Convert your video files to WebM format using FFmpeg/ mencoder

Jing (mqJing@gmail.com)

Outline

1. A Bad Solution for DVD2WebM
2. 常見影音格式
3. H.264 vs. WebM, 俄國莫斯科國立大學資訊科學系圖形與多媒體實驗室研究報告
4. 利用 FFMPEG 轉檔 webm
5. 利用 mencoder 轉檔 webm

A Solution to DVD2H264

Step 1: get the title information

```
isdvd video.iso
```

Step 2: get the subtitle information

```
mplayer -v dvd://<Title> -dvd-device <video.iso>
```

ex:

```
mplayer -v dvd://2 -dvd-device video.iso
```

Step 3: extract the subtitle

```
rm output.idx output.sub
mencoder dvd://2 -dvd-device video.iso -o output.xxx -ovc frameno -oac mp3lame -lameopts vbr=3 -vobsubout output -vobsuboutindex 0  -vobsuboutid zh -sid 11
mencoder dvd://2 -dvd-device video.iso -o output.xxx -ovc frameno -oac mp3lame -lameopts vbr=3 -vobsubout output -vobsuboutindex 1  -vobsuboutid en -sid 3
```

Step 4: encode to H.264/mp3

```
mencoder dvd://2 -dvd-device video.iso -o output.avi -ovc x264 -x264encopts bitrate=1800 pass=1 -oac mp3lame -lameopts q=0:aq=0
```

Further Information

- A lots of transcode knowledge, http://www.mplayerhq.hu/DOCS/HTML/en/menc-feat-dvd-mpeg4.html#menc-feat-dvd-mpeg4-preparing-encode
-

A Bad Solution for DVD2WebM

Basic Flow

1. Extract the subtitles (Chinese and English)
2. Copy the video stream from a specified title of DVD
3. Transcoding the stream into WebM format

Get DVD title information

lsdvd video.iso

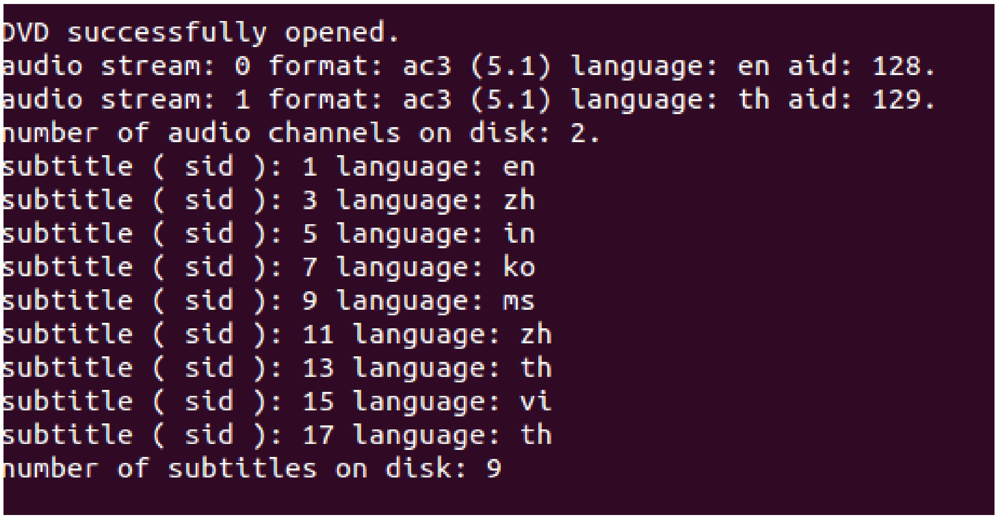
Get subtitle information

Command:

```
mplayer -v dvd://<Title #> -dvd-device <video.iso>
```

ex:

```
mplayer -v dvd://1 -dvd-device video.iso
```



```
DVD successfully opened.
audio stream: 0 format: ac3 (5.1) language: en aid: 128.
audio stream: 1 format: ac3 (5.1) language: th aid: 129.
number of audio channels on disk: 2.
subtitle ( sid ): 1 language: en
subtitle ( sid ): 3 language: zh
subtitle ( sid ): 5 language: in
subtitle ( sid ): 7 language: ko
subtitle ( sid ): 9 language: ms
subtitle ( sid ): 11 language: zh
subtitle ( sid ): 13 language: th
subtitle ( sid ): 15 language: vi
subtitle ( sid ): 17 language: th
number of subtitles on disk: 9
```

step 1: Extract the subtitle

mencoder dvd://<Title#> -dvd-device <DVD Image> -o output.xxx -ovc frameno -oac mp3lame -lameopts vbr=3 -vobsubout <Ouput Subtitle Name> -vobsuboutindex <The Subtitle Index in the Subtitle File> 0 -vobsuboutid <Specify the language name for the subtitle index, show in the subtitle track> -sid **<Specify the subtitle we want to extract>**

ex:

Add two subtitles in the subtitle file

(a) Clear

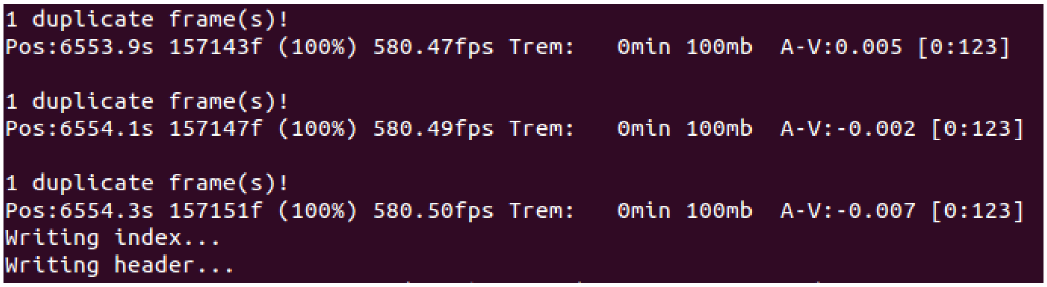
```
rm output.idx output.sub   // see the note
```

(b) Extract the Tranditional Chinese subtitle and assign the it as zh

```
mencoder dvd://2 -dvd-device video.iso -o output.xxx -ovc frameno -oac mp3lame -lameopts vbr=3 -vobsubout output -vobsuboutindex 0  -vobsuboutid zh -sid 11
```

(c) Extract the English subtitle and assign the index as en mencoder dvd://2 -dvd-device video.iso -o output.xxx -ovc frameno -oac mp3lame -lameopts vbr=3 -vobsubout output -vobsuboutindex 1 -vobsuboutid en -sid 1

Note: you should remove the idx and sub file or mencoder will append the result at the previous subtitle files.



```
1 duplicate frame(s)!
Pos:6553.9s 157143f (100%) 580.47fps Trem:   0min 100mb  A-V:0.005 [0:123]

1 duplicate frame(s)!
Pos:6554.1s 157147f (100%) 580.49fps Trem:   0min 100mb  A-V:-0.002 [0:123]

1 duplicate frame(s)!
Pos:6554.3s 157151f (100%) 580.50fps Trem:   0min 100mb  A-V:-0.007 [0:123]
Writing index...
Writing header...
```

step 2: Copy the dvd title 2

mencoder dvd://2 -dvd-device video.iso -o output.mpg -ovc copy -oac mp3lame -lameopts q=0:aq=0

note: please don’t use the “-oac copy”. it will causing audio/video doesn’t sync issue)

step 3: mpg -> webm

ffmpeg -i output.mpg -b 1500k -acodec libvorbis -ab 160k -ac 2 -f webm output.webm

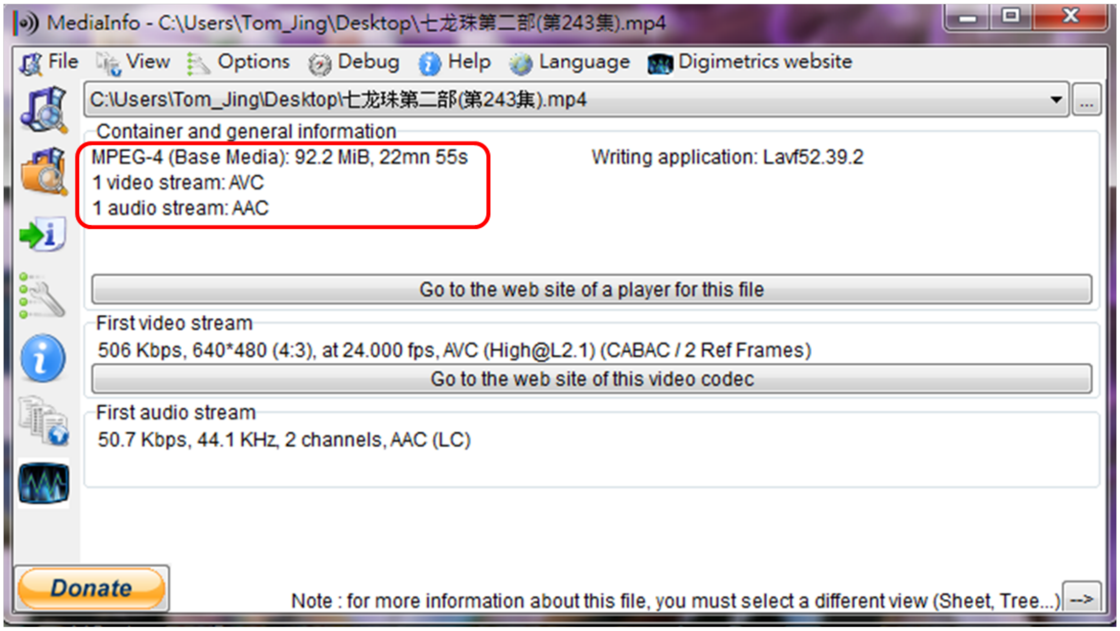
Note:

因為技術的關係, 直接使用 mencoder 轉出來的 web 格式檔, 有字幕但沒有聲音. 所以上面的方法, 是使用 ffmpeg 轉 web.

常見影音格式

- **mp4 file format**: 這個檔案格式, 被設計用來封裝 MPEG-4 影音串流的壓縮資料, see MPEG-4 Part 14. 其中影像壓縮技術標準 ITU H.264, 已經被包含在 MPEG-4 裡面. see MPEG-4 Part 10 [9].

  ex: 在下圖中, 你會看到 video stream 是用 Advanced Video Coding, AVC (== H.264)


MediaInfo - C:\Users\Tom_Jing\Desktop\七龙珠第二部(第243集).mp4
File
View
Options
Debug
Help
Language
Digimetrics website
Go to the web site of a player for this file
Go to the web site of this video codec
Donate
-->

C:\Users\Tom_Jing\Desktop\七龙珠第二部(第243集).mp4

Container and general information

MPEG-4 (Base Media): 92.2 MiB, 22mn 55s
1 video stream: AVC
1 audio stream: AAC

Writing application: Lavf52.39.2

First video stream

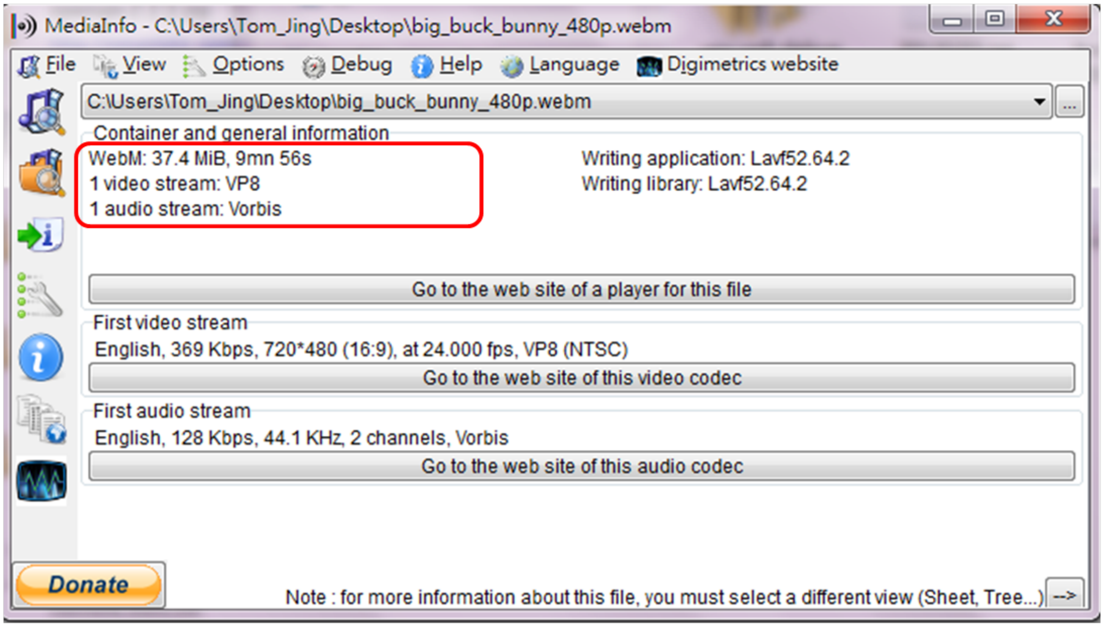
506 Kbps, 640*480 (4:3), at 24.000 fps, AVC (High@L2.1) (CABAC / 2 Ref Frames)

First audio stream

50.7 Kbps, 44.1 KHz, 2 channels, AAC (LC)

Note : for more information about this file, you must select a different view (Sheet, Tree...)

- **webm file format:** 這個檔案格式, 被設計用來封裝 vp8 與 vorbis 影音串流的壓縮資料 [10].


MediaInfo - C:\Users\Tom_Jing\Desktop\big_buck_bunny_480p.webm
File
View
Options
Debug
Help
Language
Digimetrics website
Go to the web site of a player for this file
Go to the web site of this video codec
Go to the web site of this audio codec
Donate
-->

C:\Users\Tom_Jing\Desktop\big_buck_bunny_480p.webm

Container and general information

WebM: 37.4 MiB, 9mn 56s
1 video stream: VP8
1 audio stream: Vorbis

Writing application: Lavf52.64.2
Writing library: Lavf52.64.2

First video stream

English, 369 Kbps, 720*480 (16:9), at 24.000 fps, VP8 (NTSC)

First audio stream

English, 128 Kbps, 44.1 KHz, 2 channels, Vorbis

Note : for more information about this file, you must select a different view (Sheet, Tree...)

H.264 vs. WebM: 請閱讀 2011 年五月的俄國莫斯科國立大學資訊科學系圖形與多媒體實驗室的研究報告 MPEG-4 AVC/H.264 Video Codecs Comparison [11]. 下面是節錄的簡單結果.

影像品質的比較:
下面是在不同的 bitrate 下, 每一個 codec 的品質比較圖. Y-SSIM 越大越好. 由該報告做出的結果可以知道. 第一名是 x264. MS Expression Encoder 表現也不賴.

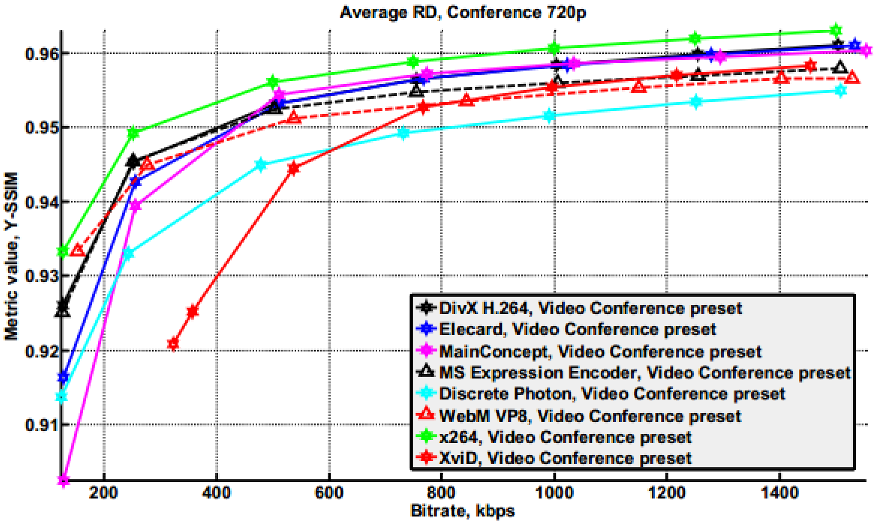


Figure 2. Bitrate/quality—usage area "Video Conference," 720p sequence, Y-SSIM metric

編碼速度的比較: WebM 比 x264 快

VIDEO MPEG-4 AVC/H.264 CODECS COMPARISON ***MOSCOW, MAY 2011***
CS MSU GRAPHICS & MEDIA LAB VIDEO GROUP

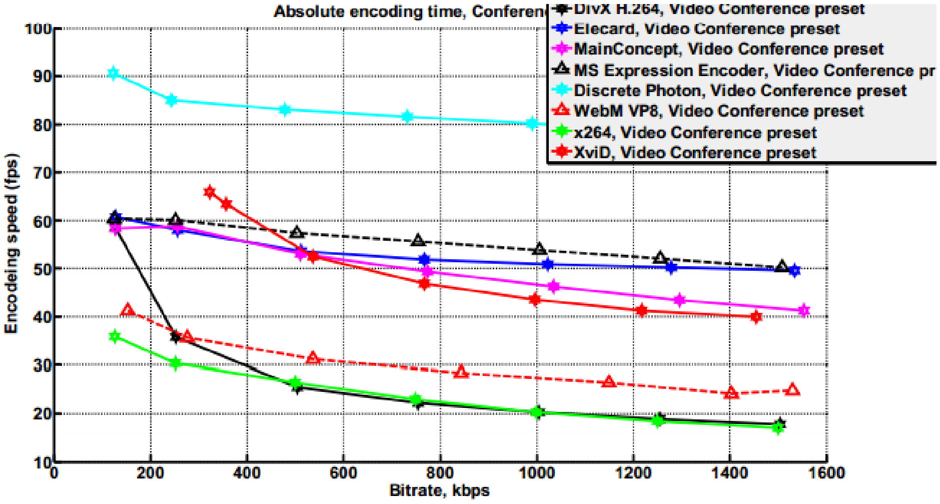


Figure 5. Encoding speed—usage area "Video Conference"

Misc

x264 presets

- Movie, High Quality:
  - 1-st pass: --tune ssim --pass 1 --keyint 500 --preset slow
  - 2-nd pass: --tune ssim --pass 2 --keyint 500 --preset slow

dvd2h264

```
mencoder dvd://<title #> -dvd-device <video.iso> -o <output.avi> -ovc
x264 -x264encopts bitrate=1800 pass=1 -oac mp3lame -lameopts q=0:aq=0
```

ex:

```
mencoder dvd://2 -dvd-device video.iso -o output.avi -ovc x264 -x264encopts
bitrate=1800 pass=1 -oac mp3lame -lameopts q=0:aq=0
```

FFMPEG part

Merge all vob files to a single one

```
ffmpeg -i 1.vob 2.vob -vcodec copy -acodec copy output.vob
```

Transcoding the video file

Windows

Batch file: webm.bat

```
REM webm (VP8 / Vorbis)
"ffmpeg.exe" -i %1 -b 2000k -vcodec libvpx -acodec libvorbis -ab 160000 -ac 2 -f webm -g 30  %1_2000.webm

"ffmpeg.exe" -i %1 -b 500k -vcodec libvpx -acodec libvorbis -ab 160000 -ac 2 -f webm -g 30  %1_500.webm
```

Usage

```
c:>web.bat <File_name>.mp4
```

Linux

syntax:

```
ffmpeg -i <input> -b <bit rate>  -acodec <audio codec> -ab <audio bit rate> -ac <audio channel number> -f <format> <output_filename>
```

example:

```
ffmpeg -i ./VTS_02_1.VOB -b 1500k  -acodec libvorbis -ab 160k -ac 2 -f webm VTS_02_1.VOB_1500.webm
```

Select audio channel

Step 1: get the audio stream information

```
ffmpeg -i jing.vob
```

Step 2: extract the audio

```
-map <input>:<stream>
```

ex:

```
-map 0:3
```

Speedup

```
-thread 4
```

Mplayer Part

dvd2xvid

```
mencoder dvd://<title #> -dvd-device <video.iso> -o <output.avi> -ovc xvid -xvidencopts bitrate=1800 -oac mp3lame -lameopts q=0:aq=0
```

ex:

```
mencoder dvd://2 -dvd-device video.iso -o output.avi -ovc xvid -xvidencopts bitrate=1800 -oac mp3lame -lameopts q=0:aq=0
```

dvd2h264

```
mencoder dvd://<title #> -dvd-device <video.iso> -o <output.avi> -ovc x264 -x264encopts bitrate=1800 pass=1 -oac mp3lame -lameopts q=0:aq=0
```

ex:

```
mencoder dvd://2 -dvd-device video.iso -o output.avi -ovc x264 -x264encopts bitrate=1800 pass=1 -oac mp3lame -lameopts q=0:aq=0
```

dvd2webm (no sound) (with idx subtitle: no delay)

```
mencoder dvd://<title #> -dvd-device <video.iso> -o <output.webm> -ovc lavc -oac lavc -of lavf -lavfopts format=webm -lavcopts acodec=vorbis:vcodec=libvpx -ffourcc VP80
```

copy (no subtitle)

```
mencoder dvd://2 -dvd-device video.iso -o output.mpg -ovc copy -oac copy
```

list supported audio/video codec

```
mencoder -ovc help -oac help
```

[5]

Extracting DVD subtitles to a Vobsub file

dvd2h264 + extract subtitle with default lanuage

```
remove subtitles.idx subtitles.sub
mencoder dvd://2 -dvd-device video.iso -o output.avi -ovc x264 -x264encopts bitrate=1800 pass=1 -oac mp3lame -lameopts q=0:aq=0 -vobsubout subtitles
```

[6]

verify

vlc output.avi [choose the subtitle file: subtitle.idx] http://web.njit.edu/all_topics/Prog_Lang_Docs/html/mplayer/encoding.html

ex:
mencoder dvd://2 -dvd-device video.iso -o output.webm -ovc lavc -oac lavc -of lavf -lavfopts format=webm -lavcopts acodec=vorbis:vcodec=libvpx -ffourcc VP80

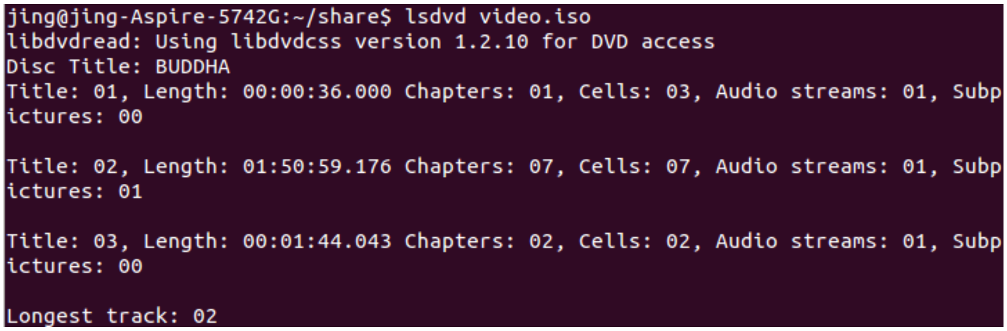
Gather Information

DVD Information

lsdvd <video.iso> or <mount point>

ex:

lsdvd video.iso

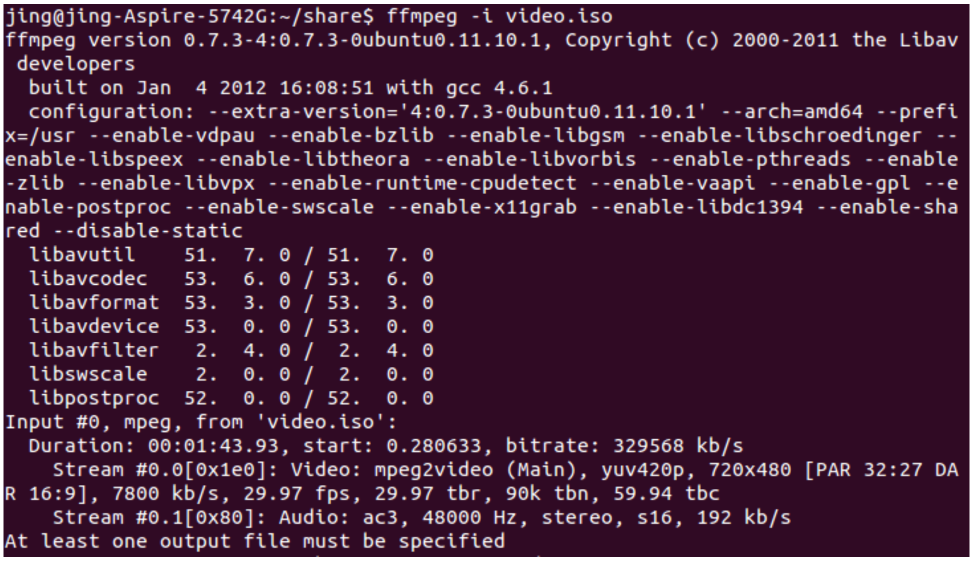


```
jing@jing-Aspire-5742G:~/share$ lsdvd video.iso
libdvdread: Using libdvdcss version 1.2.10 for DVD access
Disc Title: BUDDHA
Title: 01, Length: 00:00:36.000 Chapters: 01, Cells: 03, Audio streams: 01, Subp
ictures: 00

Title: 02, Length: 01:50:59.176 Chapters: 07, Cells: 07, Audio streams: 01, Subp
ictures: 01

Title: 03, Length: 00:01:44.043 Chapters: 02, Cells: 02, Audio streams: 01, Subp
ictures: 00

Longest track: 02
```

[4]

Video Information

ffmpeg -i video.iso



```
jing@jing-Aspire-5742G:~/share$ ffmpeg -i video.iso
ffmpeg version 0.7.3-4:0.7.3-0ubuntu0.11.10.1, Copyright (c) 2000-2011 the Libav
 developers
  built on Jan  4 2012 16:08:51 with gcc 4.6.1
  configuration: --extra-version='4:0.7.3-0ubuntu0.11.10.1' --arch=amd64 --prefi
x=/usr --enable-vdpau --enable-bzlib --enable-libgsm --enable-libschroedinger --
enable-libspeex --enable-libtheora --enable-libvorbis --enable-pthreads --enable
-zlib --enable-libvpx --enable-runtime-cpudetect --enable-vaapi --enable-gpl --e
nable-postproc --enable-swscale --enable-x11grab --enable-libdc1394 --enable-sha
red --disable-static
  libavutil    51.  7. 0 / 51.  7. 0
  libavcodec   53.  6. 0 / 53.  6. 0
  libavformat  53.  3. 0 / 53.  3. 0
  libavdevice  53.  0. 0 / 53.  0. 0
  libavfilter   2.  4. 0 /  2.  4. 0
  libswscale    2.  0. 0 /  2.  0. 0
  libpostproc  52.  0. 0 / 52.  0. 0
Input #0, mpeg, from 'video.iso':
  Duration: 00:01:43.93, start: 0.280633, bitrate: 329568 kb/s
    Stream #0.0[0x1e0]: Video: mpeg2video (Main), yuv420p, 720x480 [PAR 32:27 DA
R 16:9], 7800 kb/s, 29.97 fps, 29.97 tbr, 90k tbn, 59.94 tbc
    Stream #0.1[0x80]: Audio: ac3, 48000 Hz, stereo, s16, 192 kb/s
At least one output file must be specified
```

How to build WebM library

Step 1: Pull the source

git init

git pull http://git.chromium.org/webm/libvpx.git

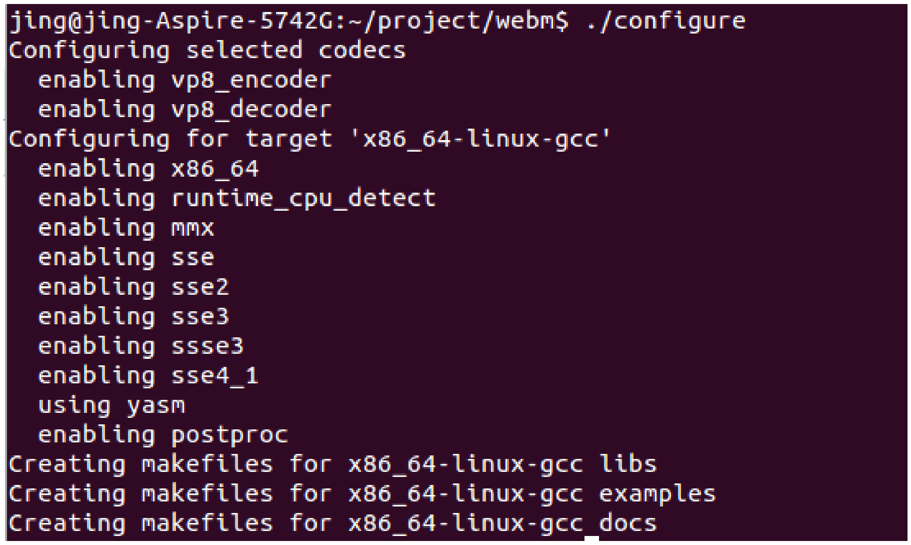
[7]

Step 2: install necessary tool chain

yasm

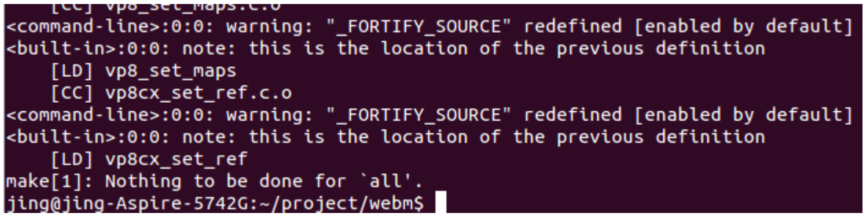
Step 3: configure your option

$ ./configure

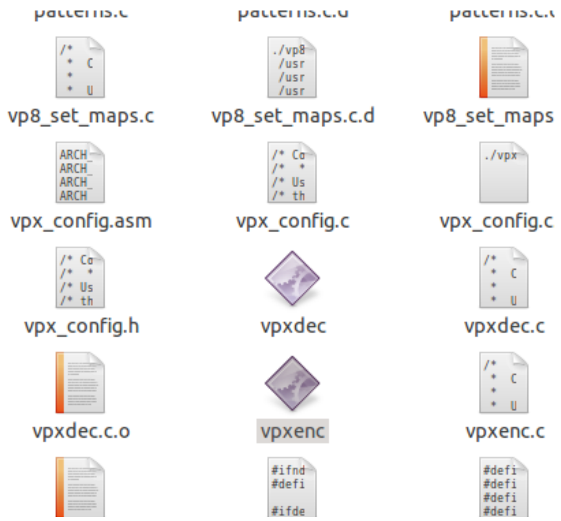


```
jing@jing-Aspire-5742G:~/project/webm$ ./configure
Configuring selected codecs
  enabling vp8_encoder
  enabling vp8_decoder
Configuring for target 'x86_64-linux-gcc'
  enabling x86_64
  enabling runtime_cpu_detect
  enabling mmx
  enabling sse
  enabling sse2
  enabling sse3
  enabling ssse3
  enabling sse4_1
  using yasm
  enabling postproc
Creating makefiles for x86_64-linux-gcc libs
Creating makefiles for x86_64-linux-gcc examples
Creating makefiles for x86_64-linux-gcc docs
```

Step 4: make



```
    [CC] vp8_set_maps.c.o
<command-line>:0:0: warning: "_FORTIFY_SOURCE" redefined [enabled by default]
<built-in>:0:0: note: this is the location of the previous definition
    [LD] vp8_set_maps
    [CC] vp8cx_set_ref.c.o
<command-line>:0:0: warning: "_FORTIFY_SOURCE" redefined [enabled by default]
<built-in>:0:0: note: this is the location of the previous definition
    [LD] vp8cx_set_ref
make[1]: Nothing to be done for `all'.
jing@jing-Aspire-5742G:~/project/webm$
```



Test

(input.mpg should be a deinterlaced video)

ffmpeg -i input.mpg -an input.y4m

```
 ../project/webm/vpxenc --end-usage=cbr --target-bitrate=500 --rt --threads=0 --
output=output.webm input.y4m
```

pipe version

```
ffmpeg -i input.mpg -pix_fmt yuv420p -f yuv4mpegpipe - 2>/dev/null | ../
project/webm/vpxenc --end-usage=cbr --target-bitrate=500 --rt --threads=0 - -o
output.webm
```

[8]

References

1. http://howto-pages.org/ffmpeg/
2. http://wiki.quakeworld.nu/Mencoder_howto
3. mencoder webm: http://wiki.samat.org/WebMv
4. list dvd information, http://savvyadmin.com/dvd-to-xvid-encoding-with-mencoder/
5. Mcodec How to, http://wiki.quakeworld.nu/Mencoder_howto
6. Extracting DVD subtitles to a Vobsub file, http://web.njit.edu/all_topics/Prog_Lang_Docs/html/mplayer/encoding.html
7. VP8 Encoder Parameter Guidelines, http://www.webmproject.org/tools/encoder-parameters/
8. LGPL project covering MPEG video compression technology - User discussions, http://comments.gmane.org/gmane.comp.video.ffmpeg.user/32907 .. for mpeg 2 webm
9. MPEG-4, http://en.wikipedia.org/wiki/MPEG-4
10. WebM Project, http://www.webmproject.org/
11. MPEG-4 AVC/H.264 Video Codecs Comparison, http://www.compression.ru/video/codec_comparison/h264_2011/mpeg-4_avc_h264_video_codecs_comparison.pdf
12. FFmpeg document, http://ffmpeg.org/ffmpeg.html

VLC Part -- Transcode Guide (VOB)

The context is try to answer the following question: How to transcode the VOB file using VLC with no window, subtitle, terminiate when the job done! All instructions are tested and verified.

Transcode, x264 video only

```
vlc C:\dvd\VIDEO_TS\VTS_01_2.VOB --sout="#transcode{venc=x264{keyint=10,
tune=zerolatency, vbv-maxrate=512}, vb=512,
acodec=none}:standard{access=file,mux=avi,dst="e:\output_file.avi"}"
```

Transcode, ffmpeg video only

```
vlc C:\dvd\VIDEO_TS\VTS_01_1.VOB --
sout="#transcode{vcodec=mp4v,acodec=mpga,vb=3000,ab=256,venc=ffmpeg{keyint=10, hurry-
up,vt=800000}}:standard{access=file,mux=avi,dst="e:\output_file.avi"}"
```

Transcode, h264 ts (audio + video)

```
vlc C:\dvd\VIDEO_TS\VTS_01_2.VOB --
sout="#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,samplerate
=44100}:standard{access=file,mux=avi,dst="e:\output_file.avi"}"
```

Transcode, h264 ts (audio + video) with dummy interface

```
vlc -I dummy C:\dvd\VIDEO_TS\VTS_01_2.VOB --
sout="#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,samplerate
=44100}:standard{access=file,mux=avi,dst="e:\output_file.avi"}"
```

Transcode, h264 (audio + video + subtitle) with dummy interface

```
vlc -I dummy C:\dvd\VIDEO_TS\VTS_01_2.VOB --sub-track=0 --
sout="#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,samplerate
=44100,soverlay}:standard{access=file,mux=avi,dst="e:\output_file.avi"}" vlc://quit
```

Transcode, h264 (audio + video + subtitle) with no window

```
vlc -I dummy --dummy-quiet C:\dvd\VIDEO_TS\VTS_01_2.VOB --sub-track=0 --
sout="#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,samplerate
=44100,soverlay}:standard{access=file,mux=avi,dst="e:\output_file.avi"}" vlc://quit
```

activate the verbose mode

```
        -vvv
```

no window

```
        -I dummy --dummy-quiet
```

play VOB with subtitle

```
        vlc --sub-track=4 C:\dvd\VIDEO_TS\VTS_01_2.VOB
```

kill the thread when done

```
vlc -I dummy --dummy-quiet  e:\input_file.avi --
sout="#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,samplerate
=44100}:standard{access=file,mux=avi,dst="e:\output_file.avi"}" vlc://quit
```

kill the thread when done (script version)

```
call "C:\Program Files\VideoLAN\VLC\vlc.exe" -I dummy --dummy-quiet  e:\input_file.avi --
sout="#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,samplerate
=44100}:standard{access=file,mux=avi,dst="e:\output_file.avi"}" vlc://quit
```

VLC References

1. VLC Command Line Help, http://wiki.videolan.org/VLC_command-line_help
2. Advanced Use of Vlc, http://wiki.videolan.org/Documentation:Play_HowTo/Advanced_Use_of_VLC
3. Advanced Streaming Using Command Line of Vlc, http://wiki.videolan.org/Documentation:Streaming_HowTo/Advanced_Streaming_Using_the_Command_Line
4. VLC Video Streaming, https://sites.google.com/site/cipolettasjuliatechnology/video-streaming/vlc
5. VLC Command Line Batch file, http://www.autohotkey.com/board/topic/84403-vlc-command-line-parameters-not-going-through/
6. VLC Command Line Batch file 2, http://superuser.com/questions/409990/vlc-command-line-batch-video-conversion
7. Language Tag and Code, http://en.wikipedia.org/wiki/Language_localisation

PowerShell Part

init

step 1: launch cmd.exe with administrator mode & into the powershell environment

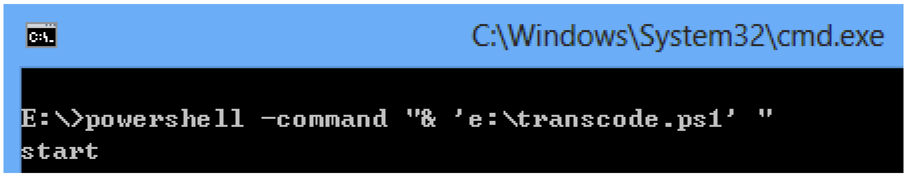
```
>powershell.exe
```

step 2: Enable non-interactive mode

```
ps >set-executionpolicy RemoteSigned
```

step 3: run the ps1 file from Dos

```
>powershell -command "& 'e:\transcode.ps1' "
```



```
E:\>powershell -command "& 'e:\transcode.ps1' "
start
```

special character

```
`0  Null
`a  Alert bell/beep
`b  Backspace
`f  Form feed
`n  New line
`r  Carriage return
`t  Horizontal tab
`v  Vertical tab
`'  Single quote
`"  Double quote
```

example:

Call native execution file

transcode.ps1

```
$cmd = "& `"C:\Program Files\VideoLAN\VLC\vlc.exe`" -I dummy --dummy-
quiet  e:\input_file.avi --
sout=`"#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,sampler
ate=44100}:standard{access=file,mux=avi,dst=`"e:\output_file.avi`"}`" vlc://quit"

write-host $cmd
Invoke-Expression $cmd | out-null
```

Dos Prompt

```
powershell -command "& 'e:\transcode.ps1' "
```

Wait a job complete and then run the others (Jing’s suggestion)

transcode.ps1

```
$cmd = "& `"C:\Program Files\VideoLAN\VLC\vlc.exe`" -I dummy --dummy-
quiet  e:\input_file.avi --
sout=`"#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,sampler
ate=44100}:standard{access=file,mux=avi,dst=`"e:\output_file.avi`"}`" vlc://quit"
$cmd2 = "& `"C:\Program Files\VideoLAN\VLC\vlc.exe`" -I dummy --dummy-
quiet  e:\input_file2.avi --
sout=`"#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,sampler
ate=44100}:standard{access=file,mux=avi,dst=`"e:\output_file2.avi`"}`" vlc://quit"

# step 1: do the transcode
write-host "transcoding ..."
write-host $cmd
Invoke-Expression $cmd

# step 2: wait the job done
$nid = (get-process vlc).id
write-host "wait for vlc. nid=" $nid
wait-process -id $nid

# step 3: do the next job (transcode ...)
write-host "transcoding ..."
write-host $cmd2
Invoke-Expression $cmd2

# step 4: wait the job done
$nid = (get-process vlc).id
write-host "wait for vlc. nid=" $nid
```

```
wait-process -id $nid

write-host "ok"
```

Dos Prompt

```
        powershell -command "& 'e:\transcode.ps1' "
```

Foreach Version (影音不同步)

vob2h264.ps1

```
# 把指定目錄下的 vob 檔案, 批次轉換成 H.264 格式

$SourceDir="I:\VIDEO_TS\"
$DestDir="e:\"
$Source=@("vts_02_1.vob", "vts_02_2.vob","vts_02_3.vob","vts_02_4.vob")

#$cmd = "& `"C:\Program Files\VideoLAN\VLC\vlc.exe`" -I dummy --dummy-
quiet  e:\input_file.avi --
sout=`"#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,sampler
ate=44100}:standard{access=file,mux=avi,dst=`"e:\output_file.avi`"}`" vlc://quit"


foreach ($item in $Source){
        # step 1: do the transcode
        write-host "Transcode processing:" $SourceDir$item " to " $DestDir$item".avi"
        $trans_cmd="& `"C:\Program Files\VideoLAN\VLC\vlc.exe`" -I dummy --dummy-quiet
$SourceDir$item --
sout=`"#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,samplera
te=44100, deinterlace}:standard{access=file,mux=avi,dst=`"$DestDir$item.avi`"}`" vlc://quit"
        write-host $trans_cmd
        Invoke-Expression $trans_cmd

        # step 2: wait the job done
        $nid = (get-process vlc).id
        write-host "wait for vlc. nid=" $nid
        wait-process -id $nid
}

write-host "Job done!"
```

run.bat

```
powershell -command "& '.\vob2h264.ps1' "
```

DOS Batch Part

Call DOS bat example

transcode.ps1

```
$filename = "test1.vob"
$cmd3 = "& cmd /k e:\run.bat "+$filename


write-host $cmd3
Invoke-Expression $cmd3
write-host "done! 1"


$filename = "test2.vob"
$cmd3 = "& cmd /k e:\run.bat "+$filename


write-host $cmd3
Invoke-Expression $cmd3
write-host "done! 1"
```

run.bat

```
rem %1 = argument 1. Here is the “test1.vob”
echo %1
call "C:\Program Files\VideoLAN\VLC\vlc.exe" -I dummy --dummy-quiet  %1 --
sout="#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,samplera
te=44100}:standard{access=file,mux=avi,dst="e:\output_file.avi"}" vlc://quit

rem exit for powershell cmd.exe
exit
```

Call DOS bat example 2

transcode.ps1

```
$filename = "e:\input_file2.avi"
$cmd3 = "& cmd /k e:\run2.bat "+$filename
write-host $cmd3
Invoke-Expression $cmd3
write-host "done! 1"

$filename = "e:\input_file.avi"
$cmd3 = "& cmd /k e:\run2.bat "+$filename
write-host $cmd3
Invoke-Expression $cmd3
write-host "done! 1"
```

run2.bat

```
call "C:\Program Files\VideoLAN\VLC\vlc.exe" -I dummy --dummy-quiet  %1 --sout="#transcode{vcodec=h264,vb=800,scale=1,acodec=mpga,ab=128,channels=2,samplerate=44100}:standard{access=file,mux=avi,dst="e:\output_file.avi"}" vlc://quit
exit
```

Dos Prompt

```
E:\>powershell -command "& 'e:\transcode.ps1' "
```

PowerShell References

1. Building GUI Applications in PowerShell, http://www.drdobbs.com/windows/building-gui-applications-in-powershell/240049898

Requirement

1	basic	vob -> h264, transcode script	單一VOB檔轉檔
2	basic	subtitle (with default language)	單一VOB檔轉檔包含字幕

3	basic batch	current folder transcode	對目前目錄中，所有 VOB 檔進行轉檔
4	basic batch	specified a folder and nest transcoding all VOB files	對指定目錄中，所有 VOB 檔(包含子目錄內)

			全部進行轉檔
5	Advanced	Configure file	提供設定檔功能
6	Advanced	Basic transcode function with simple GUI	提供視窗介面
7	Advanced	select target folder with simple GUI	提供指定轉檔後的 H264 檔位置