:vbitrate=bitrate
```

This is the first pass of video encoding. Optionally specify the video bitrate MEncoder printed at the end of the previous pass.

4. *Third pass:*

```
mencoder file/DVD -oac copy -pass 2 \
    -ovc divx4 -divx4opts br=bitrate
```

This is the second pass of video encoding. Specify the same bitrate as in the previous pass unless you really know what you are doing. In this pass, audio from `frameno.avi` will be inserted into the destination file...and it's all ready!

### Example 7.1. Example of 3-pass encoding

```
rm frameno.avi
```

remove this file, which can come from a previous 3-pass encoding (it interferes with current one)

```
mencoder dvd://2 -ovc frameno -o frameno.avi -oac mp3lame -lameopts vbr=3
mencoder dvd://2 -ovc lavc -lavcopts vcodec=mpeg4:vpass=1 -oac copy -o movie.avi
mencoder dvd://2 -ovc lavc -lavcopts vcodec=mpeg4:vpass=2 -oac copy -o movie.avi
```

## 7.2. Encoding to MPEG format

MEncoder can create MPEG (MPEG-PS) format output files. It's probably useful only with [libavcodec](#)'s *mpeg1video* codec, because players – except MPlayer – expect MPEG1 video, and MPEG1 layer 2 (MP2) audio streams in MPEG files.

This feature is not very useful right now, aside that it probably has many bugs, but the more importantly because MEncoder currently cannot encode MPEG1 layer 2 (MP2) audio, which all other players expect in MPEG files.

To change MEncoder's output file format, use the `-of mpeg` option.

Example:

```
mencoder -of mpeg -ovc lavc -lavcopts vcodec=mpeg1video -oac copy other_options media.avi -o outp
```

## 7.3. Rescaling movies

Often the need to resize movie images' size emerges. Its reasons can be many: decreasing file size, network bandwidth, etc. Most people even do rescaling when converting DVDs or SVCDs to DivX AVI. This is **bad**. Instead of even you doing so, read the [Preserving aspect ratio](#) section.

The scaling process is handled by the `scale` video filter: `-vf scale=width:height`. Its quality can be set with the `-sws` option. If it's not specified, MEncoder will use 0: fast bilinear.

Usage:

```
mencoder input.mpg -ovc lavc -lavcopts vcodec=mpeg4 -vf scale=640:480 -o output.avi
```

## 7.4. Stream copying

MEncoder can handle input streams in two ways: **encode** or **copy** them. This section is about **copying**.

- **Video stream** (option `-ovc copy`): nice stuff can be done :) Like, putting (not converting!) FLI or VIVO or MPEG1 video into an AVI file! Of course only MPlayer can play such files :) And it probably has no real life value at all. Rationally: video stream copying can be useful for example when only the audio stream has to be encoded (like, uncompressed PCM to MP3).
- **Audio stream** (option `-oac copy`): straightforward. It is possible to take an external audio file (MP3, WAV) and mux it into the output stream. Use the `-audiofile filename` option for this.

## 7.5. Fixing AVIs with broken index or interleaving

Easiest thing. We simply copy the video and audio streams, and MEncoder generates the index. Of course this cannot fix possible bugs in the video and/or audio streams. It also fixes files with broken interleaving, thus the `-ni` option won't be needed for them anymore.



Command:

```
mencoder -idx input.avi -ovc copy -oac copy -o output.avi
```

### 7.5.1. Appending multiple AVI files

As a side-effect, the broken AVI fixer function enables MEncoder to append 2 (or more) AVI files:

Command:

```
cat 1.avi 2.avi | mencoder -noidx -ovc copy -oac copy -o output.avi -
```

#### Note

This expects 1.avi and 2.avi to use the same codecs, resolution, stream rate etc, and at least 1.avi must not be broken. You may need to fix your input AVI files first, as described [above](#).

## 7.6. Encoding with the libavcodec codec family

[libavcodec](#) provides simple encoding to a lot of interesting video and audio formats (currently its audio codecs are unsupported). You can encode to the following codecs (more or less up to date):

Codec name	Description
mjpeg	Motion JPEG
ljpeg	Lossless JPEG
h263	H263
h263p	H263 Plus
mpeg4	ISO standard MPEG-4 (DivX 5, XVID compatible)
msmpeg4	pre-standard MPEG-4 variant by MS, v3 (aka DivX3)
msmpeg4v2	pre-standard MPEG-4 by MS, v2 (used in old asf files)
wmv1	Windows Media Video, version 1 (aka WMV7)
wmv2	Windows Media Video, version 2 (aka WMV8)
rv10	an old RealVideo codec
mpeg1video	MPEG1 video
mpeg2video	MPEG2 video
huffyuv	lossless compression
asv1	ASUS Video v1
asv2	ASUS Video v2
ffv1	FFmpeg's lossless video codec

The first column contains the codec names that should be passed after the vcodec config, like: `-lavcopts vcodec=msmpeg4`

An example, with MJPEG compression:

```
mencoder dvd://2 -o title2.avi -ovc lavc -lavcopts vcodec=mjpeg -oac copy
```

## 7.7. Encoding from multiple input image files (JPEGs, PNGs or TGAs)

MEncoder is capable of creating movies from one or more JPEG, PNG or TGA files. With simple framecopy it can create MJPEG (Motion JPEG), MPNG (Motion PNG) or MTGA (Motion TGA) files.

### Explanation of the process:

1. MEncoder *decodes* the input image(s) with `libjpeg` (when decoding PNGs, it will use `libpng`).
2. MEncoder then feeds the decoded image to the chosen video compressor (DivX4, Xvid, `ffmpeg` `msmpeg4`, etc.).

**Examples.** The explanation of the `-mf` option can be found below in the man page.

Creating a DivX4 file from all the JPEG files in the current dir:

```
mencoder -mf on:w=800:h=600:fps=25 -ovc divx4 -o output.avi \*.jpg
```

Creating a DivX4 file from some JPEG files in the current dir:

```
mencoder -mf on:w=800:h=600:fps=25 -ovc divx4 -o output.avi frame001.jpg,frame002.jpg
```

Creating a Motion JPEG (MJPEG) file from all the JPEG files in the current dir:

```
mencoder -mf on:w=800:h=600:fps=25 -ovc copy -o output.avi \*.jpg
```

Creating an uncompressed file from all the PNG files in the current dir:

```
mencoder -mf on:w=800:h=600:fps=25:type=png -ovc raw -o output.avi \*.png
```

### Note

Width must be integer multiple of 4, it's a limitation of the RAW RGB AVI format.

Creating a Motion PNG (MPNG) file from all the PNG files in the current dir:

```
mencoder -mf on:w=800:h=600:fps=25:type=png -ovc copy -o output.avi \*.png
```

Creating a Motion TGA (MTGA) file from all the TGA files in the current dir:

```
mencoder -mf on:w=800:h=600:fps=25:type=tga -ovc copy -o output.avi \*.tga
```

## 7.8. Extracting DVD subtitles to Vobsub file

MEncoder is capable of extracting subtitles from a DVD into Vobsub formatted files. They consist of a pair of files ending in `.idx` and `.sub` and are usually packaged in a single `.rar` archive. MPlayer can play these with the `-vobsub` and `-vobsubid` options.

## MPlayer – The Movie Player for LINUX

You specify the basename (i.e without the `.idx` or `.sub` extension) of the output files with `-vobsubout` and the index for this subtitle in the resulting files with `-vobsuboutindex`.

If the input is not from a DVD you should use `-ifo` to indicate the `.ifo` file needed to construct the resulting `.idx` file.

If the input is not from a DVD and you do not have the `.ifo` file you will need to use the `-vobsubid` option to let it know what language id to put in the `.idx` file.

Each run will append the running subtitle if the `.idx` and `.sub` files already exist. So you should remove any before starting.

### Example 7.2. Copying two subtitles from a DVD while doing 3-pass encoding

```
rm subtitles.idx subtitles.sub
mencoder dvd://1 -vobsubout subtitles -vobsuboutindex 0 -sid 2 -o frameno.avi -ovc frameno
mencoder dvd://1 -oac copy -ovc divx4 -pass 1
mencoder dvd://1 -oac copy -ovc divx4 -pass 2 -vobsubout subtitles -vobsuboutindex 1 -sid 5
```

### Example 7.3. Copying a french subtitle from an MPEG file

```
rm subtitles.idx subtitles.sub
mencoder movie.mpg -ifo movie.ifo -vobsubout subtitles -vobsuboutindex 0 -vobsuboutid fr -sid 1
```

## 7.9. Preserving aspect ratio

DVDs and SVCDs (i.e. MPEG1/2) files contain an aspect ratio value, which describes how the player should scale the video stream, so humans won't have egg heads (ex.:  $480 \times 480 + 4:3 = 640 \times 480$ ). However when encoding to AVI (DivX) files, you have to be aware that AVI headers don't store this value. Rescaling the movie is disgusting and time consuming, there has to be a better way!

There is

MPEG4 has a unique feature: the video stream can contain its needed aspect ratio. Yes, just like MPEG1/2 (DVD, SVCD) and H263 files. Regrettably, there are **no** video players outside which support this attribute of MPEG4, except MPlayer.

This feature can be used only with [libavcodec](#)'s `mpeg4` codec. Keep in mind: although MPlayer will correctly play the created file, other players will use the wrong aspect ratio.

You seriously should crop the black bands over and below the movie image. See the man page for the usage of the `cropdetect` and `crop` filters.

Usage

```
mencoder sample-svcd.mpg -ovc lavc -lavcopts vcodec=mpeg4:autoaspect -vf crop=714:548:0:14 -oac c
```

## 7.10. Custom inter/intra matrices

With this feature of [libavcodec](#) you are able to set custom inter (I-frames/key frames) and intra (P-frames/predicted frames) matrices. It is supported by many of the codecs: `mpeg1video` and

mpeg2video are reported as working.

A typical usage of this feature is to set the matrices preferred by the KVCD specifications.

**The KVCD "Notch" Quantization Matrix:**

Intra:

```

8  9 12 22 26 27 29 34
9 10 14 26 27 29 34 37
12 14 18 27 29 34 37 38
22 26 27 31 36 37 38 40
26 27 29 36 39 38 40 48
27 29 34 37 38 40 48 58
29 34 37 38 40 48 58 69
34 37 38 40 48 58 69 79

```

Inter:

```

16 18 20 22 24 26 28 30
18 20 22 24 26 28 30 32
20 22 24 26 28 30 32 34
22 24 26 30 32 32 34 36
24 26 28 32 34 34 36 38
26 28 30 32 34 36 38 40
28 30 32 34 36 38 42 42
30 32 34 36 38 40 42 44

```

Usage:

```

$ mencoder input.avi -o output.avi -oac copy -ovc lavc -lavcopts inter_matrix=...:intra_matrix=..

$ mencoder input.avi -ovc lavc -lavcopts
vcodec=mpeg2video:intra_matrix=8,9,12,22,26,27,29,34,9,10,14,26,27,29,34,37,
12,14,18,27,29,34,37,38,22,26,27,31,36,37,38,40,26,27,29,36,39,38,40,48,27,
29,34,37,38,40,48,58,29,34,37,38,40,48,58,69,34,37,38,40,48,58,69,79
:inter_matrix=16,18,20,22,24,26,28,30,18,20,22,24,26,28,30,32,20,22,24,26,
28,30,32,34,22,24,26,30,32,32,34,36,24,26,28,32,34,34,36,38,26,28,30,32,34,
36,38,40,28,30,32,34,36,38,42,42,30,32,34,36,38,40,42,44 -oac copy -o svcd.mpg

```

## 7.11. Making a high quality MPEG4 ("DivX") rip of a DVD movie

Ripping a DVD title into a maximally high quality MPEG4 (DivX) file involves many considerations. Below is an example of the process when there is no file size goal (other than perhaps fitting the result into 2GB). libavcodec will be used for the video, and the audio will be copied as is without any changes.

### 7.11.1. Cropping

Play the DVD and run the crop detection filter (`-vf cropdetect`) on it. This gives you a crop rectangle to use for encoding. The reason for cropping is that many movies are not shot in a standard DVD aspect ratio (16/9 or 4/3), or, for whatever reason, the picture does not properly fill the frame. So you want to crop out the pointless black bars when you rip. It also improves the quality of the rip since the sharp edge of the black bars wastes a lot of bits. A common aspect is 2.35, which is cinescope. Most big blockbuster movies have this

aspect ratio.

### 7.11.2. Quality level

Next you need to choose the desired quality level. When there is no need to fit the resulting file on CDs or the like, using constant quantizing AKA constant quality is a good choice. That way each frame is given as much bits as its needs to keep the quality at the desired level, but multiple encoding passes are not needed. With `libavcodec`, you get constant quality by using `-lavcopts vqscale=N`. `vqscale=3` should give you a file below 2GB in size, depending mainly on the movie length and video noisiness (the more noise, the harder it is to compress.)

### 7.11.3. Files over 2GB

If the file resulting from constant quality encoding is over 2GB big, you will have to create an index to be able to view it properly. Either

- play the file with `-forceidx` to create an index on the fly or
- use `-saveidx` to write an index to a file once and `-loadidx` to use it when playing the file.

If this bothers you, you may want to keep the file size below 2GB.

There are two ways to avoid this. You can try encoding again using `vqscale=4` and see if both the resulting file size and picture quality are acceptable. You can also use 2 pass encoding. As you will be copying the audio track as is and hence know its bitrate, and you know the running time of the movie, you can compute the required bitrate to give to the `-lavcopts vbitrate=bitrate` option without using 3 pass encoding.

### 7.11.4. Deinterlacing

If the movie is interlaced, you may want to deinterlace it as part of the ripping. It is debatable whether deinterlacing should be done at this stage. The benefit is that deinterlacing when converting to MPEG4 makes compression better, and viewing easier and less CPU intensive on computer monitors as no deinterlacing is required at that stage.

If deinterlacing at the ripping stage is a good idea depends on the DVD. If the DVD is made from film, which was shot at 24 fps, you can as well deinterlace while ripping. If, however, the original was 50/60 fps video, converting into deinterlaced 23.976/25 fps video will lose information. If you do decide to interlace, you can further experiment with different deinterlacing filters. See <http://www.wieser-web.de/MPlayer/> for examples. A good starting point is `-vf pp=fd`.

If you are both cropping and deinterlacing, deinterlace *before* cropping. Actually, this is not necessary if the crop offset is vertically a multiple of 2 pixels. However with some other filters like `dering` you should always crop last, so it's a good habit to put the crop filter last.

### 7.11.5. Inverse telecine

If you are ripping a PAL DVD, which is 25 fps, you do not need to think about the fps. Just use 25 fps. NTSC DVDs on the other hand are 29.97 fps (often rounded to 30 fps, but that is not what they are). If the movie was shot for TV, you again do not need to touch the fps. But if the movie was shot on film, and hence at (exactly) 24 fps, it has been converted to 29.97 fps when making the DVD. That conversion where 12 fields are added

## MPlayer – The Movie Player for LINUX

to each 24 frames of film is called telecine. For more info about telecine, see a [Google search for "telecine field 23.976"](#).

In case you have such a telecined DVD, you will want to do inverse telecine, that is convert the movie to 23.976 fps (29.97\*4/5). Otherwise camera panning will look jerky and awful. You can use `-ofps 23.976` for this. Anything that is shown in theatres is shot on film and needs inverse telecine, TV shows do not.

### 7.11.6. Scaling and aspect ratio

For best quality, do not scale the movie while ripping. Scaling causes artifacts and makes the file larger. Pixels in DVD movies are not square, so DVD movies include info about the correct aspect ratio. It is possible to store the aspect ratio in the MPEG4 header of the output file. Most video players ignore this info, but MPlayer honors it. So if you are only going to use MPlayer for viewing the ripped file, you do not need to scale the movie, just pass `-lavcopts autoaspect` to MEncoder and things will automatically work right. If you must scale the movie, be careful about getting the size right especially if you do cropping.

### 7.11.7. Summing it up

With all of the above mentioned in mind, a suitable encoding command might be

```
mencoder dvd://1 -aid 128 -oac copy -ovc lavc -lavcopts vcodec=mpeg4:vqscale=3:vhq:v4mv:trell:auto  
-ofps 23.976 -vf crop=720:364:0:56 -o Harry_Potter_2.avi
```

Here `dvd://1` gives the DVD title to rip. Option `-aid 128` says to use audio track 128, and `-oac copy` to copy it as is. You'll have to use MPlayer to find out the right values for these options.

Options `vhq:v4mv:trell` for `-lavcopts` improve quality versus bitrate, but make encoding take longer. Especially `trell` slows encoding down but also increases quality visibly. If you want to deinterlace, add a `pp` filter to `-vf`, for example `-vf pp=fd,crop=720:364:0:56` (in that order). If you don't need inverse telecine, leave out the `-ofps 23.976`.

## Appendix A. Mailing lists

There are some public mailing lists on MPlayer. Unless explicitly stated otherwise the language of these lists is **English**. Please do not send messages in other languages or HTML mail! Message size limit is 80k. If you have something bigger put it up for download somewhere. Click the links to subscribe. On the mailing lists, the same rules about writing and quoting apply as on usenet. Please follow them, it makes the life of those who read your mails a lot easier. If you do not know them please read [HOWTO edit messages](#) or (if you are in a hurry) [Quoting HOWTO](#).

- MPlayer announce list: <http://mplayerhq.hu/mailman/listinfo/mplayer-announce> List for MPlayer announcements. Subscribe here if you want to get announcements about new features.
- MPlayer developers list: <http://mplayerhq.hu/mailman/listinfo/mplayer-dev-eng> This list is about MPlayer development! Talking about interface/API changes, new libraries, code optimization, configure changes is ontopic here. Send patches but not bug reports, user questions, feature requests or flames here to keep the list traffic low.
- MPlayer DOCS list: <http://mplayerhq.hu/mailman/listinfo/mplayer-docs> This list is about MPlayer documentation and translation discussions. Documentation and homepage related CVS logs are also sent here.

## MPlayer – The Movie Player for LINUX

- MPlayer users list: <http://mplayerhq.hu/mailman/listinfo/mplayer-users>
  - ◆ Send bug reports here after reading the [Known Bugs](#) and [bug reporting](#) section).
  - ◆ Send feature requests here (after reading the **whole documentation**).
  - ◆ Send user questions here (after reading the **whole documentation**).
- MPlayer Hungarian users list: <http://mplayerhq.hu/mailman/listinfo/mplayer-felhasznalok>
  - ◆ Hungarian language list
  - ◆ Topic? We'll see about it... mostly flame and RTFM questions up to now :(
- MPlayer & Matrox G200/G400/G450/G550 users:  
<http://mplayerhq.hu/mailman/listinfo/mplayer-matrox> Matrox related questions like
  - ◆ things about mga\_vid
  - ◆ Matrox's official beta drivers (for X 4.x.x)
  - ◆ matroxfb-TVout stuff
- MPlayer & DVB card users: <http://mplayerhq.hu/mailman/listinfo/mplayer-dvb> Things related to the hardware decoder card called DVB (not **DXR3!**).
- MPlayer CVS-log: <http://mplayerhq.hu/mailman/listinfo/mplayer-cvslog> All changes in MPlayer code are automatically sent to this list. Only questions about these changes belong here (if you do not understand why a change is required or you have a better fix or you have noticed a possible bug/problem in the commit).
- MPlayer Cygwin-porting list: <http://mplayerhq.hu/mailman/listinfo/mplayer-cygwin> List for discussion about MPlayer's Cygwin and MinGW port.
- MPlayer OS/2-porting list: <http://mplayerhq.hu/mailman/listinfo/mplayer-os2> List for discussion about MPlayer's OS/2 port.
- MPlayer Weekly News' editors and translators list:  
<http://mplayerhq.hu/mailman/listinfo/mplayer-mwn> List for discussion about the Weekly News releases.

### Note

You can reach the searchable mailing list archives at <http://www.mplayerhq.hu/cgi-bin/htsearch>.

## Appendix B. How to report bugs

### Table of Contents

*B.1. [How to fix bugs](#)*

*B.2. [How to report bugs](#)*

*B.3. [Where to report bugs](#)*

*B.4. [What to report](#)*

*B.4.1. [System Information](#)*

*B.4.2. [Hardware and drivers](#)*

*B.4.3. [Configure problems](#)*

*B.4.4. [Compilation problems](#)*

*B.4.5. [Playback problems](#)*

*B.4.6. [Crashes](#)*

*B.4.6.1. [How to conserve information about a reproducible crash](#)*

*B.4.6.2. [How to extract meaningful information from a core dump](#)*

*B.5. [I know what I am doing...](#)*

Good bug reports are a very valuable contribution to the development of any software project. But just like writing good software, good problem reports involve some work. Please realize that most developers are

extremely busy and receive obscene amounts of email. So while your feedback is crucial in improving MPlayer and very much appreciated, please understand that you have to provide **all** of the information we request and follow the instructions in this document closely.

## B.1. How to fix bugs

If you feel have the necessary skills you are invited to have a go at fixing the bug yourself. Or maybe you already did that? Please read [this short document](#) to find out how to get your code included in MPlayer. The people on the [mplayer-dev-eng](#) mailing list will assist you if you have questions.

## B.2. How to report bugs

First of all please try the latest CVS version of MPlayer as your bug might already be fixed there. Development moves extremely fast, most problems in official releases are reported within days or even hours, so please use **only CVS** to report bugs. This includes binary packages of MPlayer. CVS instructions can be found at the bottom of [this page](#) or in the README. If this did not help please refer to the list of [known bugs](#) and the rest of the documentation. If your problem is not known or not solvable by our instructions, then please report the bug.

Please do not send bug reports privately to individual developers. This is community work and thus there might be several people interested in it. Sometimes other users already experienced your troubles and know how to circumvent a problem even if it is a bug in MPlayer code.

Please describe your problem in as much detail as possible. Do a little detective work to narrow down the circumstances under which the problem occurs. Does the bug only show up in certain situations? Is it specific to certain files or file types? Does it occur with only one codec or is it codec independent? Can you reproduce it with all output drivers? The more information you provide the better are our chances at fixing your problem. Please do not forget to also include the valuable information requested below, we will be unable to properly diagnose your problem otherwise.

An excellent and well written guide to asking questions in public forums is [How To Ask Questions The Smart Way](#) by [Eric S. Raymond](#). There is another called [How to Report Bugs Effectively](#) by [Simon Tatham](#). If you follow these guidelines you should be able to get help. But please understand that we all follow the mailing lists voluntarily in our free time. We are very busy and cannot guarantee that you will get a solution for your problem or even an answer.

## B.3. Where to report bugs

Subscribe to the mplayer-users mailing list: <http://mplayerhq.hu/mailman/listinfo/mplayer-users> and send your bug report to: <mailto:mplayer-users@mplayerhq.hu>

The language of this list is **English**. Please follow the standard [Netiquette Guidelines](#) and **do not send HTML mail** to any of our mailing lists. You will only get ignored or banned. If you do not know what HTML mail is or why it is evil, read this [fine document](#). It explains all the details and has instructions for turning HTML off. Also note that we will not individually CC (carbon-copy) people so it is a good idea to subscribe to actually receive your answer.



## B.4. What to report

You may need to include log, configuration or sample files in your bug report. If some of them are quite big then it is better to upload them to our FTP server in a compressed format (gzip and bzip2 preferred) and include only the path and file name in your bug report. Our mailing lists have a message size limit of 80k, if you have something bigger you have to compress or upload it.

### B.4.1. System Information

- Your Linux distribution or operating system and version e.g.:
  - ◆ Red Hat 7.1
  - ◆ Slackware 7.0 + devel packs from 7.1 ...
- kernel version:

```
uname -a
```

- libc version:

```
ls -l /lib/libc[.]*
```

- gcc and ld versions:

```
gcc -v
ld -v
```

- binutils version:

```
as --version
```

- If you have problems with fullscreen mode:

- ◆ Window manager type and version

- If you have problems with XVIDIX:

- ◆ X colour depth:

```
xdpyinfo | grep "depth of root"
```

- If only the GUI is buggy:

- ◆ GTK version
- ◆ GLIB version
- ◆ libpng version
- ◆ GUI situation in which the bug occurs

### B.4.2. Hardware and drivers

- CPU info (this works on Linux only):

```
cat /proc/cpuinfo
```

- Video card manufacturer and model, e.g.:

- ◆ ASUS V3800U chip: nVidia TNT2 Ultra pro 32MB SDRAM
- ◆ Matrox G400 DH 32MB SGRAM

- Video driver type & version, e.g.:

- ◆ X built-in driver
- ◆ nVidia 0.9.623
- ◆ Utah-GLX CVS 2001-02-17
- ◆ DRI from X 4.0.3
- Sound card type & driver, e.g.:
  - ◆ Creative SBLive! Gold with OSS driver from [oss.creative.com](http://oss.creative.com)
  - ◆ Creative SB16 with kernel OSS drivers
  - ◆ GUS PnP with ALSA OSS emulation
- If in doubt include `lspci -vv` output on Linux systems.

### B.4.3. Configure problems

If you get errors while running `./configure`, or autodetection of something fails, read `configure.log`. You may find the answer there, for example multiple versions of the same library mixed on your system, or you forgot to install the development package (those with the `-dev` suffix). If you think there is a bug, include `configure.log` in your bug report.

### B.4.4. Compilation problems

Please include these files:

- `config.h`
- `config.mak`

Only if compilation fails below one of these directories, include these files:

- `Gui/config.mak`
- `libvo/config.mak`
- `libao2/config.mak`

### B.4.5. Playback problems

Please include the output of MPlayer at verbosity level 1, but remember to **not truncate the output** when you paste it into your mail. The developers need all of the messages to properly diagnose a problem. You can direct the output into a file like this:

```
mplayer -v options filename > mplayer.log 2>&1
```

If your problem is specific to one or more files, then please upload the offender(s) to:  
<ftp://mplayerhq.hu/MPlayer/incoming/>

Also upload a small text file having the same base name as your file with a `.txt` extension. Describe the problem you are having with the particular file there and include your email address as well as the output of MPlayer at verbosity level 1. Usually the first 1–5 MB of a file are enough to reproduce the problem, but to be sure we ask you to:

```
dd if=yourfile of=smallfile bs=1024k count=5
```

It will take the first five megabytes of '**your-file**' and write it to '**small-file**'. Then try again on this small file and if the bug still shows up your sample is sufficient for us. Please **do not ever** send such files via mail!

## MPlayer – The Movie Player for LINUX

Upload it, and send only the path/filename of the file on the FTP-server. If the file is accessible on the net, then sending the **exact** URL is sufficient.

### B.4.6. Crashes

You have to run MPlayer inside **gdb** and send us the complete output or if you have a `core` dump of the crash you can extract useful information from the Core file. Here's how:

#### B.4.6.1. How to conserve information about a reproducible crash

Recompile MPlayer with debugging code enabled:

```
./configure --enable-debug=3  
make
```

and then run MPlayer within `gdb` using:

```
gdb ./mplayer
```

You are now within `gdb`. Type:

```
run -v options-to-mplayer filename
```

and reproduce your crash. As soon as you did it, `gdb` will return you to the command line prompt where you have to enter

```
bt  
disass $pc-32 $pc+32  
info all-registers
```

#### B.4.6.2. How to extract meaningful information from a core dump

Create the following command file:

```
bt  
disass $pc-32 $pc+32  
info all-registers
```

Then simply execute this command:

```
gdb mplayer --core=core -batch --command=command_file > mplayer.bug
```

## B.5. I know what I am doing...

If you created a proper bug report following the steps above and you are confident it is a bug in MPlayer, not a compiler problem or broken file, you have already read the documentation and you could not find a solution, your sound drivers are OK, then you might want to subscribe to the `mplayer-advusers` list and send your bug report there to get a better and faster answer.

Please be advised that if you post newbie questions or questions answered in the manual there, you will be ignored or flamed instead of getting an appropriate answer. So do not flame us and subscribe to `-advusers` only if you really know what you are doing and feel like being an advanced MPlayer user or developer. If you meet these criteria it should not be difficult to find out how to subscribe...

## Appendix C. Known bugs

### Table of Contents

*C.1. Special system/CPU-specific bugs/problems*

*C.2. Various A-V sync and other audio problems*

*C.2.1. General audio delay or jerky sound (exists with all or many files)*

*C.2.2. Audio delay/de-sync specific to one or a few files*

*C.2.3. No sound at all*

*C.2.4. No picture at all (just plain grey/green window)*

*C.2.5. Video-out problems*

### C.1. Special system/CPU-specific bugs/problems

- SIGILL (signal 4) on P3 using 2.2.x kernels:

Problem: kernel 2.2.x doesn't have proper (working) SSE support

Solution: upgrade kernel to 2.4.x

Workaround: `./configure --disable-sse`

- General SIGILL (signal 4):

Problem: you compiled and run MPlayer on different machines (for example compiled on P3 and running on Celeron)

Solution: compile MPlayer on the same machine where you will use it!

Workaround: `./configure --disable-sse` etc. options

- "Internal buffer inconsistency" during MEncoder run:

Problem: known problem when lame < 3.90 was compiled with gcc 2.96 or 3.x.

Solution: use lame >=3.90.

Workaround: compile lame with gcc 2.95.x and remove any already installed lame packages, they may have been compiled with gcc 2.96.

- Messed up MP2/MP3 sound on PPC:

Problem: known GCC miscompilation bug on PPC platforms, no fix yet.

Workaround: use FFmpeg's (slow) MP1/MP2/MP3 decoder (`-ac ffmpeg`)

- sig11 in libmpeg2, when scaling+encoding:

Problem: known GCC 2.95.2 MMX bug, upgrade to 2.95.3.

### C.2. Various A-V sync and other audio problems

### C.2.1. General audio delay or jerky sound (exists with all or many files)

- most common: buggy audio driver! – try to use different drivers, try ALSA 0.9 OSS emulation with `-ao oss`, also try `-ao sdl`, sometimes it helps. If your file plays fine with `-nosound`, then you can be sure it's sound card (driver) problem.

- audio buffer problems (buffer size badly detected)

Workaround: MPlayer's `-abs` option

- samplerate problems – maybe your card doesn't support the samplerate used in your files – try the resampling plugin (`-af resample=...`)

- slow machine (CPU or VGA)

try with `-vo null`, if it plays well, then you have slow VGA card/driver

Workaround: buy a faster card or read this documentation about how to speed up

Also try `-framedrop`

### C.2.2. Audio delay/de-sync specific to one or a few files

- bad file

Workaround:

- ◆ `-ni` or `-nobps` option (for non-interleaved or bad files)

and/or

- ◆ `-mc 0`(required for files with badly interleaved VBR audio)

and/or

- ◆ `-delay` or `+/-` keys at runtime to adjust delay

If none of these help, please upload the file, we'll check (and fix).

- your sound card doesn't support 48kHz playback

Workaround: buy a better sound card... or try to decrease fps by 10% (use `-fps 27` for a 30fps movie) or use the resample plugin

- slow machine(if A-V is not around 0, and the last number in the status line increasing)

Workaround: `-framedrop`

### C.2.3. No sound at all

- your file uses an unsupported audio codec

read the documentation and help us adding support for it

## C.2.4. No picture at all (just plain grey/green window)

- your file uses an unsupported video codec

Workaround: read the documentation and help us adding support for it

- auto-selected codec can't decode the file, try to select another using `-vc` or `-vfm` options
- you try to play DivX 3.x file with OpenDivX decoder or XviD (`-vc odivx`) – install Divx4Linux and recompile player

## C.2.5. Video-out problems

First note: options `-fs` `-vm` and `-zoom` are just recommendations, not (yet) supported by all drivers. So it isn't a bug if it doesn't work. Only a few driver supports scaling/zooming, don't expect this from x11 or dga.

### OSD/sub flickering.

- x11 driver: sorry, it can't be fixed now
- xv driver: use `-double` option

### Green image using `mga_vid` (`-vo mga` / `-vo xmga`).

- `mga_vid` misdetects your card's RAM amount, reload it using `mga_ram_size` option

# Appendix D. MPlayer skin format

## Table of Contents

### D.1. Overview

#### D.1.1. Directories

#### D.1.2. Image formats

#### D.1.3. Skin components

#### D.1.4. Files

### D.2. The skin file

#### D.2.1. Main window and playbar

#### D.2.2. Subwindow

#### D.2.3. Skin menu

### D.3. Fonts

#### D.3.1. Symbols

### D.4. GUI messages

The purpose of this document is to describe the MPlayer skin format. The information contained here might be wrong, for

1. It is not me who wrote the GUI.
2. The GUI is not finished.
3. I might be wrong.

So do not be surprised if something does not work as described here.

Thanks to *Zoltán Ponekker* for his help.

András Mohari <mayday@freemail.hu>

## D.1. Overview

It does not really have anything to do with the skin format, but you should know that MPlayer has **no** builtin skin, so **at least one skin must be installed in order to be able to use the GUI.**

### D.1.1. Directories

The directories searched for skins are (in order):

1. `$(DATADIR)/Skin/`
2. `$(PREFIX)/share/mplayer/Skin/`
3. `~/ .mplayer/Skin/`

Note that the first path may vary according to the way MPlayer was configured (see the `--prefix` and `--datadir` arguments of the **configure** script).

Every skin is installed into its own directory under one of the directories listed above, for example:

```
$(PREFIX)/share/mplayer/Skin/default/
```

### D.1.2. Image formats

Images must be truecolor (24 or 32 bpp) PNGs.

In the main window and in the playbar (see below) you can use images with `transparency': Regions filled with the color #FF00FF (magenta) are fully transparent when viewed by MPlayer. This means that you can even have shaped windows if your X server has the XShape extension.

### D.1.3. Skin components

Skins are quite free-format (unlike the fixed-format skins of Winamp/XMMS, for example), so it is up to you to create something great.

Currently there are three windows to be decorated: the main window, the subwindow, the playbar, and the skin menu (which can be activated by a right click).

- The **main window** and/or the **playbar** is where you can control MPlayer. The background of the window is an image. Various items can (and must) be placed in the window: *buttons*, *potmeters* (sliders) and *labels*. For every item, you must specify its position and size.

A **button** has three states (pressed, released, disabled), thus its image must be divided into three parts vertically. See the button item for details.

A **potmeter** (mainly used for the seek bar and volume/balance control) can have any number of phases by dividing its image into different parts below each other. See hpotmeter and potmeter for details.

## MPlayer – The Movie Player for LINUX

**Labels** are a bit special: The characters needed to draw them are taken from an image file, and the characters in the image are described by a [font description file](#). The latter is a plain text file which specifies the x,y position and size of each character in the image (the image file and its font description file form a font *together*). See [dlabel](#) and [slabel](#) for details.

### Note

All images can have full transparency as described in the section about [image formats](#). If the X server doesn't support the XShape extension, the parts marked transparent will be black. If you'd like to use this feature, the width of the main window's background image must be dividable by 8.

- The **subwindow** is where the movie appears. It can display a specified image if there is no movie loaded (it is quite boring to have an empty window :-)) **Note:** transparency is **not allowed** here.
- The **skin menu** is just a way to control MPlayer by means of menu entries. Two images are required for the menu: one of them is the base image that shows the menu in its normal state, the other one is used to display the selected entries. When you pop up the menu, the first image is shown. If you move the mouse over the menu entries, the currently selected entry is copied from the second image over the menu entry below the mouse pointer (the second image is never shown as a whole).

A menu entry is defined by its position and size in the image (see the section about the [skin menu](#) for details).

There is an important thing not mentioned yet: For buttons, potmeters and menu entries to work, MPlayer must know what to do if they are clicked. This is done by [messages](#) (events). For these items you must define the messages to be generated when they are clicked.

### D.1.4. Files

You need the following files to build a skin:

- The configuration file named [skin](#) tells MPlayer how to put different parts of the skin together and what to do if you click somewhere in the window.
- The background image for the main window.
- Images for the items in the main window (including one or more font description files needed to draw labels).
- The image to be displayed in the subwindow (optional).
- Two images for the skin menu (they are needed only if you want to create a menu).

With the exception of the skin configuration file, you can name the other files whatever you want (but note that font description files must have a `.font` extension).

## D.2. The skin file

As mentioned above, this is the skin configuration file. It is line oriented; comment lines start with a `';` character at the beginning of the line (only spaces and tabs are allowed before the `';`).

The file is made up of sections. Each section describes the skin for an application and has the following form:

```
section = section name
:
```



## MPlayer – The Movie Player for LINUX

```
.  
end
```

Currently there is only one application, so you need only one section: its name is **movieplayer**.

Within this section each window is described by a block of the following form:

```
window = window name  
.br/>.br/>.br/>end
```

where *window name* can be one of these strings:

- **main** – for the main window
- **sub** – for the subwindow
- **menu** – for the skin menu
- **playbar** – playbar

(The sub and menu blocks are optional – you do not need to create a menu or decorate the subwindow.)

Within a window block, you can define each item for the window by a line in this form:

```
item = parameter
```

Where *item* is a string that identifies the type of the GUI item, *parameter* is a numeric or textual value (or a list of values separated by commas).

Putting the above together, the whole file looks something like this:

```
section = movieplayer  
  window = main  
  ; ... items for main window ...  
  end  
  
  window = sub  
  ; ... items for subwindow ...  
  end  
  
  window = menu  
  ; ... items for menu ...  
  end  
  
  window = playbar  
  ; ... items for playbar ...  
  end  
end
```

The name of an image file must be given without leading directories – images are searched for in the *Skin* directory. You may (but you need not) specify the extension of the file. If the file does not exist, MPlayer tries to load the file `<filename>.<ext>`, where `png` and `PNG` are tried for `<ext>` (in this order). The first matching file will be used.

Finally some words about positioning. The main window and the subwindow can be placed in the different

## MPlayer – The Movie Player for LINUX

corners of the screen by giving X and Y coordinates. 0 is top or left, -1 is center and -2 is right or bottom, as shown in this illustration:

```
(0, 0)----(-1, 0)----(-2, 0)
|         |         |
|         |         |
(0,-1)----(-1,-1)----(-2,-1)
|         |         |
|         |         |
(0,-2)----(-1,-2)----(-2,-2)
```

Here is an example to make this clear. Suppose that you have an image called `main.png` that you use for the main window:

```
base = main, -1, -1
```

MPlayer tries to load `main`, `main.png`, `main.PNG` files.

### D.2.1. Main window and playbar

Below is the list of entries that can be used in the 'window = main' ... 'end', and the 'window = playbar' ... 'end' blocks.

```
base = image, X, Y
```

Lets you specify the background image to be used for the main window. The window will appear at the given X, Y position on the screen The window will have the size of the image.

#### Note

These coordinates do not currently work for the display window.

#### Warning

Transparent regions in the image (colored #FF00FF) appear black on X servers without the XShape extension. The image's width must be dividable by 8.

```
button = image, X, Y, width, height, message
```

Place a button of width \* height size at position X, Y. The specified message is generated when the button is clicked. The image given by image must have three parts below each other (according to the possible states of the button), like this:

```
+-----+
|  pressed  |
+-----+
|  released |
+-----+
| disabled  |
+-----+
```

```
decoration = enable/disable
```

Enable or disable window manager decoration of the main window. Default is **disable**.

## Note

This doesn't work for the display window, there is no need to.

*hpotmeter = button, bwidth, bheight, phases, numphases, default, X, Y, width, height, message*

*vpotmeter = button, bwidth, bheight, phases, numphases, default, X, Y, width, height, message*

Place a horizontal (hpotmeter) or vertical (vpotmeter) potmeter of width \* height size at position X,Y. The image can be divided into different parts for the different phases of the potmeter (for example, you can have a pot for volume control that turns from green to red while its value changes from the minimum to the maximum.). hpotmeter can have a button that can be dragged horizontally. The parameters are:

- *button* – the image to be used for the button (must have three parts below each other, like in case of [button](#))
- *bwidth,bheight* – size of the button
- *phases* – the image to be used for the different phases of the hpotmeter. A special value of NULL can be used if you want no such image. The image must be divided into *numphasesparts* vertically like this:

```
+-----+
| phase #1 |
+-----+
| phase #2 |
+-----+
...
+-----+
| phase #n |
+-----+
```

- *numphases* – number of phases stored in the *phases* image
- *default* – default value for hpotmeter (in the range 0 to 100)
- *X,Y* – position for the hpotmeter
- *width,height* – width and height of the hpotmeter
- *message* – the message to be generated when the value of hpotmeter is changed

*potmeter = phases, numphases, default, X, Y, width, height, message*

A hpotmeter without a button. (I guess it is meant to be turned around, but it reacts to horizontal dragging only.) For the description of the parameters see [hpotmeter](#). *phases* can be NULL, but it is quite useless, since you cannot see where the potmeter is set.

*font = fontfile, fontid*

Defines a font. *fontfile* is the name of a font description file with a .fnt extension (do not specify the extension here). *fontid* is used to refer to the font (see [dlabel](#) and [slabel](#)). Up to 25 fonts can be defined.

*slabel = X, Y, fontid, "text"*

Place a static label at the position X,Y. *text* is displayed using the font identified by *fontid*. The text is just a raw string (\$x variables do not work) that must be enclosed between double quotes (but the " character cannot be part of the text). The label is displayed using the font identified by *fontid*.

*dlabel = X, Y, length, align, fontid, "text"*

Place a dynamic label at the position X,Y. The label is called dynamic because its text is refreshed periodically. The maximum length of the label is given by *length* (its height is the height of a character). If the text to be displayed is wider than that, it will be scrolled, otherwise it is aligned within the specified space by the value of the *align* parameter: 0 is for right, 1 is for center, 2 is for left.

## MPlayer – The Movie Player for LINUX

The text to be displayed is given by `text`: It must be written between double quotes (but the " character cannot be part of the text). The label is displayed using the font identified by `fontid`. You can use the following variables in the text:

Variable	Meaning
\$1	play time in <i>hh:mm:ss</i> format
\$2	play time in <i>mmm:ss</i> format
\$3	play time in <i>hh</i> format (hours)
\$4	play time in <i>mm</i> format (minutes)
\$5	play time in <i>ss</i> format (seconds)
\$6	movie length in <i>hh:mm:ss</i> format
\$7	movie length in <i>mmm:ss</i> format
\$8	play time in <i>h:mm:ss</i> format
\$v	volume in <i>xxx.xx%</i> format
\$V	volume in <i>xxx.xx</i> format
\$b	balance in <i>xxx.xx%</i> format
\$B	balance in <i>xxx.xx</i> format
\$\$	the \$ character
\$a	a character according to the audio type (none: n, mono: m, stereo: t)
\$t	track number (in playlist)
\$o	filename
\$f	filename in lower case
\$F	filename in upper case
\$T	a character according to the stream type (file: f, Video CD: v, DVD: d, URL: u)
\$p	the <b>p</b> character (if a movie is playing and the font has the <b>p</b> character)
\$s	the <b>s</b> character (if the movie is stopped and the font has the <b>s</b> character)
\$e	the <b>e</b> character (if playback is paused and the font has the <b>e</b> character)
\$x	movie width
\$y	movie height
\$C	name of the codec used

### Note

The `$a`, `$T`, `$p`, `$s` and `$e` variables all return characters that should be displayed as special symbols (for example, `e` is for the pause symbol that usually looks something like ||). You should have a font for normal characters and a different font for symbols. See the section about [symbols](#) for more information.

### D.2.2. Subwindow

The following entries can be used in the `'window = sub' ... 'end'` block.

*base = image, X, Y, width, height*

The image to be displayed in the window. The window will appear at the given `X`, `Y` position on the screen (`0, 0` is the top left corner). You can specify `-1` for center and `-2` for right (`X`) and bottom (`Y`). The window will be as large as the image. `width` and `height` denote the size of the window; they

## MPlayer – The Movie Player for LINUX

are optional (if they are missing, the window is the same size as the image).

*background* = *R, G, B*

Lets you set the background color. It is useful if the image is smaller than the window. *R, G* and *B* specifies the red, green and blue component of the color (each of them is a decimal number from 0 to 255).

### D.2.3. Skin menu

As mentioned earlier, the menu is displayed using two images. Normal menu entries are taken from the image specified by the *base* item, while the currently selected entry is taken from the image specified by the *selected* item. You must define the position and size of each menu entry through the *menu* item.

The following entries can be used in the 'window = menu'..'end' block.

*base* = *image*

The image for normal menu entries.

*selected* = *image*

The image showing the menu with all entries selected.

*menu* = *X, Y, width, height, message*

Defines the *X, Y* position and the size of a menu entry in the image. *message* is the message to be generated when the mouse button is released over the entry.

## D.3. Fonts

As mentioned in the section about the parts of a skin, a font is defined by an image and a description file. You can place the characters anywhere in the image, but make sure that their position and size is given in the description file exactly.

The font description file (with *.fnt* extension) can have comment lines starting with *;*. The file must have a line in the form

```
image = image
```

Where *image* is the name of the image file to be used for the font (you do not have to specify the extension).

```
"char" = X, Y, width, height
```

Here *X* and *Y* specify the position of the *char* character in the image (0, 0 is the upper left corner). *width* and *height* are the dimensions of the character in pixels.

This example defines the A, B, C characters using *font.png*.

```
; Can be "font" instead of "font.png".
```

```
image = font.png
```

```
; Three characters are enough for demonstration purposes :-)
```

```
"A" = 0,0, 7,13
```

```
"B" = 7,0, 7,13
```

```
"C" = 14,0, 7,13
```

### D.3.1. Symbols

Some characters have special meanings when returned by some of the variables used in `dlabel`. These characters are meant to be shown as symbols so that things like a nice DVD logo can be displayed instead of the character 'd' for a DVD stream.

The following table lists all the characters that can be used to display symbols (and thus require a different font).

Character	Symbol
<b>p</b>	play
<b>s</b>	stop
<b>e</b>	pause
<b>n</b>	no sound
<b>m</b>	mono sound
<b>t</b>	stereo sound
<b>f</b>	stream is a file
<b>v</b>	stream is a Video CD
<b>d</b>	stream is a DVD
<b>u</b>	stream is a URL

### D.4. GUI messages

These are the messages that can be generated by buttons, potmeters and menu entries.

#### Note

Some of the messages might not work as expected (or not work at all). As you know, the GUI is under development.

#### Playback control:

##### *evNext*

Jump to next track in the playlist.

##### *evPause*

Forms a switch together with `evPlaySwitchToPause`. They can be used to have a common play/pause button. Both messages should be assigned to buttons displayed at the very same position in the window. This message pauses playing and the image for the `evPlaySwitchToPause` button is displayed (to indicate that the button can be pressed to continue playing).

##### *evPlay*

Start playing.

##### *evPlaySwitchToPause*

The opposite of `evPauseSwitchToPlay`. This message starts playing and the image for the `evPauseSwitchToPlay` button is displayed (to indicate that the button can be pressed to pause playing).

##### *evPrev*

Jump to previous track in the playlist.

##### *evStop*

Stop playing.

**Seeking:**

*evBackward10sec*

Seek backward 10 seconds.

*evBackward1min*

Seek backward 1 minute.

*evBackward10min*

Seek backward 10 minutes.

*evForward10sec*

Seek forward 10 seconds.

*evForward1min*

Seek forward 1 minute.

*evForward10min*

Seek forward 10 minutes.

*evSetMoviePosition*

Seek to position (can be used by a potmeter; the relative value (0–100%) of the potmeter is used).

**Video control:**

*evDoubleSize*

Set the movie window to double size.

*evFullScreen*

Switch fullscreen mode on/off.

*evNormalSize*

Set the movie window to its normal size.

**Audio control:**

*evDecAudioBufDelay*

Decrease audio buffer delay.

*evDecBalance*

Decrease balance.

*evDecVolume*

Decrease volume.

*evIncAudioBufDelay*

Increase audio buffer delay.

*evIncBalance*

Increase balance.

*evIncVolume*

Increase volume.

*evMute*

Mute/unmute the sound.

*evSetBalance*

Set balance (can be used by a potmeter; the relative value (0–100%) of the potmeter is used).

*evSetVolume*

Set volume (can be used by a potmeter; the relative value (0–100%) of the potmeter is used).

**Miscellaneous:**

***evAbout***

Open the about window.

***evDropSubtitle***

Disables the currently used subtitle.

***evEqualizer***

Turn the equalizer on/off.

***evExit***

Quit the program.

***evIconify***

Iconify the window.

***evLoad***

Load a file (by opening a file browser window, where you can choose a file).

***evLoadPlay***

Does the same as `evLoad`, but it automatically starts playing after the file is loaded.

***evLoadSubtitle***

Loads a subtitle file (with the fileselector)

***evLoadAudioFile***

Loads an audio file (with the fileselector)

***evNone***

Empty message, it has no effect (except maybe in CVS versions :-)).

***evPlayList***

Open/close the playlist window.

***evPlayDVD***

Tries to open the disc in the given DVD-ROM drive.

***evPlayVCD***

Tries to open the disc in the given CD-ROM drive.

***evPreferences***

Open the preferences window.

***evSetAspect***

Sets displayed image aspect.

***evSetURL***

Displays the URL dialog window.

***evSkinBrowser***

Open the skin browser window.

## Appendix E. Developer cries

### Table of Contents

*E.1. [GCC 2.96](#)*

*E.2. [Binary distribution](#)*

*E.3. [nVidia](#)*

*E.4. [Joe Barr](#)*

## E.1. GCC 2.96

**The background:** The GCC 2.95 series is an official GNU release and version 2.95.3 of GCC is the most bug-free in that series. We have never noticed compilation problems that we could trace to gcc-2.95.3. Starting with Red Hat Linux 7.0, **Red Hat** included a heavily patched CVS version of GCC in their distribution and named it **2.96**. Red Hat included this version in the distribution because GCC 3.0 was not



## MPlayer – The Movie Player for LINUX

finished at the time, and they needed a compiler that worked well on all of their supported platforms, including IA64 and s390. The Linux distributor **Mandrake** also followed Red Hat's example and started shipping GCC 2.96 with their Linux–Mandrake 8.0 series.

**The statements:** The GCC team disclaimed any link with GCC 2.96 and issued an official response to GCC 2.96. Many developers around the world began having problems with GCC 2.96, and started recommending other compilers. Examples are MySQL, and avifile. Other interesting links are Linux kernel news flash about kernel 2.4.17 and Voy Forum. MPlayer also suffered from intermittent problems that were all solved by switching to a different version of GCC. Several projects started implementing workarounds for some of the 2.96 issues, but we refused to fix other people's bugs, especially since some workarounds may imply a performance penalty.

GCC 2.96 does not allow | (pipe) characters in assembler comments because it supports Intel as well as AT&T Syntax and the | character is a symbol in the Intel variant. The problem is that it *silently* ignores the whole assembler block. This is supposedly fixed now, GCC prints a warning instead of skipping the block.

**The present:** Red Hat says that GCC 2.96–85 and above is fixed. The situation has indeed improved, yet we still see problem reports on our mailing lists that disappear with a different compiler. In any case it does not matter any longer. Hopefully a maturing GCC 3.x will solve the issue for good. If you want to compile with 2.96 give the `--disable-gcc-checking` flag to `configure`. Remember that you are on your own and **do not report any bugs**. If you do, you will only get banned from our mailing list because we have had more than enough flame wars over GCC 2.96. Please let the matter rest.

If you have problems with GCC 2.96, you can get 2.96–85 packages from the Red Hat ftp server, or just go for the 3.0.4 packages offered for version 7.2 and later. You can also get gcc-3.2.3-11 packages (unofficial, but working fine) and you can install them along the `gcc-2.96` you already have. MPlayer will detect it and use 3.2 instead of 2.96. If you do not want to or cannot use the binary packages, here is how you can compile GCC 3 from source:

1. Go to the GCC mirrors page page and download `gcc-core-XXX.tar.gz` where `XXX` is the version number. This includes the complete C compiler and is sufficient for MPlayer. If you also want C++, Java or some of the other advanced GCC features `gcc-XXX.tar.gz` may better suit your needs.
2. Extract the archive with

```
tar -xvzf gcc-core-XXX.tar.gz
```

3. GCC is not built inside the source directory itself like most programs, but needs a build directory outside the source directory. Thus you need to create this directory via

```
mkdir gcc-build
```

4. Then you can proceed to configure gcc in the build directory, but you need the configure from the source directory:

```
cd gcc-build
../gcc-3.XXX/configure
```

5. Compile GCC by issuing this command in the build directory:

```
make bootstrap
```

6. Now you can install GCC (as root) by typing

```
make install
```

## E.2. Binary distribution

MPlayer previously contained source from the OpenDivX project, which disallows binary redistribution. This code has been removed in version 0.90-pre1 and the remaining file `divx_vbr.c` that is derived from OpenDivX sources has been put under the GPL by its authors as of version 0.90pre9. You are now welcome to create binary packages as you see fit.

Another impediment to binary redistribution was compiletime optimizations for CPU architecture. MPlayer now supports runtime CPU detection (pass the `--enable-runtime-cpudetection` to **configure**). It is disabled by default because it implies a small speed sacrifice, but it is now possible to create binaries that run on different members of the Intel compatible CPU family.

## E.3. nVidia

We dislike the fact that nVidia only provides binary drivers (for use with XFree86), which are often buggy. We have had many reports on mplayer-users about problems related to these closed-source drivers and their poor quality, instability and poor user and expert support. Many of these problems/issues keep appearing repeatedly. We have been contacted by nVidia lately, and they said these bugs do not exist, instability is caused by bad AGP chips, and they received no reports of driver bugs (like the purple line). So if you have a problem with your nVidia card, you are advised to update the nVidia driver and/or buy a new motherboard or ask nVidia to supply open-source drivers. In any case, if you are using the nVidia binary drivers and facing driver related problems, please be aware that you will receive very little help from our side because we have little power to help in this matter.

## E.4. Joe Barr

Joe Barr became infamous in december 2001 by writing a less than favorable MPlayer review called MPlayer: The project from hell. He found MPlayer hard to install, and concluded that the developers were unfriendly and the documentation incomplete and insulting. You be the judge of that. He went on to mention Arpi negatively in his 10 Linux predictions for 2002. In a followup review of xine called A streaming media player for the rest of us he continued stirring up controversy. Ironically at the end of that article he quotes his exchange with Günter Bartsch, the original author of xine, that perfectly summarizes the whole situation:

However, he also went on to say that he was "surprised" by my column about MPlayer and thought it was unfair, reminding me that it is a free software project. "If you don't like it," Bartsch said, "you're free not to use it."

Almost two years later in october 2003 he wrote another review called Mplayer revisited. In it he came to the following conclusions:

I would have to say that there have been improvements in the number of features, in performance, and in documentation. It's still not the easiest install in the world, especially for newbies, but it's a little better than it used to be.

and

But more importantly, I didn't notice any recent comments about user abuse. I think I deserve some of the credit for that, even if I do say so myself. Arpi and the rest of the project team must feel that way too, because they have taken care to remember me in a special section of

## MPlayer – The Movie Player for LINUX

the documentation included in the tarball. Like I said at the start, some things haven't changed at all.

We could not have summarized our feelings towards Joe Barr better: "It's still not the fairest or best researched article in the world, but it's better than it used to be." Hopefully the next time around we will meet each other's expectations. However, the credit for maturity goes to our increasing age only, and maybe to being weary of flame wars.

## **Appendix F. How to send patches**

Please see [this file](#).