

# Codifica animazione con Xvid

## Da Avidemux2

= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorial = Glossario =

---

Questa è una guida per configurare le opzioni del Xvid ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Xvid&prev=\\_t&rurl=translate.google.it&usg=ALkJrhi20j-mx6xwGcKulD1GoqWj8HI3\\_w](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Xvid&prev=_t&rurl=translate.google.it&usg=ALkJrhi20j-mx6xwGcKulD1GoqWj8HI3_w)) (versione 1.xx) codec per la codifica. In particolare queste impostazioni sono ottimizzate per la codifica di animazione, come cartoni animati ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Cartoon\\_animation&prev=\\_t&rurl=translate.google.it&usg=ALkJrhikNa\\_CTeStAg\\_RiCnS0imPS6ym9Q](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Cartoon_animation&prev=_t&rurl=translate.google.it&usg=ALkJrhikNa_CTeStAg_RiCnS0imPS6ym9Q)) e anime. ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Anime&prev=\\_t&rurl=translate.google.it&usg=ALkJrhhuVELq1Q-etdSth7wykP1Obw7XpQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Anime&prev=_t&rurl=translate.google.it&usg=ALkJrhhuVELq1Q-etdSth7wykP1Obw7XpQ))

Queste opzioni di codifica dovrebbe essere disponibile in qualsiasi sistema operativo in cui il codificatore Xvid è installato.

Le opzioni **in grassetto** sono le modifiche specifiche che sono raccomandati. Il *corsivo* sono i commenti sulle opzioni.

### PRINCIPALI:

- Encoding Type: **Two Pass**
- Target size (**MBytes**):?? *(Qualsiasi dimensione del video che si desidera)*
- Quantizzazione:?? *(Utile solo per quantizzazione singolo passaggio based)*
  - ☐ Interlaced *(utile solo se avete bisogno di deinterlacciamento - di solito non uso; filtri sono migliori per il deinterlacciamento)*
  - ☐ Modalità Cartoon *(Solo per i cartoni animati molto colore piatto, in genere si usa NON)*
  - ☒ Turbo Mode *(aumenta la velocità in 2 modalità 2pass pass)*
  - ☐ Scala di grigi
  - ☒ Chroma Optimizer
  - ☐ Packed bitstream

### Stima del movimento:

- Cerca moto di precisione: **"6 - Ultra High"**
- VHQ Mode: **"4 - ricerca Wide"**
  - ☒ Chroma Motion
  - ☒ 4MV
  - ☒ HQ AC
  - I-frame interval - Min: **1** Max: **300**
  - **NOTA DI CONFIGURAZIONE** - Numeri di framerate
    - NTSC [Americano e Giapponese] = 300
    - PAL [europea] o FILM [alcuni DVD] = 240
    - HDTV = 1200
  - Advanced Simple Profile
    - Al numero di fotogrammi B: **3** *(tre Usa come il numero minimo, migliorando notevolmente la qualità video)*
    - ☐ QPel *(Supporti per pixel trimestre)*
    - ☒ GMC *(Se si intende solo la riproduzione su PC usarlo, se si vuole essere compatibili con il lettore DVD standalone non)*
    - ☒ bvhq *(produce più bello che cercano B-frame)*

**Quantizzazione:**

- ☒ H.263 *(migliore per i cartoni animati e anime)*
- ☐ MPEG *(generalmente migliore per l'azione dal vivo)*
- ☒ quantizzazione Treillis
- Quantization restrizioni
  - I-frame quantizer Min: **2** *(o 1 per CD completo, richiede la sperimentazione)*
  - I-frame quantizer Max: **31**
  - P-frame quantizer Min: **2** *(o 1 per CD completo, richiede la sperimentazione)*
  - P-frame quantizer Max: **31**
  - B-frame quantizer Min: **2** *(o 1 per CD completo, richiede la sperimentazione)*
  - B-frame quantizer Max: **31**

**Secondo passo:**

- A due passi tuning
  - I-frame Boost (%): **10**
  - I-frame più vicino ... (frames): **1**
  - ... sono ridotti (%): **20**
  - Miglioramento overflow Max (%): **5 (default)**
  - Degrado overflow Max (%): **5 (default)**
- Curve di compressione
  - Scene ad alto bitrate (%): **0**
  - A basso bitrate scene di miglioramento (%): **0**
  - Forza il controllo di overflow (%): **5** *(20 è un bene per i film completo - richiede ulteriori test)*

---

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorial = Glossario =-  
 Estratto da "[http://www.avidemux.org/admWiki/index.php?title=Encoding\\_animation\\_with\\_Xvid](http://www.avidemux.org/admWiki/index.php?title=Encoding_animation_with_Xvid)"

---

- Ultima modifica 01:12, 20 gennaio 2008.
- Questa pagina è stata letta 35.630 volte.
- Tutela della privacy
- A proposito di Avidemux2
- Disclaimer

# Conversione in DVD

## Da Avidemux2

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-

---

Un passo interattivo-by-step creato da LoRd\_MuldeR è disponibile anche in [1] ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://mulder.dummwiedeutsch.de/etc/adm\\_wink/avi2dvd.htm&prev=\\_t&rurl=translate.google.it&usg=ALkJrhgc-YA3y9PKNxIpBlm7BuQuji0u5g](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://mulder.dummwiedeutsch.de/etc/adm_wink/avi2dvd.htm&prev=_t&rurl=translate.google.it&usg=ALkJrhgc-YA3y9PKNxIpBlm7BuQuji0u5g)) (Flash obbligatorio).

### Sommario

- 1 Creare un DVD
- 2 Il modo in gran parte automatico
- 3 Preparare il video
- 4 Selezione del codec video
- 5 Preparare l'audio
- 6 Selezionare il formato di uscita, semplice caso
- 7 Selezione del formato di output, elaborate caso

## Creazione di un DVD

Rivediamo un paio di fatti, prima di entrare nei dettagli succosi.

- Avidemux verrà creato il file MPEG per voi.
- Avrete bisogno di un programma di authoring per convertire MPEG che per la gerarchia dei file DVD. La maggior parte di loro sono con dvdauthor. Ad esempio è possibile utilizzare Varsha, KDvdAuthor o qualsiasi altra cosa.

## Il modo in cui la maggior parte automatico

Utilizzare la funzione Auto-> DVD menu. Sarà l'installazione la maggior parte alle inadempienze ragionevole. L'unica cosa che a sinistra vi sarà quello di impostare il codec video di 2pass modalità e immettere la dimensione finale desiderata.

## Preparare il video

Per essere compatibile con DVD, la risoluzione deve essere:

- 352 \* 480 o 720 \* 480 \* 704 o 480 per NTSC
- 352 \* 576 o 720 o 576 \* 704 \* 576 per PAL / SECAM

Come faccio a sapere se è PAL o NTSC?

Fare clic sul pulsante Info (File-> Proprietà). Se il "Frame Rate" campo è 25, è PAL o SECAM. Se si tratta di 23,976 o 29.97, è NTSC.

Se non è esattamente uno di questi valori, si dovrà utilizzare il Ricampiona fps filtro per convertire in PAL o NTSC.

Ora che avete individuato il tipo di video, è necessario ridimensionare ad una adeguata larghezza / altezza. Il modo più

semplice è quello di utilizzare la res "DVD" il pulsante nella finestra di dialogo Filtri video che lo farà automaticamente.

**Avvertenza:** Non è possibile ridimensionare ciecamente a 720 \* 480 o qualsiasi altra cosa. Si deve prendere in considerazione il rapporto di aspetto di farlo correttamente. E il rapporto di aspetto non è la stessa per PAL e NTSC per. La cosa migliore è lasciare che "res DVD" di farlo per voi.

## Selezionando il codec video

Si dispone di due famiglie di codec, che sono adatti per la creazione di DVD: "DVD (lavc)", che sta usando lavcodec, o "DVD", che sta utilizzando libmpeg2enc. Entrambi hanno i loro pro e contro:

- DVD (lavc): più veloce, una buona, utilizzare solo **2 PASSA IN MODALITA 'PER DVD.**
- DVD: lento, buono, può essere tranquillamente utilizzato in modalità CQ o in modalità CBR.

Per i DVD il bitrate totale deve essere inferiore a 9,8 Mb / s. Quindi, per dirla semplicemente un bitrate massimo di circa 9 Mb va bene. Assumendo che si intende utilizzare la modalità 2 pass con DVD (lavc) è necessario inserire il video una sola dimensione finale. Perciò è meglio usare il primo calcolatore, che calcola il video taglia da la dimensione finale.

## Preparare l'audio

Per i DVD, audio deve essere di 48 kHz e una delle:

- AC3 (FFM AC3)
- MP2 (FFM MP2 o twolame)
- LPCM (WAV LPCM)

Se la traccia audio dal video di origine è adatta, la migliore potrebbe essere quello di usarlo e lasciare che l'audio da copiare. Else scegliere il formato preferito (per me sarebbe twolame).

Non dimenticare di impostare il filtro audio. Setup downmix, se necessario, e non dimenticare che l'audio deve essere di 48 kHz. Utilizzare un filtro di ricampionamento se necessario.

Così, per esempio, potrete impostare downmix Dolby Prologic 2, ricampionamento a 48 kHz, twolame codec a 160 kb / s.

## Selezionando il formato di uscita, semplice caso

Per i DVD, se si è soddisfatti con una sola traccia audio e nessun sottotitolo, il migliore sarebbe quella di selezionare "MPEG PS A + V" come formato di output utilizzando il menu Formato. Il file MPEG risultante sarà adatto per l'autoring. Basta premere il pulsante Salva e otterrete foo.mpeg che può essere scritto in seguito con il vostro programma di authoring preferito.

## Selezionando il formato di output, elaborate caso

Facendo una traccia audio multi è un po 'diversa, ma non complicati. In primo luogo è salvare l'audio, con l'Audio-> menu Save (naturalmente, tutto quanto sopra audio relativo è necessaria). Otterrete foo\_1.mp2 file per esempio.

Ripetere l'operazione per tutte le tracce che si desidera convertire, cambiare la traccia audio con Audio-> Main Track. Alla fine avrete foo\_1.mp2, foo\_2.mp2 ecc ..

Ora selezionate "Video MPEG" come contenitore di uscita e premere il pulsante Salva. Si salva come foo.m2v per esempio.

Una volta che hai tutti i file, il tempo di accendere mplex per creare il file MPEG. La sintassi sarà

```
mplex -f 9 -i final.mpg foo.m2v foo_1.mp2 foo_2.mp2 ecc ..
```

final.mpg è quindi adatto per l'autoring.

---

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-  
Estratto da "[http://www.avidemux.org/admWiki/index.php?title=Converting\\_to\\_DVD](http://www.avidemux.org/admWiki/index.php?title=Converting_to_DVD)"

---

- Ultima modifica 07:10, 2 dicembre 2007.
- Questa pagina è stata letta 110.460 volte.
- Tutela della privacy
- A proposito di Avidemux2
- Disclaimer

# Deinterlacciamento Video

## Da Avidemux2

- = Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorial = Glossario = -

---

Deinterlacciamento video può essere una situazione difficile, soprattutto con l'intreccio numerose / deinterlacciamento filtri disponibili in Avidemux. Questa guida è qui per offrire un aiuto nel decidere quali filtri sono meglio per voi, e come si deve loro.

### Sommario

- 1 Panoramica
- 2 Operazioni preliminari
- 3 DVD Video
  - 3.1 Per framerate NTSC (30 fps)
  - 3.2 Per NTSC a PAL framerate: cambiare 30 fps a 24 fps
- 4 video registrato
  - 4.1 MythTV MPEG Non-interlacciato
- 5 Cfr. anche

## Panoramica

In poche parole, l'interlacciamento è una tecnica per memorizzare i fotogrammi del video a metà è divisa tra immagini separate. Il quadro è suddiviso in più righe e le righe dispari andare a un fotogramma, mentre le linee di andare anche l'altra cornice. Questo è chiamato interlacciamento video. Se non correttamente Deinterlace il tuo video, le immagini video può sembrare strano o nella peggiore delle ipotesi, essere rotto.

## Getting Started

### IMPORTANTE:

- Quando si deinterlacciamento video, i filtri di deinterlacciamento video dovrebbe sempre andare **in primo luogo**, anche prima del ritaglio filtri.
- **Disabilitare** completamente Post Processing di de-block orizzontale e verticale. Questi dovrebbero essere inutile per il tuo video, se altri filtri e le impostazioni video codec siano corrette.

## DVD Video

Per i video DVD, un buon fine generale Filtro di deinterlacciamento è Decomb Telecide. Sarà necessario selezionare un tipo di *strategia* dal menu a tendina. Selezionando 'Nessuna strategia' effettivamente ottenere rimuovere tutte le linee si intrecciano e il risultato in un quadro adeguato. Questo è raccomandato di default, ma a volte le altre opzioni sono adatti per l'uso. Vedere che l'articolo (o inferiore) per informazioni più dettagliate su questo filtro.

## Per framerate NTSC (30 fps)

Se il video di origine è 29.97 fps (o 30 fps), e si desidera mantenere tale framerate, quindi tutto quello che dovete fare è deinterlacciare il video con Decomb Telecide e nient'altro.

- Prima aggiungere il filtro **Decomb Telecide**.
- Per la *strategia* di configurazione del filtro di selezionare l'opzione *Nessuna strategia*. Questo sarà solo deinterlacciare il video e lasciare inalterato il framerate.
- **NON** devi usare Decomb Decimate perché non si sta cercando di ridurre il numero di fotogrammi del video.

## Per NTSC a PAL framerate: cambiare 30 fps a 24 fps

Se il video di origine è 29.97 fps (o 30 fps), e si desidera trasformare a 23.976 fps (o 24 fps), effettuare le seguenti operazioni:

- Prima aggiungere il filtro **Decomb Telecide**.
- Per la *strategia* di configurazione del filtro di selezionare l'opzione *3:2 Pulldown*. Questo deinterlacciare il video e preparare framerate a 24 fps correttamente, ma in realtà non farlo.
- È inoltre necessario utilizzare Decomb Decimate dopo la Telecide Decomb per ottenere il giusto numero di fotogrammi per 24 fps. Utilizzando le opzioni di default Decomb Decimate sono normalmente abbastanza buono. Ciò consentirà di ridurre il numero di fotogrammi del video a circa 24 fps.

## Video registrato

### MythTV MPEG Non-interlacciato

Deinterlacciamento di video internamente, nell'ambito MythTV non è sempre la più efficace, né la scelta di altissima qualità. A volte è meglio lasciare il video registrato in forma non interlacciato e poi deinterlacciare dentro di Avidemux utilizzando un filtro più specializzato e potente. Il Yadif filtro video con le impostazioni di default funziona molto bene nella maggior parte delle situazioni.

## Vedi anche

Filtri video Interlacciamento

---

- = Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario = -  
Estratto da "[http://www.avidemux.org/admWiki/index.php?title=Deinterlacing\\_Video](http://www.avidemux.org/admWiki/index.php?title=Deinterlacing_Video)"

---

- Ultima modifica 06:52, 11 gennaio 2008.
- Questa pagina è stata letta 30.430 volte.
- Tutela della privacy
- A proposito di Avidemux2
- Disclaimer

# DVD to AVI

## Da Avidemux2

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-

---

Questo tutorial spiega il processo di conversione di un file MPEG-1 o MPEG-2 DVD file in un file AVI contenente video MPEG-4 ASP (questo è spesso **erroneamente** chiamato "DivX" o "Xvid" - vedi il comune miti articolo per una spiegazione di le parole DivX e Xvid e la differenza tra software e del formato).

Ci sono diversi programmi disponibili on-line che permettono di convertire il DVD MPEG-4 AVI, utilizzando strumenti come il popolare MEncoder. La differenza tra Avidemux e questi altri progetti è che Avidemux permette di modificare il file prima della codifica, e di fare un controllo visivo di quello che stai facendo.

## Sommario

- 1 Capire i tipi di file MPEG
- 2 Ottenere un file VOB utilizzabile con MPlayer
- Loading 3 e indicizzare i tuoi file MPEG
- 4 Editing
  - 4,1 versus NTSC FILM
  - 4,2 Cropping
  - 4,3 Ridimensionamento
  - 4,4 filtri video Maggiori informazioni e il taglio
- 5 video Configurazione
  - 5,1 Scelta di un encoder
  - 5,2 Configurazione l'encoder
  - 5,3 dimensioni del video Calcolo
- 6 Configurazione audio
  - 6,1 Selezione traccia audio
  - 6,2 Controllo sincronismo A / V
  - 6,3 Scelta codificatore audio
  - 6,4 Audio filtri
- 7 Salvataggio

## Comprendere i tipi di file MPEG

- m1v significa "MPEG-1 Video", cioè un file che contiene solo un flusso cosiddetto video elementari, senza audio.
- m2v significa "MPEG-2 Video".
- mpg è un flusso di programma che contiene il video multiplexati e flussi audio.
- vob è un flusso DVD sistema che contiene le informazioni video, audio e supplementari, è un anche un flusso di programma.
- VDR è un flusso di trasporto che contiene video e audio (s). Si è sostenuto, ma senza correzione di sincronizzazione.

## Ottenere un file VOB utilizzabile con MPlayer

```
mplayer dvd://1 -dumpstream -dumpfile rippeddvd.vob
```



Questo creerà un file nella directory di lavoro chiamato rippeddvd.vob. Questo è un file Avidemux VOB in formato MPEG compatibili con i vari flussi audio incluso nel DVD.

## Caricamento e indicizzazione dei tuoi file MPEG

Caricare questo file in Avidemux rippeddvd.vob facendo clic "Apri" icona della cartella nella barra degli strumenti, oppure andando File-> Apri.

Ora verrà presentato con una finestra di dialogo con un elenco a discesa di scelte flusso audio. Qui è dove scegliere quale traccia audio del DVD che si desidera utilizzare per il video. In generale l'impostazione predefinita, o il primo della lista è la scelta migliore. Fare clic su "OK" per avviare l'indicizzazione del MPEG. Questa operazione potrebbe richiedere un certo numero di minuti a seconda della velocità della vostra macchina.

**Nota:** Avidemux non legge i flussi MPEG. E 'stato progettato per leggere un indice stream MPEG. Un indice stream MPEG è un file di testo contenente una descrizione del MPEG e la posizione dei fotogrammi in tutto il flusso. Questo file permette di Avidemux casuale cercare e soggiorno accurati. Detto altrimenti, senza l'indice, Avidemux non in grado di gestire i file MPEG.

## Editing

### NTSC versus FILM

Alcuni DVD sono codificati in 23,976 aka FILM (maggior parte dei film in realtà). Alcuni altri sono codificati in 29,96 (NTSC), sapone per esempio. Nel primo caso, il lettore DVD non un'operazione di convertire al volo in formato NTSC (telecine). Così l'intestazione MPEG dice sempre 29,96 in quanto sarà sempre il formato finale.

Avidemux utilizza mpeg2dec per decodificare i flussi MPEG (con una piccola patch). mpeg2dec non fa il telecine su Movie Film (e che è meglio così).

Ciò significa che Avidemux non può dire la differenza tra cinema e NTSC. Quindi, se il MPEG guarda progressivo (non interlacciato) e desincronizzazione appare evidente (e sempre peggio), l'uso Video-> Frame Rate e impostarlo a 23.976.

Per PAL MPEG, non c'è nessun problema, è sempre 25 fps.

Se l'audio è presente, Avidemux cercherà di stimarne se il video è 23,976 confrontando la durata audio e video.

### Ritaglio

Ritaglio rimuove i bordi neri lungo la parte superiore e inferiore del video in formato DVD widescreen. Consente maggiore quantità di dati da utilizzare per la codifica dell'informazione immagine reale. Senza tagliare i bordi taglienti sarebbe ridurre drasticamente la qualità delle immagini in queste zone, come il codec MPEG basato non gestiscono bene. Accertarsi sempre di rimuovere completamente tutti i confini, anche se sono solo a metà nero o impuro.

1. Per ritagliare il video, è necessario utilizzare i filtri video. Prima di selezionare i filtri tuttavia, utilizzare la barra di scorrimento in basso per selezionare un punto al centro del film. La ragione di questo, è che la funzione di coltura auto regolerà il taglio in base al frame corrente.
2. Ora premere il pulsante Video filtri a comparsa la lista dei video dei filtri.
3. Aggiungi Trasformazione -> Ritaglia.
4. Ora, fare clic sul AutoCrop pulsante "". Noterete le aree nere apparirà in verde per visualizzare le aree in cui il video verrà tagliata.
5. Se ti piace il modo in cui sembra fare clic su "OK" e quindi chiudere la lista dei filtri video.

### Ridimensionamento

Probabilmente si desidera ridimensionare il video a qualcosa di più piccolo. A risoluzione inferiore significa bit più alto rapporto segnale / rumore di pixel, che possono migliorare la qualità a bitrate più bassi, che sono tipici per MPEG-4.

Inoltre, i file AVI non contengono le informazioni il rapporto di aspetto (ma può essere memorizzata all'interno del bitstream MPEG-4), quindi dovrete ridimensionare il video dopo il ritaglio di ottenere l'aspect ratio corretto. Il rapporto di aspetto è la forma di un pixel. Su un PC su cui è in gran parte quadrato, però, su un DVD potrebbe essere 4:3 o 16:9.

Quindi, riportare l'elenco del filtro video.

1. Selezionare il rapporto di aspetto di ingresso (16:9 è il più comune per i DVD) e il rapporto di aspetto di destinazione (1:1 per i file AVI).
2. Quindi selezionare il metodo di ridimensionamento - bilineare non è tagliente come bicubico o lanczos, ma è generalmente più comprimibile. Quindi dipende dal bitrate (metodi più nitida per i bitrate più alti), di compressione video di origine e il vostro gusto (alcune persone preferiscono nitida video, anche se questo significa artefatti di compressione più come la chiamata).
3. Si desidera controllare il ciclo di 16 fino ad essere sicuro la larghezza e l'altezza finale sono multipli di 16.
4. Quindi spostare il cursore fino a raggiungere la larghezza desiderata.

## Più filtri video e taglio

A seconda della fonte, è possibile aggiungere più filtri (sottotitoli, Denoiser, deinterlacer, IVTC ecc.)

A questo punto, il video è pronto per qualsiasi modifica / taglio si potrebbe desiderare di farlo.

## Configurazione video

### La scelta di un encoder

Ora scegliere il codec video dal menu a tendina. MPEG-4 encoder due supportato da Avidemux sono Xvid e ffmpeg (libavcodec) MPEG-4. Entrambi sono molto buone e più o meno analoga, quindi sta a voi quale preferite. Useremo Xvid in questo tutorial, come è attualmente supportata meglio in Avidemux ed ha pertanto raccomandato.

### Configurazione del codificatore

Fare clic sul pulsante Configura e ti verrà presentata la finestra *Xvid opzioni di codifica*. E 'altamente raccomandato l'uso di encoding a 2 passi in modo da poter selezionare le dimensioni del file finale. Non preoccuparti per le dimensioni finali del file, e impostare le opzioni come si vorrebbe avere e fare clic su OK per salvare le impostazioni.

### Calcolo dimensioni del video

Ora usare la calcolatrice, eseguire Format che è AVI, selezionare il proprio mezzo (cioè la dimensione del file finale), e fare clic su Applica. Questa consiste nel riempire la dimensione del file nella finestra di *opzioni di codifica Xvid*. Come pure i risultati calcolati verrà visualizzato nella finestra. Puoi chiudere questa finestra ora.

## Configurazione audio

### Selezione traccia audio

Il primo passo è quello di scegliere quale traccia audio da codificare. Di default la prima traccia audio è selezionata, che su un DVD 'dovrebbe' essere la lingua principale del DVD (cioè inglese per la Regione 1). Per scegliere un'altra traccia audio principale vai a Audio -> Main Track e utilizzare la casella a discesa per selezionare la traccia audio interno si desidera, o se si desidera si può scegliere una fonte audio esterna. Se si desidera una seconda traccia audio da usare audio codificato -> seconda pista.

## Controllo A / V sync

A questo punto si desidera controllare la sincronizzazione audio. Attivare la modalità di anteprima in uscita filtrato e assicurarsi che l'audio è sincronizzato, se non lo è, utilizzare la funzione di spostamento per correggere questo.

## Scelta codificatore audio

Successiva che si deve decidere quale codec usare. MP3 tramite LAME è il codec più supportato. Tenere la traccia AC3 di default è anche una buona opzione per il suono 5.1. Ricompressione da una compressione lossy già non potrà mai suonare come una buona, i risparmi dimensione del file non sarà troppo grande utilizzando un altro codec, e vi farà risparmiare tempo per non comprimere l'audio.

## Filtri audio

Infine si deve decidere se è necessario o si desidera alcun filtro.

## Risparmio

Ora selezionate File -> Save -> Save Video e aspettare. Avidemux farà passo 1 poi passare a passare 2 (compreso l'audio). Il risultato sarà un file AVI con video MPEG-4 ASP e all'interno audio MP3.

---

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-  
Estratto da "[http://www.avidemux.org/admWiki/index.php?title=DVD\\_to\\_AVI](http://www.avidemux.org/admWiki/index.php?title=DVD_to_AVI)"

---

- Ultima modifica 07:51, 12 novembre 2006.
- Questa pagina è stata letta 106.858 volte.
- Tutela della privacy
- A proposito di Avidemux2
- Disclaimer

# Editing MPEG cattura (DVB o IVTV)

## Da Avidemux2

= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorial = Glossario =

---

Questa pagina cerca di dare suggerimenti su come modificare i file MPEG catturati. Questi MPEGs catturati sono generalmente da DVB S / T (in formato MPEG TS) o da schede IVTV base o di qualsiasi altra carta con hardware di codifica MPEG-2 (in formato MPEG PS).

## Il problema

Questi coglie spesso contengono errori di trasmissione, che finiscono come fotogrammi mancanti o rotti. Un giocatore (MPlayer, xine, VLC) verrà costantemente i flussi di risincronizzazione utilizzando le informazioni di temporizzazione incorporato nel flusso. Avidemux non lo faranno.

A parte lo spostamento costante, che è facilmente recuperabile con il filtro TimeShift, che si tradurrà in un problema crescente sincronizzazione durante la codifica o la transcodifica. Anche il risparmio senza re-encoding saranno async.

Registrazioni MythTV sono un esempio lampante di questo problema. Se il processo di seguito non è seguita l'audio sarà compensato da circa -330 ms con l'inizio della registrazione e la deriva per tutta la durata della registrazione. Si prega di notare che non tutte le registrazioni MythTV hanno questo problema, solo alcuni a seconda della configurazione software e hardware.

## La soluzione

L'unico modo affidabile al 100% per farlo è usare ProjectX ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://project-x.sourceforge.net/&prev=\\_t&rurl=translate.google.it&usg=ALkJrhhXSkPnWvFwFMycqk4sxG41Y\\_Fxg](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://project-x.sourceforge.net/&prev=_t&rurl=translate.google.it&usg=ALkJrhhXSkPnWvFwFMycqk4sxG41Y_Fxg)) (vedi tutorial ProjectX). Ci vuole un po 'di tempo, ma è facilmente script, uno script di esempio è disponibile qui. ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.wslove.org/remux2&prev=\\_t&rurl=translate.google.it&usg=ALkJrhhB37rW96yBN8pZOVekxLzyGdrqdQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.wslove.org/remux2&prev=_t&rurl=translate.google.it&usg=ALkJrhhB37rW96yBN8pZOVekxLzyGdrqdQ)) Diciamo che hai un file chiamato 2537\_20060819203500.mpg DVB in formato MPEG TS (DVB-T) di cattura.

Prima si demultiplex il file in flussi elementari, ma *sincronizzati* flussi elementari

```
ProjectX 2537_20060819203500.mpg
```

Che genererà un file per stream elementari con le seguenti estensioni:

- m2v: per streaming video
- mp2: per MPEG audio stream
- ac3: per Dolby Digital aka stream AC3

Ora dobbiamo ricombinare questi flussi, in formato PS, per esempio.

```
mplex-f 8-o output_file.mpg 2537_20060819203500.ac3 2537_20060819203500.m2v 2537_20060819203500.mp2
```

Il file risultante (output\_file.mpg) possono essere modificati in Avidemux (rimuovere le pubblicità, per esempio) o di

transcodifica di formati senza alcun problema di sincronizzazione.

ProjectX è molto affidabile per risincronizzare i flussi. Per informazioni, si dupe missing frames audio o video creare cornici vuote quando sono stati eliminati.

Fino a versioni aggiornate di mplex per i sistemi Win32 sono difficili da trovare, tuttavia ImagoMPEG-Muxer ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.videohelp.com/tools%3Ftool%3DImagoMPEG-Muxer&prev=\\_t&rurl=translate.google.it&usg=ALkJrhjKKyD4fyI7qOowRST6ko3oA0rnDQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.videohelp.com/tools%3Ftool%3DImagoMPEG-Muxer&prev=_t&rurl=translate.google.it&usg=ALkJrhjKKyD4fyI7qOowRST6ko3oA0rnDQ)) è una sostituzione perfetto.

## Tocco finale

Se si codifica di altro formato (MPEG-4 ,...), probabilmente si vuole andare a Preferenze e selezionare Input / Usa libavcodec decodificatore MPEG. libavcodec contiene il codice per tentare di nascondere gli errori di decodifica. Così invece di avere un blocco verde, si avrà una corretta, ma la maggior parte offuscata blocco, che è molto più piacevole per gli occhi.

---

- = Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario = -  
Estratto da "[http://www.avidemux.org/admWiki/index.php?title=Editing\\_MPEG\\_capture\\_%%28DVB\\_or\\_IVTV\\_29](http://www.avidemux.org/admWiki/index.php?title=Editing_MPEG_capture_%%28DVB_or_IVTV_29)"

---

- Ultima modifica 09:57, 2 settembre 2007.
- Questa pagina è stata letta 43.068 volte.
- Tutela della privacy
- A proposito di Avidemux2
- Disclaimer

# Guida di tipi di codifica

## Da Avidemux2

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorial = Glossario =-

---

Questa è una piccola guida per capire i termini usati per i tipi di codifica video.

### Sommario

- 1 a due passi
- 2 Stessa Qz come input
- 3 Single pass - bitrate
- 4 pass unico - quantizzazione

## A due passi

Questo metodo ha due passa sopra l'intero video. Il passo 1 analizza l'intero video. Fa una complessa analisi di ogni fotogramma e ogni scena, e decide quali parti del video che hanno bisogno di più bitrate e quelli che richiedono meno. Durante il secondo passaggio il video è in realtà codificato utilizzando le informazioni che si trovano nel passo precedente.

Perché sono due passaggi necessari? Perché il codec non può vedere in futuro. Così non si può sapere quanti bit molto da dare alle immagini attuali.

Supponiamo che la quantizzazione \* dimensioni di codifica è più o meno costante. Se fai il 1 ° passaggio di codifica con quantizzazione = 2, si otterrà una dimensione di S. Ora, si voleva conoscere la dimensione finale (F), che dovrebbe essere inferiore a S, altrimenti hai appena sprecato il vostro tempo:.)

La quantizzazione media è quindi  $Q = (2 * S) / F$ . E 'quindi sintonizzati tutti insieme per dare più bit a scene in cui si farà una grande differenza, e meno per le scene in cui dare bit non aumenterà la qualità Quanta.

Così, con due passaggi in modalità vi darà:

1. Un dato filesize
  2. Una buona distribuzione di bit
- **Qualità:** qualità molto alta
  - **Time:** richiede la maggior parte del tempo (ovviamente perché fa due passaggi). Il primo passo può essere più veloce del secondo perché è in qualche modo più leggero. Il calcolo non ha bisogno di numeri precisi al lavoro.
  - **La dimensione del video:** Se si utilizza l'opzione video dimensioni di destinazione, di solito accurati entro il 2% delle dimensioni di destinazione.
  - **Raccomandato:** Sì. Vale la pena il tempo supplementare.

## Stesso Qz come input

Un metodo rapido per la ri-codifica un file che si codifica ogni frame con la quantizzazione come la quantizzazione del fotogramma di ingresso è stato codificato con. L'output potrebbe essere un file con bitrate / dimensione del file simile a quello originale. Questo esempio può essere utile per ri-codifica MPEG-4 video codec senza opzioni, come QPel, che

alcuni lettori non supportano l'hardware (si noti che da ri-codifica, si perde sempre di qualità, a meno che il formato di destinazione è lossless). Questo metodo consente di estrarre gli 1-passare le informazioni di analisi già memorizzate nel frame video e riutilizzare, in modo efficace per evitare un passaggio complesso e risparmio di tempo.

- **Nota:** Questo metodo non può garantire che la dimensione del file finale / bitrate sarà lo stesso come l'originale, ma può arrivare molto vicino, ed è più veloce di *due-pass*.
- **Qualità:** di solito piuttosto vicino al file originale.
- **Tempo:** il tempo lineare in funzione della lunghezza del file. Richiede circa la metà del tempo di *due-pass* encoding, di solito abbastanza veloce.
- **La dimensione del video:** in genere lo stesso del file di input originale, ma non sempre.
- **Raccomandata:** solo utile per modificare un file esistente codificato.

## Single pass - bitrate

Questa codifica il video una sola volta, utilizzando una gamma limitata bitrate video su tutte le scene, indipendentemente dal fatto che o non hanno bisogno di meno, uguale o superiore a bitrate. Si noti che questo non è costante encoding bitrate.

- **Qualità:** I video di qualità più bassa.
- **Time:** Richiede il minor tempo quantità di codificare.
- **La dimensione del video:** Dimensione di video varia e non può essere previsto con precisione.
- **Raccomandato:** questo metodo non è raccomandato perché produce i video di qualità più bassa. *Two-pass* è raccomandato.

## Single pass - quantizzazione

Conosciuto anche come quantizzatore costante o modalità a qualità costante. Selezionando un numero, ogni fotogramma otterrà lo stesso livello di compressione in base al livello scelto. Buona per la compressione semplice e veloce, a scapito magari di non raggiungere le dimensioni grandi o piccole di video desiderato. I bassi livelli di parità di compressione inferiore, e quindi tradursi in una qualità video superiore. Alti livelli di compressione più uguali, e quindi tradursi in una qualità video inferiore. Questo è molto utile per qualcosa velocemente la compressione basata su un sistema semplice valore.

- **Qualità:** variabile, a seconda della quantizzazione.
- **Tempo:** il tempo lineare in funzione della lunghezza del file. Di solito abbastanza veloce.
- **La dimensione del video:** variabile a seconda della quantizzazione, non può essere previsto con facilità.
- **Raccomandato:** consigliata quando la compressione semplice e veloce è desiderato o quando più alta è la qualità di codifica desiderata - che è, che codifica per il video completo con quantizer = 2.

---

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorial = Glossario =-  
 Estratto da "[http://www.avidemux.org/admWiki/index.php?title=Guide\\_to\\_encoding\\_types](http://www.avidemux.org/admWiki/index.php?title=Guide_to_encoding_types)"

---

- Ultima modifica 13:29, 25 luglio 2006.
- Questa pagina è stata letta 49.290 volte.
- Tutela della privacy
- A proposito di Avidemux2
- Disclaimer

# H264

## Da Avidemux2

- = Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario = -

---

Questo articolo descrive brevemente cosa H.264 è e come ottenere il supporto di codifica H.264 per Avidemux. Esso riassume e spiega anche l'x264 opzioni disponibili in Avidemux. Questo può essere considerato un (semplice) guida per il codec.

## Sommario

- 1 Panoramica
  - 1,1 x264 Introduzione
  - 1,2 Get x264 per Avidemux
- 2 opzioni H.264/AVC spiegato
  - 2.1 disponibile x264 opzioni in Avidemux
  - 2.2 Generale
    - 2.2.1 Rate Control
    - 2.2.2 Pixel Aspect Ratio
    - 2.2.3 Multi-Threading
  - 2,3 Motion
    - 2.3.1 Motion Estimation
    - 2.3.2 Motion Vector
    - 2.3.3 Prediction
  - 2,4 Partition
    - 2.4.1 Partition Search
  - 2,5 Frame
    - 2.5.1 Frame Encoding
    - 2.5.2 B-Frames
    - 2.5.3 I-Frame
  - 2,6 Analysis
    - 2.6.1 Analisi di configurazione
    - 2.6.2 Luma Quantization Deadzones
    - 2.6.3 Quantization Matrix
    - 2.6.4 Psycho-visivo ottimizzato RDO & Trellis
  - 2,7 quantizzazione
    - 2.7.1 quantizzazione di controllo
    - 2.7.2 Curve di quantizzazione di compressione
    - 2.7.3 La quantizzazione adattiva
  - 2,8 Advanced
    - 2.8.1 Video Buffer Verifier
    - 2.8.2 Zone
  - 2,9 Output
    - 2.9.1 IMPOST. USCITA
    - 2.9.2 Video Usability Informazioni
- 3 disponibile x264 opzioni in Avidemux
  - 3,1 obsoleti Opzioni
- 4 Profili H.264/AVC e Livelli
  - 4.1 Elenco di tutti i profili H.264/AVC



- 4.2 Elenco di tutti i livelli H.264/AVC
- 5 GPU di sostegno
- 6 fotogrammi IDR
- 7 Elenco dei riferimenti
- 8 Cfr. anche

## Panoramica

**H.264**, che è anche conosciuto come "MPEG-4 Part-10" o "MPEG-4 Advanced Video Coding (AVC)", è uno standard di compressione video digitale, che si distingue per il conseguimento molto elevata compressione dei dati. Mentre H.264 richiede in genere di alimentazione della CPU più per la riproduzione di video codificati con le grandi MPEG-4 Part-2 standard (usati per XviD e DivX), l'efficienza di compressione è molto meglio! Ciò significa: Con H.264/AVC è possibile ottenere una migliore qualità notevole con le stesse dimensioni di file-o-si può ottenere la stessa qualità in un file di piccole dimensioni significative (rispetto al formato MPEG-4 ASP). Mentre H.264 comprime molto più efficiente di MPEG-4 Part-2, il vantaggio su MPEG-2 è ancora maggiore.

Informazioni più dettagliate su H.264 possono essere trovate nella Wikipedia ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://de.wikipedia.org/wiki/H.264&prev=\\_t&rurl=translate.google.it&usg=ALkJrhiDFyoyqk3yq1jhAv0YdW1EcwNv\\_A](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://de.wikipedia.org/wiki/H.264&prev=_t&rurl=translate.google.it&usg=ALkJrhiDFyoyqk3yq1jhAv0YdW1EcwNv_A)) corrispondente. Un confronto tra i codificatori H.264 vari contro MPEG-4 Part-2, MPEG-2 e altri formati video possono essere trovate qui. ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://mirror05.x264.nl/Dark/website/compare.html&prev=\\_t&rurl=translate.google.it&usg=ALkJrhg6xtF8iBOK5XA12wKu7D2Z\\_pTTSg](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://mirror05.x264.nl/Dark/website/compare.html&prev=_t&rurl=translate.google.it&usg=ALkJrhg6xtF8iBOK5XA12wKu7D2Z_pTTSg))

## x264 Introduzione

Mentre utilizza Avidemux "built-in" libavcodec da FFmpeg per la *decodifica* H.264, ha bisogno di un supplementare (esterno) libreria per la codifica H.264. Quindi Avidemux utilizza **x264**. ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.videolan.org/developers/x264.html&prev=\\_t&rurl=translate.google.it&usg=ALkJrhiuWJOUNmqEI49UMiU0W0MvgoJ1TQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.videolan.org/developers/x264.html&prev=_t&rurl=translate.google.it&usg=ALkJrhiuWJOUNmqEI49UMiU0W0MvgoJ1TQ)) x264 è una libreria per la codifica di flussi video H.264/AVC. Il codice è stato scritto da zero da Laurent Aimar, Loren Merritt, Eric Petit (OS X), Min Chen (VfW / ASM), Justin Clay (VfW), Måns Rullgård, Radek Czyz, Christian Heine (ASM), Alex Izvorski (asm), e Alex Wright. È rilasciato sotto i termini della licenza GPL. ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/GNU\\_General\\_Public\\_License&prev=\\_t&rurl=translate.google.it&usg=ALkJrhhKraM2tYuP2wUxQTNiELQoqkBRA](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/GNU_General_Public_License&prev=_t&rurl=translate.google.it&usg=ALkJrhhKraM2tYuP2wUxQTNiELQoqkBRA)) Quindi, per chiarire, la libreria si chiama encoder *x264*, mentre lo standard di compressione che usa è chiamato *H.264* (o *MPEG-4 AVC*). In altre parole: Il software crea x264 encoder video H.264/AVC. Va notato che x264 pur essendo "libero" il software in grado di competere con encoder H.264 commerciali in termini di qualità e velocità. Le aziende più importanti nel settore video, come YouTube e Facebook, sono noti per utilizzare l'encoder x264.

## Get x264 per Avidemux

Se x264 non è disponibile nella versione di Avidemux, vi è una guida su come scaricare e compilare x264 da soli. E 'nella sezione H264 Compile.

Dopo aver compilato x264, dovrete ricompilare Avidemux costruire in funzione di x264. Vi è anche una guida su come effettuare questa operazione nel Installa -> Compile Avidemux da SVN (Subversion) sezione.

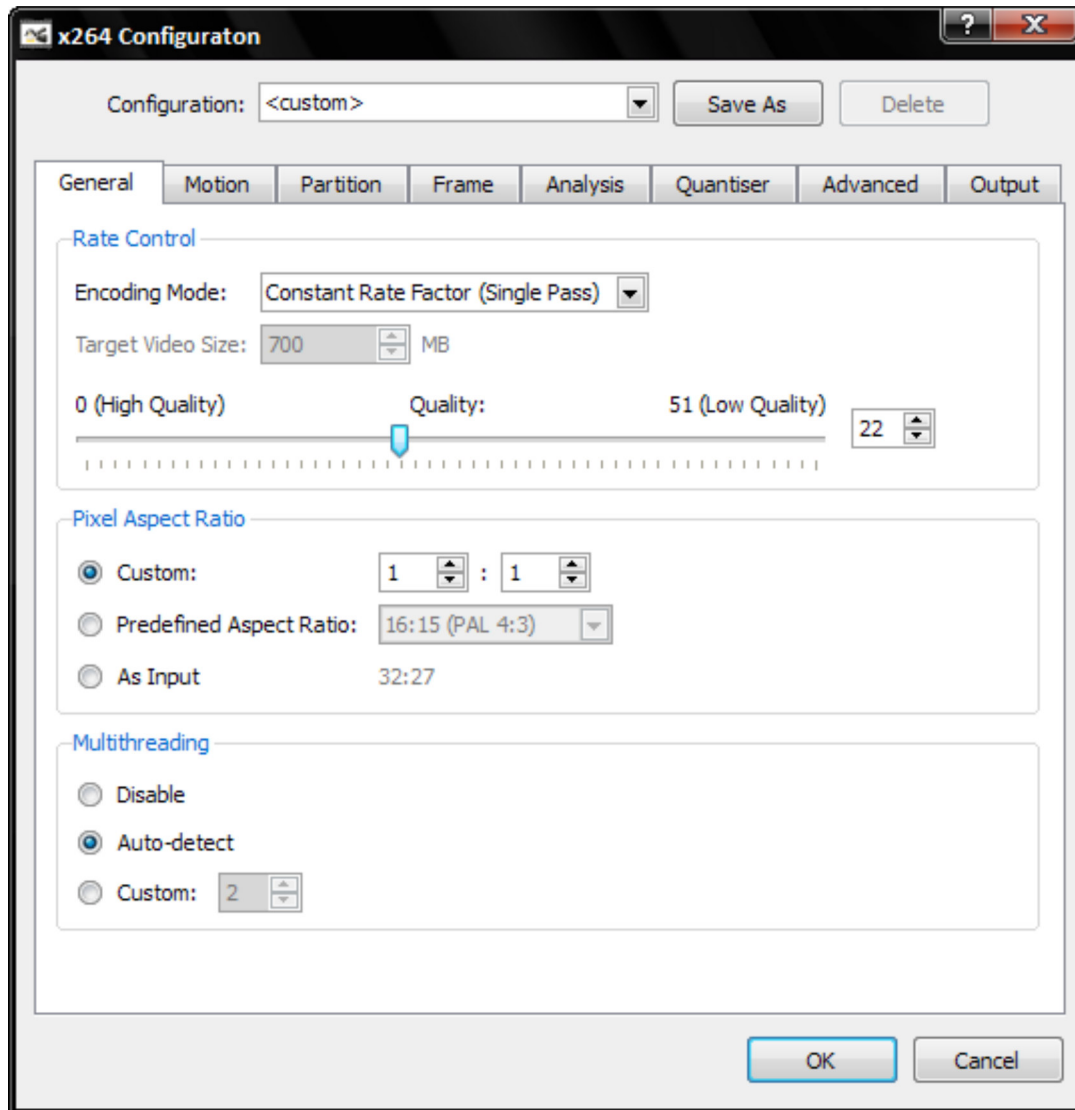
Si noti che se si utilizza il pre-compilati Avidemux costruisce per *Microsoft Windows*, la richiesta x264 navi biblioteca con il programma di installazione. Quindi **nessun** software aggiuntivo è richiesto! Roba come "Codec Pack", "VFW Codec" o "DirectShow Filters" **non funziona** con Avidemux! In ogni caso, l'ultima build della libreria x264 per Avidemux può essere trovato in questo ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://avidemux.org/admForum/viewtopic.php%3Fid%3D5615&prev=\\_t&rurl=translate.google.it&](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://avidemux.org/admForum/viewtopic.php%3Fid%3D5615&prev=_t&rurl=translate.google.it&))

usg=ALkJrhfhfXAW9ozSYaXNA3sfdHjkXC1akvw) thread (assicuratevi di navigare al posto *all'ultimo!*). Queste build di solito sono più recenti - e meno provato - di quelli che viene fornito con Avidemux.

## Opzioni H.264/AVC spiegato

### X264 opzioni disponibili in Avidemux

Avidemux contiene la maggior parte delle opzioni disponibili per il codec H264 tramite la libreria **x264**. Per le opzioni *non sono* ancora disponibili, vedere la "non disponibile" in questo articolo.



## Generale

### Rate Control

- **Modalità di codifica:**
  - **Single Pass - costante di quantizzazione:** questa modalità è anche conosciuta come la "QP Mode". Sarà

codificare il video ad un *quantizzatore costante*, quindi si sceglie la *quantizzazione* di destinazione **non**, il *bitrate* target. La quantizzazione è una misura per la quantità di perdita di dati: un quantizzatore *più alto* significa che i dati saranno più persi, il che si traduce in una compressione migliore (file più piccoli), ma offre anche peggiorare la qualità visiva. In contrasto con un quantizzatore *più bassa* significa che i dati meno andranno persi, che si traduce in una migliore qualità visiva, ma anche comprime peggio (file di dimensioni maggiori). H.264 utilizza una scala di quantizzazione tra 0 e 52. Il valore di quantizzazione di default è **26**. Se ci si rivolge per un certo *livello di qualità* e non si preoccupano molto della dimensione finale del file, allora si potrebbe prendere in considerazione utilizzando la modalità *QP*. Ma se ci si rivolge per una certa dimensione di file (o un certo *bitrate* medio), quindi tenere lontano da *QP Mode*! Questo perché la dimensione finale (il *bitrate* medio) è del tutto *imprevedibile*, in questa modalità.

- **Si prega di notare:** La modalità *CRF* dovrebbe essere preferito il modo *QP*! Anche la *quantizzazione adattiva* (AQ) verrà disattivata nella modalità di *QP*, mentre è abilitato (di default) in modalità di *CRF*.
- **Single Pass - Fattore di velocità costante:** questa modalità è anche conosciuta come la "CRF Mode" o "Constant Quality" mode. Funziona sostanzialmente analogo al modo *QP* (vedi sopra), ma sarà codificare con una quantizzazione *media*, invece di una costante. Per essere più precisi, questa modalità di codifica in un costante fattore "velocità", che deriva dalla quantizzazione specificato. Internamente modalità *CRF* utilizza lo stesso algoritmo di controllo della frequenza, come x264 modalità *ABR*, solo *senza* un *bitrate* target. Il vantaggio della modalità di *CRF* è che si adatta alla percezione umana molto meglio di quanto la modalità di *QP*. Per esempio, aumenterà il quantizzatore in "veloce" scene in cui la perdita non sarà visibile e comunque inferiore al quantizzatore in "slow" scene. Quindi il modo *CRF* dovrebbe dare la stessa qualità *soggettiva*, come modalità di *QP*, ma si ottiene di solito una compressione nettamente superiore. Si consiglia di preferire la modalità *CRF* su modalità di *QP*, *CRF*, anche se è un po' più lento. Quando si passa da *QP* alla modalità *CRF*, si consiglia di ridurre leggermente la quantizzazione. Questo dovrebbe dare circa la stessa dimensione del file come prima, ma migliore qualità visiva! Un altro importante vantaggio della modalità di *CRF* è che beneficerà di *quantizzazione adattiva*, qualcosa che la modalità *QP* non può fare.
  - **Si prega di notare:** anche la modalità di *CRF*, non garantisce un "perfetto" qualità costante! Un valore specifico *CRF* solo offrire una qualità *simile* per varie fonti, fino a quando non modificare le altre impostazioni. Uso di "lento" impostazioni con lo stesso valore *CRF* sarà né produrre un file di dimensioni inferiori a stessa qualità o di un produrre un file delle stesse dimensioni di una migliore qualità. È anche possibile che sia, la dimensione e la qualità, sarà aumentato. La qualità "per misura" rapporto sarà migliorato comunque.
- **Commento:** La scelta del *quantizzatore* corretta impostazione per un *CRF* (o *PQ*) codifica *non* è banale! Questo perché la qualità visiva è molto *soggettiva*: ciò che alcuni considerano "buona qualità" altre persone prenderà in considerazione "orribile di qualità" - e viceversa. Inoltre, la quantizzazione impostazione dipende fortemente sui contenuti *del* video. Tuttavia, un quantizzatore impostazione nella fascia tra **16** e **32** dovrebbe dare risultati soddisfacenti nella maggior parte dei casi. Utilizzando un quantizzatore *inferiore* a **16** di solito è eccessivo, tranne che per fini di mastering. Utilizzando un quantizzatore *più alto* di **32** si tradurrà in video quasi *inguardabile*. La quantizzazione del **22** sembra essere ragionevole per la maggior parte. Tuttavia il materiale con texture pochi, come Anime e cartoni animati, in grado di affrontare con quantizzatori molto più *elevato*. Allo stesso tempo "vita reale", riprese con un sacco di texture quantizzatori potrebbe richiedere molto più *basso*, soprattutto nelle scene buie. Vi è anche una regola empirica: abbassando il valore *CRF* da 6 a raddoppiare la dimensione del file, abbassando *CRF* entro il 1° alzerà dalla dimensione del file ~ 12,5% (molto approssimativamente). Inoltre, la pratica comune è la seguente: Inizia con un valore basso *CRF*, come ad esempio **16**. Quindi aumentare il valore *CRF* a passi di *uno*, fino a quando la qualità diventa intollerabile. In questo modo si trova il valore più alto *CRF* possibile che conferisce una qualità ancora accettato per i tuoi occhi. Una volta che l'hai trovato, è possibile utilizzare tale valore per *tutti i* tuoi codifica per il futuro.
- **Single Pass - Bitrate:** Questa modalità codificare il video ad un *bitrate medio* con un solo singolo passaggio. Quindi questa modalità richiede solo la metà del tempo di un "Two Pass" codificare. In contrasto con modalità *CRF* (e modalità *QP*), il *bitrate* medio risultante è conosciuto in anticipo. Quindi è facile prevedere le dimensioni finali del file. Un *bitrate più alto* si tradurrà in una migliore qualità visiva, ma naturalmente sarà anche risultato in un file di maggiori dimensioni. Un *bitrate più bassi* saranno i risultati in un piccolo file, ma sarà anche tradursi in una qualità visiva peggiore. Purtroppo il codificatore non "conoscere" il contenuto del video in anticipo quando si codifica con un solo passaggio. Quindi la capacità del codificatore di regolare il *bitrate* per quanto riguarda il contenuto del video è *estremamente limitato* in

questo modo! Solo "locale" ottimizzazioni sono possibili. Questo si traduce in una *qualità video* piuttosto *male* (rispetto a un "Two Pass" encode), soprattutto a medio e bitrate più bassi! Quindi è *vivamente* consigliato di **non** utilizzare questa modalità, a meno che non si ha realmente bisogno di farlo in un solo passaggio.

- **A due passi - Bitrate medio:** Questa modalità codificare il video ad un bitrate *medio* e utilizzerà *due* passaggi di codifica. Di conseguenza, questa modalità richiede il doppio del tempo di un "Single Pass" encode (circa). In contrasto con modalità *CRF* (e modalità *QP*), il bitrate medio risultante è conosciuto in anticipo. Quindi è facile prevedere le dimensioni finali del file. Un bitrate *più alto* si tradurrà in una migliore qualità visiva, ma naturalmente sarà anche risultato in un file di maggiori dimensioni. Un bitrate *più bassi* saranno i risultati in un piccolo file, ma sarà anche tradursi in una qualità visiva peggiore. Durante il *primo* passaggio il codificatore esegue una analisi dettagliata del video e creare una cosiddetta "stats" file. Poi, durante il *secondo* passaggio della codifica reale ha luogo e il file finale è creato. Il vantaggio di utilizzare *due* passi, è che durante il *secondo* passaggio il codificatore può fare affidamento sui dati raccolti durante il *primo* passaggio. Questo permette al codificatore di distribuire i bit disponibili, tra *l'intero* video. Ad esempio "ad alto movimento" scene otterrà un significativo bitrate superiore a quello "statico" scene. Questo viene fatto per mantenere la qualità visiva costante durante tutto il film. Ugly "blocco" sul fast / movimento spontaneo (come si vede in "Single Pass" codifica) è evitata. Quindi un "Two Pass" codificare fornisce il *massimo* della qualità visiva per il bitrate dato obiettivo (dimensione del file). Si raccomanda di usare sempre *questo* modo, se ci si rivolge per un certo bitrate medio!
- **A due passi - File Size:** Questa modalità sarà effettivamente utilizzare il "Two Pass - Average Bitrate" mode. L'unica differenza è che Avidemux calcolerà *automaticamente* il bitrate necessario per voi. In questo modo una dimensione specifica del file di destinazione può essere facilmente colpito. Basta inserire le dimensioni del file desiderato (ad esempio "700 MB" per un supporto CD-R o "4700 MB" per i supporti DVD-R) e il gioco è fatto! Tutto il resto funziona esattamente come descritto per il "Two Pass - Average Bitrate" mode.
  - **Si prega di notare:** x264, **non** prenderà il bitrate *audio* e il carico di container in considerazione. Quindi le dimensioni di destinazione specificate nella finestra di dialogo x264 effetti solo la parte *video* del file. Se il file contiene almeno una traccia audio, quindi il file effettivo uscirà *più grande* della dimensione specificata. Anche il *contenitore* aggiunge alcuni overhead aggiuntivo al file. Quindi, per favore usare Avidemux ' "Calcolatrice" strumento per impostare le dimensioni del file di destinazione correttamente!
- **Commento:** La scelta del bitrate buon bersaglio per un *bitrate a base di* codifica ( "Two Pass" o "Single Pass" mode) **non** è banale a tutti! Questo perché il bitrate necessario per ottenere risultati *soddisfacenti* dipende molto la comprimibilità "del tuo video e anche alle preferenze personali. Ad esempio "pulite" le fonti di farla franca con un bitrate significativa minore rumorosi / fonti granulosa. Anche filmati animati in genere riescono a farla franca con bitrate molto più piccolo di "vita reale" metraggio. Comunque, nella maggioranza dei casi un bitrate medio nell'intervallo tra i **500 kbps** ei **2500 kbps** dovrebbe dare risultati accettabili per il materiale SD più, come il Video-DVD di backup. Bitrate medio oltre i **2500 kbps** sono considerati "Overkill" per il materiale SD. Delle uscite di eccezioni ovviamente! Anche essere consapevoli che quando si tratta di materiale HD (720p o 1080p) significativa bitrate più alti saranno necessari. Bitrate di **10 Mbps** e soprattutto non sono eccezionali per codifica HD. Si noti che il trattamento pre-, come denoising, può ridurre il requisito bitrate di una sorgente.
  - **Si prega di notare:** Dal momento che è piuttosto difficile per decidere su un bitrate specifico, si sono di solito meglio utilizzando la modalità di *CRF*, invece di uno dei *bitrate basato su* modalità.
- **Lossless Mode:** x264 supporta anche vero "lossless" compressione. Con questa modalità non ci sarà *alcun* degrado ai dati video. Ma la compressione *lossless* avrà un sacco di bitrate per ovvi motivi! Comunque, x264 in modalità lossless probabilmente ancora prendere bitrate inferiore rispetto ad altri encoder lossless, come ad esempio HuffYUV o ffv1. Al fine di applicare la compressione lossless, è necessario scegliere la modalità di *quantizzatore costante* e si deve impostare la quantizzazione di un valore pari a **0**. Si noti che la riproduzione di filmati H.264 senza perdita di dati richiede un decoder in grado di "Predictive Lossless" profilo. Decoder che supportano questo sono libavcodec / ffmpeg (ffdsHOW, MPlayer, ecc) e CoreAVC. Altri decoder può visualizzare l'output è disturbato (o non di uscita a tutti).

## Pixel Aspect Ratio

- Questa impostazione definisce il "Pixel Aspect Ratio" (PAR) del video. **Non** modificare il valore predefinito di **1:1** (aka "Square Pixel"), a meno che non si codifica video *anamorfico*! ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Anamorphic\\_widescreen&prev=\\_t&rurl=translate.google.it&usg=ALkJrhiP50XMEDE8snaqAHmMXEbiJld6KQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Anamorphic_widescreen&prev=_t&rurl=translate.google.it&usg=ALkJrhiP50XMEDE8snaqAHmMXEbiJld6KQ)) Nel caso in cui si codifica il materiale anamorfico e si desidera mantenere la anamorfico, allora si dovrà impostare il valore corretto PAR. In caso contrario, il video potrebbe essere visualizzato con *aspect ratio* sbagliato! Se si dispone di una fonte *anamorfico* e si desidera convertire in "Piazza Pixel" (PAR = 1:1), quindi è necessario richiamare il filtro Resize e ridimensionare il video di conseguenza. Si noti che "Pixel Aspect Ratio" **non** è uguale a "Display Aspect Ratio" (DAR). Comunque, il DAR può essere calcolato dal PAR utilizzando questa formula:  $DAR = larghezza / altezza * PAR$ . Ad esempio:  $720/576 * 64/45 = 16 / 9$ . Il vantaggio di lavorare con valori di PAR è che il PAR di un video non cambierà quando il ritaglio del video, mentre il DAR molto probabilmente *cambierà*. PAR le seguenti opzioni sono disponibili:

- **Custom:** Immettere un valore definito dall'utente PAR
- **Predefiniti Aspect Ratio:** Scegli uno dei valori più comuni PAR dalla lista
- **Come input:** Mantenere il PAR della sorgente video

## Multi-Threading

- Questa impostazione controlla come molti thread x264 verrà utilizzato per la codifica. Grazie alla sua multi-threading attuazione, x264 è in grado di utilizzare appieno la potenza di elaborazione multi-core moderno. Questo si ottiene la codifica diversi frame in parallelo (vedi [1] ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://git.videolan.org/gitweb.cgi%3Fp%3Dx264.git%3Ba%3Dblob%3Bf%3Ddoc/threads.txt&prev=\\_t&rurl=translate.google.it&usg=ALkJrhgUe7ZvGtybx CZ839490Q-IL1VKZQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://git.videolan.org/gitweb.cgi%3Fp%3Dx264.git%3Ba%3Dblob%3Bf%3Ddoc/threads.txt&prev=_t&rurl=translate.google.it&usg=ALkJrhgUe7ZvGtybx CZ839490Q-IL1VKZQ)) per i dettagli). Test hanno dimostrato che x264 scale molto bene almeno fino a 16 core. ([http://avidemux.org/admWiki/images/8/84/X264\\_16\\_cores.png](http://avidemux.org/admWiki/images/8/84/X264_16_cores.png)) Comunque, per rendere l'utilizzo ottimale della vostra macchina multi-core, il corretto numero di thread deve essere selezionata. Le opzioni disponibili sono:
- **Disabili:** Disabilita il multi-threading. Un unico filo conduttore sarà utilizzato. Questo *non fa alcuna* differenza per i single-core macchine, ma rallentare *significativamente la* x264 su macchine multi-core.
- **Auto-Detect:** Let x264 decidere il numero ottimale di thread. La formula utilizzata è la seguente:  $threads = cpu\_cores * 3 / 2$ . Test hanno dimostrato che questa formula (di solito) ha pronunciato la prestazione ottimale. Si raccomanda di mantenere questa impostazione!
- **Custom:** manuale di sovrascrivere l'individuazione x264 auto. Utilizzare solo questo, se avete una buona ragione per diffidare x264 di rilevamento.
- **Commento:** Molte persone si lamentano che il carico della CPU non raggiunge il 100% in Taskmanager durante la codifica di un video con x264, anche con il multi-threading abilitato. Questo può avere varie cause. Più probabile che qualcosa nella catena di trasformazione è risistemazione x264. Ad esempio, un single-threaded *decoder* e / o compute-intensive filtri video possono facilmente diventare il collo di bottiglia delle prestazioni. In tal caso, x264 deve attendere per l'ingresso e diventa inattivo. Quindi, in realtà non è un problema in sé x264. Un altro "problema" è la tecnologia *Hyper-Threading* di Intel, ed usati per il Pentium 4 e il Core i7 (Nehalem) del processore. Con Hyper-Threading vi sono *due* core virtuali per core fisici. Quindi il carico della CPU del 50% indica che tutti i core *fisici* sono occupati (il che equivale a carico del 100% su un None-CPU Hyper-Threading). Ultimo ma non meno importante l'efficienza del multi-threading non deve essere misurata con il carico della CPU, come mostrato dal TaskManager. Invece il throughput (che è: il numero di fotogrammi codificati al secondo) deve essere misurato. Quindi, per favore di tenere presente che il carico della CPU alto da solo non significa buona prestazione!

## Motion

### Motion Estimation

- **ME Metodo:** compressione video opere di scartare le informazioni ridondanti tra fotogrammi consecutivi. Per esempio, P-frame sarà previsto dal fotogramma precedente (s). Quindi solo la *differenza* tra il telaio e la cornice



previsto sorgente originale viene salvato nel bitstream. Che è chiamato il "residuo". La più accurata una cornice è previsto, i dati meno bisogno di essere conservati. Perché gli oggetti tendono a *spostarsi* tra le strutture vicine, l'individuazione e che compensa il movimento è essenziale per una previsione accurata! Il *Metodo ME* determina quale algoritmo viene utilizzato per la ricerca di movimento e di calcolare i cosiddetti "vettori di movimento". Utilizzando un metodo più accurato di ricerca si tradurrà in una migliore qualità visiva, ma anche più tempo per la codifica. Utilizzando un metodo più veloce sarà un'accelerazione del processo di codifica, ma anche portare a una qualità visiva peggiore. Dato che il metodo ME ha un *enorme* impatto sulla velocità di codifica e la qualità visiva del video, si dovrebbe decidere con attenzione! E 'vivamente consigliato di **non** scendere al di sotto l'impostazione predefinita. Si dovrebbe prendere in considerazione una modalità ancora più lento, se la qualità è più importante il tempo di codifica. I metodi sono disponibili i seguenti:

- **Cerca Diamond (DIA):** Quattro forma unilaterale di analisi - Questo è il metodo più veloce, ma fornisce anche la qualità peggiore. Per utilizzare questo metodo solo se la velocità di codifica è più importante della qualità.
  - Complessità:  $O(n)$  nel caso peggiore, più veloce nel caso medio.
- **Cerca esagonale (HEX):** Sei forma unilaterale di analisi - La è il metodo *predefinito*. Esso fornisce una qualità ragionevole e lavora ancora abbastanza veloce.
  - Complessità:  $O(n)$  nel caso peggiore, più veloce nel caso medio.
- **Irregolari Multi-Hexagon Ricerca (UMH):** versione più dettagliata di ricerca esagonale - Questo metodo garantisce una qualità elevata, ma le opere più *lento* di quello della semplice "HEX" metodo. Se preferite la qualità rispetto alla velocità, quindi utilizzare questo metodo.
  - Complessità:  $O(n)$ .
- **Esaustivo Search (ESA):** un'analisi completa ed esauriente - Questo metodo di forza bruta opere *molto lento*, ma la qualità risultante è di solito solo *leggermente* migliore rispetto al "UMH metodo" (se non del tutto).
  - Complessità:  $O(n^2)$ .
- **Hadamard ricerca esaustiva (TESA):** versione migliorata di "ESA" metodo con trasformata di Hadamard - Questo metodo funziona anche più lento del "ESA" metodo. Utilizzare questo metodo, se avete un sacco di tempo da perdere.
  - Complessità:  $O(n^2)$ .
- **Commento:** i test hanno dimostrato che la "ricerca esaustiva" è *significativo* più lenta "Uneven Multi-Hexagon di ricerca", ma non necessariamente produrre una migliore qualità *percepita*. Inoltre "Hadamard ricerca esaustiva" ci vorranno almeno il doppio del tempo come "irregolare Multi-Hexagon di ricerca". Pertanto, utilizzando un metodo più lento "Uneven Multi-Hexagon Search" in generale non vale il tempo supplementare di codifica.
- **Subpixel Affinamento:** Questa impostazione (noto anche come "Sub ME") controlla la precisione del processo di stima del movimento. Maggiore è la precisione, migliori saranno i risultati. Di precisione corso superiore richiede più tempo per la codifica. Si noti che, indipendentemente dal setting, **QPel** motion estimation è sempre usato. **RDO** è pari a utilizzare l'impostazione VHQ del codificatore Xvid. E 'vivamente consigliato di *non* scendere al di sotto del valore di default di **6**. Anche **Psy-RDO** richiede almeno *Sube ME* 6 per essere efficace. Se potete permettervi il tempo, allora si dovrebbe prendere in considerazione un valore ancora più elevato. E nel caso la qualità visiva è più importante la codifica in tempo, si dovrebbe anche andare con il massimo! Le seguenti modalità sono attualmente disponibili:
  - 1. QPel SAD (più veloce, di qualità peggiore)
  - 2. QPel SATD
  - 3. HPel sulla MB, quindi QPel
  - 4. Sempre QPel
  - 5. QPel & ME bidirezionale
  - 6. RD su I-e P (default, la modalità più bassa che supporta Psy-RDO)
  - 7. RD su tutti i fotogrammi
  - 8. Raffinatezza RD su I-e P-Frames
  - 9. Raffinatezza RD su tutti i fotogrammi (più lento, qualità migliore)

## Motion Vector

- **Range:** Questa impostazione definisce il modo in molti pixel vengono analizzati per la stima del movimento. Risultato valori di alta *gamma* in un'analisi più accurata, ma anche di rallentare la velocità di codifica in modo significativo. Valori più bassi accelerare il processo di codifica, ma produrrà anche in una analisi meno accurata. Si noti che il materiale ad alta risoluzione prestazioni generalmente più dalle impostazioni di *portata* superiore a bassa risoluzione marziali. Questo perché gli oggetti tendono a spostarsi più lontano (in relazione ai pixel) in video HD. In ogni caso, il valore predefinito di **16** è sufficiente per la maggior parte dei video! Il "Diamante ricerca" e "esagonale Ricerca" metodi sono ancora limitate a portata massima di *16*. Se la qualità è più importante della velocità di codifica e se si utilizza il "multi-esagono irregolare di ricerca" il metodo (o un metodo ancora più lento), si consiglia di aumentare *gamma* per un valore di **24** o addirittura **32**. A seconda del metodo scelto ME, il *valore* di fondo può essere arrotondato a un multiplo di *due* o *quattro*.
- **Massimo Motion Vector Lunghezza:** Questa impostazione può essere utilizzata per limitare la lunghezza massima di ciascun vettore di movimento. Per impostazione predefinita x264 limita la lunghezza massima dei vettori di movimento in base al livello rilevato. È possibile utilizzare *questa* opzione per sovrascrivere x264 decisionale. Si raccomanda di **non** utilizzare questa opzione, a meno che non hai una buona ragione per farlo!
- **Minimo cuscinetto tra Threads:** x264 utilizza un frame-based, multi-threading metodo. Per consentire la codifica di fotogrammi in parallelo, x264 deve garantire che ogni macroblocco dato utilizza i vettori di movimento solo da pezzi di fotogrammi di riferimento che sono stati già codificati. Questo di solito non è visibile, ma può questione di molto veloce movimento verso l'alto. Per impostazione predefinita x264 deciderà lo spazio minimo tra i thread in base al numero di thread. È possibile utilizzare *questa* opzione per sovrascrivere x264 decisionale. Si raccomanda di **non** utilizzare questa opzione, a meno che non hai una buona ragione per farlo!

## Previsione

- **B-Frame Direct Mode:** Questa caratteristica permette di B-frame da usare "predetto" vettori di movimento al posto di codifica in realtà il movimento di ogni frame. Ciò dovrebbe risparmiare un po' il bitrate e si migliora la compressione. Pertanto si consiglia di mantenere *sempre* questa impostazione è *abilitata*. Ci sono quattro diverse modalità disponibili:
  - **Nessuno:** Disabled. Per la prova unica. *Sconsigliato*.
  - **Auto:** Lasciate che il codificatore decidere l'impostazione ottimale per ogni fotogramma. Altamente raccomandato per tutti i modi di RC, ma anche più efficiente in *modalità a due passi*.
  - **Temporale:** Applicare la previsione di strutture vicine.
  - **Spaziale:** Applicare la previsione dai blocchi vicini all'interno del frame corrente (di solito preferito su *temporale*).
- **Previsione ponderata per il B-Frames:** Questa funzione consente l'encoder per la produzione di B-Frames *più accurato* da "ponderazione" i fotogrammi di riferimento in nessuna maniera simmetrica. Questo avviene a costo di qualche velocità di codifica. Poiché ponderata B-Frames in generale migliorare la qualità visiva, si consiglia di mantenere sempre questa impostazione è **abilitata**, tranne la velocità di codifica è più importante della qualità. Sembra che questa opzione è più utile per la "vita reale", filmati e meno utile per filmati animati.

## Partizione

### Partizione di ricerca

- **8x8 Adaptive DCT Trasformazione:** Questa impostazione ha consentito alla 8x8 DCT *adattativa* trasformare. Questo migliorerà *notevolmente* la qualità visiva ad un costo minore velocità. In realtà questa opzione è noto per dare la velocità *bests / commerciale* di qualità-off di tutte le opzioni. Purtroppo richiede un "alto profilo" decoder H.264 in grado. Si raccomanda di mantenere questa opzione *abilitata*, se possibile!
- **8x8, 8x16 e 16x8 P-Frame Search:** Questa impostazione consente di 8x8 partizioni su P-frame e migliora quindi la qualità visiva di questi fotogrammi. Si consiglia di mantenere questa opzione *abilitata*!
- **8x8, 8x16 e 16x8 B-Frame Search:** Questa impostazione consente di 8x8 partizioni su B-frame e migliora quindi

la qualità visiva di questi fotogrammi. Si consiglia di mantenere questa opzione abilitata!

- **4x4, 4x8 e 8x4 P-Frame Search:** Questa impostazione consente di 4x4 partizioni su P-frame, ma di solito il miglioramento della qualità sarà trascurabile. Quindi questa opzione *non* vale il tempo supplementare di codifica e quindi può *essere* spento con sicurezza.
- **8x8 I-Frame Search:** Questa impostazione consente di 8x8 partizioni su I-frame e migliora quindi la qualità visiva di questi fotogrammi, ma richiede *8x8 Adaptive DCT Transform*. Si consiglia di mantenere questa opzione abilitata, se possibile!
- **4x4 I-Frame Search:** Questa impostazione consente di 4x4 partizioni su I-frame e migliora quindi la qualità visiva di questi fotogrammi. Si consiglia di mantenere questa opzione abilitata!
- **Commento:** Durante il processo di codifica, il codificatore abbattere i video in relazione alla cosiddetta "macroblocchi". Poi si cerca di blocchi simili, al fine di scartare i dati ridondanti (vedi *Motion Estimation*). Il macroblocchi può essere suddiviso in 16x8, 8x16, 8x8, 4x8, 8x4, 4x4 e partizioni. Analizzando *più* di questi risultati partizioni in una previsione più accurata e migliora quindi la qualità visiva. Purtroppo questo avviene a costo di ulteriori tempi di codifica. In genere si consiglia di conservare *tutti* i tipi di partizione *attiva*, fatta eccezione per il "4x4 P-Frame" partizioni. Questo perché la ricerca 4x4/4x8/8x4 partizione su P-frame costi di una notevole quantità di tempo di codifica, ma il guadagno in termini di qualità di solito è trascurabile (solo video a bassa risoluzione *possono* beneficiare). Si noti che alcune delle opzioni di partizione dipendono l'uno dall'altro! Inoltre si deve tener conto del fatto che *8x8 Adaptive DCT Transform* (e di conseguenza *8x8 I-Frame Search*) sono "High Profile" caratteristiche e sarà necessario un decoder H.264 adatti, come MPlayer, ffdshow o CoreAVC. Tuttavia *Adaptive DCT 8x8* e *8x8 Transform I-Frame di ricerca* sono estremamente utili funzioni.

## Cornice

### Frame Encoding

- **CABAC:** Questa impostazione consente la codifica CABAC entropia, uno dei x264 'caratteristiche più impressionanti. CABAC (Context Adaptive Binary Arithmetic Coding) opere assolutamente *lossless*, ma dà una spinta ulteriore compressione di circa il 15%. A quantizzatori più alti CABAC può risparmiare ancora di più bitrate - fino al 50% e più è possibile (cfr. [2]). (<http://akuvian.org/src/x264/entropy.png>) Di conseguenza, con CABAC consentire vi sia ottenere un file di dimensioni inferiori a stessa qualità (*CRF* e modalità *QP*) o migliore qualità con le stesse dimensioni del file (modalità 2-Pass). Quindi è altamente raccomandato di mantenere CABAC **attivato** in tutti i casi! Tuttavia CABAC richiede tempo di CPU supplementare per entrambi, la codifica e la decodifica! Il tempo di CPU supplementare necessario per CABAC dipende molto il bitrate. Si noti che CABAC può facilmente diventare il più compute-intensive parte di decodifica H.264! Se si decide di *disattivare* CABAC (che di solito **non** dovrebbe fare), quindi la meno efficiente, ma più veloce CAVLC (Context Adaptive Variable Length Coding) saranno utilizzati.
  - **Commento:** si noti che CABAC richiede almeno un "Main" profilo decoder H.264 in grado. Se ci si rivolge per la "Baseline" o "Extended" profilo, quindi CAVLC deve essere usato!
- **Pure Interlaced Mode:** Questa impostazione consente la codifica interlacciata, in modo da attivare questa impostazione *solo* se si video è *interlacciato*. ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://100fps.com/&prev=\\_t&rurl=translate.google.it&usg=ALkJrhjY3vGIFmj6f13lhlc53azI4yb0g](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://100fps.com/&prev=_t&rurl=translate.google.it&usg=ALkJrhjY3vGIFmj6f13lhlc53azI4yb0g)) Nel caso in cui il video è *progressivo* (cioè: non interlacciato) o se non si sa cosa "interlacciata" si intende, tenere lontano da questa impostazione! Essere consapevoli del fatto che la codifica di un video *interlacciato* come *progressiva* distruggerà il contenuto! Allo stesso tempo, che codifica per un *progressivo* come *interlacciato* è fattibile, ma in modo significativo l'efficienza di codifica fa male! Ultimo ma non meno di notare che la x264 è l'attuazione di codifica *interlacciata* non è così efficace come potrebbe essere. Quindi, se avete a che fare con una sorgente *interlacciata*, si sono di gran lunga meglio utilizzare un filtro deinterlace e codificare il video come *progressivo*.
  - **Commento:** Ora che i monitor CRT stanno scomparendo e schermi LCD / Plasma sono dominare il mondo, contenuti *interlacciati* deve essere deinterlacciati in playback in tempo comunque. Purtroppo alcune schermate uso deinterlaccia piuttosto povero, con una conseguente immagine nervosa / contrasto. Quindi il modo migliore è quello di deinterlacciare *prima della* codifica del video, utilizzando un deinterlacer di alta



qualità / bobber, come Yadif o TDeint.

- **Loop Filter:** Questa impostazione controlla uno dei x264 'caratteristiche più importanti: il filtro **Inloop Deblocking**. A differenza di MPEG-4 ASP (DivX, Xvid, ecc), lo sblocco Inloop è una caratteristica *obbligatoria* dello standard H.264. Così l'encoder, x264 in questo caso, può contare su il *decoder* per eseguire una corretta deblocking. Inoltre, tutti i P e B-Frames di flussi H.264 si riferiscono ai fotogrammi *sbloccati*, invece di quelli non lavorati, che migliora la comprimibilità. Non c'è assolutamente **alcuna** ragione per la disattivare completamente la Deblocking Inloop, quindi è altamente consigliato di tenerlo **attivato** in tutti i casi. Ci sono due impostazioni disponibili per configurare il filtro Inloop Deblocking:
  - **Forza:** Questa impostazione è anche chiamato "Alpha Deblocking". Esso controlla quanto il filtro Deblocking si *liscia* il video, quindi ha un effetto importante sulla *nitidezza* complessiva del vostro video. Il valore predefinito è **0** e dovrebbe essere sufficiente per appianare tutti i blocchi dal tuo video, in particolare in modalità di quantizzazione (QP o CRF). Valori *negativi* darà un video più forte, ma si aumenta anche il pericolo di blocco di artefatti visibili! In contrasto con i valori *positivi* si tradurrà in un video fluido, ma anche eliminare ulteriori dettagli.
  - **Soglia:** Questa impostazione è anche chiamata "Beta Deblocking" ed è più difficile da gestire rispetto alla Alpha Deblocking. Esso controlla la soglia per il rilevamento di blocco. Il valore predefinito è **0** e dovrebbe essere sufficiente per individuare tutti i blocchi nel tuo video. Valori *negativi* "salvare" i dettagli, ma i blocchi più potrebbe scivolare attraverso (soprattutto in aree pianeggianti). In contrasto con i valori *positivi* rimuoverà i dettagli e le catture più blocchi.
  - **Commento:** In generale non vi è alcuna necessità di modificare l'impostazione predefinita di **0:0** per forza: limite, in quanto dà risultati molto buoni per una vasta gamma di video. Tuttavia è possibile provare diverse impostazioni per trovare le impostazioni ottimali per i *tuo*i occhi. Se ti piace un video più forte e non mi importa a pochi isolati qua e là, allora potreste essere felice con **-2: -1**. Questo potrebbe anche essere la pena di provare per MPEG-4 ASP (DivX, Xvid, ecc) gli utenti! Se ti piace un immagine liscia e pulita o codificare un sacco di roba Anime, allora si può provare qualcosa di simile a **1:2**. Tuttavia *non* si deve lasciare l'intervallo tra **-3** e **2** per entrambe le impostazioni!
- **Max. Ref. frames:** A differenza di MPEG-4 ASP, H.264 permette di fotogrammi di riferimento *moltiplici*. Questa impostazione controlla il numero di fotogrammi possono essere referenziate P e B-Frames. I valori più alti si otterrà un effetto di compressione più efficiente, il che significa una migliore qualità visiva a dimensioni del file stesso. Purtroppo quadri di riferimento più richiederà più tempo per la codifica (e di potenza della CPU anche un po' più piccolo per la riproduzione). Per impostazione predefinita il numero di fotogrammi di riferimento è limitato a **1**. Si raccomanda di aumentare il numero di riferimenti ad almeno **3**. Tuttavia, utilizzando più di **4** o **5** fotogrammi di riferimento per la "vita reale" riprese dovrebbero essere evitati, in quanto non migliorerà ulteriormente i risultati! Allo stesso tempo, Anime e cartoni animati beneficiare molto dal frame di riferimento supplementari. A volte anche il massimo di **16** fotogrammi di riferimento può essere utile per tale materiale.
  - **Commento:** Anche se il "software" di solito i giocatori supportare *qualsiasi* numero di fotogrammi di riferimento, "hardware" i giocatori sono limitate a un numero massimo di fotogrammi di riferimento! Il numero massimo di fotogrammi di riferimento può essere calcolato a partire dalla "Max Decoded Picture Buffer Size" (MaxDPB) e la risoluzione del video. Il valore MaxDPB è definito dai singoli H.264 profilo supportato da parte del giocatore (per maggiori dettagli vedi l'allegato A delle specifiche H.264).

## B-Frames

- **Max consecutive:** Questa impostazione controlla il numero *massimo* di consecutive *B-Frames*. B-frame si riferiscono ad entrambi, il precedente *ed* il successivo I-Frame (o P-frame). In questo modo, B-Frames in grado di comprimere anche più efficiente di P-Frames. B-frame possono migliorare significativamente la qualità visiva del video con le stesse dimensioni del file. Pertanto di utilizzare fotogrammi B è altamente raccomandato. Si noti inoltre che permettano *maggior*e B-Frames non farà male alla qualità: Si può anche *tranquillamente* scegliere il massimo di **16** consecutive B-Frames. Questo perché solo specificare il limite superiore per il numero dei *consecutive* B-Frames. x264 sarà ancora decidere quante consecutive B-Frames sono effettivamente utilizzati. Così, anche se si consente fino a 16 consecutive B-Frames, l'encoder raramente andare così in alto. Tuttavia, la limitazione del numero massimo di B-Frames a *meno* di 16 è ragionevole, perché la maggior parte video non potranno beneficiare di utilizzare più ~ **4** consecutive B-Frames comunque! Alzare il B-Frame limite superiore a quello che sarebbe solo

un rallentamento del processo di codifica per *alcun* vantaggio reale! Se si imposta il limite di B-Frame a **0** (di default), B-Frames verrà *disattivato*. Naturalmente disabilitando B-frame **non** è raccomandato!

- **B-Frame Bias:** Questa impostazione controlla la probabilità che un B-frame viene utilizzato al posto di un P-Frame. Il valore predefinito è **0**, che è anche l'impostazione consigliata. Un valore *positivo* aumenta la probabilità che un B-Frame è impostato. In contrasto con un valore *negativo* diminuisce la probabilità che un B-Frame è impostato. Naturalmente l'encoder non sarà mai violare il limite *massimo consecutiva*, non importa quale impostazione *diagonale* viene utilizzato.
- **Adaptive B-Frame decisione:** Questa opzione controlla come l'encoder sceglie il numero dei consecutive B-Frames. Non importa quali impostazioni si sceglie, l'encoder *non* sarà *mai* violare la massima consecutiva B-Frame limite (ma si può decidere di utilizzare *meno* B-frame). Le modalità following sono disponibili:
  - **Fast:** Questa modalità utilizza un B veloce e ottimale algoritmo di decisione Frame. Si utilizza di solito un numero molto basso di B-Frames, anche con una B piuttosto elevato limite di Frame. Utilizzare questa modalità solo se si favoriscono la velocità su qualità!
  - **Ottimali:** Questa modalità è anche nota come "Trellis B-Frame decisione", ma non è legato alla possibilità di quantizzazione *Trellis* a tutti. Questo metodo è significativo più lento del "Fast" B-Frame metodo di decisione, ma trovare il numero ottimale di B-Frames. In particolare si affievolisce vengono gestiti molto meglio con questo metodo. In quanto la velocità di questo metodo dipende *in* larga misura dal limite di B-Frame, si dovrebbe limitare il numero massimo di consecutive B-Frames ad un valore ragionevole!
  - **Disabili:** Questa opzione consente di *disattivare* la B-adaptive decisione Frame. Utilizzare questo solo per i test!
- **Utilizzare come riferimento:** Questa caratteristica è spesso chiamata "B-Pyramid". Se si attiva questa impostazioni, B-Frames sono autorizzati a fare riferimento non-lineare, al fine di migliorare l'utilizzo bitrate e la qualità. In questo modo, B-Frames può fare riferimento a B-Frames. Di solito si consiglia di mantenere questa funzione è **abilitata** come dovrebbe migliorare il risultato. Tuttavia si deve tenere conto che questa è una "High Profile" caratteristica e richiede pertanto un decoder adatto, come MPlayer, ffdshow o CoreAVC.

## I-Frames

- **Dimensioni minime GOP:** Questa impostazione controlla il numero *minimo* di fotogrammi tra due fotogrammi IDR. Fotogrammi IDR sono simili ai fotogrammi chiave in video MPEG-4 ASP: la riproduzione può essere avviata solo in un fotogramma IDR, come nessun fotogramma *dopo* il fotogramma IDR farà riferimento a un frame *prima* del fotogramma IDR. In H.264 questo *non* è possibile con "normale" I-Frame, a causa di molteplici riferimenti. Così fotogrammi IDR sono necessari per permettere alla ricerca nel video. Tuttavia fotogrammi IDR troppi causerebbe una codifica inefficiente, quindi c'è un intervallo *minimo* per i fotogrammi IDR. Come regola generale, questo valore dovrebbe essere uguale il *framerate* del video. Ad esempio, un 25 fps video dovrebbe usare un valore di **25**, a 29.97 fps video dovrebbe usare un valore di **30** e così via.
- **Dimensioni massime del GOP:** A differenza di "Min intervallo fotogramma IDR" questa impostazione controlla il numero *massimo* di fotogrammi tra due fotogrammi IDR. Un valore *più alto* si tradurrà in un intervallo di IDR più frame e quindi cerca di rallentamento, un valore *più basso* si tradurrà in un intervallo di IDR più brevi e quindi migliorare la ricerca. Come regola generale, l'intervallo fotogramma IDR non deve essere inferiore al framerate del video moltiplicato per un fattore *10*. Ad esempio, un 25 fps video devono utilizzare almeno un valore di **250**, 29.97 fps video devono utilizzare almeno un valore di **300** e così via. Utilizzando i valori anche superiori a migliorare la compressione a costo di qualche ricerca di prestazioni. Naturalmente marziali con molti "prende tempo" e lungo "colpi di monitoraggio" potranno beneficiare molto di più da long GOP's a marziale che consiste principalmente di scene molto brevi. Si prega di notare che il long GOP's farà male di recupero degli errori, che possono essere un problema per la cottura a vapore di media (e anche per l'autoring Blu-Ray)
- **Scene Cut Soglia:** Questa impostazione controlla la soglia per l'x264 '*cambio di scena*'. In questo modo l'encoder può mettere un I-frame ad ogni cambio di scena (invece di un P-o B-Frame), che dovrebbe portare a più bello scena tagli. Un risultato di soglia *più bassa* in un rilevamento più aggressivo cambio di scena, che potrebbero essere utili per video molto scuro. In contrasto con una soglia *più elevata* in grado di rilevare cambiamenti di scena di meno. Il

valore predefinito è **40** e deve essere adatta per la maggior parte dei video.

## Analisi

### Analisi di configurazione

- **Refs misto:** Se questa impostazione è attivata, ogni blocco 16x16 macro può scegliere il proprio (ottimale) fotogramma di riferimento. Questo rallenterà il processo di codifica, ma permette una compressione più efficiente. Specialmente se si usa un elevato numero di *fotogrammi di riferimento*, questa impostazione vi darà grande miglioramento e vale il tempo supplementare di codifica. Se si utilizzano sistemi di riferimento pochi, *Rif. mista* sarà meno efficace. Si dovrebbe mantenere questa impostazione è **abilitata**, se la qualità visiva è più importante della velocità di codifica.
- **Chroma ME (Motion Estimate):** Se questa impostazione è abilitata, quindi le informazioni sul colore (chroma) saranno presi in considerazione per il rilevamento del movimento, altrimenti non sarà. Con "Chroma ME" attivato il rilevamento del movimento sarà più lento ma più preciso. In modo che di solito produce una qualità visiva *superiore* al costo di qualche tempo di codifica. Pertanto si consiglia di mantenere sempre questa impostazione è **abilitata**, tranne la velocità di codifica è più importante della qualità visiva.
- **La quantizzazione trellis:** Questa impostazione consente la quantizzazione *trellis* RD. Fondamentalmente Trellis eseguirà un ulteriore passo di quantizzazione: Si terrà alcuni "dettagli" che sarebbe stato rimosso in altro modo. Allo stesso tempo sarà eliminare alcuni "dettagli" che altrimenti sarebbero rimasti altrimenti. Di solito Trellis *migliorerà* la qualità complessiva in modo notevole, ma provoca un significativo *rallentamento* del processo di codifica. Prima che le *ottimizzazioni Psy* sono stati aggiunti a x264, Trellis **2** è stato detto di avere una tendenza a rimuovere piccoli dettagli e migliorare i bordi. Pertanto Trellis **1** spesso è stata considerata la scelta migliore. Ma ora quando si usa **Psy RDO**, è altamente consigliato di utilizzare Trellis **2**, anche se è più lento rispetto a traliccio significativo **1**. Se la velocità è più importante della qualità, fissati Trellis a **0** per disattivarlo. Si noti che **Psy-traliccio** richiede la quantizzazione *trellis*, così *Psy-traliccio* saranno disattivati quando a traliccio è impostato a **0**. Si noti inoltre che *a traliccio* richiede *CABAC*. Le modalità disponibili sono:
  - **2: Always On** (lento, miglior qualità)
  - **1: Final Makroblock solo** (più veloce, di qualità media)
  - **0: Disabled** (più veloce, di qualità peggiore)
- **P Fast-salto:** Se questa impostazione è *selezionata*, "P-Fast Skip" verrà utilizzato. Fast P-Salta è un aumento di velocità di ottimizzazione che il processo di codifica a costo di qualche qualità visiva. Tuttavia, la perdita di qualità causate da P Fast-di solito è *trascurabile*, mentre l'aumento di velocità è distinto Salta. Pertanto si consiglia di mantenere *P Fast-Salta* controllati. Purtroppo, in rari casi P Fast-Salta la causa artefatti in "flat" scene, quindi si consiglia di deselezionare *P Fast-Salta* nel caso in cui la qualità visiva è più importante della velocità di codifica.
- **DCT Decimate:** Se questa impostazione è *selezionata*, *DCT decimazione* verranno utilizzati. Questa caratteristica permette x264 di disfarsi "inutile" blocchi DCT. Tali blocchi DCT non verrà scritto il flusso di bit, che consente di risparmiare circa il bitrate e migliora l'efficienza di codifica. Naturalmente ci sarà una perdita di qualità sottile, ma di solito l'effetto è trascurabile. Poiché DCT decimazione porta a significativi file più *piccoli* in modalità basata quantizzazione (QP o CRF) si consiglia di mantenere questa impostazione è *abilitata*. *Non* si deve disabilitare la *decimazione DCT*, meno che non abbiate una buona ragione per farlo. Rumors dicono che *DCT decimazione* non dovrebbero essere utilizzare insieme con la quantizzazione **trellis**, ma questo è stato smentito!
- **Riduzione del rumore:** Questa impostazione controlla x264 filtro 'Denoise interno. Si prega di notare che denoising *non fa* parte delle specifiche H.264! Quindi questo è essere considerato un ulteriore *pre-prcoessing* funzionalità. Il valore predefinito è **0**, che sarà completamente *disattivare* il filtro Denoise in x264. Non vi è alcuna necessità di modificare questa impostazione, ad eccezione esplicitamente desidera applicare denoising supplementari al tuo video *prima della* codifica. Di solito i valori buoni per la riduzione del rumore *non* sono superiori a **1000**. Comunque si di solito sarà meglio con un buon "stand-alone" filtro Denoise come FluxSmooth o denoise3d

MPlayer. Se si utilizza uno di questi, si prega di assicurarsi x264 'la riduzione del rumore è spento!

## Luma Quantization Deadzones

- **Intra Luma zona morta di quantizzazione**
  - [TO-DO] *Se sai quali sono le informazioni a mettere qui, non esitate a contattarci!*
- **Inter Luma zona morta di quantizzazione**
  - [TO-DO] *Se sai quali sono le informazioni a mettere qui, non esitate a contattarci!*

## Quantization Matrix

- **Appartamento di Matrix:** La quantizzazione è la parte *con perdita* di compressione video: I coefficienti sarà diviso attraverso la *matrice di quantizzazione* e poi arrotondato. La "Flat Matrix" è la matrice di quantizzazione di default delle specifiche H.264 - tutte le voci sono semplicemente riempite con 16's. Questa matrice è noto per dare risultati *abbastanza buoni* per una vasta gamma di video e bitrate. Ciò significa che la qualità *soggettiva* così come valori di PSNR.
- **JVT Matrix:** Questa è la matrice alternativa quantizzazione delle specifiche H.264. Test hanno dimostrato che il "JVT Matrix" *prestazioni* scarse, anche se fa parte delle specifiche ufficiali. Quindi è vivamente consigliato di **non** usare questa matrice, fatta eccezione per il collaudo e il confronto! Vi sarà molto meglio con la "Flat Matrix" in quasi ogni caso.
- **Custom Matrix:** Questa impostazione consente di caricare le tue matrici di quantizzazione *proprio*. La creazione di matrici di quantizzazione come un compito complesso, uno ha bisogno di una *profonda* comprensione di come funziona la compressione video in dettaglio. Così la creazione di matrici di quantizzazione nuovo dovrebbe essere riservato ai guru H.264. Tuttavia è possibile trovare un elenco di matrici adatte a questa ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://forum.doom9.org/showthread.php%3Fp%3D887125&prev=\\_t&rurl=translate.google.it&usg=ALkJrhxOLw765SNZqAdaQFhoAeC1rxN3Q#post887125](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://forum.doom9.org/showthread.php%3Fp%3D887125&prev=_t&rurl=translate.google.it&usg=ALkJrhxOLw765SNZqAdaQFhoAeC1rxN3Q#post887125)) e questa ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://forum.doom9.org/showthread.php%3Fp%3D677092&prev=\\_t&rurl=translate.google.it&usg=ALkJrhgK2QvtQrDI\\_5wGMW2CsaoFCaRGyA#post677092](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://forum.doom9.org/showthread.php%3Fp%3D677092&prev=_t&rurl=translate.google.it&usg=ALkJrhgK2QvtQrDI_5wGMW2CsaoFCaRGyA#post677092)) posizione. Si prega di notare che il "filtro di deblock" impostazioni di avere un enorme impatto sul modo in buono / cattivo una certa matrice di quantizzazione esegue! Anche la maggior parte delle matrici personalizzate sono destinati a una certa fascia bitrate (ad esempio, ultra elevata o ultra basso bitrate) e si esibirà *male* al di fuori di questo intervallo. Ultimo ma non meno importante *non* si deve usare ogni matrici personalizzate, ad eccezione di sapere cosa si sta facendo. Nella maggior parte dei casi si ottengono risultati soddisfacenti semplicemente attaccando con la "Flat Matrix" (default).
  - **Commento:** Ora che x264 contiene *psico-varie ottimizzazioni visiva* (Adaptive Quantization, Psy-RDO, Psy-Trellis) matrici di quantizzazione personalizzate sono diventati obsoleti! La maggior parte delle cose che la gente ha cercato di ottenere con matrici personalizzate, come ad esempio i dettagli e la conservazione del grano, ora può essere raggiunto da ottimizzazioni Psy in un modo più sofisticato. Inoltre, le ottimizzazioni Psy sono sintonizzati per l'impostazione predefinita matrice *piana*. Pertanto l'utilizzo di "estrema" delle matrici personalizzate possono provocare effetti indesiderati quando ottimizzazioni Psy sono coinvolti! Pertanto vi raccomandiamo di rimanere con il "piatto" a matrice, a meno che non hai una buona ragione per usare una matrice personalizzata.

## Psico-visivo ottimizzato RDO & Trellis

- **Commento:** L'occhio umano non solo desidera che l'immagine simile a quello originale, che vuole l'immagine di avere *complessità* simili. Pertanto, avremmo preferito vedere un po 'distorta, ma ancora a blocchi dettagliato di un non-distorto, ma completamente offuscata blocco. Il risultato è una tendenza verso una dettagliata e / o immagini sgranate di uscita, un po 'come xvid, tranne che i suoi dettagli reali, piuttosto che brutto blocco (cfr. [3]). ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://forum.doom9.org/showpost.php%3Fp%3D1144270%26postcount%3D1&prev=\\_t&rurl=translate.google.it&](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://forum.doom9.org/showpost.php%3Fp%3D1144270%26postcount%3D1&prev=_t&rurl=translate.google.it&)

usg=ALkJrhhiM\_W\_NaYrHcJnSLBWtaRRPjgdXQ) Lo scopo di **Psy RDO** è quello di mantenere la complessità di un blocco codificato simile alla complessità del blocco originario. In questo modo *Psy RDO* produce un'immagine che sembra molto più nitida e dettagliata, in molti casi (rispetto a *nessuno* Psy RDO). Aiuta anche a preservare la grana della pellicola molto! Si prega di notare che Psy RDO comporterà necessariamente un *male* metriche, come PSNR e SSIM. Appena psico-visiva ottimizzazioni sono coinvolti, la metrica classica diventa inutile! Si noti inoltre che Psy RDO lavorerà con modalità *RDO* solo: se la *partizione decisione* è impostato su **6** (o superiore), quindi Psy RDO sarà per impostazione predefinita, altrimenti sarà disattivato. In aggiunta a *Psy RDO* c'è anche **Psy-traliccio** ora. Questo è ancora considerato un "sperimentale" e *disattivare* la funzione di default, ma sembra di grande aiuto per il mantenimento delle trame ai video. Si noti che Psy traliccio si basa sulla quantizzazione *trellis*. Di conseguenza, sarà efficace solo con la quantizzazione trellis abilitato troppo (a traliccio **1** è sufficiente ora, ma **2** sarà più efficace).

- **Psy RDO Forza:** Questa impostazione controlla la forza di *Psy RDO*. Si noti che l'ultima patch *Psy RDO* automaticamente scala la forza di Psy RDO, in base alla quantizzazione del fotogramma! Pertanto, la "forza" stetting è semplicemente un ulteriore fattore, che sarà moltiplicato per il fattore di scala *interna*. Il valore predefinito per *Psy RDO Forza* è **1.0** e *non* si dovrebbe modificare tale valore se non avete una buona ragione per farlo!
- **Psy Trellis Forza:** Questa impostazione controlla la forza di *Trellis Psy*. Il valore predefinito è **0.0** al momento, in modo da Trellis Psy sarà *disabilitato* per default. Comunque *può* essere utile per utilizzare Psy traliccio per il tuo video. Ma attenzione! I test hanno dimostrato che un valore di **1.0** di solito è troppo forte per Trellis Psy. Per le fonti più un valore di **0,2** fino a **0,4** dovrebbe essere sufficiente. Anche i valori più alti possono introdurre artefatti!
- **Nota:** Psy RDO e Trellis Psy sono stati ufficialmente impegnato ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://git.videolan.org/gitweb.cgi%3Fp%3Dx264.git%3Ba%3Dcommit%3Bh%3Dabe641635e9dc1209af4e201f9308edd33fe3096&prev=\\_t&rurl=translate.google.it&usg=ALkJrhjaAZrkTb3LrNTpg7BAIVhY1\\_CR4g](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://git.videolan.org/gitweb.cgi%3Fp%3Dx264.git%3Ba%3Dcommit%3Bh%3Dabe641635e9dc1209af4e201f9308edd33fe3096&prev=_t&rurl=translate.google.it&usg=ALkJrhjaAZrkTb3LrNTpg7BAIVhY1_CR4g)) e quindi sono diventati parte integrante della x264. Le più recenti versioni di Windows Avidemux conterrà x264 con Psy RDO e Trellis Psy, anche se non ci sono controlli GUI per regolare Psy RDO / a traliccio (in Avidemux) ancora. Quindi le impostazioni di default saranno utilizzati: Mentre Psy RDO è attivata (a forza 1.0), Psy Trellis è *disattivato*. Patched x264 costruisce il mio utilizzo dei valori predefiniti diversi però!

## Quantizzatore

### Quantizzazione di controllo

- **Minimo di quantizzazione:** Specifica il valore *minimo* di quantizzazione deve essere eseguita. Questo significa che *ogni frame ottenere almeno* questa quantità di perdita di dati. Il valore predefinito è **10**, il che assicura non bitrate è sprecato il quantizzatore troppo bassa. Questo valore deve essere a posto, anche per i video di alta qualità.
- **Massimo di quantizzazione:** Specifica il *massimo* di quantizzazione di essere ammessi. Ciò significa che *nessuno* dei Frames otterrà un importo più elevato di perdita di dati di questo. Il valore predefinito è **51**, che è la quantizzazione massimo possibile. Così "Max Qp" *non* è limitato per impostazione predefinita. Naturalmente l'encoder solo andare così in alto, quando è veramente necessario, quindi non preoccupatevi!
- **Passo massimo di quantizzazione:** Specifica quanto la quantizzazione può cambiare tra due fotogrammi consecutivi. Il valore di default è **4**. Questo fa sì che due fotogrammi consecutivi non avranno quantizzatore *troppo* diversi. Se si consente una maggiore significativi passi QP, questo potrebbe tradursi in qualità visibile "salti" tra i fotogrammi, in modo da non farlo.
- **Media Bitrate Tollerance:** Questa impostazione effetti il "Single Pass" bitrate modalità basata solo. Si controlla come *precisa* l'encoder colpirà il bitrate di destinazione (o le dimensioni del file di destinazione). L'obiettivo di "Bitrate varianza" è quello di avvicinarsi il più possibile la qualità di un modo di codificare CRF, pur essendo da qualche parte vicino la dimensione del file di destinazione. Un valore di **0.0** limiterebbe l'encoder a colpire *esattamente* il bitrate desiderato. Il valore di default di **1.0** permette uno *scarto dell'1%*, che è ancora abbastanza

restrittivo, ma già molto meglio di CBR puro. Si prega di notare che la discrepanza di solito dovrebbe essere all'interno di una gamma di 30%. Inoltre, la modalità CRF dà sempre risultati molto migliori rispetto al bitrate a base di modalità e quindi è il metodo consigliato!

- **Fattore compreso tra I-e P-Quants Frame:** Questa impostazione controlla quante *più forte* P-frame sarà compresso rispetto a I-Frame. Un valore di **1.0** potrebbe assegnare lo *stesso* quantizzatore fotogrammi P e I-Frame, mentre il valore di default di **1,4** quantizzatori assegna *il 40%* superiore a P-frame (rispetto a I-Frame). Ciò equivale al "I-frame Boost" opzione di Xvid. Compressione P-frame più forte di I-frame è consigliato, come I-frame servono come riferimento iniziale di una scena, e quindi hanno un impatto enorme sulla qualità del frame successivo. Pertanto *non* si dovrebbe modificare il valore predefinito, a meno che non hai una buona ragione per farlo!
- **Fattore compreso tra P e B-Quants Frame:** Questa impostazione controlla quante *più forte* B-frame sarà compresso rispetto al P-Frames. Un valore di **1.0** potrebbe assegnare lo *stesso* quantizzatore a B-frame e P-frame, mentre il valore di default di **1,3** assegna un quantizzatore *30%* più elevato è il B-Frames (rispetto al P-frame). Compressione di B-Frames più forte del P-frame si raccomanda, come B-frame *non* sono referenziate da altri fotogrammi (tranne che per il B-Pyramid), mentre il P-frame come server di riferimento per il frame successivo. Pertanto *non* si dovrebbe modificare il valore predefinito, a meno che non hai una buona ragione per farlo!
- **Chroma di Luma scostamento di quantizzazione:** Questa impostazione controlla come molto più *forti* le informazioni sul colore (chroma) sarà compresso, rispetto alle informazioni di luminosità (luminanza). A volte si rende sens di comprimere le informazioni sul colore più forte le informazioni di luminosità, come la perdita di dati in informazioni sul colore è *meno* visibile all'occhio umano che la perdita di dati in informazioni di luminosità. L'offset specificato sarà *aggiunto* il quantizzatore chroma. Può essere configurato tra **-12** e **+12**. Il valore predefinito è **0** e di solito si consiglia di mantenere il valore di default! Si noti che entrambi, *Psy-RDO* e *Psy-traliccio*, riduca il compensato da uno o da due, se abilitato. Così si può finire con un offset di **-4** usando ottimizzazioni Psy.

## Quantizzatore Curve di compressione

- **Quantizzatore Curve di compressione (%):** questa impostazione è anche chiamata "qcomp" o "variabilità bitrate" (da *non* confondere con la *varianza bitrate*). Esso controlla quanto il bitrate può fluttuare sopra il video intero. L'impostazione di questa a **0%** sarebbe l'esecuzione di un flusso *costante di bitrate*, mentre un valore del **100%** si tradurrebbe in un flusso *costante di quantizzazione*. Il valore predefinito è **60%**, che dà buoni risultati per la maggior parte dei video. Quindi, *non* cambiare l'impostazione predefinita, a meno che non hai una buona ragione per farlo! Nota che la *quantizzazione adattiva* (AQ) sostituisce in parte l'effetto di *qcomp* e x264 sarà internamente sollevare qcomp a compensare sulla base della forza di quantizzazione adattiva. Si noti inoltre che la modalità *CRF* utilizzando insieme ad una *qcomp* del 100% è tecnicamente equivalente a modalità di *QP*, salvo che la modalità *CRF* è ancora in grado di utilizzare AQ (QP modalità che non può fare). Quindi più si sollevano *qcomp*, la modalità di *CRF* più vicino arriva a una codifica *QP*.
- **Ridurre la fluttuazione prima della curva di compressione:** Questa impostazione si applica una sfocatura gaussiana temporale della curva di quantizzazione *prima* della "quantizzazione di compressione" passo. Questo viene fatto allo scopo di appiattire le fluttuazioni del quantizzatore indesiderati, che dovrebbe rendere la qualità visiva più stabile, soprattutto in relazione ai contenuti Anime. Il valore predefinito è **20,0** e di solito non ha bisogno di essere cambiato.
- **Ridurre la fluttuazione dopo curva di compressione:** Questa impostazione si applica una sfocatura gaussiana temporale della curva di quantizzazione *dopo* la "quantizzazione di compressione" passo. Questo viene fatto allo scopo di appiattire *ulteriormente* le fluttuazioni del quantizzatore indesiderati. Il valore di default è di **0,5** e di solito non ha bisogno di essere cambiato.
- **Commento:** Per ulteriori informazioni su come x264 'controllo della velocità di opere in dettaglio si possono trovare in questa ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://git.videolan.org/gitweb.cgi/%3Fp%3Dx264.git/%3Ba%3Dblob%3Bf%3Ddoc/ratecontrol.txt&prev=\\_t&rurl=translate.google.it&usg=ALkJrhjSE-bRQ9D6QRnNgJPVNAT8BYUkPg](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://git.videolan.org/gitweb.cgi/%3Fp%3Dx264.git/%3Ba%3Dblob%3Bf%3Ddoc/ratecontrol.txt&prev=_t&rurl=translate.google.it&usg=ALkJrhjSE-bRQ9D6QRnNgJPVNAT8BYUkPg)) posizione.

## Adaptive Quantization

- **Commento:** *Adaptive Quantization (AQ)* consente a ogni macroblocco all'interno della cornice di scegliere un quantizzatore *diverso*, invece di assegnare la quantizzazione *stessa* a *tutti i* blocchi all'interno del telaio. Lo scopo di AQ si sta muovendo in più di bit in "flat" macroblocchi. Questo è fatto da adattativa abbassando il quantizzatore dei blocchi di alcuni (e aumentando il quantizzatore dei blocchi di altro tipo). Senza AQ, aree piane e scure dell'immagine tende a mostrare il brutto blocco o banding. Grazie al nuovo algoritmo di AQ, il blocco e bandaggio possono essere notevolmente ridotti! Con AQ abilitato, ci si può aspettare un aumento significativo (!) Guadagnano in qualità complessiva dell'immagine. Soprattutto nelle scene buie e scene con "flat" sfondi (cielo, erba, muri, ecc) i dettagli molto di più può essere conservato. AQ, tuttavia, sembra di eseguire meno efficienti su Cartoon / Anime materiale (ma comunque aiuta a prevenire banding). Si noti che AQ può essere utilizzato con il bitrate modalità di base (*Single-Pass* e *Two-Pass*), nonché con le modalità di *CRF*. Essa **non** può essere utilizzato con le modalità *QP*! Questo perché la modalità QP usa il quantizzatore costante per definizione, che è uno dei motivi per cui modalità PQ in generale deve essere evitata al giorno d'oggi.
- **Varianza AQ:** Questa impostazione controlla se *Adaptive quantizzazione* è utilizzato o meno. Attualmente vi è un solo AQ modalità disponibili in x264: *varianza AQ* (VAQ). Mantenere AQ **abilitato**, se possibile!
- **AQ Forza:** Questa impostazione controlla la quantità di AQ che viene applicata ai fotogrammi. Il valore predefinito è **1,0** *AQ Forza* ora, in modo AQ sarà *abilitato* di default. Il valore predefinito deve essere ben bilanciata e dare buoni risultati AQ per la maggior parte delle fonti. Se pensate che il vostro video richiede più forte AQ, allora si può aumentare la *Forza AQ*. Un valore di **1,5** è considerato "forte" AQ. Se pensate che l'effetto AQ è troppo forte, si può abbassare la *Forza AQ*. Un valore di **0.5** è considerato "basso" AQ.
- **Esempio:** x264 con VAQ-vs-No AQ ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://mplayer.somestuff.org/misc/vaq.html&prev=\\_t&rurl=translate.google.it&usg=ALkJrhioekAqMPxl6WIN8L-gWNVLax4yrg](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://mplayer.somestuff.org/misc/vaq.html&prev=_t&rurl=translate.google.it&usg=ALkJrhioekAqMPxl6WIN8L-gWNVLax4yrg))

## Advanced

### Video Buffer Verifier

- **Massimo VBV Bitrate:** Specifica il bitrate massimo *locale* supportata dal dispositivo. VBV assicura che questo bitrate non sarà superato. Il valore **0** indica che il bitrate massimo *non* è limitato (di default).
- **Dimensione massima del buffer:** Specifica la dimensione del buffer del dispositivo. VBV assicura che non sarà né il buffer overflow, né underflow. Un valore **0** indica che il controllo della frequenza non ha bisogno di cure per il buffer di dispositivo (default).
- **Iniziale VBV Buffer Occupancy:** Specifica il livello di riempimento del buffer dispositivo all'inizio della riproduzione.
- **Commento:** VBV (Video Buffering Verifier) sono richiesti *solo* per la compatibilità di periferiche hardware. H.264 usa anche il termine "ipotetico Riferimento Decoder" (HRD) invece di VBV. Si raccomanda di **non** utilizzare VBV, a meno che non si può aggirare l'ostacolo. VBV *possono* danneggiare la qualità del video in modo significativo, ma non potrà mai migliorare la qualità! Inoltre si raccomanda di codificare in modalità *2-Pass* VBV quando è coinvolto. Con *1-Pass* VBV può portare a significativi di qualità inferiore! (Cfr. [4] ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Video\\_buffering\\_verifier&prev=\\_t&rurl=translate.google.it&usg=ALkJrhjgLf4aun9eLN6dRZOxm6duWhB9iA](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Video_buffering_verifier&prev=_t&rurl=translate.google.it&usg=ALkJrhjgLf4aun9eLN6dRZOxm6duWhB9iA)) per i dettagli)

### Zone

- **Aggiungere:** Aggiungere una nuova zona alla lista.
- **Modifica:** Modifica una zona esistente.



- **Aggiungere:** rimuovere una zona dalla lista.
- **Commento:** Zone può essere utilizzato per assegnare *manualmente* un bitrate inferiore o superiore ad una certa sezione del video (ad esempio l'esecuzione di un bitrate più basso per i crediti finale). Ci sono due modi per controllare il bitrate di una zona: utilizzando un fattore di "Bitrate" è possibile modificare il bitrate *rispetto* alla decisione di encoder e utilizzando una "quantizzazione" è possibile sovrascrivere la decisione encoder con un valore *costante* di quantizzazione.

## Uscita

### IMPOST. USCITA

- **IDC Livello:**
  - Per impostazione predefinita x264 rileverà il **livello** del flusso risultante H.264 in base alle impostazioni encoder che avete scelto (e basati sulle proprietà del video). *Questa* opzione può essere usata per sovrascrivere x264 decisione. Essere consapevoli del fatto che x264 **non** applicare il livello selezionato per voi! Basta specificare quale livello sarà segnalato nell'impostazione del vostro flusso H.264. Ma questo **non** significa che il vostro flusso effettivamente conforme a quel livello! Quindi si può facilmente produrre un flusso *non valida* specificando un livello improprio. Quindi è *altamente* raccomandato di mantenere il *livello IDC* impostazione su **Auto** e lasciare x264 di rilevare il corretto livello. Se si desidera che il flusso H.264 per conformarsi ad uno specifico livello di H.264, allora è *necessario* scegliere le impostazioni del codificatore di conseguenza. Inoltre è necessario assicurarsi che la risoluzione del tuo video e framerate non superino i limiti del livello. In breve: non modificare *questa* opzione, a meno che non hai una buona ragione per farlo!
- **Sequenza di parametri Set Identifier:**
  - [TO-DO] *Se sai quali sono le informazioni a mettere qui, non esitate a contattarci!*
- **Applicare Ripetibilità:**
  - [TO-DO] *Se sai quali sono le informazioni a mettere qui, non esitate a contattarci!*
- **Utilizzare Access Unit Delimitatori:**
  - [TO-DO] *Se sai quali sono le informazioni a mettere qui, non esitate a contattarci!*

### Video Usability Informazioni

- **Commento:** Queste impostazioni sono solo suggerimenti per l'apparecchio di riproduzione. Usarli a proprio rischio e pericolo!
- **Overscan**
- **Formato video**
- **Color Primarie**
- **Caratteristiche di trasferimento**
- **Color Matrix**
- **Chroma Sample Location**
- **Full Range Campioni**

## Disponibile x264 opzioni in Avidemux

- Tutte le partizioni 2x2
- **Psy-RD e Psy-opzioni traliccio** - Psy-RD verrà attivata comunque (con forza di default)



- PSNR ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/PSNR&prev=\\_t&rurl=translate.google.it&usg=ALkJrhAhYLCs89MudenH04KIMZyB1mVEw](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/PSNR&prev=_t&rurl=translate.google.it&usg=ALkJrhAhYLCs89MudenH04KIMZyB1mVEw)) e SSIM ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/SSIM&prev=\\_t&rurl=translate.google.it&usg=ALkJrhhybtQoIlod38fiFXG6drp6CglBdQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/SSIM&prev=_t&rurl=translate.google.it&usg=ALkJrhhybtQoIlod38fiFXG6drp6CglBdQ)) calcoli

## Obsoleti Opzioni

- **Pre-scenecut:** Dal r1117 x264 userà *sempre* pre-scenecut, scenecut poiché è generalmente migliore rispetto scenecut regolare in termini di precisione e regolare non ha funzionato in anyways modalità threaded.
- **B-RDO:** RD decisione del modo in base per il B-Frames. Questa opzione è stata rimossa in r996. E 'ora abilitato implicitamente al *Sub ME 7* o superiore.
- **Bidirezionale ME:** congiunta ottimizzare sia i vettori di movimento in B-Frames. Questa opzione è stata rimossa in r996. E 'ora abilitato implicitamente al *Sub ME 5* o superiore.
- **AQ Sensibilità:** Questa opzione *non è mai* esistito in x264 ufficiali. E 'stato utilizzato solo in experimental patch di *quantizzazione adattiva*. Attuale AQ non usarlo.

## Profili H.264/AVC e Livelli

Le specifiche H.264/AVC definire una serie di **profili** diversi. Ogni profilo, che specifica *le caratteristiche* di H.264 è permesso (o non consentito). Se si desidera che il flusso video H.264 per essere conforme ad un certo profilo, quindi si può solo le funzioni abilitate consentito in *questo* profilo. I profili sono necessari per assicurarsi che il file video viene riprodotto bene su un decoder certa. Per esempio, un "Main" video profilo compatibile svolgerà ammenda 100% su *ogni* "Principale" decoder profilo in grado / lettore. Quando si lavora con l'encoder **x264**, ci sono fondamentalmente due profili si deve prendere cura di: "Main" e il profilo di "High" profilo. X264, tuttavia, *manca* la funzione Error Resilience dal "Baseline Profile", così come il supporto interlacciamento da "Extended profilo". Se si desidera riprodurre il tuo video su lettori *software*, quindi non c'è bisogno di preoccuparsi di profili più di tanto. Il decoder H.264 da "libavcodec", che viene utilizzato in MPlayer, VLC Player, ffdshow e molti altri, supporta tutte x264 'caratteristiche, tra cui la "High" e "Predictive Lossless" Funzioni del profilo. Lo stesso per i decoder di proprietà, come CoreAVC. Tuttavia, se si prendono di mira un giocatore di *hardware*, quindi profili sono molto importanti, come i giocatori di hardware sono molto *restrittive* su ciò che il profilo che essi supportano.

In aggiunta ai *profili*, le specifiche H.264/AVC anche definire una serie di **livelli**. Mentre *i profili* che definiscono le caratteristiche di compressione del formato H.264 possono (o non può) essere utilizzati, *i livelli* di porre ulteriori restrizioni su altre proprietà del video. Queste restrizioni includono la massima risoluzione, il bitrate massimo, il framerate massimo (per una data risoluzione) e il numero massimo di fotogrammi di riferimento (anche se indirettamente limitato MaxDPB). In ordine di riproduzione video H.264 su un giocatore di *hardware* specifico, che il giocatore non deve solo supporto per il *profilo* del tuo video, ma anche il *livello* del vostro video (o una più alta). Ancora una volta i giocatori *del software* in genere non hanno tali restrizioni, finché CPU è abbastanza potente.

**Nota:** La notazione comune per profili e dei livelli è "Profile @ Level", per esempio *High@4.1*. Inoltre, non vi è *alcun* modo per codificare direttamente il video ad un livello specifico e / o nel profilo. Se volete che il vostro video per conformarsi ad un certo profilo / livello, è necessario scegliere le impostazioni del codificatore di conseguenza. Preset possono essere utili per trovare le impostazioni corrette. Comunque, può ancora essere necessario ridimensionare il tuo video e / o modificare il framerate.

## Elenco di tutti i profili H.264/AVC

	Basale	Extended	Principale	Alto	Alto 10	Alta 4:2:2	Alta 4:4:4 Predictive
I e P Fette	SI	SI	SI	SI	SI	SI	SI
B Slices	NO	SI	SI	SI	SI	SI	SI
SI e SP Fette	NO	SI	NO	NO	NO	NO	NO
Riferimento a più Frames	SI	SI	SI	SI	SI	SI	SI

In-Loop filtro di deblock	SI	SI	SI	SI	SI	SI	SI
CAVLC Entropy Coding	SI	SI	SI	SI	SI	SI	SI
CABAC Entropy Coding	NO	NO	SI	SI	SI	SI	SI
Flessibile Macroblock Ordinazione (FMO)	SI	SI	NO	NO	NO	NO	NO
Arbitraria Slice Ordinazione (ASO)	SI	SI	NO	NO	NO	NO	NO
Redundant Slices (RS)	SI	SI	NO	NO	NO	NO	NO
Partizionamento dei dati	NO	SI	NO	NO	NO	NO	NO
Interlacciato Coding (PicAFF, MBAFF)	NO	SI	SI	SI	SI	SI	SI
4:2:0 Chroma Format	SI	SI	SI	SI	SI	SI	SI
Formato video in bianco e nero (4:0:0)	NO	NO	NO	SI	SI	SI	SI
4:2:2 Chroma Format	NO	NO	NO	NO	NO	SI	SI
4:4:4 Chroma Format	NO	NO	NO	NO	NO	NO	SI
8 Bit Sample Depth	SI	SI	SI	SI	SI	SI	SI
9 e 10 Bit Sample Depth	NO	NO	NO	NO	SI	SI	SI
11-14 Bit Sample Depth	NO	NO	NO	NO	NO	NO	SI
Vs 4x4 Trasformazione 8x8 Adattività	NO	NO	NO	SI	SI	SI	SI
Quantizzazione Scaling Matrici	NO	NO	NO	SI	SI	SI	SI
Separa Cb e Cr controllo QP	NO	NO	NO	SI	SI	SI	SI
Distinti colori Plane Coding	NO	NO	NO	NO	NO	NO	SI
Predictive Lossless Coding	NO	NO	NO	NO	NO	NO	SI
	Basale	Extended	Principale	Alto	Alto 10	Alta 4:2:2	Alta 4:4:4 Predictive

Da Wikipedia, l'enciclopedia libera

## Elenco di tutti i livelli H.264/AVC

Livello di numero	Macroblocchi Max al secondo	Dimensione del frame Max (macroblocchi)	Max bit rate video (VCL) per Baseline, Extended e Main Profiles	Max bit rate video (VCL) per High Profile	Max bit rate video (VCL) per 10 High Profile	Max bit rate video (VCL) per High 4:2:2 e High 4:4:4 Predictive Profili	Esempi per l'alta risoluzione @ frame rate (max memorizzati i fotogrammi) in Level
1	1485	99	64 kbit/s	80 kbit/s	192 kbit/s	256 kbit/s	128x96@30.9 (8) 176x144@15.0 (4)
1b	1485	99	128 kbit / s	160 kbit/s	384 kbit/s	512 kbit/s	128x96@30.9 (8) 176x144@15.0 (4)
1,1	3000	396	192 kbit/s	240 kbit/s	576 kbit/s	768 kbit/s	176x144@30.3 (9) 320x240@10.0 (3) 352x288@7.5 (2)
1,2	6000	396	384 kbit/s	480 kbit/s	1152 kbit/s	1536 kbit/s	320x240@20.0 (7) 352x288@15.2 (6)
1,3	11880	396	768 kbit/s	960 kbit/s	2304 kbit/s	3072 kbit/s	320x240@36.0 (7) 352x288@30.0 (6)
2	11880	396	2 Mbit/s	2.5 Mbit/s	6 Mbit/s	8 Mbit/s	320x240@36.0 (7) 352x288@30.0 (6)
2,1	19800	792	4 Mbit/s	5 Mbit/s	12 Mbit/s	16 Mbit/s	352x480@30.0 (7) 352x576@25.0 (6)
2,2	20250	1620	4 Mbit/s	5 Mbit/s	12 Mbit/s	16 Mbit/s	352x480@30.7(10) 352x576@25.6 (7) 720x480@15.0 (6) 720x576@12.5 (5)
3	40500	1620	10 Mbit/s	12.5 Mbit/s	30 Mbit/s	40 Mbit/s	352x480@61.4 (12) 352x576@51.1 (10) 720x480@30.0 (6) 720x576@25.0 (5)

<b>3,1</b>	108000	3600	14 Mbit/s	14 Mbit/s	42 Mbit/s	56 Mbit/s	720x480@80.0 (13) 720x576@66.7 (11) 1280x720@30.0 (5)
<b>3,2</b>	216000	5120	20 Mbit / s	25 Mbit / s	60 Mbit/s	80 Mbit/s	1280x720@60.0 (5) 1280x1024@42.2 (4)
<b>4</b>	245760	8192	20 Mbit / s	25 Mbit / s	60 Mbit/s	80 Mbit/s	1280x720@68.3 (9) 1920x1080@30.1 (4) 2048x1024@30.0 (4)
<b>4,1</b>	245760	8192	50 Mbit/s	62.5 Mbit/s	150 Mbit/s	200 Mbit/s	1280x720@68.3 (9) 1920x1080@30.1 (4) 2048x1024@30.0 (4)
<b>4,2</b>	522240	8704	50 Mbit/s	62.5 Mbit/s	150 Mbit/s	200 Mbit/s	1920x1080@64.0 (4) 2048x1080@60.0 (4)
<b>5</b>	589824	22080	135 Mbit/s	168.75 Mbit/s	405 Mbit/s	540 Mbit/s	1920x1080@72.3 (13) 2048x1024@72.0 (13) 2048x1080@67.8 (12) 2560x1920@30.7 (5) 3680x1536@26.7 (5)
<b>5,1</b>	983040	36864	240 Mbit/s	300 Mbit/s	720 Mbit/s	960 Mbit/s	1920x1080@120.5 (16) 4096x2048@30.0 (5) 4096x2304@26.7 (5)
<b>Level number</b>	<b>Max macroblocks per second</b>	<b>Max frame size (macroblocks)</b>	<b>Max video bit rate (VCL) for Baseline, Extended and Main Profiles</b>	<b>Max video bit rate (VCL) for High Profile</b>	<b>Max video bit rate (VCL) for High 10 Profile</b>	<b>Max video bit rate (VCL) for High 4:2:2 and High 4:4:4 Predictive Profiles</b>	<b>Esempi per l'alta risoluzione @ frame rate (max stored frames) in Level</b>

Da Wikipedia, l'enciclopedia libera

For more detailed information, please refer to "Annex A" in the official ITU-T H.264 specifications!

## GPU sostegno

Since GPGPU ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://de.wikipedia.org/wiki/GPGPU&prev=\\_t&rurl=translate.google.it&usg=ALkJrhgI59S-AYvhKar6hNZgiYW-inw-xg](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://de.wikipedia.org/wiki/GPGPU&prev=_t&rurl=translate.google.it&usg=ALkJrhgI59S-AYvhKar6hNZgiYW-inw-xg)) has become a hot topic, people began asking for GPU support in Avidemux. These people need to understand that Avidemux cannot offer GPU support for H.264 encoding, until GPU support is implemented in the *x264* library. There is a project scheduled to add CUDA ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://de.wikipedia.org/wiki/CUDA&prev=\\_t&rurl=translate.google.it&usg=ALkJrhiSDPoXL9N5tZGi0069FMjAYG-oqg](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://de.wikipedia.org/wiki/CUDA&prev=_t&rurl=translate.google.it&usg=ALkJrhiSDPoXL9N5tZGi0069FMjAYG-oqg)) support to *x264* (see [5] ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://wiki.videolan.org/SoC\\_x264\\_2009&prev=\\_t&rurl=translate.google.it&usg=ALkJrhgS8SI78C0L4v6aKGzfru7H9wLDbw#GPU\\_Motion\\_Estimation](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://wiki.videolan.org/SoC_x264_2009&prev=_t&rurl=translate.google.it&usg=ALkJrhgS8SI78C0L4v6aKGzfru7H9wLDbw#GPU_Motion_Estimation))), but there are no results yet (May 2009). We know that there are *commercial* H.264 encoders with GPU support available already. But if you look at these encoders closely, you will notice that their speed-up claims are marketing blabber. These encoders may be fast, but their quality isn't anywhere near *x264*'s quality! Also note that marketing people tend to compare their encoders to the completely unoptimized *H.264 Reference Encoder*. *x264* is faster than the reference encoder by several orders of magnitude, which renders these speed comparisons meaningless. *x264* can run extremely fast on a CPU and scales up to at least 16 cores. So don't believe everything that marketing people claim!

## IDR-frames

IDR frames are: An IDR frame is what has been traditionally known as an I frame. An IDR frame, just like an I frame in

MPEG-1/2 and MPEG-4 ASP, starts with a clean slate, and all subsequent frames will make reference to the IDR frame and subsequent frames. Non IDR I frames should be rare, but since they cannot be ruled out, enforcing a minimal IDR interval can help improve compression in some high motion scenes. In H.264/AVC you can also have I frames inside a GOP, which are not seekable, since the long time references introduced in H.264/AVC could result in a P frame after the I frame to reference a P frame before the I frame.

Max IDR-keyframe interval indicates the maximum distance between two IDR frames. Similarly, Min IDR-keyframe interval indicates the minimum distance between two IDR frames.

## List of References

- Official ITU-T H.264 Specifications ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.neuron2.net/library/avc/T-REC-H%255B1%255D.264-200711-I!!PDF-E.pdf&prev=\\_t&rurl=translate.google.it&usg=ALkJrhMaj\\_dSIAP0yvXp6MJ02V7kZhJKw](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.neuron2.net/library/avc/T-REC-H%255B1%255D.264-200711-I!!PDF-E.pdf&prev=_t&rurl=translate.google.it&usg=ALkJrhMaj_dSIAP0yvXp6MJ02V7kZhJKw)) - provided by Neuron2
- x264 - A high performance H.264/AVC encoder ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://akuvian.org/src/x264/overview\\_x264\\_v8\\_5.pdf&prev=\\_t&rurl=translate.google.it&usg=ALkJrhi0kexnfD7\\_A7bWrvfpvteAmrLLtw](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://akuvian.org/src/x264/overview_x264_v8_5.pdf&prev=_t&rurl=translate.google.it&usg=ALkJrhi0kexnfD7_A7bWrvfpvteAmrLLtw)) - by Loren Merritt and Rahul Vanam
- H.264/AVC Thread on Doom9's Forum ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://forum.doom9.org/forumdisplay.php%3Fp%3D77&prev=\\_t&rurl=translate.google.it&usg=ALkJrhjQWUtnPc5heWJtfueUbwfP7m7BKg](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://forum.doom9.org/forumdisplay.php%3Fp%3D77&prev=_t&rurl=translate.google.it&usg=ALkJrhjQWUtnPc5heWJtfueUbwfP7m7BKg)) - especially posts by *akupenguin*, *Dark Shikari* and *\*.mp4 guy*
- A qualitative overview of x264's ratecontrol methods ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://git.videolan.org/gitweb.cgi%3Fp%3Dx264.git%3Ba%3Dblob%3Bf%3Ddoc/ratecontrol.txt&prev=\\_t&rurl=translate.google.it&usg=ALkJrhjSE-brQ9D6QRrNgJPVNAT8BYUkPg](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://git.videolan.org/gitweb.cgi%3Fp%3Dx264.git%3Ba%3Dblob%3Bf%3Ddoc/ratecontrol.txt&prev=_t&rurl=translate.google.it&usg=ALkJrhjSE-brQ9D6QRrNgJPVNAT8BYUkPg)) - by Loren Merritt
- The x264 multi-threading threading method ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://git.videolan.org/gitweb.cgi%3Fp%3Dx264.git%3Ba%3Dblob%3Bf%3Ddoc/threads.txt&prev=\\_t&rurl=translate.google.it&usg=ALkJrhgUe7ZvGtybxCZ839490Q-IL1VKZQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://git.videolan.org/gitweb.cgi%3Fp%3Dx264.git%3Ba%3Dblob%3Bf%3Ddoc/threads.txt&prev=_t&rurl=translate.google.it&usg=ALkJrhgUe7ZvGtybxCZ839490Q-IL1VKZQ)) - by Loren Merritt
- x264 ffmpeg mapping and options guide ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://sites.google.com/site/linuxencoding/x264-ffmpeg-mapping&prev=\\_t&rurl=translate.google.it&usg=ALkJrhxfPC0IxNxiPHL0bKM47of22H3EQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://sites.google.com/site/linuxencoding/x264-ffmpeg-mapping&prev=_t&rurl=translate.google.it&usg=ALkJrhxfPC0IxNxiPHL0bKM47of22H3EQ))
- Wikipedia, the free encyclopedia ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/X264&prev=\\_t&rurl=translate.google.it&usg=ALkJrhgfEtbDqYjK7PKaQ6FHgdFSPZDFoQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/X264&prev=_t&rurl=translate.google.it&usg=ALkJrhgfEtbDqYjK7PKaQ6FHgdFSPZDFoQ)) - article about the "x264" encoder ( German version ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://de.wikipedia.org/wiki/X264&prev=\\_t&rurl=translate.google.it&usg=ALkJrhqhQIBbNb\\_a4Nl8-QtZPKvFXrQbBA](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://de.wikipedia.org/wiki/X264&prev=_t&rurl=translate.google.it&usg=ALkJrhqhQIBbNb_a4Nl8-QtZPKvFXrQbBA)) )
- DeathTheSheep's AVC VfW Guide ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://gabextreme.googlepages.com/x264vfwunited&prev=\\_t&rurl=translate.google.it&usg=ALkJrhAczQx\\_bldVde6iA8SYFcdVL2U2A](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://gabextreme.googlepages.com/x264vfwunited&prev=_t&rurl=translate.google.it&usg=ALkJrhAczQx_bldVde6iA8SYFcdVL2U2A))
- MeWiki - x264 settings ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://mewiki.project357.com/wiki/X264\\_Settings&prev=\\_t&rurl=translate.google.it&usg=ALkJrhj\\_AVlyzf2AA3-t4BtlSo\\_z366mMA](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://mewiki.project357.com/wiki/X264_Settings&prev=_t&rurl=translate.google.it&usg=ALkJrhj_AVlyzf2AA3-t4BtlSo_z366mMA))
- Selur's man x264 (Hilfe zum x264 CLI) ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.flaskmpeg.info/board/thread.php%3Fthreadid%3D5571&prev=\\_t&rurl=translate.google.it&usg=ALkJrhgSB2UMNsm4\\_orMXbjYVJizXUw0fg](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.flaskmpeg.info/board/thread.php%3Fthreadid%3D5571&prev=_t&rurl=translate.google.it&usg=ALkJrhgSB2UMNsm4_orMXbjYVJizXUw0fg)) - German documentation
- Digital Digest - x264 Options Explained ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.digital-digest.com/articles/x264\\_options\\_page1.html&prev=\\_t&rurl=translate.google.it&usg=ALkJrhOyUGTQLLOmwuVcWvu7Kq85HC5Zw](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.digital-digest.com/articles/x264_options_page1.html&prev=_t&rurl=translate.google.it&usg=ALkJrhOyUGTQLLOmwuVcWvu7Kq85HC5Zw))

## Vedi anche

- Compiling H264

---

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-  
Retrieved from " <http://www.avidemux.org/admWiki/index.php?title=H264> "

---

- This page was last modified 20:45, 11 May 2009.
- This page has been accessed 100,258 times.
- Tutela della privacy
- A proposito di Avidemux2
- Disclaimer

# Preset

## Da Avidemux2

- = Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorial = Glossario = -

---

Avidemux permette di memorizzare audio corrente / encoder video e le impostazioni dei filtri, che possono essere utilizzati come ulteriore "profili" per i formati di determinate categorie (ad esempio per la riproduzione su dispositivi portatili), o semplicemente come le opzioni di encoder preferito (in modo che non si è necessario impostare manualmente più e più volte).

## Come conservare le opzioni encoder

Per memorizzare l'encoder preset, è necessario creare uno script personalizzato. Al contrario di file di progetto, che contengono la completa modifica dell'elenco, script personalizzati può contenere solo le impostazioni particolare si è interessato a. Il modo più semplice di creare uno script personalizzato con le opzioni encoder preferito è l'impostazione delle opzioni nell'interfaccia utente grafica, il risparmio il progetto e quindi la modifica del file di progetto (che è uno script ECMAScript). Rimuovere tutto ciò che non è necessario (per esempio il file corrente caricato - si vuole essere in grado di applicarla a qualsiasi tipo di file), lasciare solo le impostazioni del codificatore nello script. Salvarlo nella directory Avidemux con script personalizzati:

- **Linux / BSD:** ~ / .avidemux / personalizzato ( ". Avidemux" è una directory nascosta della propria home directory, in modo da verificare la "Mostra file nascosti" nel vostro file manager)
- **Windows 2000 e XP:** \ Documents and Settings \ \$ USER \$ \ Local Settings \ Application Data \ avidemux \ custom
- **Windows Vista:** \ Users \ \$ USER \$ \ AppData \ Roaming \ avidemux \ custom

Nomi di file personalizzati script deve terminare con l'. Estensione js. Poi potete caricare lo script dal menu personalizzati in Avidemux.

## Esempio - salvare le opzioni di x264

Diciamo che si desidera memorizzare il vostro favorito x264 opzioni encoder come un preset:

1. Aprire un file video.
2. Selezionare x264 come l'encoder video e configurarlo.
3. Salvare uno script: File-> Save Project.
4. Aprire lo script in un editor di testo (ad esempio Blocco note di Windows, gedit in GNOME o KDE in Kate). Si vedrà qualcosa di simile a questa:

```

/ / AD <- necessarie per identificare / /
/ / - create automaticamente --
/ / - Progetto: / home/user/x264.js

var app = new Avidemux ();

/ / ** Video **
/ / 01 video source
( "app.load / home / user / video.mpeg");
/ / 01 segmenti
app.clearSegments ();
app.addSegment (0,0,1863);
app.markerA = 0;
app.markerB = 1.862;

/ / ** ** Postproc
app.video.setPostProc (0,0,0);

app.video.setFps1000 (28.000);

/ / ** ** Filtri

/ / ** Conf Video Codec **
app.video.codec ( "X264", "AQ = 26", "188 00 00 00 00 00 00 00 00 00 00 00 00 00 00 28 00 00 00 00 1e 00 00 00 3c 00
00 0a 00 00 00 33 00 00 00 04 00 00 00 03 00 00 00 28 00 00 00 19 00 00 00 32 00 00 00 00 00 00 00 01 00
00 00 01 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 01 00 00 00 04 00 00 00 10 00 00 00 01 00 00 00 01 00 00 00 01 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 5a 00 00 00 ");

/ / ** Audio **
app.audio.reset ();
app.audio.codec ( "AAC", 128,4, "80 00 00 00");
app.audio.normalizeMode = 0;
app.audio.normalizeValue = 0;
app.audio.delay = 0;
app.audio.mixer ( "Nessuno");
app.setContainer ( "MP4");
setSuccess (1);
/ / app.Exit ();

/ / End of script

```

Quello che vi serve è i 2 linee:

```
var app = new Avidemux ();
```

creare un'istanza Avidemux

```

app.video.codec ( "X264", "AQ = 26", "188 00 00 00 00 00 00 00 00 00 00 00 00 00 00 28 00 00 00 00 1e 00 00 00 3c 00
00 0a 00 00 00 33 00 00 00 04 00 00 00 03 00 00 00 28 00 00 00 19 00 00 00 32 00 00 00 00 00 00 00 01 00
00 00 01 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 01 00 00 00 04 00 00 00 10 00 00 00 01 00 00 00 01 00 00 00 01 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 5a 00 00 00 ");

```

impostare l'encoder per x264 e memorizzare tutte le sue opzioni (come potete vedere, salvare lo script dalla GUI è molto più facile che dover scrivere manualmente tutte le impostazioni)

Così il vostro script personalizzato sarà simile a questa:

```

var app = new Avidemux ();
app.video.codec ( "X264", "AQ = 26", "188 00 00 00 00 00 00 00 00 00 00 00 00 00 00 28 00 00 00 00 1e 00 00 00 3c 00
00 0a 00 00 00 33 00 00 00 04 00 00 00 03 00 00 00 28 00 00 00 19 00 00 00 32 00 00 00 00 00 00 00 01 00
00 00 01 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 01 00 00 00 04 00 00 00 10 00 00 00 01 00 00 00 01 00 00 00 01 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 5a 00 00 00 ");

```

Salvare il file come x264.js nella directory personalizzato Avidemux (vedere la sezione precedente). Quando si avvia Avidemux, x264.js dovrebbe apparire nel menu personalizzato. Ora si dovrebbe sempre essere in grado di ripristinare le opzioni preferite x264 aprendo Custom-> x264.js.

---

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-  
Estratto da "<http://www.avidemux.org/admWiki/index.php?title=Presets>"

---

- Ultima modifica 20:26, 24 luglio 2008.
- Questa pagina è stata letta 20.853 volte.
- Tutela della privacy
- A proposito di Avidemux2
- Disclaimer



# Project X

## Da Avidemux2

- = Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorial = Glossario = -

---

### Sommario

- 1 Generale
- 2 Installazione di Java
  - 2,1 Ubuntu / Kubuntu / Xubuntu
- 3 Download Project X
- 4 Estrazione Project X
- 5 esegue Project X
  - 5,1 Linux / BSD
  - 5.2 Windows
- 6 Problemi
  - 6,1 Ubuntu
- 7 Sourceforge

## Generale

Project X cerca del suo meglio per gestire e riparare i tipi di flusso di molti e mostra che cosa è andato storto alla reception. Questo è molto utile per la riparazione di file MPEG, in particolare con problemi di sincronizzazione audio e video.

## Installazione di Java

Java deve essere installata per eseguire il file binario eseguibile Project X. Come questo è fatto per il vostro computer può variare. Ecco alcune linee guida per specifici sistemi operativi su come installare Java.

### Ubuntu / Kubuntu / Xubuntu

Consentire l'universo nel tuo elenco di fonti per APT. Da una console di root o tramite l'uso `sudo nano / vim / gedit` o qualsiasi altro editor di testo che si desidera aprire: `/ etc / apt / sources.list`. Dovresti avere l'universo "" attivato su una delle vostre linee repository.

Dopo aver assicurato l'universo "" repository è attivato nel tuo apt-file fonti, da una console di root o tramite `sudo` eseguire questo comando:

```
sudo apt-get update
```

Prossimo verrà installato Java stesso. Eseguire questa riga di comando dalla console di root o con `su` o `sudo`:

```
sudo apt-get install sun-java6-bin
```

## Download Project X

Project X è scritto in Java e può essere eseguito su qualsiasi piattaforma o quasi.

È possibile scaricare binari eseguire file jar da entrambi:

- Doom9.org ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.doom9.org/&prev=\\_t&rurl=translate.google.it&usg=ALkJrhj5z7OrunSZAQ8SySrmHkLCbfGSiQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.doom9.org/&prev=_t&rurl=translate.google.it&usg=ALkJrhj5z7OrunSZAQ8SySrmHkLCbfGSiQ)) nella loro sezione **Downloads**

Oppure si può afferrare la directory dei file da un mirror creato per gli utenti Avidemux.

- ProjectX, ([http://students.washington.edu/cdobrich/ProjectX\\_0.90.4.zip](http://students.washington.edu/cdobrich/ProjectX_0.90.4.zip)) questo è lo stesso del file di Doom9. La sua unica leechers qui per scaricare e salvare la larghezza di banda Doom9.org.

## Estrazione di Project X

Dopo aver scaricato il file zip Project X binario, è necessario per estrarlo. Come si esegue questa operazione dipenderà dal vostro sistema operativo. Nella maggior parte dei sistemi Linux / BSD può essere fatto con il comando *unzip*. Come in questo esempio:

```
unzip ProjectX_0.90.4.zip
```

## Esegue Project X

Dopo aver scaricato Project X, è possibile semplice esecuzione. Esattamente come si esegue il programma dipenderà dal sistema operativo ..

### Linux / BSD

In Linux e BSD sistema di variante, che hanno contengono un completo e valido console ambiente, è possibile eseguire il progetto X utilizzando la riga di comando. Da una console di comando, assicurarsi che la corrente prompt della console è in realtà di lavoro della stessa directory del X Project. File jar o che il file. Vaso è nel tuo percorso di sistema, quindi eseguire questo comando:

```
java-jar ProjectX.jar
```

Project X dovrebbe ora aperta.

### Windows

Se si utilizza Windows, di solito è sufficiente *fare doppio clic* sul **file. Jar**. In alternativa è possibile selezionarlo e quindi *fare clic destro del mouse* per aprire il menu e selezionare **Apri** per avviare il file. Per impostazione predefinita Java, se si è installato correttamente, dovrebbe eseguire l'applicazione per voi.

## Problemi

### Ubuntu

Se si sta utilizzando Ubuntu e si era di errore quando si è tentato di eseguire il progetto X dalla console, potrebbe essere necessario eseguire i seguenti comandi per configurare Java (sia come **root** o usare **sudo**:

```
sudo update-alternatives - install / usr / bin / java java / usr / local / java / bin / java 3
sudo update-alternatives - config java
```

Se ciò non risolve il vostro errori durante l'esecuzione, si prega di fare riferimento alla pagina di installazione di Java qui.  
(<https://help.ubuntu.com/community/Java>)

## SourceForge

Project X è un progetto su SourceForge. Notizie e aggiornamenti al programma può essere trovato qui.  
([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://sourceforge.net/projects/project-x&prev=\\_t&rurl=translate.google.it&usg=ALkJrhgf\\_FHKoxZNvBrBKJQve21uUsMx4g](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://sourceforge.net/projects/project-x&prev=_t&rurl=translate.google.it&usg=ALkJrhgf_FHKoxZNvBrBKJQve21uUsMx4g)) Questo sito non contiene un pre-compilato file eseguibile e si può richiedere di creare, se si desidera utilizzare la loro versione. Di solito i file binari sono regolari aggiornamento per data e con le versioni attuali di codice molto recenti.

---

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-  
Estratto da "[http://www.avidemux.org/admWiki/index.php?title=Project\\_X](http://www.avidemux.org/admWiki/index.php?title=Project_X)"

---

- Ultima modifica 15:59, 8 luglio 2008.
- Questa pagina è stata letta 41.781 volte.
- Tutela della privacy
- A proposito di Avidemux2
- Disclaimer

# Salvare solo l'audio

## Da Avidemux2

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-

---

Questa è una guida rapida per se si vuole salvare solo l'audio da un file. Questo è molto semplice, ma molto utili.

1. Caricare il file video.
2. **Optional:** Impostare la *A* & *B* posizioni se si desidera solo una parte della traccia audio.
3. Dal pannello laterale, la sezione audio, scegliere il codec audio che si desidera utilizzare - ad esempio MP3, Ogg Vorbis, WAV, AAC, Copia o se si vuole estrarre un pezzo della colonna sonora originale.
4. Configurare le opzioni per il vostro codec audio utilizzando il pulsante *Configura*.
5. Vai al menu: *Audio* -> *Salva*.
6. Salvare il file audio.

Una nota di avvertimento: utilizza lo stesso metodo si salva la traccia audio grezzi. Se è AAC o Ogg Vorbis, vi ritroverete con un file ingiocabile come hanno bisogno di un contenitore per essere giocabile.

---

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-

Estratto da "[http://www.avidemux.org/admWiki/index.php?title=Save\\_only\\_audio](http://www.avidemux.org/admWiki/index.php?title=Save_only_audio)"

---

- Ultima modifica 15:52, 4 giugno 2006.
- Questa pagina è stata letta 28.875 volte.
- Tutela della privacy
- A proposito di Avidemux2
- Disclaimer

# Salva solo il video

## Da Avidemux2

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-

---

## Salvare il solo il video da un file

Questa è una guida rapida per se si vuole salvare solo il video da un file. Questo è molto semplice, ma molto utili.

- Caricare il file video
  - **Optional:** Impostare la **A & B** posizioni se si desidera solo una parte del video
  - Aprire il menu: *Audio -> Main Track*
    - Selezionare: Nessuno
  - Salvare il file video
- 

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-

Estratto da "[http://www.avidemux.org/admWiki/index.php?title=Save\\_only\\_video](http://www.avidemux.org/admWiki/index.php?title=Save_only_video)"

---

- Ultima modifica 00:25, 14 giugno 2006.
- Questa pagina è stata letta 24.626 volte.
- Tutela della privacy
- A proposito di Avidemux2
- Disclaimer

# Scripting tutorial

## Da Avidemux2

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-

---

Questa pagina cerca di spiegare come funziona scripting all'interno Avidemux.

Il motore di scripting utilizzato da Avidemux è SpiderMonkey, ed è un motore di ECMAScript / JavaScript. Si cammina attraverso un semplice script che rimuove il bitstream male imballati da una directory intera.

### Sommario

- 1 Headers
- 2 directories Impostazione
- 3 ciclo principale
- 4 Trasformazione funzione
- 5 script completo
- 6 Vedi anche

### Intestazioni

Il seguente sarà probabilmente le due prime linee di qualsiasi script Avidemux:

```
// AD <--  
var app = new Avidemux ();
```

Il primo è utilizzato da Avidemux per identificare il file, e devono essere presenti (almeno per i quattro primi caratteri: *// AD*).

La seconda linea è dove creare un'istanza della classe Avidemux. Quasi tutte le funzioni Avidemux saranno accessibili attraverso di esso, nel nostro caso app.

### La creazione di directory

Ci dichiariamo un sacco di variabili e di impostare la dirs in entrata e in uscita

```
// AD <--  
/*  
Semplice script che esegue la scansione della directory orgDir  
e decomprimere tutti i file AVI  
Il file risultante viene messo nella directory destdir
```

```
Utilizzando directorySearch nuova API
```

```
* /
var app = new Avidemux ();
var ds = new DirectorySearch ();
var orgDir;
var destDir;
var reg = new RegExp (".\$");
/ *
    Questo è per Unix
    Per il cambiamento di Windows
    sep = "\\ \\";
```

```
* /
var settembre = "/";
var extension;
var target;
```

Se si utilizza direttamente i percorsi nello script, è molto importante non dimenticare di rimuovere l'ultimo /. Per esempio, /foo / bar è ok, / foo / bar / non è.

Chiediamo l'utente a dare input e output dir. Per fare ciò, ci chiediamo per i file e solo mantenere la parte della directory di esso.

```
orgDir = fileReadSelect ();
DESTDIR = fileWriteSelect ();
orgDir = pathOnly (orgDir);
DESTDIR = pathOnly (DESTDIR);
orgDir.replace = orgDir (reg, "");
destDir.replace = destDir (reg, "");
```

Si noti che usiamo la regexp *reg* per rimuovere il carattere / o \.

## Main loop

Ora che le directory sono impostati, siamo in grado di interpretarle. Let's get il file interessanti:

```
if (ds.Init (orgDir))
(
/ / Solo i file processo vogliamo cioè AVI
if (! ds.isNotFile & &! ds.isDirectory & &! ds.isHidden & &! ds.isArchive & &! ds.isSingleDot & &! ds.is)
(
    ds.GetExtension extension = ();
    if (estensione == "avi")
    (
        target = ds.GetFileName ();
        = destDir target + settembre + target;
        ProcessFile (orgDir + settembre + ds.GetFileName (), target);
    )
)
```

Ci permette di mantenere solo i file che non sono directory e hanno un'estensione *avi*. Naturalmente è possibile sostituire l'estensione alle vostre esigenze. Next usiamo l'espressione regolare REG2 per rimuovere il percorso.

Se il file è in arrivo / dump / you\_are\_under / foo.avi, e directory di destinazione è / tmp, si verificherà il seguente:

target = foo.avi

target = / tmp + / + foo.avi = / tmp / foo.avi

Poi si antepone la directory di destinazione e il file processo di chiamata con entrambe le file di origine e di destinazione.

Prendendo l'esempio stesso

```
= source / dump / you_are_under + / + foo.avi = / you_are_under / dump / foo.avi
```

```
target = / tmp + / + foo.avi = / tmp / foo.avi
```

## Funzione di trasformazione

Il nostro ciclo principale è finita. Il divertimento continua nella funzione ProcessFile che farà il lavoro reale. Sarà un processo un file ed è chiamato per ogni file dal nostro ciclo principale.

```
ProcessFile funzione (filename, targetfile)
(
    / / Carica il file
    app.forceUnpack ();
    app.load (filename);
    app.rebuildIndex ();
    app.audio.delay = 0;
    app.audio.mixer ( "Nessuno");
    app.audio.scanVBR ();
    app.setContainer ( "AVI_UNP");
    app.save (targetfile);
    return 1;
)
```

Abbiamo appena chiamata forceunpack per rimuovere il bitstream imballato, e chiamare scanVBR nel caso in cui sia MP3 VBR. Quindi impostare il formato di output per AVI\_UNP e salvare.

Naturalmente si tratta di un semplice esempio. Se avete trasformato un file e volete fare lo stesso trattamento di un gruppo di file: ecco come fare:

1. Modificare il file a proprio piacimento
2. Salvare il progetto come / tmp / prj.js per esempio
3. Copia incolla / tmp / prj.js file in ProcessFile
4. Modifica per usare il file nomefile e targetfile come origine e destinazione

Si può perfettamente applicare filtri audio e video, cambiare codec, ecc ..

## Completa script

L'intero script simile a questa:



```

/ / AD <--
/ *
    Semplice script che esegue la scansione della directory orgDir
    e decomprimere tutti i file AVI
    Il file risultante viene messo nella directory destdir
    Utilizzando directorySearch nuova API
* /
var app = new Avidemux ();
var ds = new DirectorySearch ();
var orgDir;
var destdir;
var reg = new RegExp (".$");
/ *
    Questo è per Unix
    Per il cambiamento di Windows
    sep = "\ \";
    REG2 = new RegExp ( "\ \ .* \ \");
* /
var settembre ="/";
var REG2 = new RegExp ( "\ / .* \ /");
var extension;
var target;
/ /
/ /
/ /
/ / Selezionare i file da directory originali e di destinazione
/ /
    orgDir = fileReadSelect ();
    DESTDIR = fileWriteSelect ();
    orgDir = pathOnly (orgDir);
    DESTDIR = pathOnly (DESTDIR);
/ /
/ / Ultimo lembo \ \ o / /
/ /
    orgDir.replace = orgDir (reg , "");
    destDir.replace = destdir (reg , "");
    print ( "orgDir: <" + orgDir + ">");
    print ( "destdir: <" + destdir + ">");
/ /
/ / Go
/ /
if (ds.Init (orgDir))
(
    while (ds.NextFile ())
    (
        / / Solo processo di file che vogliamo cioè AVI
        if (! ds.isNotFile & &! ds.isDirectory & &! ds.isHidden & &! ds.isArchive & &! ds.isSingleDot & &! d
        (
            ds.GetExtension extension = ();
            if (estensione == "avi")
            (
                target = ds.GetFileName ();
                = destdir target + settembre + target;
                Stampa ("****" + ds.GetFileName ()+"-->" + target);
                ProcessFile (orgDir + settembre + ds.GetFileName (), target);
            )
            / / print ( "File:" + ds.GetFileName () + "è" + ds.GetFileSize () + "bytes, estensione" + estens
        )
    )
    print ( "Abbiamo solo cercato nella directory \" " + ds.GetFileDirectory () + " \ ");
    ds.Close ();
)
altro
    displayError ( "Errore durante l'inizializzazione della ricerca di directory");
displayInf ( "Finito!");
ProcessFile funzione (filename, targetfile)
(
    / / Carica il file
    app.forceUnpack ();
    app.load (filename);
    app.rebuildIndex ();
    app.audio.delay = 0;
    app.audio.mixer ( "Nessuno");
    app.audio.scanVBR ();
    app.setContainer ( "AVI_UNP");
    app.save (targetfile);
    return 1;
)

```

## Vedi anche

- [Scripting](#)

Estratto da "[http://www.avidemux.org/admWiki/index.php?title=Scripting\\_tutorial](http://www.avidemux.org/admWiki/index.php?title=Scripting_tutorial)"

---

- Ultima modifica 10:39, 4 marzo 2007.
- Questa pagina è stata letta 38.417 volte.
- [Tutela della privacy](#)
- [A proposito di Avidemux2](#)
- [Disclaimer](#)

# Scrivi il tuo filtro

## Da Avidemux2

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario =-

---

Questa pagina cerca di spiegare come scrivere il tuo filtro

### Sommario

- 1 Setup & Tools
- 2 Tutto ciò è una classe
- 3 Il costruttore
- 4 printconf
- 5 GetCouple
- 6 Configurazione
- 7 setFrameNoAlloc
- 8 Dichiarare il filtro
  - 8,1 filtro in bundle
  - 8,2 Plugin
- 9 parole finali

## Setup & Tools

Prima di tutto avete bisogno di fonti avidemux (se fai plugin che non hanno nemmeno bisogno di compilare avidemux in sé, ma hai ancora bisogno di fonti)

Per Windows, è necessario MinGW gcc / g++ (versione 4.x) o Cygwin gcc / g++ (versione 4.x). MSVC non funziona come ci saranno alcuni c++ link e MSVC C++ non è compatibile con G++.

## Tutto è una classe

Tutti i filtri all'interno avidemux sono derivati di classe AVDMGenericVideoStream. Solo un paio di metodi devono essere definiti:

## Il costruttore

It Takes 2 argomenti

- Il filtro precedente nella catena, sarà copiato in `_in`
- Coppie \* CONFcouple che contiene la configurazione. Esso è nullo significa che hai di utilizzare il valore predefinito.

Il costruttore deve aggiornare il suo campo `_info` che contiene una descrizione del formato di output. Se non fps cambiare né larghezza / altezza è sufficiente copiare l'uno dal filtro precedente utilizzando `_in-> getInfo ()`

Si dovrebbe avere un campo di classe di nome `_param` che è una struttura che descrive i parametri. Per esempio

```
typedef struct rotateParam
(
    uint32_t angolo;
    rotateParam);
```

Nel costruttore si può usare la comoda `** GET` macro Recupera il parametro da `CONFCouple` ad esempio `GET (angolo);`

## printconf

il metodo successivo è `printconf`. E 'utilizzato per stampare la configurazione nel filtro nella finestra di dialogo.

```
char * yourClassName:: printconf (void)
(
    static char buf [50];
    sprintf ((char *) buf, "Il mio ruotare il filtro, l'angolo:% d", _param-> angle);
    return buf;
)
```

Si noti che il buffer è statico al fine di evitare l'assegnazione / liberando così come essa può essere complicato.

## GetCouple

Questo metodo viene utilizzato per recuperare la configurazione attuale. Il modo più semplice è quello di utilizzare la macro `CSET` come questo

```
yourClassName uint8_t:: getCoupledConf (CONFCouple Coppie **)
(
    ADM_assert (_param);
    * =coppie CONFCouple nuovo (1); // Numero di param nella vostra struttura
    # CSET undef
    # define CSET (x) (* coppie) -> setCouple (# x, (_param-> x))
    CSET (angolo);
    return 1;
)
```

`getCoupledConf` e il costruttore sono una coppia. E 'essenziale che utilizzano lo stesso numero di parametri e gli stessi nomi di parametro. Else avrete afferma popping.

## Configurare

```
uint8_t yourClassName:: configurare (AVDMGenericVideoStream * a)
```

Questo metodo è usato per configurare il filtro. Il parametro è lo stesso per il costruttore. E 'più sicuro per aggiornare `_in` con essa però. Si consiglia (vivamente) di utilizzare `dialogFactory` come lavorerà su entrambe le GTK e Windows ed è molto semplice da usare. Alcune informazioni:

- Restituendo 0 non significa nulla è cambiato
- Tornando 1 significa che hai qualcosa cambia e che la catena dei filtri potrebbe essere necessario ricostruire.

IMPORTANTE: Il cambiamento deve essere fatto immediatamente in configurazione!

## getFrameNoAlloc

Questo è il metodo centrale come il lavoro sarà fatto qui.

```
uint8_t yourClassName:: getFrameNumberNoAlloc (inframe uint32_t,
                                             len uint32_t *,
                                             * Dati ADMImage, uint32_t * flags)
```

len è la dimensione in byte dell'immagine di output. bandiere possono essere ignorati. inframe è la cornice che hai a rendere. dei dati è il luogo in cui si deve rendere il telaio.

Ricordate che i filtri sono una catena, in modo che ciascun filtro può chiedere (qualsiasi fotogramma) al suo predecessore. Nel caso in cui la maggior parte si vuole allocare un buffer temporaneo nel costruttore (che viene spesso chiamato \_uncompressed) e chiedere il filtro precedente per riempire per voi. Ad esempio, il verticalFlip può essere semplificata così:

```
uint8_t yourClassName:: getFrameNumberNoAlloc (inframe uint32_t,
                                             len uint32_t *,
                                             * Dati ADMImage, uint32_t * flags)
{
    ADM_assert (inframe < _info.nb_frames); // Assicurarsi che non andiamo fuori dai limiti
    // Frame di lettura per il precedente
    ADM_assert (_in-> getFrameNumberNoAlloc (inframe, len, _uncompressed, flags)); // Ora _uncompressed con
    // Flip da _uncompressed ai dati, che contiene l'immagine finale
    Return 1; // OK
}
```

Alcune note:

- Tutte le immagini sono in formato YV12
- Se avete bisogno di più immagini per rendere un'immagine out (3:2 pulldown, soppressione del rumore ,...), si deve istanziare un oggetto VideoCache

## Dichiara il filtro

Ci sono 2 modi di dichiarare il filtro: in bundle o plug-in. Puoi fare entrambe le cose allo stesso tempo, nello stesso file.

### Bundle filtro

Si è un po 'più complicato. Dovete:

- Dichiarare un tag per il filtro in ADM\_filter / video\_filters.h
- Registra il tuo filtro in ADM\_filter / filter\_declaration.cpp utilizzando

```
REGISTERX ( "Ruota", "Ruota", "Ruota l'immagine di 90, 180 o 270 gradi." VF_ROTATE, 1, rotate_create, rota
```

Il primo parametro è il nome interno (deve essere univoco), il secondo e 3 i parametri sono quello utilizzato per la visualizzazione a, d breve il quarto è il tag che ha dichiarato in precedenza. DONT riutilizzare un tag esistente. Si avrebbe

grandi problemi, mi creda. I 2 ultimi deve essere la stessa come la funzione descritta di seguito

Torniamo al nostro filtro di file C ++. Prima cosa è necessario dichiarare una const che assomiglia

```
yourfilter_param FILTER_PARAM static = (1, "angolo");
```

Il primo parametro è il numero di parametri nella struttura param seguita dal nome parametri, esattamente come nella struttura. Quindi utilizzare questi due macro:

```
BUILD_CREATE (rotate_create, yourClassName);  
SCRIPT_CREATE (rotate_script, yourClassName, yourfilter_param);
```

Assicurarsi che il primo parametro di questi due macro corrispondere a quello utilizzato in REGISTERX.

That's all:).

## Plugin

E 'più semplice. Hai ancora dichiarare la const che assomiglia a

```
yourfilter_param FILTER_PARAM static = (1, "angolo");
```

Allora

```
extern "C"  
{  
    SCRIPT_CREATE (FILTER_create_fromscript, yourClassName, yourfilter_param);  
    BUILD_CREATE (FILTER_create, yourClassName);  
    char * FILTER_getName (void)  
    {  
        return "DynRotate"; // Nome del tuo filtro. deve essere unico!  
    }  
    char * FILTER_getDesc (void)  
    {  
        return "blah blah"; // Breve descrizione del tuo filtro  
    }  
    uint32_t FILTER_getVersion (void)  
    {  
        return 1; // versione del filtro  
    }  
    uint32_t FILTER_getAPIVersion (void)  
    {  
        return ADM_FILTER_API_VERSION;  
    }  
}
```

È autonomo, non si deve modificare qualsiasi file in ADM\_filter / xxx se è solo un plug-in. Puoi fare entrambe le cose dichiarazione nello stesso file, in modo che possa essere fornito in bundle all'interno avidemux o compilato come un filtro plug-in.

## Final parole

Il modo più semplice per avviare dal ADM\_filter / ADM\_vidDummyFilter.cpp che è un filtro molto semplice, si può costruire con

```
sh builddummy.sh su Unix
```

0

```
builddummy_win32.sh sh su Windows (potrebbe essere necessario modificare il percorso / nomi per il vostro
```

Le principali differenze con AviSynth sono:

- Formato immagine è sempre YV12
- stride = width
- Tu non restituisci il telaio, è dato a voi dal chiamante come argomento

- = Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorials = Glossario = -

Estratto da "[http://www.avidemux.org/admWiki/index.php?title=Writing\\_your\\_own\\_filter](http://www.avidemux.org/admWiki/index.php?title=Writing_your_own_filter)"

---

- Ultima modifica 16:38, 14 aprile 2007.
- Questa pagina è stata letta 13.869 volte.
- Tutela della privacy
- A proposito di Avidemux2
- Disclaimer

# Standalone MPEG-4 giocatori

## Da Avidemux2

-= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorial = Glossario =-

---

### Sommario

- 1 Introduzione
- 2 Il Pranzo VOP hack
- 3 GMC e qpel
- 4 Muxing

## Introduzione

Questo articolo cerca di aiutarvi a rendere il vostro video in riproduzione su MPEG-4 standalone lettori DVD, siano essi DivX Certified ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.divx.com/company/partner/certified\\_program.php&prev=\\_t&rurl=translate.google.it&usg=ALkJrhjWggDs-o72KK\\_koXFGJPzKfDkwEQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.divx.com/company/partner/certified_program.php&prev=_t&rurl=translate.google.it&usg=ALkJrhjWggDs-o72KK_koXFGJPzKfDkwEQ)) o meno.

## Il Pranzo VOP hack

B-frame sono piacevoli, ma induce un cambiamento un ritardo fotogramma perché un B-frame ha bisogno sia il suo precedente e successiva / fotogrammi P da decodificare.

Che non gestire molto bene con VFW (Video for Windows), limitazione che, più o meno assume un fotogramma in / out di un fotogramma.

Per aggirare questo, DivX, Inc. ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.divx.com/company/&prev=\\_t&rurl=translate.google.it&usg=ALkJrhilpJWwIqbBJ1\\_nTiThmfBD\\_p2BaQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://www.divx.com/company/&prev=_t&rurl=translate.google.it&usg=ALkJrhilpJWwIqbBJ1_nTiThmfBD_p2BaQ)) ha introdotto PB fotogrammi in formato AVI. Per spiegarlo, vedere la B-frame articolo. Fondamentalmente si tratta di aggiungere una immagine non codificati (N-VOP) dopo un frame che contiene un P e un frame B. Alcuni (la maggior parte?) Giocatori soffocamento o balbettare su tale in quanto crea artificialmente un frame aggiuntivo. Per rilevare che, hanno anche messo un marcatore specifico nella sezione di dati utente del MPEG-4 stream. Alcuni giocatori di utilizzare queste informazioni per passare alla modalità VOP sacco, ma non è affidabile al 100%. Più il flusso non è al 100% ISO compatibile.

- Regola # 1: Non utilizzare frames al sacco. Con Avidemux non è possibile, ma con qualche altro software è possibile.
- Regola # 2: Se si dispone di alcuni, depack loro.

Al fine di farlo, quando viene richiesto se Avidemux dovrebbe depack frame quali, dire di sì. Avidemux terrà ogni fotogramma e la scansione per più di un VOP al suo interno. Se si riconosce più, si ricrea fotogrammi separati e distrugge le corrispondenti N-frame VOP.

Che si prenderà cura del Pranzo VOP, ma l'indicatore di dati utente rimarrà. Anche rimuoverlo, usare "AVI, UNP. VOP", come il formato di output, invece della solita formato AVI e salvare il video.



Che esegue la scansione e la correzione al volo il marcatore in modo che il flusso viene rilevato come non flusso compatto.

Che dovrebbe risolvere tutti i file DivX codificati sacco, ma non può funzionare per alcuni file XviD codificati.

**Attenzione:** se il file è VBR audio, non dimenticate di fare audio-> Build VBR Time Map dopo il caricamento.

Se si dispone di una quantità enorme di questi file da convertire è possibile utilizzare uno script come:

```
#!/ bin / bash
export SRC = / tmp/fma2
TGT export = / tmp/fma3
foreach.pl perl $ SRC / *. avi "avidemux2 \
- unpack-force - load $ SRC /% f \
- map-audio - save-spacchettato-vop \
$ TGT / fixed_% f - quit "
```

## GMC e qpel

XviD e libavcodec MPEG-4 è possibile utilizzare più completo GMC ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Global\\_Motion\\_Compensation&prev=\\_t&rurl=translate.google.it&usg=ALkJrhDbN45rivvYb0jPHImr2Y4kPYydQ](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Global_Motion_Compensation&prev=_t&rurl=translate.google.it&usg=ALkJrhDbN45rivvYb0jPHImr2Y4kPYydQ)) (Global Motion Compensation) e qpel ([http://translate.googleusercontent.com/translate\\_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Qpel&prev=\\_t&rurl=translate.google.it&usg=ALkJrhiLCrTXBn-43nRz3HliwQDbcUSs3w](http://translate.googleusercontent.com/translate_c?hl=it&ie=UTF-8&sl=en&tl=it&u=http://en.wikipedia.org/wiki/Qpel&prev=_t&rurl=translate.google.it&usg=ALkJrhiLCrTXBn-43nRz3HliwQDbcUSs3w)) (Quarter pixel) modelli di codec DivX.

Con queste, si sono avvitati e la necessità di ri-codificare il video. Ciò significa dual-pass encoding per mantenere la stessa dimensione.

C'è un modo per ottenere quasi lo stesso risultato utilizzando un solo passaggio. Assumendo che il video è fatto correttamente, la codifica del precedente, già distribuito quantizzatore e bitrate per ottenere una buona qualità per una determinata dimensione. Perché non riutilizzare queste informazioni?

Per fare ciò, impostare XviD (o libavcodec MPEG-4) per l'uso "Same Qz come input" mode. In tal modo, essi codificare ogni fotogramma con la quantizzazione precedentemente utilizzato per codificare. Il risultato si hanno circa la stessa dimensione / qualità rispetto agli originali, ma con un solo passaggio.

Che può anche essere fatto utilizzando la modalità a riga di comando per elaborare un sacco di file in una riga:

```
#!/ bin / bash
export SRC = / tmp/fma2
TGT export = / tmp/fma3
foreach.pl perl $ SRC / *. avi "avidemux2 \
- unpack-force - load $ SRC /% f \
- map-Audio - Video-processi - video-xvid4 codec \
- video-conf seguire = 1000 - save $ TGT / fixed_% f \
- quit "
```

## Muxing

Ho trovato che alcuni software (come alcune versioni di Nandub) non mux il file correttamente. Che fa balbettare standalone e congelare. La soluzione è semplice: basta caricare e salvare il file in Avidemux.

---

= Avidemux = codificatori video = Audio Codificatori = filtri video = filtri audio = Tutorial = Glossario =  
Estratto da "[http://www.avidemux.org/admWiki/index.php?title=Standalone\\_MPEG-4\\_players](http://www.avidemux.org/admWiki/index.php?title=Standalone_MPEG-4_players)"

- 
- Ultima modifica 18:10, 4 giugno 2006.
  - Questa pagina è stata letta 39.130 volte.
  - Tutela della privacy
  - A proposito di Avidemux2
  - Disclaimer