DivX Tutorial

Written By: Nicky

All Files and Downloads can be found at:

http://www.digital-digest.com/nickyguides/

Extracting the DVD to Hard Disk

As soon as you begin to think they cannot make a better ripper suddenly one turns up! SmartRipper is now my no.1 ripper. And, after you try it, I'm sure you will agree with me =). What makes SmartRipper so good is it can actually read a DVD just like a DVD player! It lets you choose which language you wish to rip, and it looks at the information files (.ifo) on the DVD so it can handle mult-angle movies in the same way Flask Mpeg does in DVD mode! In fact, if it wasn't for the fact that Flask Mpeg lets you add subtitles, there would be no need to bother with Flask's DVD mode at all!

***Note:** If SmartRipper says "can't unlock drive" you will need to start your DVD playing in your PC DVD player (such as PowerDVD, WinDVD etc). Start it, press pause, and then launch SmartRipper.*

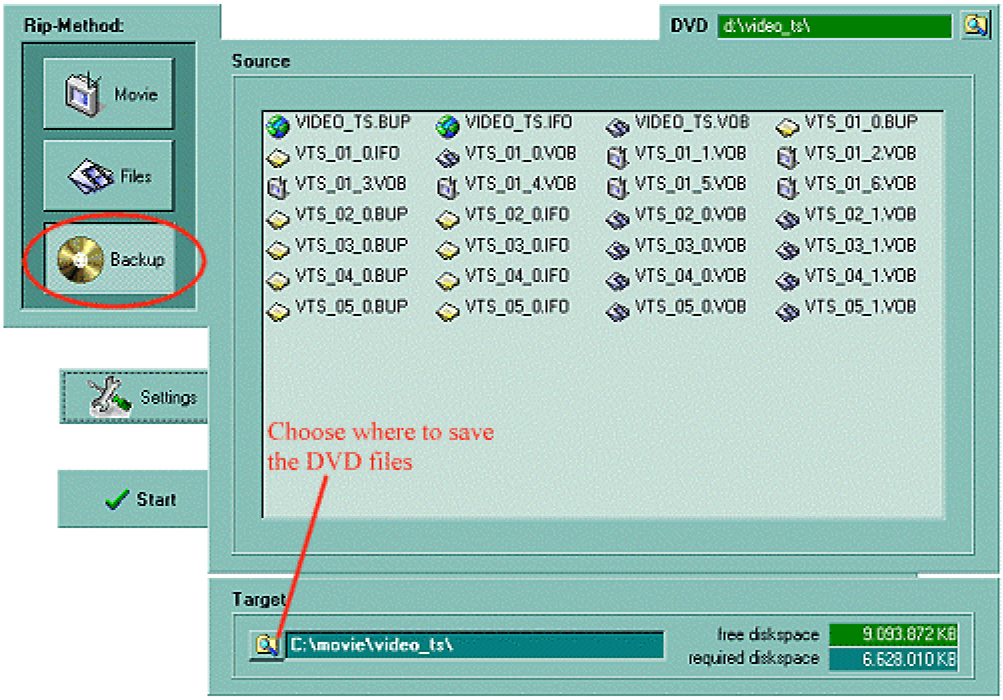
Before I start you will need:

Smartripper 2.02

SmartRipper has three basic modes:

BACKUP MODE

This mode backs up the entire DVD. It doesn't let you choose what language you want to use or what angle of the movie to rip. It is used solely for the purpose of copying exactly the whole DVD unchanged to your hard disk. It does, of course, decrypt the DVD but all files remain the same.



DVD d:\video_ts\

VIDEO_TS.BUP	VIDEO_TS.IFO	VIDEO_TS.VOB	VTS_01_0.BUP
VTS_01_0.IFO	VTS_01_0.VOB	VTS_01_1.VOB	VTS_01_2.VOB
VTS_01_3.VOB	VTS_01_4.VOB	VTS_01_5.VOB	VTS_01_6.VOB
VTS_02_0.BUP	VTS_02_0.IFO	VTS_02_0.VOB	VTS_02_1.VOB
VTS_03_0.BUP	VTS_03_0.IFO	VTS_03_0.VOB	VTS_03_1.VOB
VTS_04_0.BUP	VTS_04_0.IFO	VTS_04_0.VOB	VTS_04_1.VOB
VTS_05_0.BUP	VTS_05_0.IFO	VTS_05_0.VOB	VTS_05_1.VOB

Choose where to save the DVD files

C:\movie\video_ts\

free diskspace 9.093.872 KB
required diskspace 6.628.010 KB

Backing up a whole DVD in one go takes a ***lot*** of space, something like 6-9GB. For this reason we will *not* be using backup mode for this guide. But either way its laughingly easy to use: simply choose where you want to save your DVD file by clicking on the 'Target' button. Then click on the large Start button. That's it!

FILE MODE

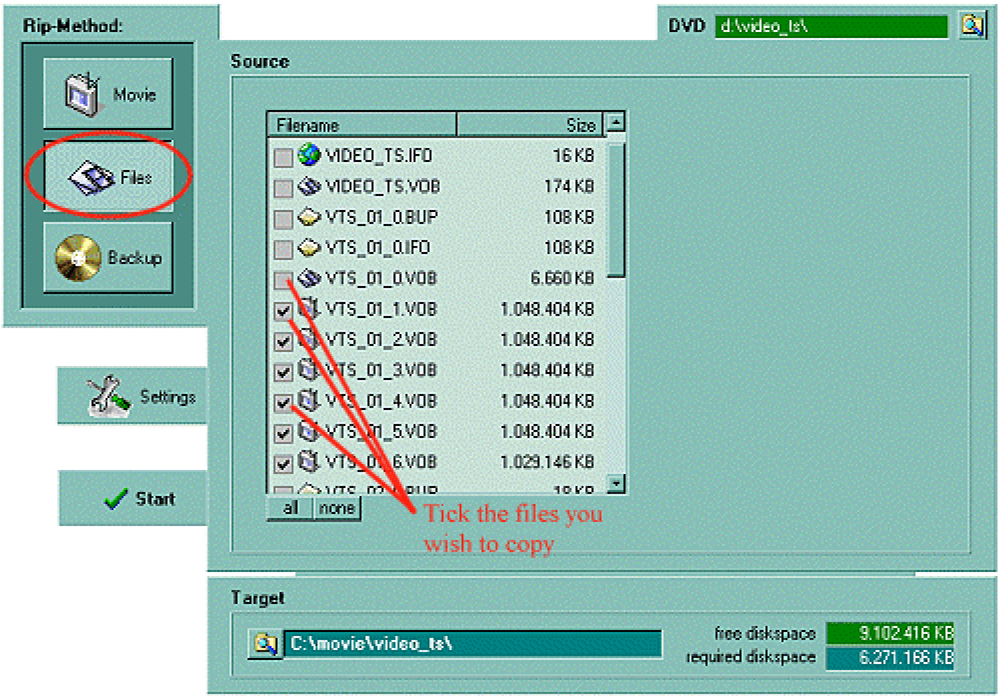
File Mode is almost the same as Backup Mode except it lets you choose which files to copy. Again, it does not read multi-angle's or let you choose which language it will rip, it literally just copies and decrypts any file you point to.

Note for Flask's DVD mode:

If you are going to use Flask Mpeg in DVD mode, then SmartRippers File Mode is the best way to rip the files for Flask to use. In File mode SmartRipper automatically selects the correct files for the main movie for you - they will have a TV screen picture by them. But don't forget to copy the IFO file that belongs to the movie as well. This will be called by the same name as the first movie file. For example, if the first movie file is called **VTS_01_0.VOB** the IFO file to copy with it will be **VTS_01_0.IFO**.

Remember, the main movie will not always have the same name. So if first file of the main movie were called **VTS_02_0.VOB** then the IFO file to copy with it will obviously also be **VTS_02_0.IFO**

For more details on this subject read the article called *"Introduction to the Structure of a DVD"* in the appendix section.



DVD d:\video_ts\

	Filename	Size
□	VIDEO_TS.IFO	16 KB
□	VIDEO_TS.VOB	174 KB
□	VTS_01_0.BUP	108 KB
□	VTS_01_0.IFO	108 KB
□	VTS_01_0.VOB	6.660 KB
☑	VTS_01_1.VOB	1.048.404 KB
☑	VTS_01_2.VOB	1.048.404 KB
☑	VTS_01_3.VOB	1.048.404 KB
☑	VTS_01_4.VOB	1.048.404 KB
☑	VTS_01_5.VOB	1.048.404 KB
☑	VTS_01_6.VOB	1.029.146 KB

Tick the files you wish to copy

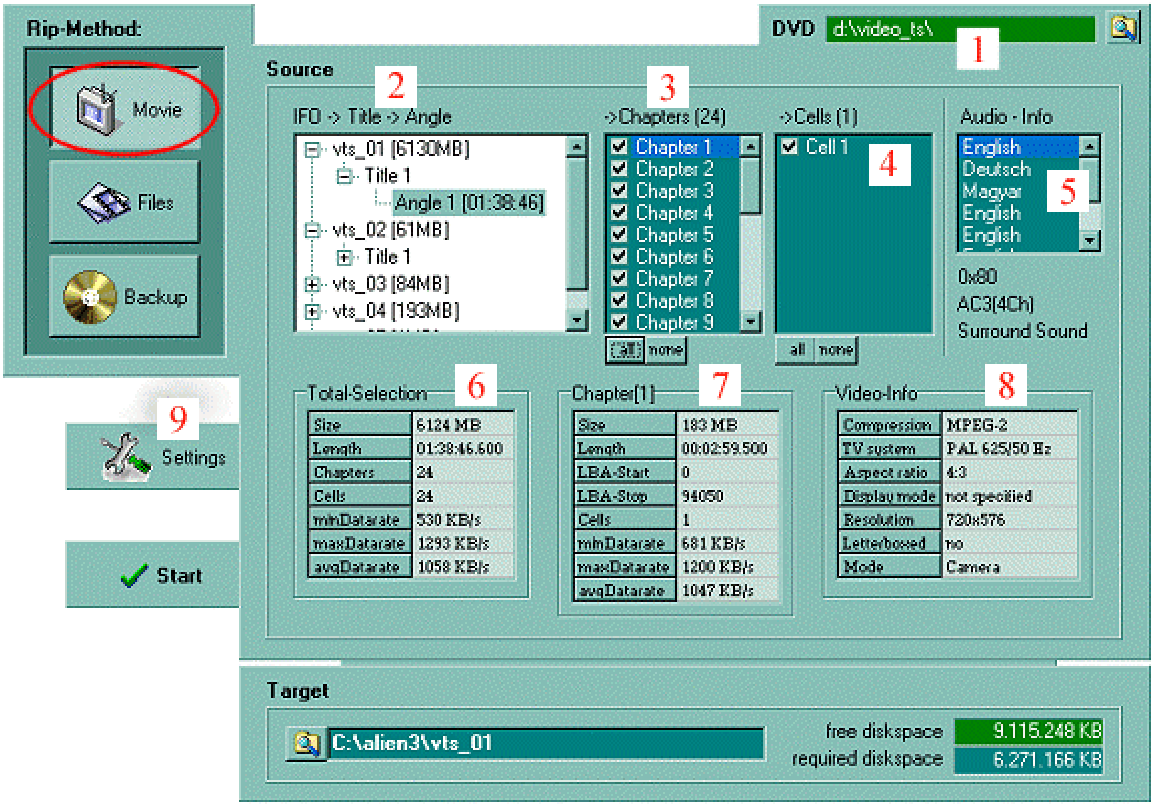
C:\movie\video_ts\

free diskspace 9.102.416 KB
required diskspace 6.271.166 KB

File mode is easy to use too; select where you wish to save your files in the 'Target' section. Tick the boxes next to the files you wish to backup. Then click on the large Start button. Your done!

MOVIE MODE

Movie mode is SmartRippers default mode and is by far the most powerful. Put in your CD and start SmartRipper and up pops the picture below. It looks complicated but I'll explain the features one by one.



DVD d:\video_ts\

Total-Selection

Size	6124 MB
Length	01:38:46.600
Chapters	24
Cells	24
minDatarate	530 KB/s
maxDatarate	1293 KB/s
avgDatarate	1058 KB/s

Chapter[1]

Size	183 MB
Length	00:02:59.500
LBA-Start	0
LBA-Stop	94050
Cells	1
minDatarate	681 KB/s
maxDatarate	1200 KB/s
avgDatarate	1047 KB/s

Video-Info

Compression	MPEG-2
TV system	PAL 625/50 Hz
Aspect ratio	4:3
Display mode	not specified
Resolution	720x576
Letterboxed	no
Mode	Camera

Target

C:\alien3\vts_01

free diskspace 9.115.248 KB

required diskspace 6.271.166 KB

1. DVD Selection Box: This lets you change the CD drive that SmartRipper is reading.

2. Movie Selection Box: This shows a list of movie titles SmartRipper has found. You can choose which movie title to rip. Since SmartRipper is reading the DVD like a DVD player, it will not rip half a movie or parts of the movie in wrong order. Each Title and Angle represents a *whole* DVD feature, be it a trailer, special feature, main movie, or even the menu selection! Each feature is identified by its DVD Vob name, such as: vts_something. The main movie is automatically chosen by SmartRipper, so you don't need to think much, just press start =). If more than one 'angle' appear for a single title, the correct one will usually be the longest in length. If they are the same length its probably a true multi-angle movie so choose the angle you prefer or just the first one.

3. Chapter Selection: This lets you extract individual chapters from each title. It is useful if you want to extract small clips for testing purposes before you spend the time converting a whole movie.

4. Cell: This doesn't seem to do much. Vob files can be divided by Cell or Vob ID, but so far I haven't had a DVD that needed this feature.

5. Audio Selection: Lets you select which audio track you wish to rip with the video.

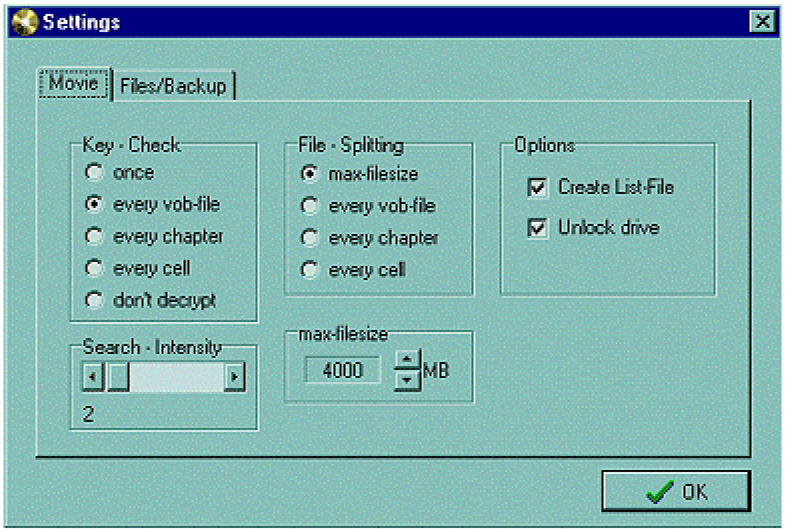
6. Total Selection: This is a display to tell you exactly what will rip when you press the start button. It offers useful info on data rate, cells, filesize and the length of the title.

7. Chapter: This offers the same information as **6** but only for the chapter you have selected in box **3**.

8. Video Info: This offers general information on the DVD. The aspect ratio is not usually correct, but such information is not needed anyway.

9. Settings: If you want to get even more technical, you can play about with the rip settings too. But you can quite happily leave this section alone if you prefer - the movie will rip perfectly without you touching it!

Finally, there is one option in the settings (**9**) that you may want to know about. SmartRipper is automatically set to 'max-filesize' in the 'File-Splitting' section. Instead of ripping every Vob separately, it will merge all the Vob files into one. *But* it will split them into separate files when it reaches the 'max-filesize' shown at the bottom. In the picture below this is set to 4000 MB (4 Gigabytes).



Key - Check
○ once
● every vob-file
○ every chapter
○ every cell
○ don't decrypt

Search - Intensity
2

File - Splitting
● max-filesize
○ every vob-file
○ every chapter
○ every cell

max-filesize
4000 MB

Options
☑ Create List-File
☑ Unlock drive

There are two reasons for this feature. Firstly, Mpeg2avi and Flask Mpeg will only convert about 7 Vob files at a time. This means if the DVD has more than that, you will not be able to convert it without problems. To avoid this, Smartripper will merge the files into as few as possible. Secondly, in Windows 95 / 98 there is a storage limit of 4GB per file. So, instead of trying to create a file larger than this, SmartRipper has the 'max-filesize' amount set to split them at 4GB. For more information on this read the '*The AVI Four Gigabyte Limit Explained*'' section in my appendix.

That's it! Because SmartRipper can handle most multi-angle movies you will be able to rip difficult movies like The Matrix without problems. There are still some very rare movies that neither Flask Mpeg nor SmartRipper are able to rip completely! One example is Terminator 2. This is a multi-branching movie and can currently only be done perfectly with vstrip. Check out my ''Repeated Sequences and multi-angle'' guide if you come across any of these movies.

Whenever you rip anything to hard drive its always a good idea to check it has been decoded correctly. The best way to do this is to play it in your computer DVD player. DVD Station, Power DVD, Win DVD ect., should all play them perfectly. They shouldn't have green or pink blocks and they shouldn't have any repeated scenes - although its quite normal for it to have a slightly squashed looking image. Check out the aspect ratio section of my appendix for an explanation of this.

DVD to DivX with Flask Mpeg

Flask Mpeg is the easiest way to rip a DVD! It takes a long time to do but that's just tough noogie! You can set it going all night and when you are sleeping and the next day it will probably be done. Or you can set it going while you are at work or both :) Flask Mpeg is a great program that can convert any Mpeg-1 or Mpeg-2 (i.e. DVD) file into a variety of formats such as DivX, AVI, Mjpeg, Mpeg-1 or Mpeg-2 etc!

Before I start here are the things you will need:

DivX ;-) Mpeg-4 Codec 3.1alpha (or greater)

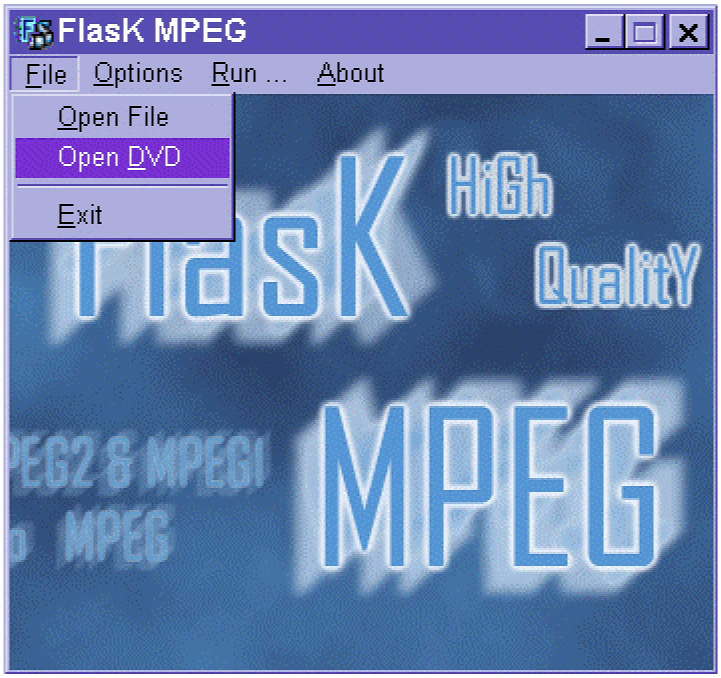
Fhg Radium MP3 codec

Advanced Bitrate Calculator 1.8

Flask Mpeg 0.594 (or greater)

If you haven't installed the Radium MP3 and the DivX codec's you may as well do so now. Remember to run the "run me first" option on the DivX install too or it will not be installed correctly. Don't use the crappy AngelPotion codec, its a bad Mpeg-4 hack, yes, thats right! A hack and not a self made Mpeg-4 codec from the specifications!

As always, you must have the DVD files ripped and decoded to your Hard Disk first if you want to convert a DVD. You can use CladDVD or VobDec to do this but I prefer SmartRipper. They are all legal (because they do not use stolen DVD codes) so there is nothing to worry about. Flask has two basic conversion modes:



Open File: This will open any mpeg file and try to convert it to whatever you want.

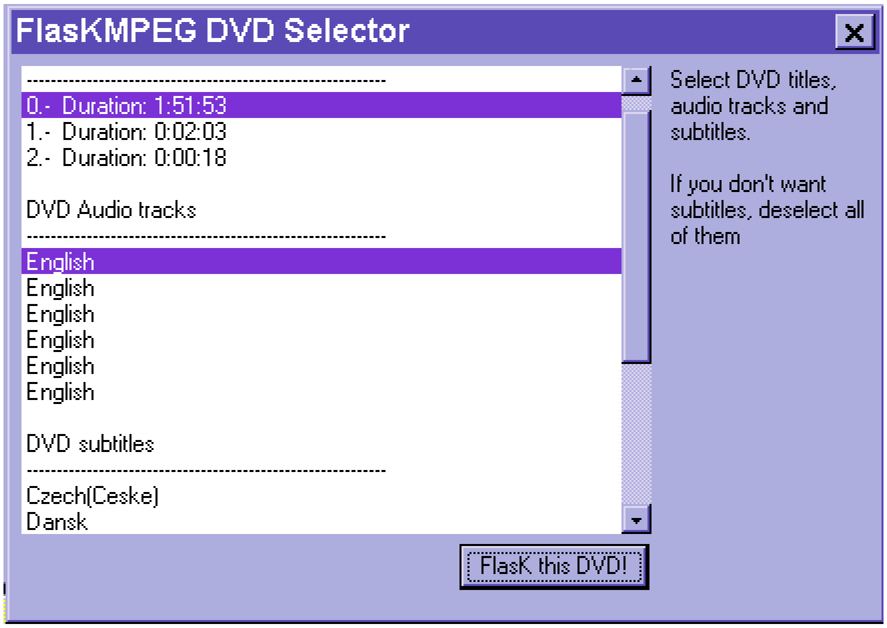
Open DVD: This is a special mode that reads a DVD in the same way a DVD player would. This means you will be able to select to convert with subtitles; it will also make sure it only rips one angle in a multi-angle DVD. This prevents repeated scenes spoiling your movie!

If you want to use Flask's 'Open DVD' mode you must copy the files exactly as they are on the DVD. This means you must rip them with SmartRippers file mode, do *not* use its movie mode. With VobDec or CladDVD you must turn off the multi-angle functions first - DeCSS is old and cannot do all DVD anymore, so don't use it!

Also, if you are using Flask's 'Open DVD' mode you must copy the IFO file that belongs to the movie you wish to convert. This will be called by the same name as the first movie file. For example, if the first movie file is called **VTS_01_0.VOB** the IFO file to copy with it will be **VTS_01_0.IFO**. Remember, the main movie will not always have the same name. So if first file of the main movie were called **VTS_02_0.VOB** then the IFO file to copy with it will obviously also be **VTS_02_0.IFO** For more details on this subject read the article called "*Introduction to the Structure of a DVD*" in the appendix section.

OPEN THE FILE WITH 'Open DVD'

Select 'Open DVD' and find the .IFO file for your movie. Up will pop something like the picture below:



FlasKMPEG DVD Selector

0.- Duration: 1:51:53
1.- Duration: 0:02:03
2.- Duration: 0:00:18

DVD Audio tracks

English
English
English
English
English
English

DVD subtitles

Czech(Ceske)
Dansk

Select DVD titles, audio tracks and subtitles.

If you don't want subtitles, deselect all of them

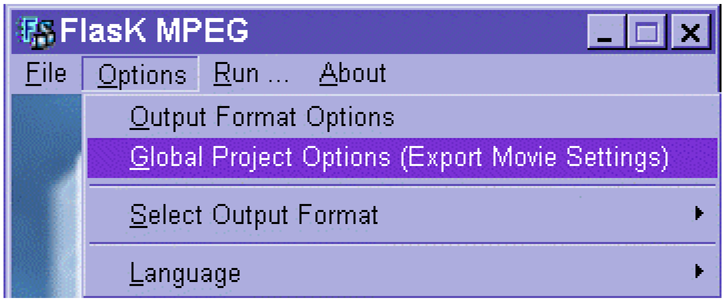
Select the movie Duration, in this case 1.51.53. This will usually be the first one in the list, but you can usually see from the length which one to choose.

Next choose the language. Obviously they cannot ALL be English so choose the first and encode a minuet of the film and listen to it. If its not English choose the next in the list, and the next and the next etc., until you find the correct one. Or open the DVD in a DVD player and see the order they are in, usually it will be the same.

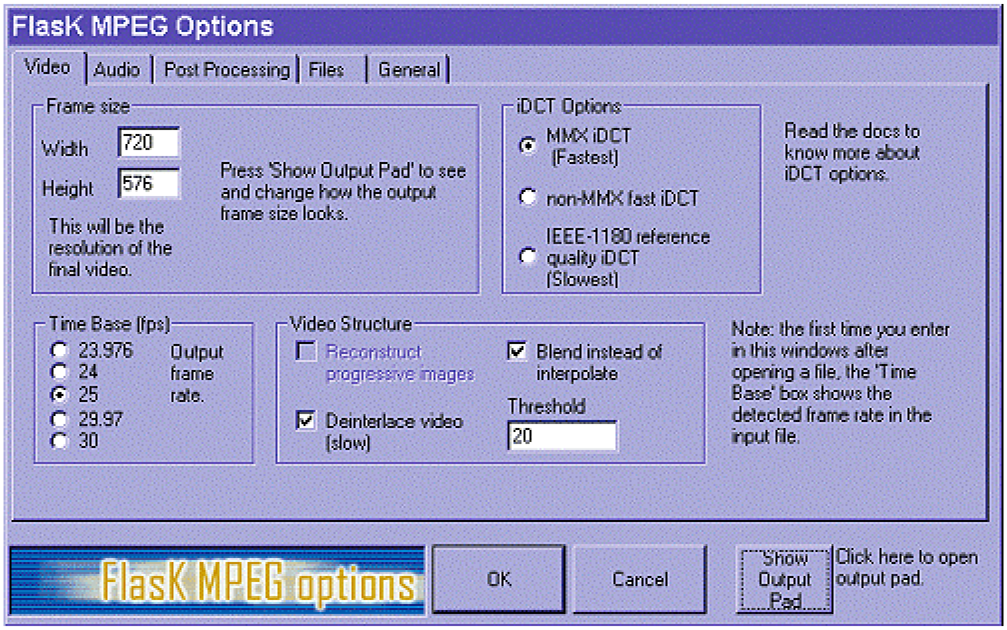
Lastly we have subtitles. I don't usually select any because I don't want them on my movie. If you choose subtitles then you will not be able to turn them off, they will ALWAYS be on your movie! Flask is not always able to do subtitles correctly, so again, you will have to try it and see. There are other ways to get subtitles, though; why not check out my subtitle ripping section for this.

Now Press FlasK this DVD!

GLOBAL PROJECT OPTIONS Select Global Project options.



VIDEO TAB



Frame size

Width 720

Height 576

This will be the resolution of the final video.

Press 'Show Output Pad' to see and change how the output frame size looks.

iDCT Options

● MMX iDCT (Fastest)

○ non-MMX fast iDCT

○ IEEE-1180 reference quality iDCT (Slowest)

Read the docs to know more about iDCT options.

Time Base (fps)

○ 23.976
○ 24
● 25
○ 29.97
○ 30

Output frame rate.

Video Structure

□ Reconstruct progressive images

☑ Deinterlace video (slow)

☑ Blend instead of interpolate

Threshold 20

Note: the first time you enter in this windows after opening a file, the 'Time Base' box shows the detected frame rate in the input file.

Click here to open output pad.

Frame Size: In the Width and Height sections you can put the size you wish your final video to be. If you wish to crop your movie too you can select the 'Show Output Pad' at the bottom right. For detailed information on how to resize a movie in Flask read section 2nd of this guide: *"resizing the video"*.

Time Base (fps): Flask will normally choose the best framerate for your movie. All PAL movies (European) are 25 frames per second (fps). So if you know your movie is PAL make sure your movie is set to 25. All North American movies are NTSC which means they are 29.97 fps. BUT because of the way they are encoded to DVD most will appear Flask will choose 23.976 fps. This is usually correct so don't change it. As always test a short clip before you do a whole movie to make sure its ok.

iDCT Options: Just leave these alone its overkill to use IEEE and the quality will not look better.

Video Structure: Do *not* check the deinterlace button unless you really need it because the video will decode very slow. For a full explanation of the interlace problem, check out my article: "Video Formats: NTSC & PAL / Telecine" in the appendix section.

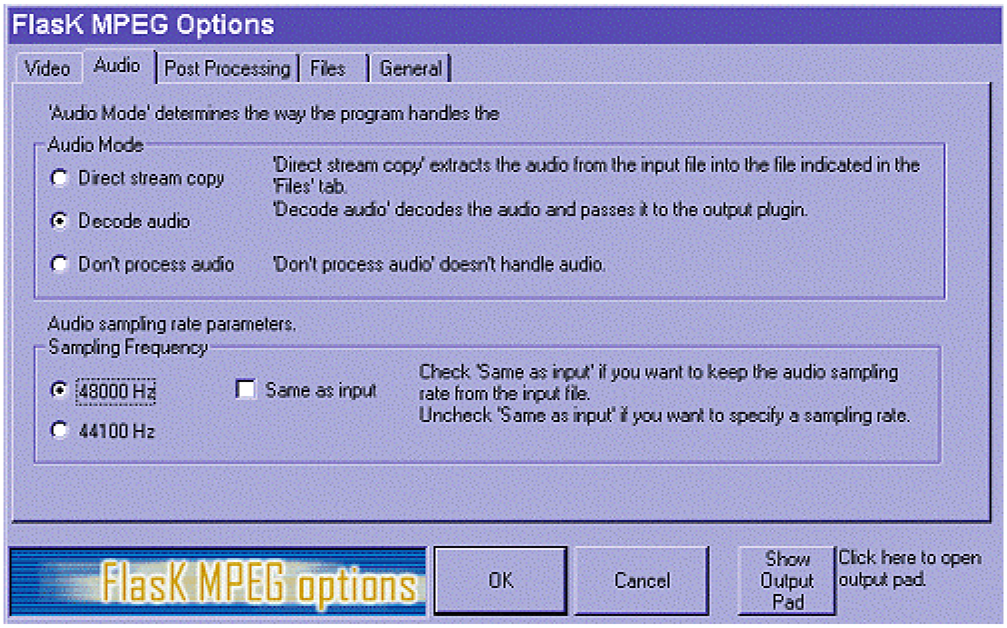
> **Deinterlace video**
>
> For this kind of Deinterlacer blend gives best quality. The Threshold setting is basically how much it blurs problem lines. The lower the number the more it blurs it. If the interlace problem is really bad you should use a setting of 1-5 instead of 20. If its not very bad you could use 15-20.

If Flask (or you) select the Time Base of 24 or 23.976 fps then there will be the extra option called 'Reconstruct progressive images'. This option is grayed out in the above picture. Flask automatically tries to detect the real framerate inside the DVD. This setting is not an IVTC (inverse telecine); it just forces Flask Mpeg to 24fps progressive mode, avoiding potential interlace problems if the movie is 24 but the playback is set to 29.976.

So you want to know what you should do with them, right?! For PAL movies, I have found that there is no point selecting anything but 25fps and you will *not* need 'reconstruct progressive images'. In fact, you will almost always get jerky playback if you select it. For NTSC I'm not sure, I think the general opinion is to let Flask choose or ignore it.

AUDIO TAB

On to the Audio tab always select 'Decode audio' if you want sound. For DVD's un-check the 'same as input box' and select 48000 Hz (just to make sure you have the right setting). For Mpeg-1 you'd use same as input or 44100Hz. Never use 44100Hz with DVD's or you will get audio synchronization problems.



'Audio Mode' determines the way the program handles the

Audio Mode

○ Direct stream copy

● Decode audio

○ Don't process audio

'Direct stream copy' extracts the audio from the input file into the file indicated in the 'Files' tab.
'Decode audio' decodes the audio and passes it to the output plugin.

'Don't process audio' doesn't handle audio.

Audio sampling rate parameters.

Sampling Frequency

● 48000 Hz

○ 44100 Hz

□ Same as input

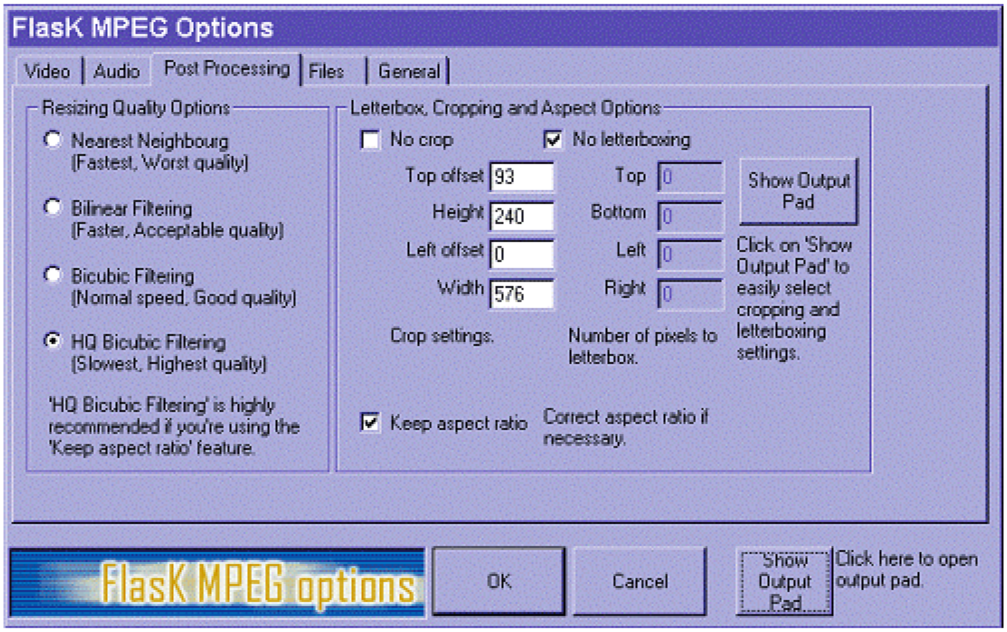
Check 'Same as input' if you want to keep the audio sampling rate from the input file.
Uncheck 'Same as input' if you want to specify a sampling rate.

Click here to open output pad.

If you just want to copy the DVD's AC3 audio across instead of converting it use the 'Direct Stream copy' option.

POST PROCESSING TAB

This section deals almost exclusively with resizing. Never use 'Nearest Neighbouring' unless you are not resizing the picture because the quality is crud. Contrary to popular belief Bilinear looks just as nice as Bicubic, except it is twice as fast. JASC (makers of Paint Shop Pro) recommend Bilinear for shrinking images and Bicubic for enlarging them. But use what you think looks best.



Resizing Quality Options

○ Nearest Neighbourg
(Fastest, Worst quality)

○ Bilinear Filtering
(Faster, Acceptable quality)

○ Bicubic Filtering
(Normal speed, Good quality)

● HQ Bicubic Filtering
(Slowest, Highest quality)

'HQ Bicubic Filtering' is highly recommended if you're using the 'Keep aspect ratio' feature.

Letterbox, Cropping and Aspect Options

□ No crop

☑ No letterboxing

Top offset 93

Height 240

Left offset 0

Width 576

Crop settings.

Top 0

Bottom 0

Left 0

Right 0

Number of pixels to letterbox.

Click on 'Show Output Pad' to easily select cropping and letterboxing settings.

☑ Keep aspect ratio

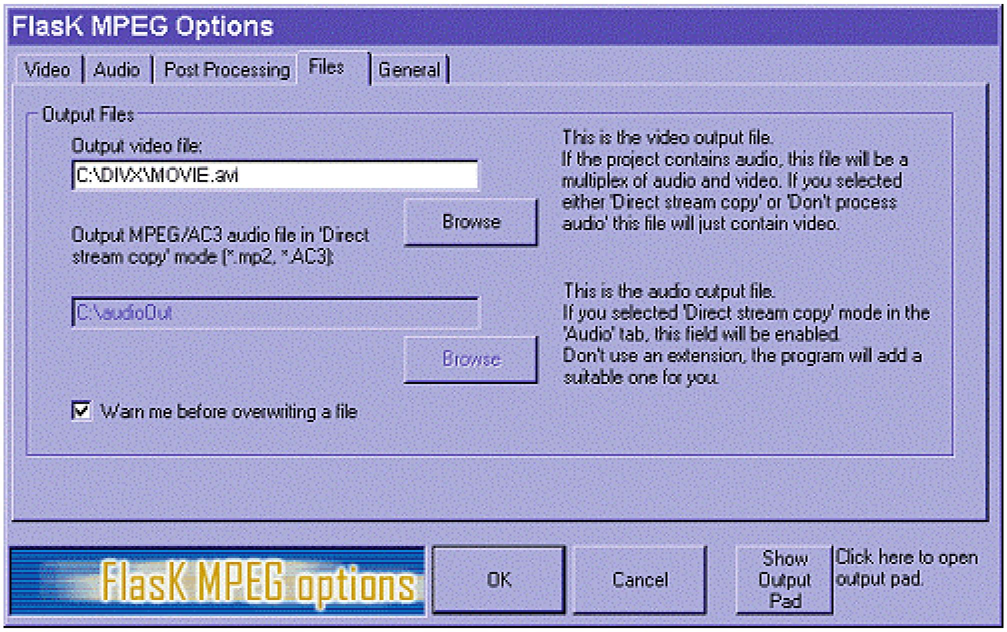
Correct aspect ratio if necessary.

Click here to open output pad.

Keep aspect ratio: PAL users should always tick this box unless you know you don't need it. This is even more important with Widescreen DVD's. If you use NTSC DVD's the image may become stretched slightly wrongly. If you notice this uncheck the 'keep aspect ratio' and work out the ratio yourself. See the article *"Resizing DVD's with Correct Aspect Ratios"* in my appendix.

Crop & Letterboxing: All the settings for cropping and letterboxing the DVD can be entered here or the output pad can be used. For detailed information on how to resize a movie in Flask read section 2nd of this guide: *"resizing the video"*.

FILES TAB



Output video file: C:\DIVX\MOVIE.avi

Output MPEG/AC3 audio file in 'Direct stream copy' mode (*.mp2, *.AC3): C:\audioOut

☑ Warn me before overwriting a file

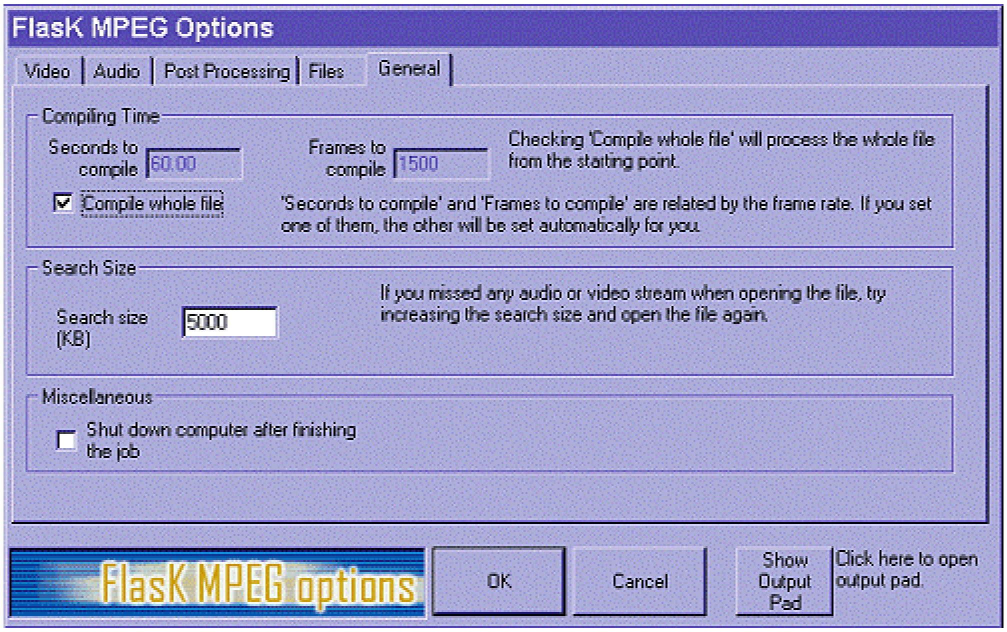
This is the video output file.
If the project contains audio, this file will be a multiplex of audio and video. If you selected either 'Direct stream copy' or 'Don't process audio' this file will just contain video.

This is the audio output file.
If you selected 'Direct stream copy' mode in the 'Audio' tab, this field will be enabled.
Don't use an extension, the program will add a suitable one for you.

Click here to open output pad.

Choose where you want to save your final movie. The audio save option is grayed out because you are encoding the video with audio in it. If you selected 'Direct Stream Copy' on the previous Audio Tab then you could say where you wanted it saved.

GENERAL TAB



Compiling Time

Seconds to compile 60.00

Frames to compile 1500

Checking 'Compile whole file' will process the whole file from the starting point.

☑ Compile whole file

'Seconds to compile' and 'Frames to compile' are related by the frame rate. If you set one of them, the other will be set automatically for you.

Search Size

Search size (KB) 5000

If you missed any audio or video stream when opening the file, try increasing the search size and open the file again.

Miscellaneous

☐ Shut down computer after finishing the job

Click here to open output pad.

Compiling Time: This speaks for itself. If you check the 'Compile whole file' box it will convert the whole DVD. If you uncheck it you can say how many seconds to encode. Obviously 60 is one minuet and 120 is two mins. The frames to compile is basically the same thing. There are 25 frames to one second of video for PAL and either 23.976 or 29.970 for NTSC. Using these options you can encode a small clip to test the quality etc. I recommend you shut down and restart Flask just before you encode a long movie to reduce chances of it crashing.

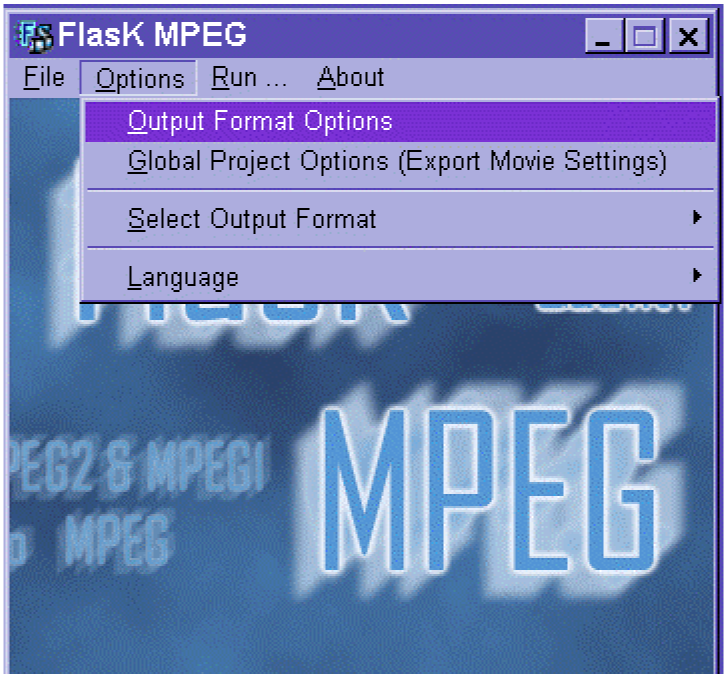
Search Size: This searches the DVD for the audio. As it says, if you have problems make the number bigger. If all else fails and you cannot find audio you can try one of the other methods I explain in the section:*"DVD Audio Extraction"*.

Miscellaneous: Like the thing says, it'll shut down the computer if it can when its done :)

That's it you're nearly ready to encode! Lets set the output options...

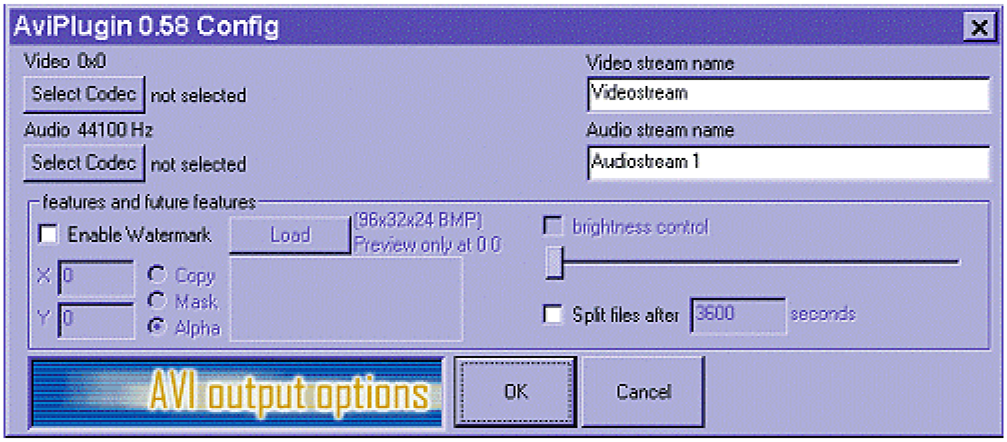
OUTPUT FORMAT OPTIONS

Go to Options > Output Format Options.



And up will pop the box shown below:

Features and Future Features: Flask has an option to 'Split files after' a certain amount of seconds. This is useful if you want a VCD to be split in half so that one part can be put on a single CD. I don't use it for Divx because I prefer to split them in VirtualDub. To use it, just work out how many seconds there are in the movie. If the movie is 131 minuets then multiply 131 by 60 and you have the seconds (7860 seconds) then just divide that in half and you have your split number (3930).


AviPlugin 0.58 Config

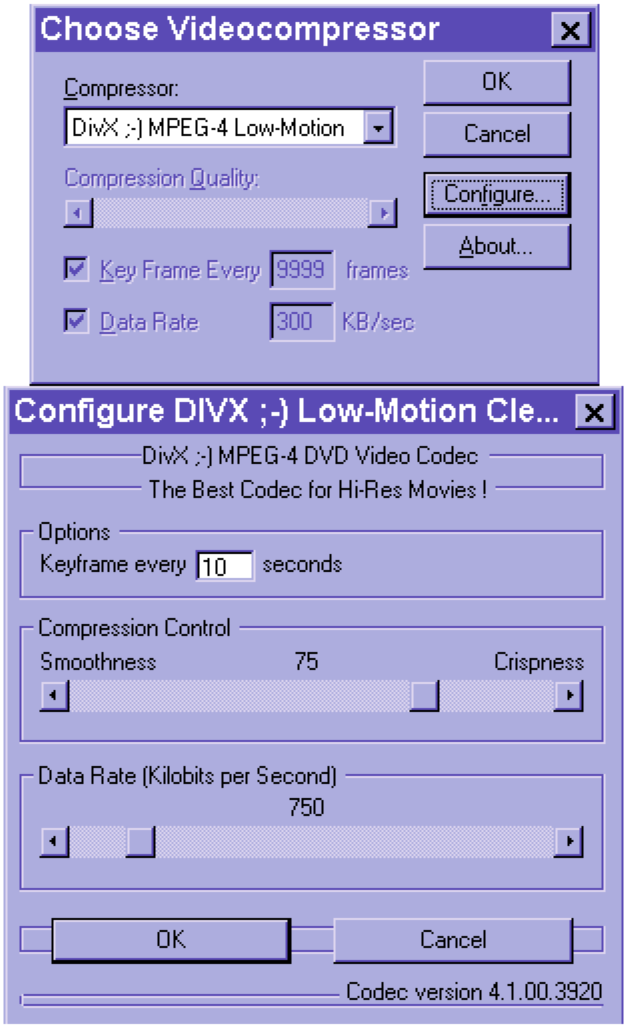
Video 0x0
Select Codec not selected
Audio 44100 Hz
Select Codec not selected

Video stream name: Videostream
Audio stream name: Audiostream 1

features and future features
□ Enable Watermark Load (96x32x24 BMP) Preview only at 0.0
X 0
Y 0
○ Copy
○ Mask
● Alpha
□ brightness control
□ Split files after 3600 seconds

Enable Watermark: This lets you use any bitmap picture you want as a logo. So you can make a little picture saying made by 'smiffy' or whatever you like. The X and Y boxes let you say where the logo will appear on the movie. Then you have three options on how it will look. It can "Copy", which just sticks the bitmap on top. It can "Mask" which lets part of the image become transparent. Or it can "Alpha" which makes the whole image transparent. I do not recommend using logos because they increase the filesize of the picture and degrade the quality.

Select Codec: These are the important buttons! Hit the top one and the following box appears:



Choose Videocompressor ☒

Compressor:
DivX ;-) MPEG-4 Low-Motion

Compression Quality:

☑ Key Frame Every 9999 frames

☑ Data Rate 300 KB/sec

Configure DIVX ;-) Low-Motion Cle... ☒

DivX ;-) MPEG-4 DVD Video Codec
The Best Codec for Hi-Res Movies !

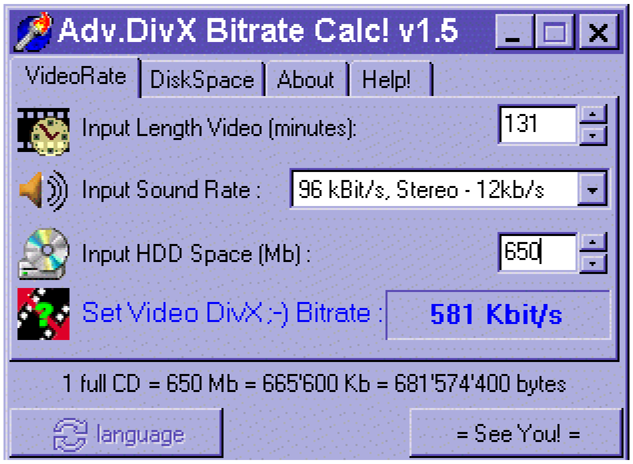
Options
Keyframe every 10 seconds

Compression Control
Smoothness 75 Crispness

Data Rate (Kilobits per Second)
750

Codec version 4.1.00.3920

Choose the Divx ;-) MPEG-4' Low or High Motion codec from the drop down list depending on what you want. For getting the best setting you can look at my article: "Best Divx Quality & Bitrate Guide" in the appendix. But in short, use a bitrate calculator to determine the best bitrate to put in here. I'm using the 'Advanced Divx Bitrate Calc! Version 1.5'. It seems to give a more reliable amount and will not make a movie larger than you intend.



Input Length Video (minutes): 131

Input Sound Rate : 96 kBit/s, Stereo - 12kb/s

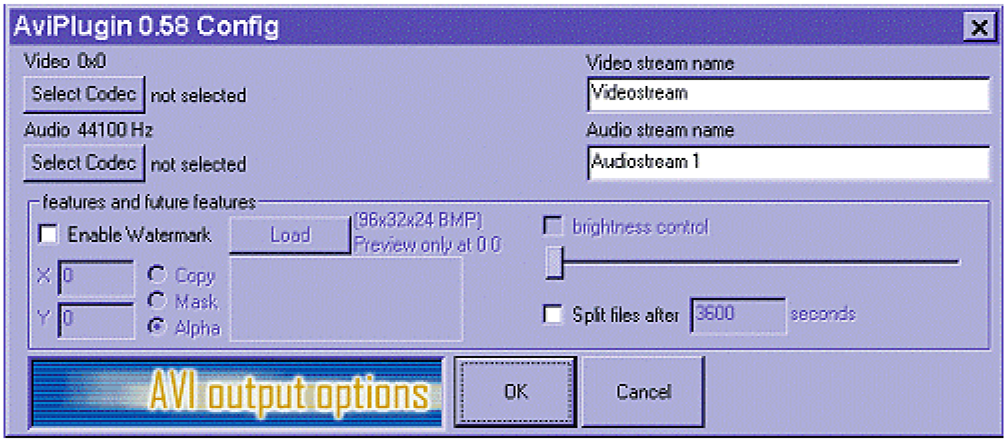
Input HDD Space (Mb) : 650

Set Video DivX ;-) Bitrate : 581 Kbit/s

1 full CD = 650 Mb = 665'600 Kb = 681'574'400 bytes

Just put the DVD movie length in minuets in the top box. Select the audio bitrate you want to use (for Flask this will almost always be 96 kBit/s Stereo). Then it will tell you the amount to use! In the above picture the calculator says 581 so you would move the 'Data Rate slider bar to 581 instead of the 750 it is in the above picture. As for keyframes, I tend to use a keyframe every 1 second, but you can use one every 10 seconds to saves more memory. Then set the 'Crispiness to 75% and you are done!

Ok. Back to our previous dialogue box. Press the bottom 'Select Codec' button to select the audio compression.



Video 0x0

not selected

Audio 44100 Hz

not selected

Video stream name: Videostream

Audio stream name: Audiostream 1

□ Enable Watermark

(96x32x24 BMP) Preview only at 0,0

X 0

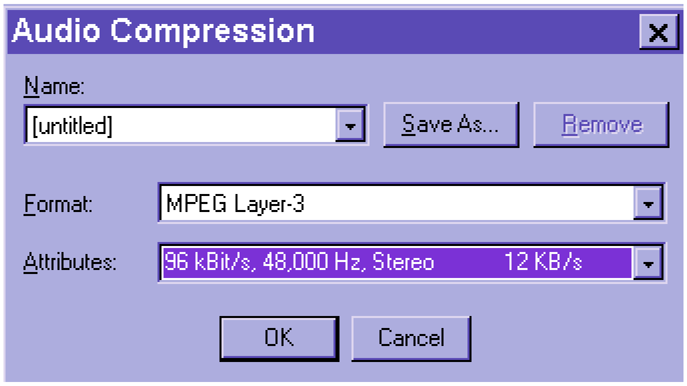
Y 0

○ Copy ○ Mask ● Alpha

□ brightness control

□ Split files after 3600 seconds

Up pops this box:



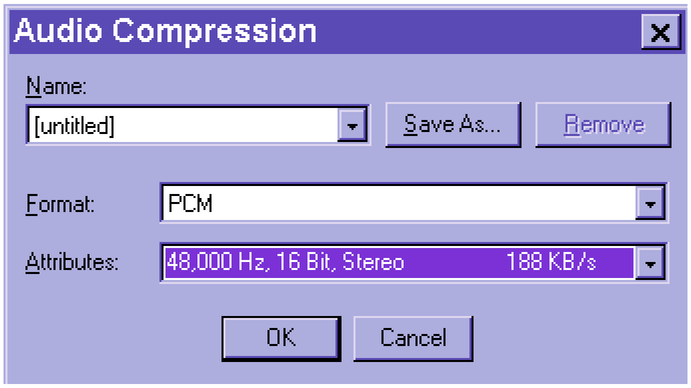
Name: [untitled]

Format: MPEG Layer-3

Attributes: 96 kBit/s, 48,000 Hz, Stereo 12 KB/s

Choose 'MPEG Layer-3' at 96 kBit/s 48,000Hz. You can choose what you like here, 128 or 190 etc., if you prefer, but there is no need it will just make the file size larger.

There are two problems with converting the audio to Mp3. First, the audio must be 48,000Hz. Some sound cards cannot handle the format and so it may play the audio back dull and buzzing. Secondly, because of the dynamic range of a DVD the audio level will sound very low and may not be loud enough for you. Check out the section *"Dull Audio, Normalization & Samplerates"* for a detailed explanation and a solution to all this. In short, you can choose PCM audio instead of mp3 and add it later. So here is where you would do that. Instead of choosing Mp3 choose PCM 48,000Hz 16bit, Stereo. *No* other audio format will work in Flask without potential problems!



Name: [untitled]

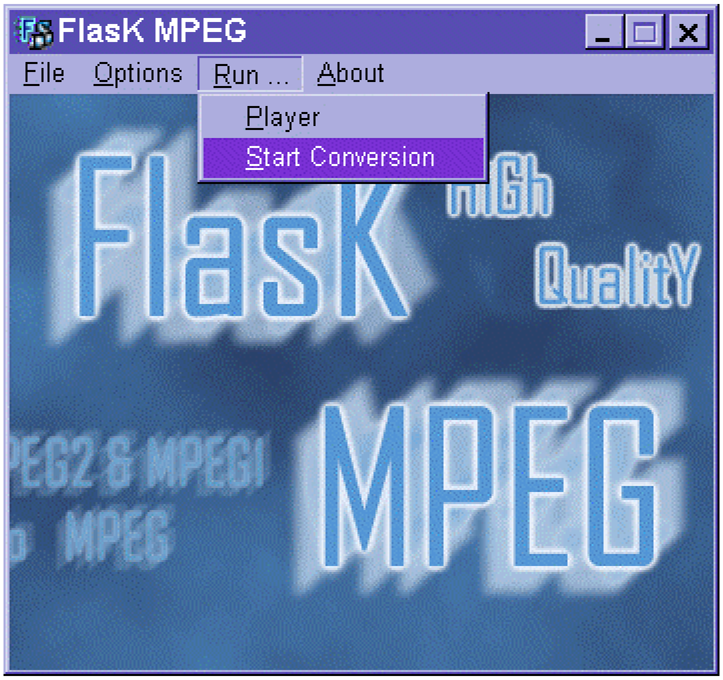
Format: PCM

Attributes: 48,000 Hz, 16 Bit, Stereo 188 KB/s

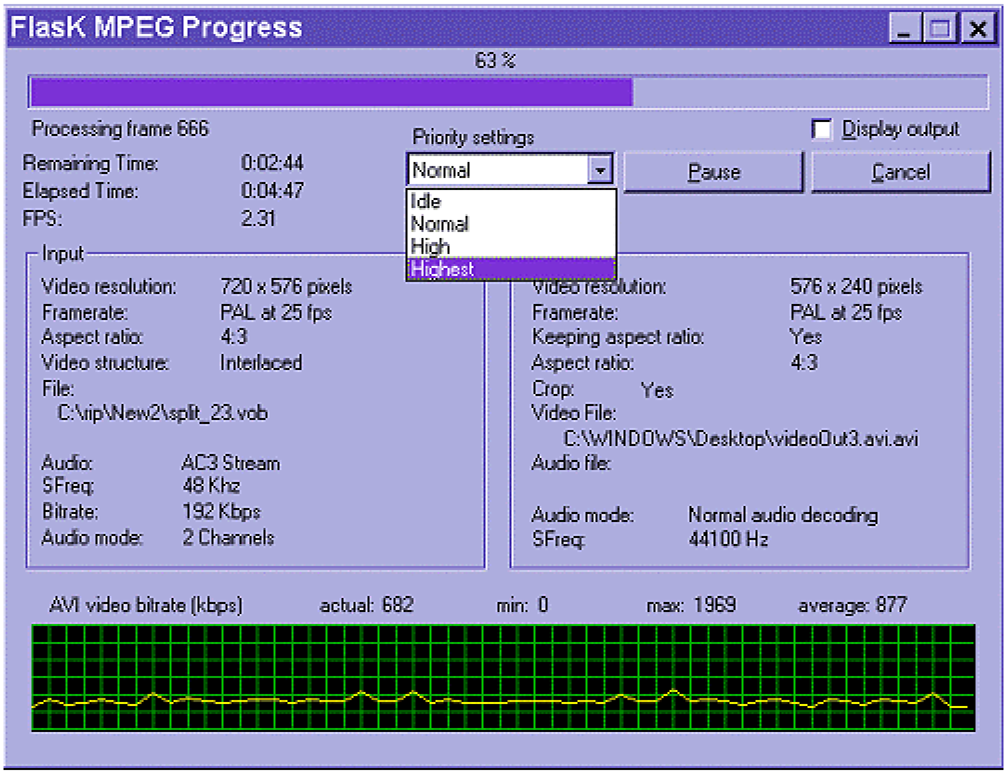
Warning! PCM Wav audio is uncompressed and creates a seriously large movie file 1-2GB! If you are making a single CD rip you will probably be okay using PCM audio. If you are making a 2 CD rip, your file may be so large that VirtualDub will not open it and Windows will not play it! This is annoying because Divx Mpeg4 can be larger than 2GB and would normally probably work okay at anything upto 4GB. But Flask seems to create a file that is corrupted at these sizes! So, the PCM Wav is an advanced option that should be treated with great respect! If you are a beginner just use Mp3!

PRESS GO!

Once you have finished setting up your movie just go to Run > Start Conversion.



Up will pop the following screen.



63 %

Processing frame 666

Remaining Time: 0:02:44
Elapsed Time: 0:04:47
FPS: 2.31

Priority settings: Normal

Input

Video resolution: 720 x 576 pixels
Framerate: PAL at 25 fps
Aspect ratio: 4:3
Video structure: Interlaced
File:
C:\rip\New2\split_23.vob

Audio: AC3 Stream
SFreq: 48 Khz
Bitrate: 192 Kbps
Audio mode: 2 Channels

Video resolution: 576 x 240 pixels
Framerate: PAL at 25 fps
Keeping aspect ratio: Yes
Aspect ratio: 4:3
Crop: Yes
Video File:
C:\WINDOWS\Desktop\videoOut3.avi.avi
Audio file:

Audio mode: Normal audio decoding
SFreq: 44100 Hz

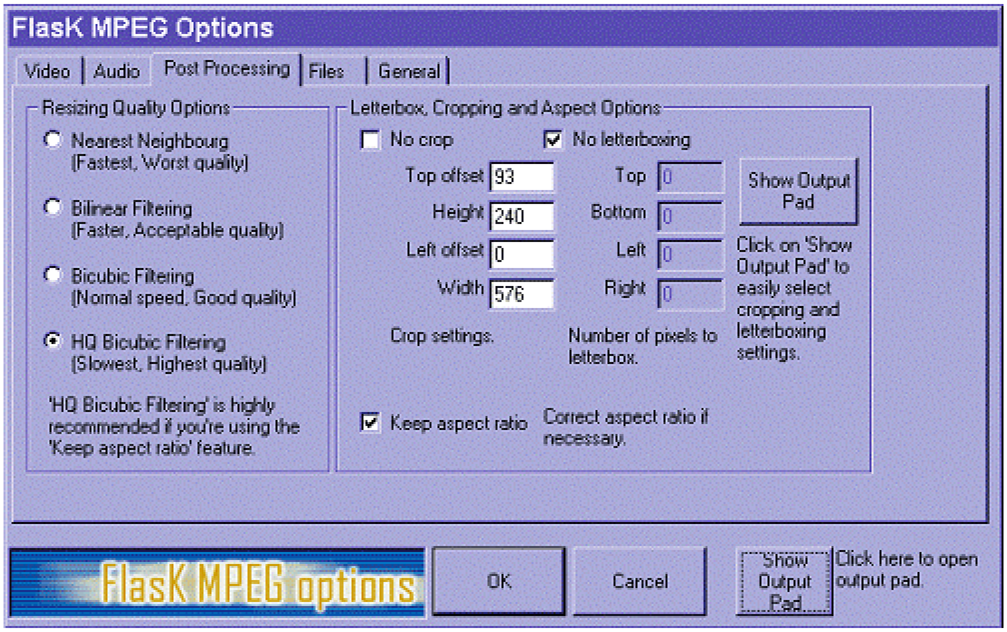
AVI video bitrate (kbps) actual: 682 min: 0 max: 1969 average: 877

If you are going to leave your computer alone to compress then select Highest in the 'Priority settings'. This will hog ALL CPU power to make compression faster. This CPU option only really works if other programs are running when Flask is though! When using idle or normal, you will be able to do basic tasks while it is compressing, such as checking Email etc. Be careful not to do too much because if your computer crashes because of another program then you'll need to start over again. Unchecking the 'Display output' box will also help speed things up a little. But if you are using a Dual Processor computer system just leave it on normal otherwise it slows it down.

That's it! In 15-20 hours on a 500Mhz machine you will have a perfect Divx movie ;).

Resizing and Cropping a DivX in Flask Mpeg

If you want to fit a movie into the smallest possible size, cropping can do a lot to help out. The most vital thing to remember for optimal compression is to always crop a few pixels into the image and delete all of the black bars. Mpeg compression works best on blurry images; so if a hard black line is seen at the bottom or top of your cropped movie it will not compress as effectively. In fact, if you cannot crop into the image then I'd say don't bother cropping at all.



Resizing Quality Options

○ Nearest Neighbourg (Fastest, Worst quality)

○ Bilinear Filtering (Faster, Acceptable quality)

○ Bicubic Filtering (Normal speed, Good quality)

● HQ Bicubic Filtering (Slowest, Highest quality)

'HQ Bicubic Filtering' is highly recommended if you're using the 'Keep aspect ratio' feature.

Letterbox, Cropping and Aspect Options

□ No crop ☑ No letterboxing

Top offset	93	Top	0
Height	240	Bottom	0
Left offset	0	Left	0
Width	576	Right	0

Crop settings. Number of pixels to letterbox.

Click on 'Show Output Pad' to easily select cropping and letterboxing settings.

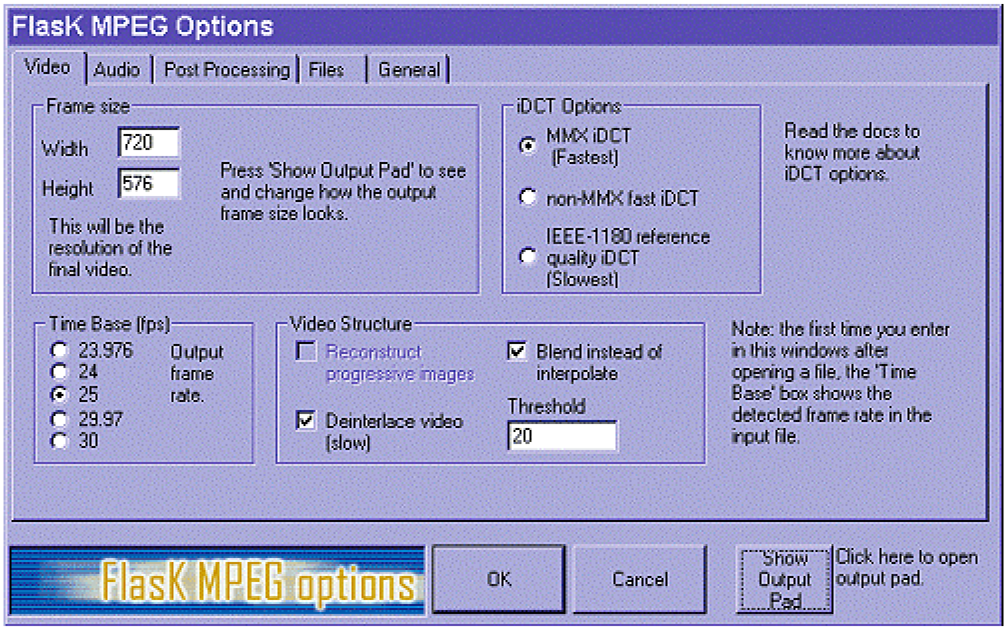
☑ Keep aspect ratio Correct aspect ratio if necessary.

Click here to open output pad.

Resizing a DVD in Flask is slightly easier than Mpeg2avi because it shows you what you are doing. PAL users should always tick Flask's Keep aspect ratio box, unless you know you don't need it. This is even more important with Widescreen DVD's. If you use NTSC DVD's the image may become stretched wrongly. If you notice this happening, uncheck the 'keep aspect ratio' and work out the ratio resizing yourself. See the article"Resizing DVD's with Correct Aspect Ratios" in my appendix for details.

RESIZING A 4:3 RATIO DVD

This requires no effort at all to do since they are basically square. Simply choose the size you prefer and put them in the Width and Height boxes in the video tab. Something like 480 x 384 works well.



Frame size

Width 720

Height 576

This will be the resolution of the final video.

Press 'Show Output Pad' to see and change how the output frame size looks.

iDCT Options

● MMX iDCT (Fastest)

○ non-MMX fast iDCT

○ IEEE-1180 reference quality iDCT (Slowest)

Read the docs to know more about iDCT options.

Time Base (fps)

○ 23.976
○ 24
● 25
○ 29.97
○ 30

Output frame rate.

Video Structure

□ Reconstruct progressive images

☑ Deinterlace video (slow)

☑ Blend instead of interpolate

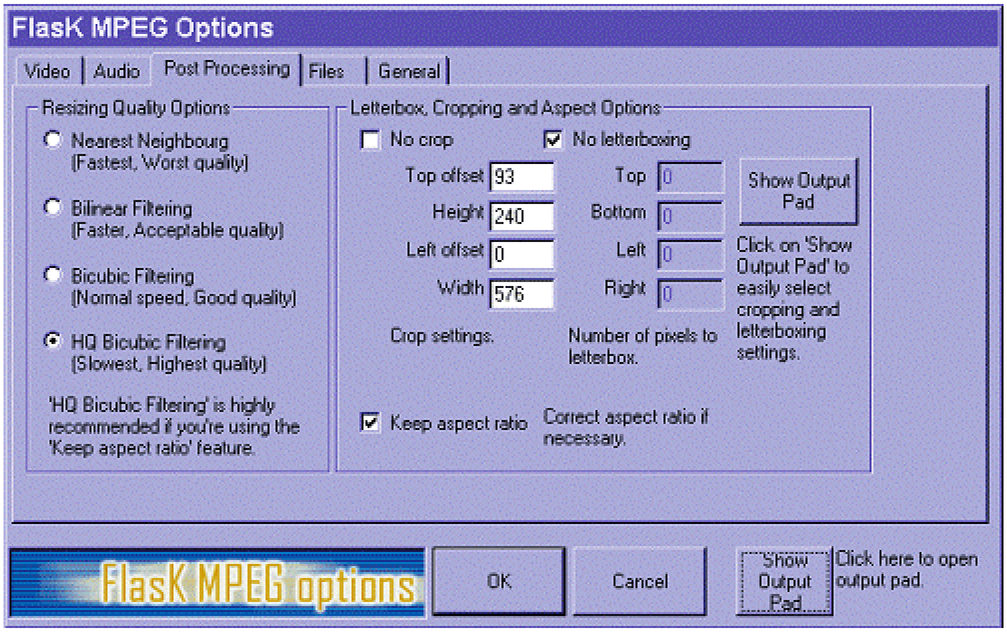
Threshold 20

Note: the first time you enter in this windows after opening a file, the 'Time Base' box shows the detected frame rate in the input file.

Click here to open output pad.

RESIZING AN ANAMORPHIC DVD

To start resizing press the 'Show Output Pad'. This will bring up a picture of the movie with controls above it that allow you to change its size and crop out black areas.



Resizing Quality Options

○ Nearest Neighbourg (Fastest, Worst quality)

○ Bilinear Filtering (Faster, Acceptable quality)

○ Bicubic Filtering (Normal speed, Good quality)

● HQ Bicubic Filtering (Slowest, Highest quality)

'HQ Bicubic Filtering' is highly recommended if you're using the 'Keep aspect ratio' feature.

Letterbox, Cropping and Aspect Options

□ No crop

☑ No letterboxing

Top offset 93

Height 240

Left offset 0

Width 576

Crop settings.

Top 0

Bottom 0

Left 0

Right 0

Number of pixels to letterbox.

Click on 'Show Output Pad' to easily select cropping and letterboxing settings.

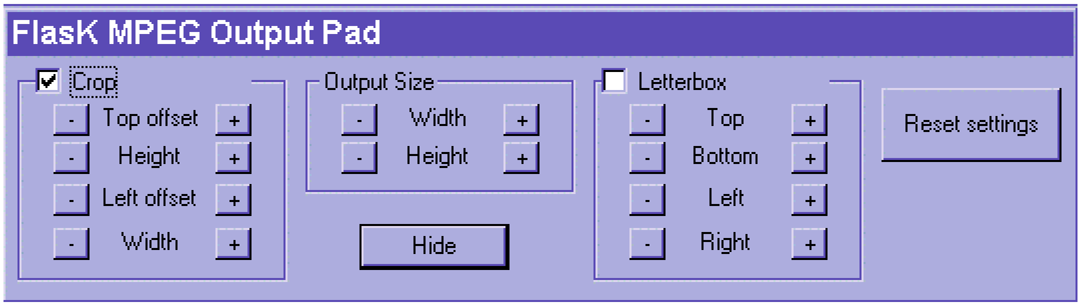
☑ Keep aspect ratio

Correct aspect ratio if necessary.

Click here to open output pad.

__Note:__ sometimes a black picture will appear. You have been unlucky here because Flask has chosen a picture for you to look at and it just happens to be a blank frame! You need to select a sinale vob file from the middle of the movie *instead of the .ifo file. This will change the picture it selects for you. Once you have the size correct reselect the .ifo file again.*

To summarize what the Output Pad controls do:



Crop - Height & Width

This setting represents an imaginary window in which the picture sits. By either increasing or decreasing the height and width you change the size of this window. This does not affect the size of the actual picture inside the window.

Height

- Cuts bits from the bottom of the picture in jumps of 16.

+ Adds them back.

Width

- Cuts bits from the right of the picture in jumps of 16.

+ Adds them Back

Crop - Top & Left Offset's

These options move the picture inside the box up, down, left and right. They do not crop the image or change its size.

Top Offset

- Moves the picture down

+ Moves picture up

Left offset

- Moves the Picture Right

\+Moves the Picture Left

Output Size

This option is important because it changes the size of the actual image inside the window.

Width

- shrinks the picture by pulling it in from the right.

+ stretches the picture by pulling it out to the right.

Height

- squashes the image by pulling it up from the bottom

+ stretches the image by pulling it up from the bottom

Reset Settings

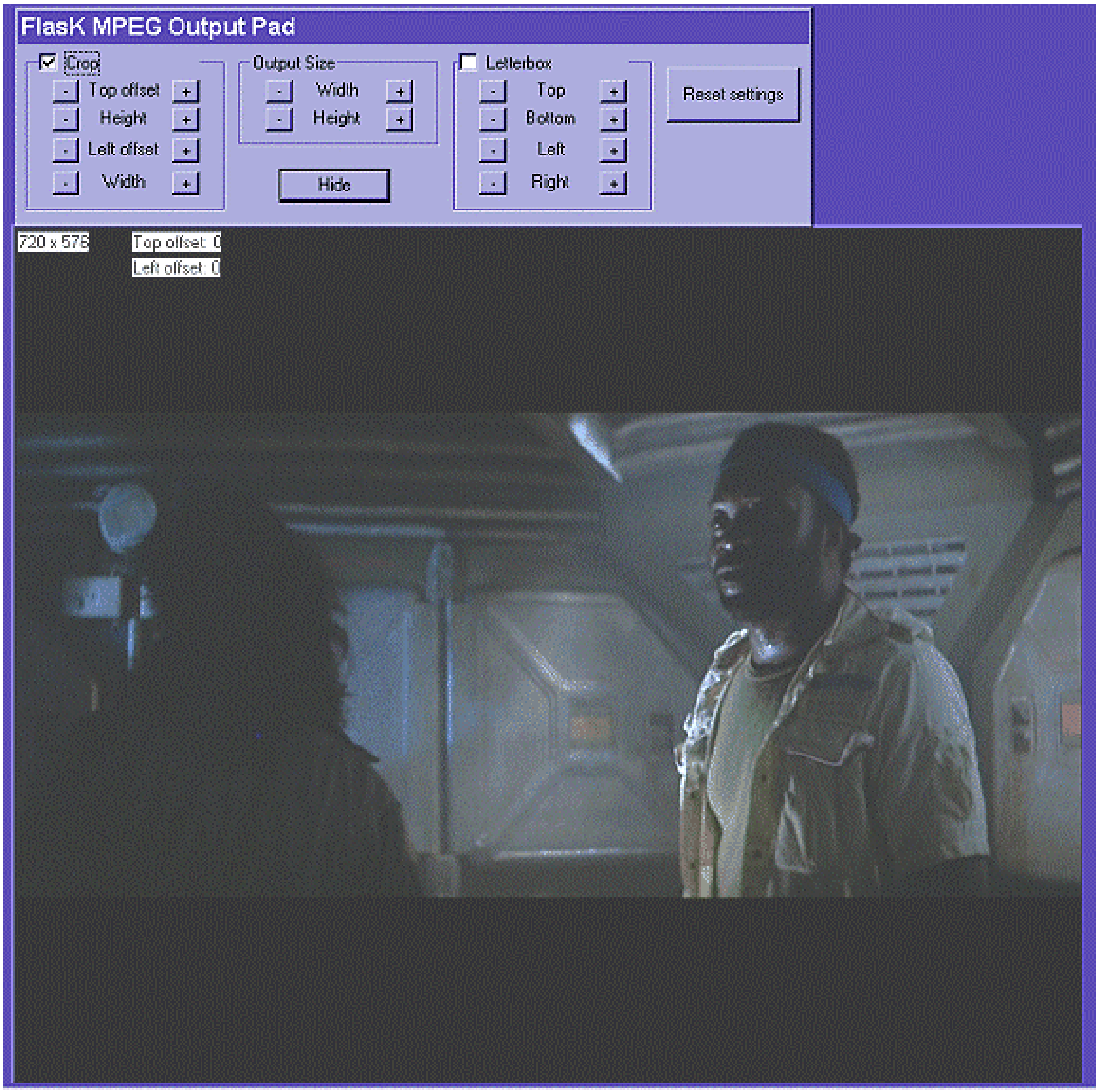
This puts the image back to FULL DVD size and is useful if you want to start again because you've the wrong sizes selected.

Letterbox

These are basically the same as crop but it resizes the whole image without cropping. Since I am cropping the image I do not use letterbox.

Okay, I have a 2.35:1 CinemaScope movie. I want not only to resize the picture but to crop out those back parts above and below the film. Doing this saves quite a bit of wasted memory and will allow you to use a higher bitrate and hence increase the quality of image playback. When played Fullscreen the cropped movie will still look the same because Media Player adds a black background anyway.

After you have clicked on the output pad button you will see a picture like the one below. Press the Reset settings button to make it appear as it is on the DVD. Bingo! This picture is far to big for a single CD DivX! Tick the 'crop' box and lets start resizing.



720 x 576

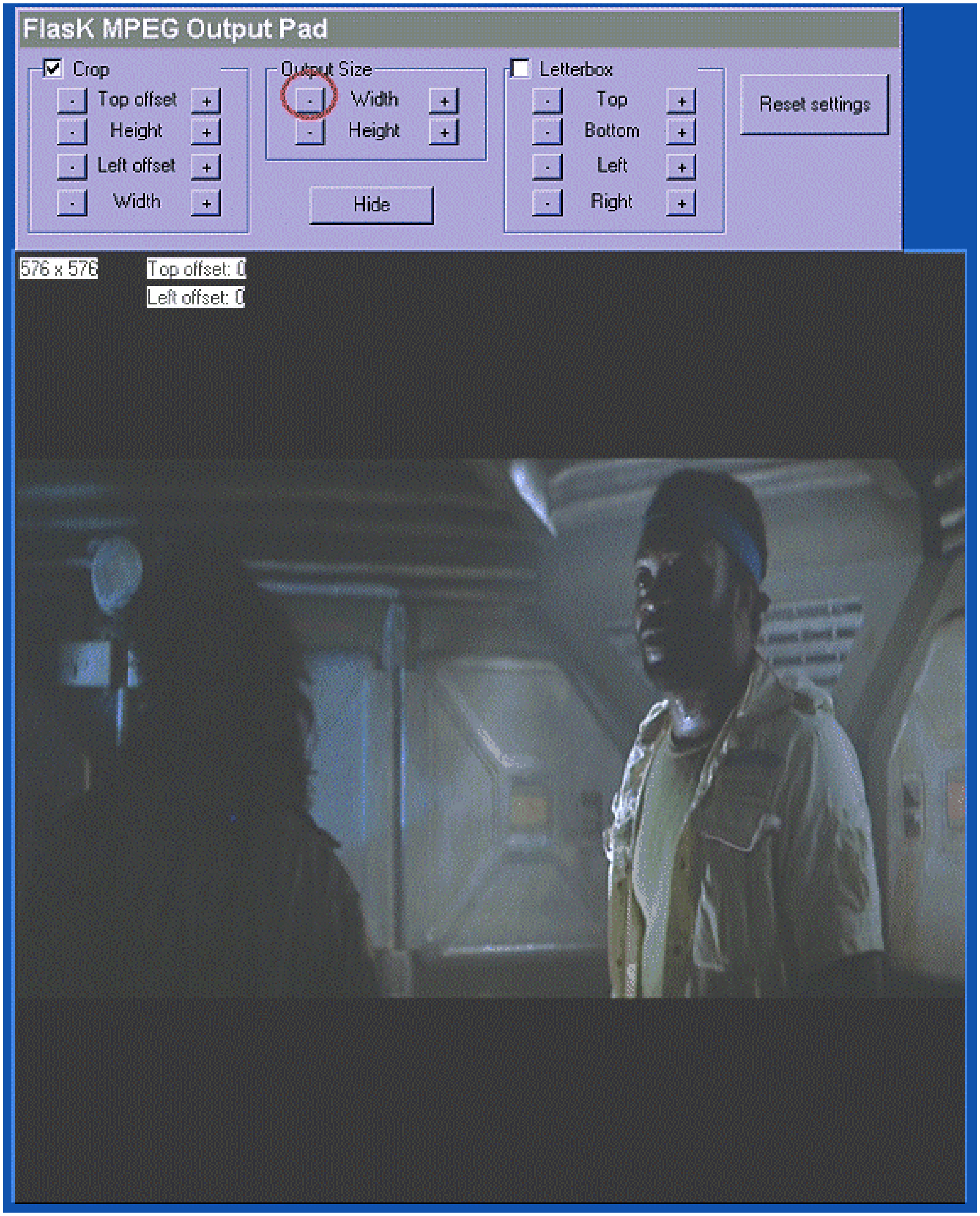
Top offset: 0

Left offset: 0

> ***Note:*** *Flask has the annoying habit of loosing parts of the picture! You will be editing it and suddenly the bottom will disappear or the whole picture will go white! This isn't actually a problem with the resizing itself, just with the preview it shows you and the DivX will be okay of course. But the easiest way to "refresh" the image and make it appear clear again is to open Windows Explorer. Maximize it once so it makes Flask disappear and then minimize it with the minimize button (don't do this by double clicking on the taskbar button). When Windows Explorer disappears and the Flask image will be as clear as a bell ;-). you need to do this quite often I'm afraid. Its also a good idea to make sure you have a light background on your desktop so you can see the edges of the picture.*

But before I crop I will squash the picture by pressing the minus (-) button in the output size section (see red circle in below picture). I pressed it 9 times until the picture was 576 wide. Notice how the size is displayed on the left of the picture. I always squash sideways first because usually the DVD will not need cropping on the sides.

Since we resized the image to the left 9 times, we will press the Height (-) button 9 times as well (just below the red circle). Flask does its best to keep in perfect aspect ratio, but if you are unsure check out my *"aspect ratio"* guide.

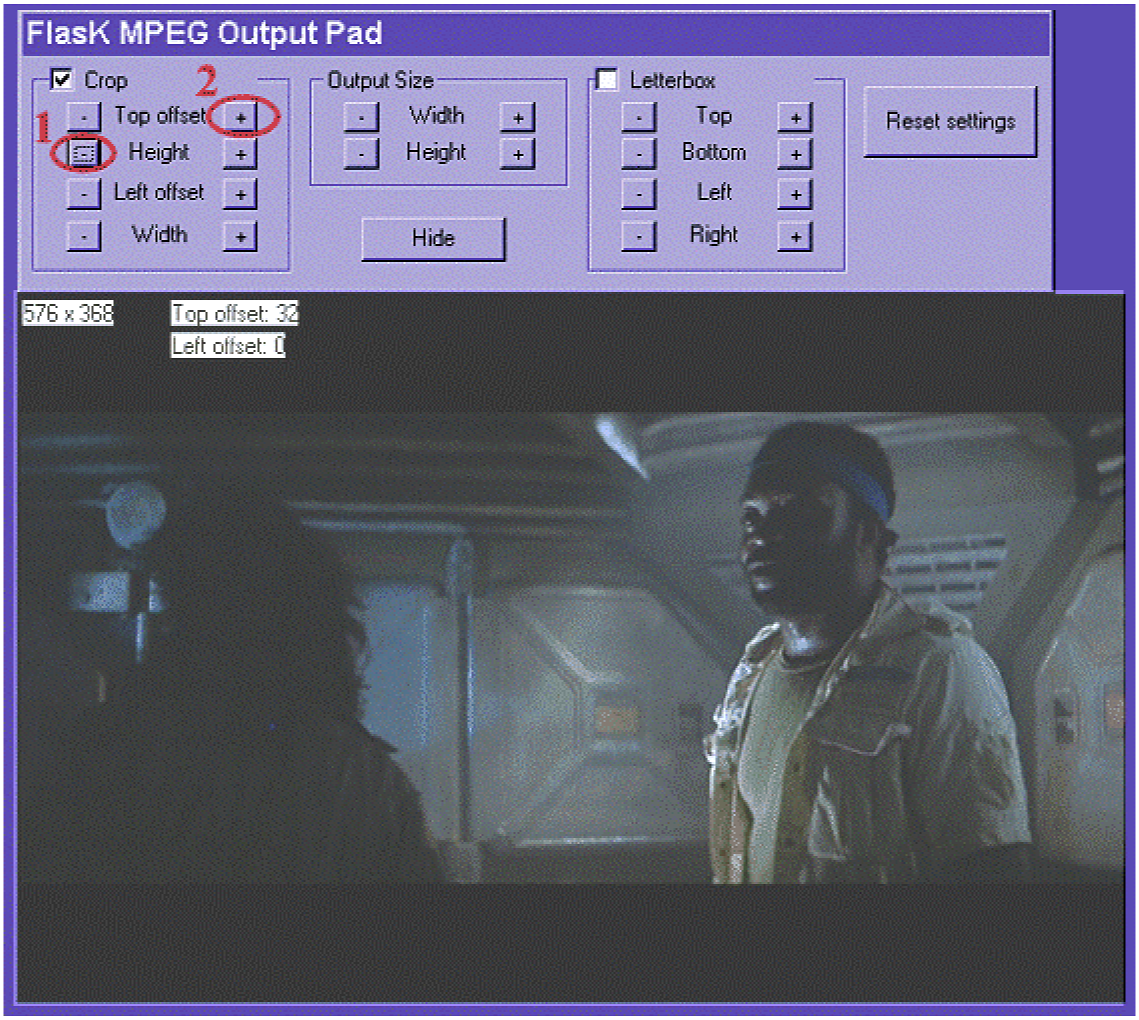


576 x 576

Top offset: 0

Left offset: 0

Okay, its not always possible to get the exact height correct, but you will still get it cropped at the best size for this particular DVD. Now we will start cropping bits off the bottom, do this a couple of times by pressing the red circle 1 (- Height). Then move the picture up until the border above disappears. Do this by pressing the red circle 2 a few times. Finally the bottom border will disappear (by using circle 1).


FlasK MPEG Output Pad

☑ Crop

576 x 368

Top offset: 32

Left offset: 0

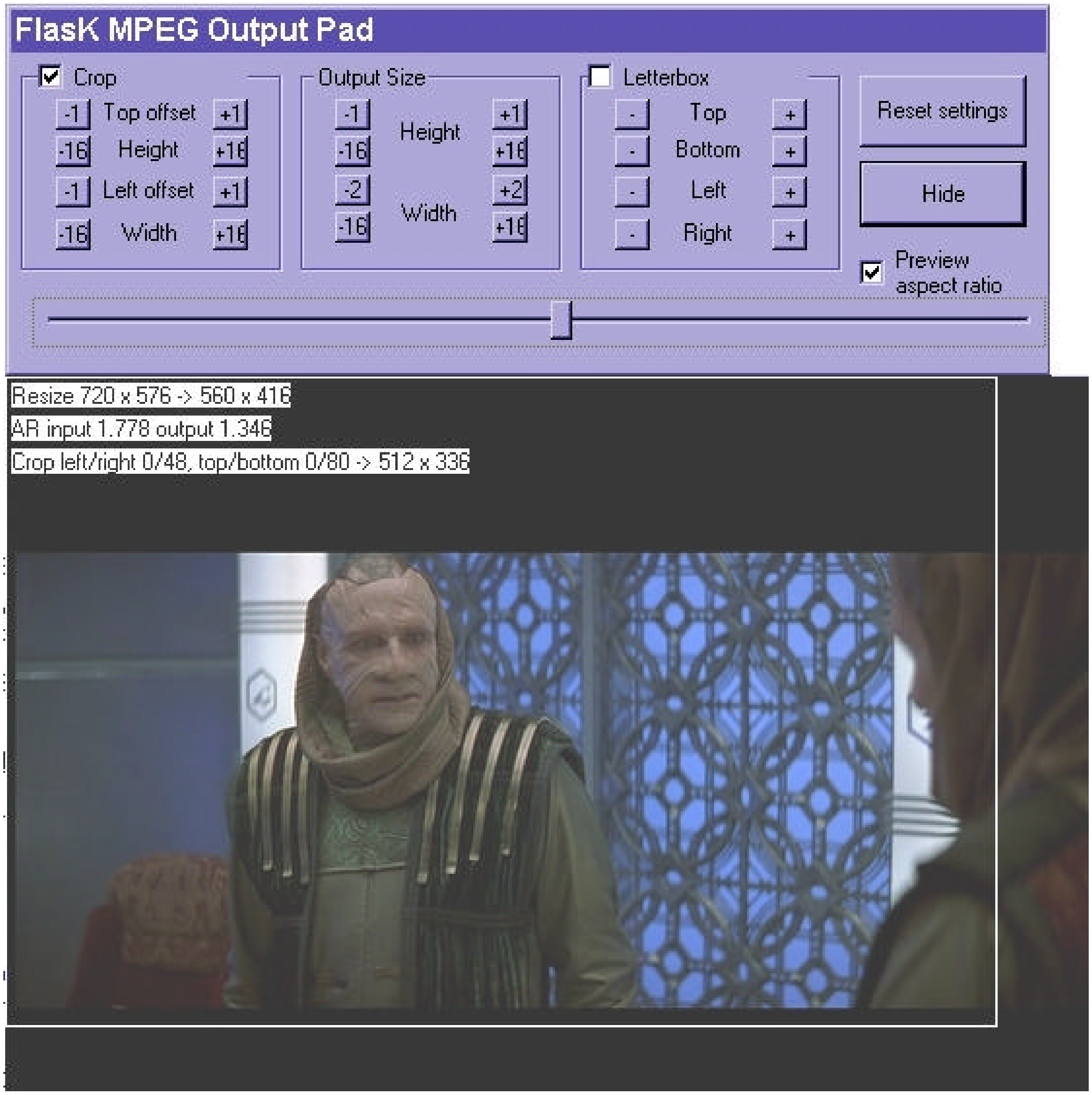
In this instance I managed to get the exact size I wanted 576 x 240, but this may not be the case for you. Use the - Top Offset (seen below) and the + Top Offset (circle 2 above) to get the picture positioned perfectly.

Okay, that's it! Don't worry if the Flask picture settings are a different size to what it says in the output pad, the cropping overrides anything Flask had set before.

Direct DVD to DivX from CD

(the new trick)

There is no such thing as a perfect solution but this one comes pretty close. For those of you who have small hard disks this is an effective way to convert a DVD to DivX directly from the CD! Its been made possible by a non-official "hacked" version of Flask Mpeg. Its called FlasKMpeg Decss 0.594h1. This version actually has some improvements on the old FlasK Mpeg. For one it seems to have fixed the aspect ratio problems for NTSC movies.



☑ Crop

□ Letterbox

☑ Preview aspect ratio

```
Resize 720 x 576 -> 560 x 416
AR input 1.778 output 1.346
Crop left/right 0/48, top/bottom 0/80 -> 512 x 336
```

Other great features are that it can outline in white where the cropping will be on you image rather than just deleting it as the old Flask did. This feature is useful because before you didn't know how much of the image you may be loosing. Also, the slider bar above the image allows you to select any part of the movie, so instead of getting some crud dark scene that is no good for resizing, you can check anywhere you like to see if it is cropped correct and in the best aspect ratio!

Before I start here are the things you will need:

DivX ;-) Mpeg-4 Codec 3.1alpha (or greater)

Fhg Radium MP3 codec

Advanced Bitrate Calculator 1.8

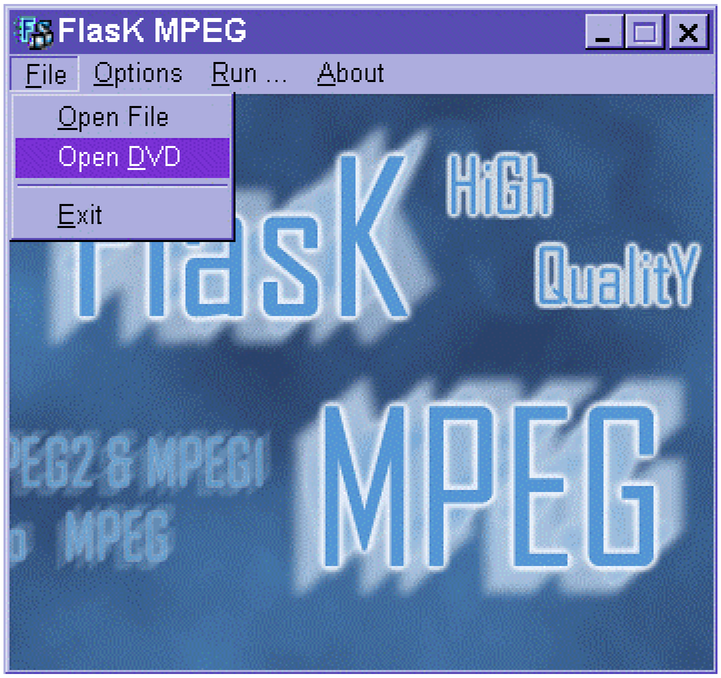
Flask Mpeg DeCSS 0.594h1 (or greater)

VobDec (without GUI)

If you haven't installed the Radium MP3 and the DivX codec's you may as well do so now. Remember to run the "run me first" option on the DivX install too or it will not be installed correctly.

Flask Mpeg has two main modes:

> **Open File:** This will open any mpeg file and try to convert it to whatever you want.
>
> **Open DVD:** This is a special mode that reads a DVD in the same way a DVD player would. This means you will be able to select to convert with subtitles; it will also make sure it only rips one angle in a multi-angle DVD. This prevents repeated scenes spoiling your movie!



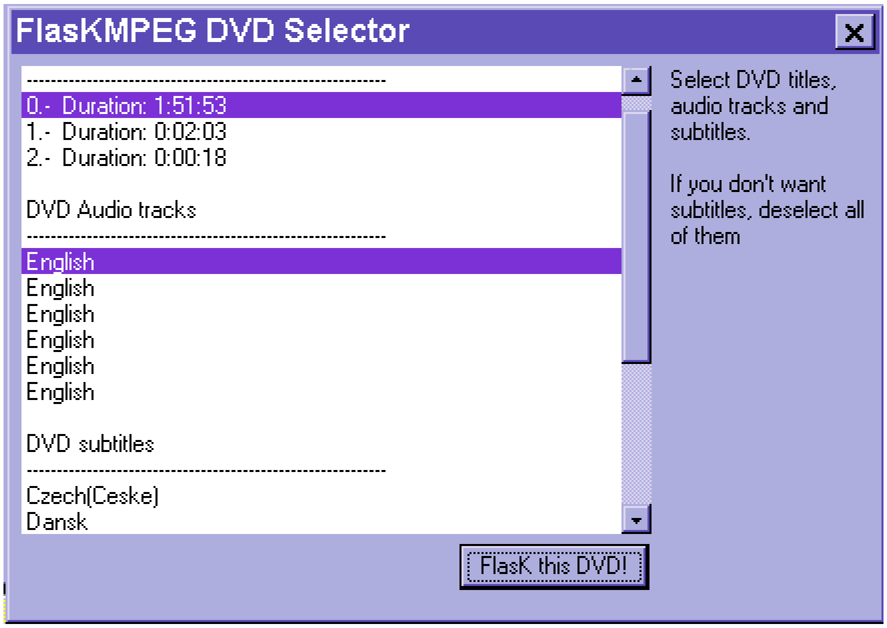
IMPORTANT NOTE!!: There have been reports about DVD drives overheating when forced to read at a much slower pace than single-speed DVD. Flask DeCSS has been tested on a few DVD-Rom's without problem. And I've never heard first hand of a drive being damaged in this way, but you might want to keep an eye on your drive the first time you transfer straight from a DVD-ROM. If you decide to try this program it is your own choice, I take *no* responsability for any damage you may do to your hardware by using this program! You have been warned!

AUTHENTICATE THE DRIVE

Basically just run SmartRipper but don't try and rip anything with SmartRipper, just run it and hit the minamize button. Bingo! Your drive is authenticated!

OPEN THE FILE WITH 'Open DVD'

Select 'Open DVD' and find the .IFO file for your movie.Up will pop something like the picture below:



```
------------------------------------------------------------
0.- Duration: 1:51:53
1.- Duration: 0:02:03
2.- Duration: 0:00:18

DVD Audio tracks
------------------------------------------------------------
English
English
English
English
English
English

DVD subtitles
------------------------------------------------------------
Czech(Ceske)
Dansk
```

Select DVD titles, audio tracks and subtitles.

If you don't want subtitles, deselect all of them

Select the movie Duration, in this case 1.51.53. This will usually be the first one in the list, but you can usually see from the length which one to choose.

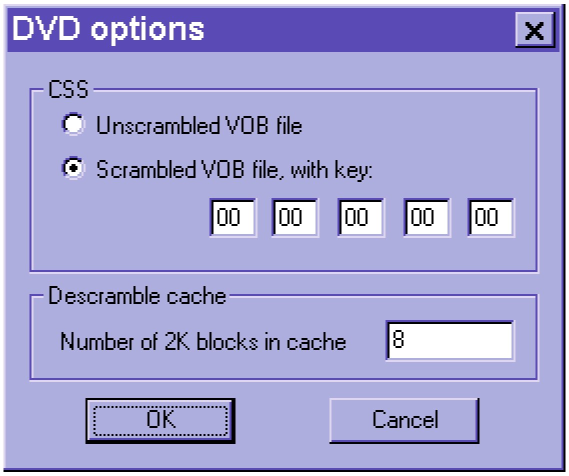
Next choose the language. Obviously they cannot ALL be English so choose the first and encode a minuet of the film and listen to it. If its not English choose the next in the list, and the next and the next etc., until you find the correct one. Or open the DVD in a DVD player and see the order they are in, usually it will be the same.

Lastly we have subtitles. I don't usually select any because I don't want them on my movie. If you choose subtitles then you will not be able to turn them off, they will ALWAYS be on your movie! Flask is not always able to do subtitles correctly, so again, you will have to try it and see. There are other ways to get subtitles, though; why not check out my subtitle ripping section for this.

Now Press FlasK this DVD!

GETTING THE DECRYPTION KEY

This is the only annoying bit :). Up pops a dialouge box asking for the decryption key to the DVD.



CSS

○ Unscrambled VOB file

● Scrambled VOB file, with key:

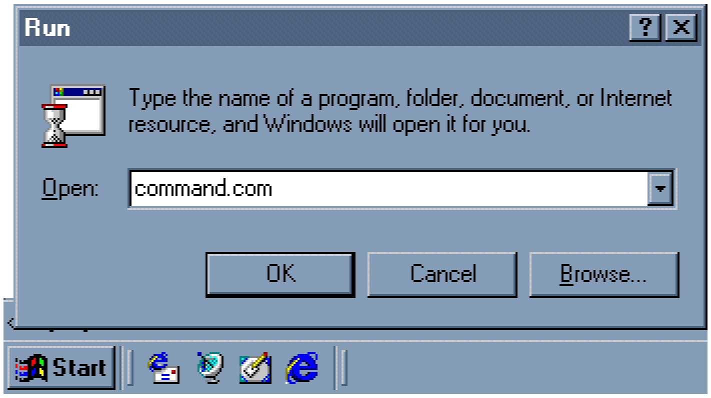
00 00 00 00 00

Descramble cache

Number of 2K blocks in cache: 8

We must use the command line utility Vobdec (which, by the way, is what 99% of GUI rippers use as their ripping engine). Anyway, its not that hard. Make a folder on your main hard drive (usually C:) and call it vobdec. Then copy the Vobdec.exe file into it and we are ready to go:

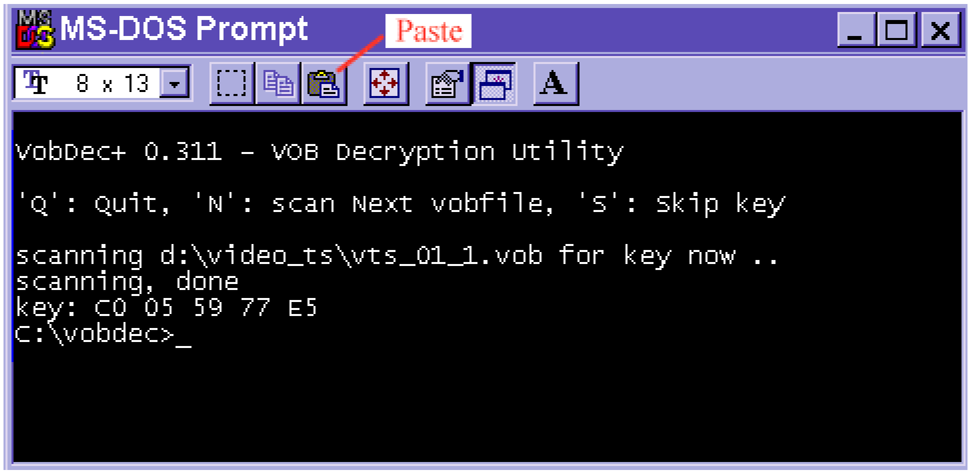
Go to Start Menu > Run and type Command.com to bring the DOS Prompt up ie:



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: command.com

Type: CD\ and press Enter. Type: CD vobdec and press Enter.



```
VobDec+ 0.311 - VOB Decryption Utility

'Q': Quit, 'N': scan Next vobfile, 'S': Skip key

scanning d:\video_ts\vts_01_1.vob for key now ..
scanning, done
key: C0 05 59 77 E5
C:\vobdec>_
```

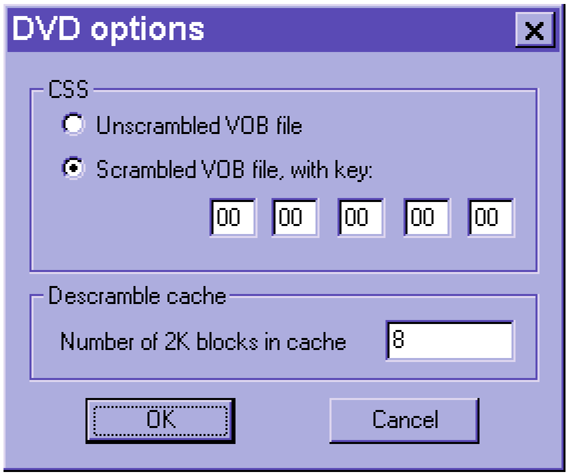
Now, cut and paste the following line into the DOS prompt and press Enter:

vobdec **d**:\video_ts\vts_01_1.vob

Note that the green **d** is the name of the drive you have the DVD in. Its usually d: but change it to whatever your DVD drive says. After searching vobdec will find the key and give a number like this:

E1 69 03 06 00

Just type them into the boxes below.



DVD options

CSS

○ Unscrambled VOB file

● Scrambled VOB file, with key:

00 00 00 00 00

Descramble cache

Number of 2K blocks in cache: 8

Its easy to find any key with vobdec because all you do is type the location of the vob file and it will decrypt it. Often the main movie will be on the vts_01_1.vob file. But if it is not, just change the vob name to whatever vob file you wish to find the key. To illustrate, if you wanted to find the key to the vob file vts_02_1.vob then you paste into the DOS Prompt:

```
vobdec d:\video_ts\vts_02_1.vob
```

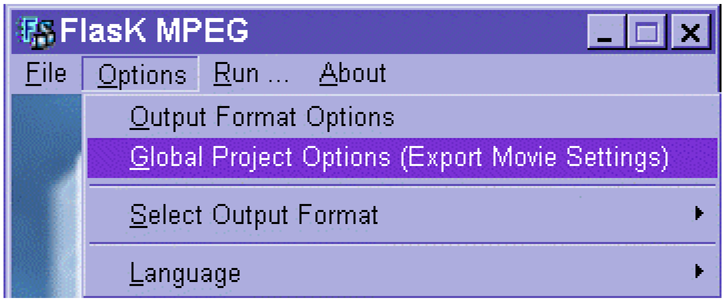
If you wanted to check out the second in the list you put:

```
vobdec d:\video_ts\vts_02_2.vob
```

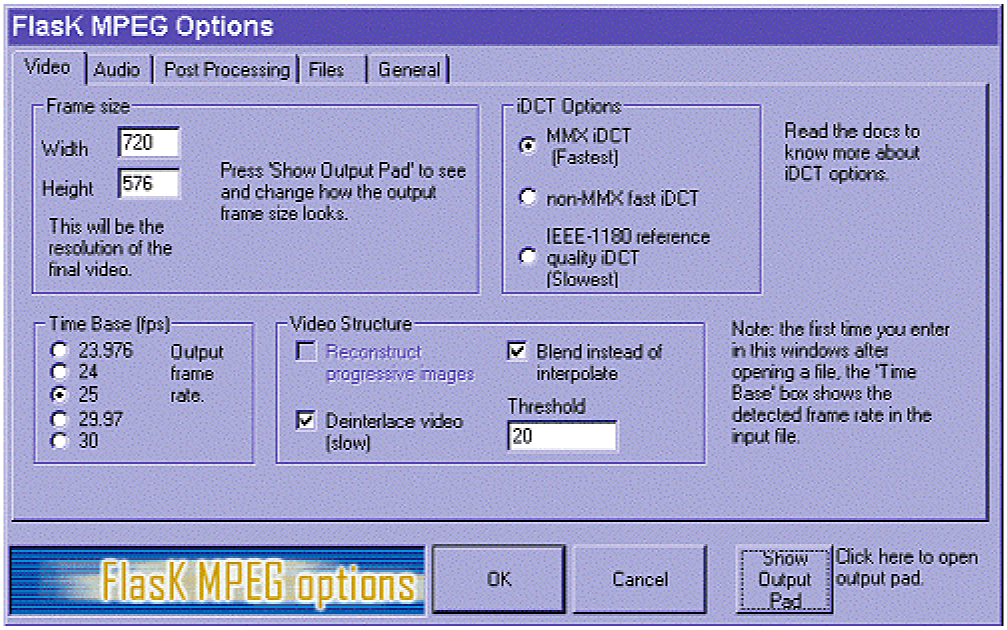
When you have a key, encode a 30 second clip of the movie. If you see bright green mess on the screen and / or Flask crashes you have been given the wrong Key! You have just been unlucky. Keep trying different vob files on the DVD and writing down the key's, eventually one of them should do the trick.

SETTING UP YOUR RIP - GLOBAL PROJECT OPTIONS

Select Global Project options.



VIDEO TAB



Frame size

Width 720

Height 576

This will be the resolution of the final video.

Press 'Show Output Pad' to see and change how the output frame size looks.

iDCT Options

● MMX iDCT (Fastest)

○ non-MMX fast iDCT

○ IEEE-1180 reference quality iDCT (Slowest)

Read the docs to know more about iDCT options.

Time Base (fps)

○ 23.976

○ 24

● 25

○ 29.97

○ 30

Output frame rate.

Video Structure

□ Reconstruct progressive images

☑ Deinterlace video (slow)

☑ Blend instead of interpolate

Threshold 20

Note: the first time you enter in this windows after opening a file, the 'Time Base' box shows the detected frame rate in the input file.

Click here to open output pad.

Frame Size: In the Width and Height sections you can put the size you wish your final video to be. If you wish to crop your movie too you can select the 'Show Output Pad' at the bottom right. For detailed information on how to resize a movie in Flask read section 2nd of this guide: *"resizing the video"*.

Time Base (fps): Flask will normally choose the best framerate for your movie. All PAL movies (European) are 25 frames per second (fps). So if you know your movie is PAL make sure your movie is set to 25. All North American movies are NTSC which means they are 29.97 fps. BUT because of the way they are encoded to DVD most will appear Flask will choose 23.976 fps. This is usually correct so don't change it. Flask DeCSS sometimes chooses PAL instead of NTSC so be careful. And as always test a short clip before you do a whole movie to make sure its ok.

iDCT Options: Just leave these alone its overkill to use IEEE and the quality will not look better.

Video Structure: Do *not* check the deinterlace button unless you really need it because the video will decode very slow. For a full explanation of the interlace problem, check out my article: "Video Formats: NTSC & PAL / Telecine" in the appendix section.

Deinterlace video

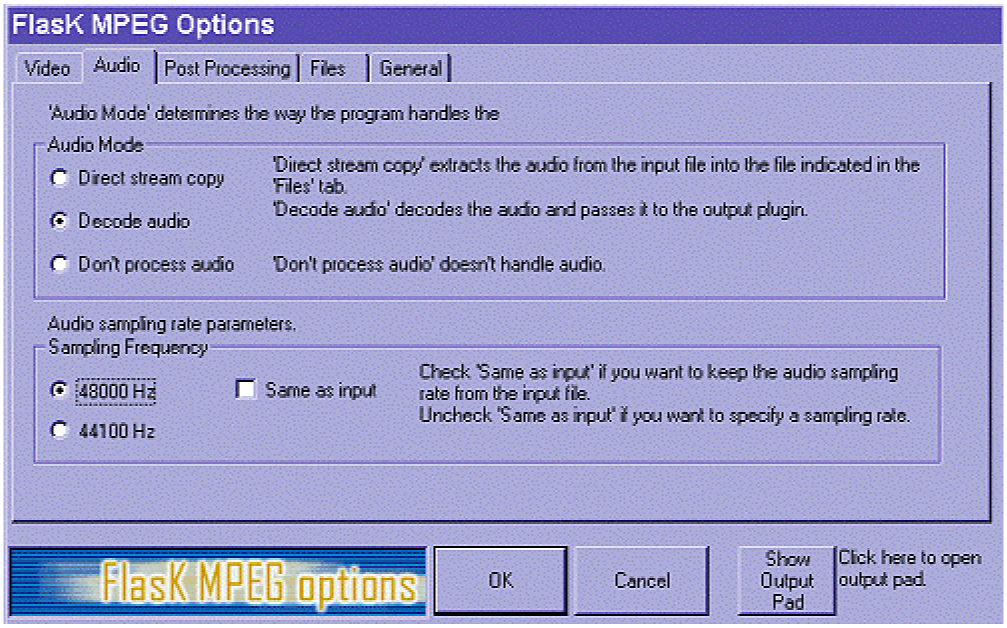
For this kind of Deinterlacer blend gives best quality. The Threshold setting is basically how much it blurs problem lines. The lower the number the more it blurs it. If the interlace problem is really bad you should use a setting of 1-5 instead of 20. If its not very bad you could use 15-20.

If Flask (or you) select the Time Base of 24 or 23.976 fps then there will be the extra option called 'Reconstruct progressive images'. This option is grayed out in the above picture. Flask automatically tries to detect the real framerate inside the DVD. This setting is not an IVTC (inverse telecine); it just forces Flask Mpeg to 24fps progressive mode, avoiding potential interlace problems if the movie is 24 but the playback is set to 29.976.

So you want to know what you should do with them, right?! For PAL movies, I have found that there is no point selecting anything but 25fps and you will *not* need 'reconstruct progressive images'. In fact, you will almost always get jerky playback if you select it. For NTSC I'm not sure, I think the general opinion is to let Flask choose or ignore it.

AUDIO TAB

On to the Audio tab always select 'Decode audio' if you want sound. For DVD's un-check the 'same as input box' and select 48000 Hz (just to make sure you have the right setting). For Mpeg-1 you'd use same as input or 44100Hz. Never use 44100Hz with DVD's or you will get audio synchronization problems.



'Audio Mode' determines the way the program handles the

Audio Mode

○ Direct stream copy
● Decode audio
○ Don't process audio

'Direct stream copy' extracts the audio from the input file into the file indicated in the 'Files' tab.
'Decode audio' decodes the audio and passes it to the output plugin.

'Don't process audio' doesn't handle audio.

Audio sampling rate parameters.

Sampling Frequency

● 48000 Hz
○ 44100 Hz

□ Same as input

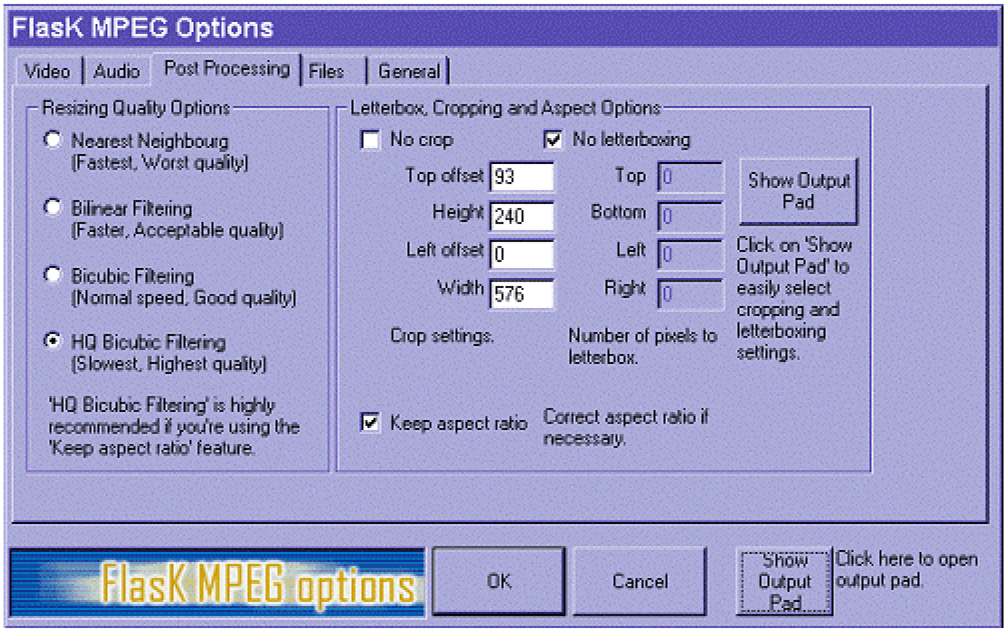
Check 'Same as input' if you want to keep the audio sampling rate from the input file.
Uncheck 'Same as input' if you want to specify a sampling rate.

Click here to open output pad.

If you just want to copy the DVD's AC3 audio across instead of converting it use the 'Direct Stream copy' option.

POST PROCESSING TAB

This section deals almost exclusively with resizing. Never use 'Nearest Neighbouring' unless you are not resizing the picture because the quality is crud. Contrary to popular belief Bilinear looks just as nice as Bicubic, except it is twice as fast. JASC (makers of Paint Shop Pro) recommend Bilinear for shrinking images and Bicubic for enlarging them. But use what you think looks best.



Resizing Quality Options

○ Nearest Neighbourg (Fastest, Worst quality)

○ Bilinear Filtering (Faster, Acceptable quality)

○ Bicubic Filtering (Normal speed, Good quality)

● HQ Bicubic Filtering (Slowest, Highest quality)

'HQ Bicubic Filtering' is highly recommended if you're using the 'Keep aspect ratio' feature.

Letterbox, Cropping and Aspect Options

□ No crop

☑ No letterboxing

Top offset 93

Height 240

Left offset 0

Width 576

Crop settings.

Top 0

Bottom 0

Left 0

Right 0

Number of pixels to letterbox.

Click on 'Show Output Pad' to easily select cropping and letterboxing settings.

☑ Keep aspect ratio

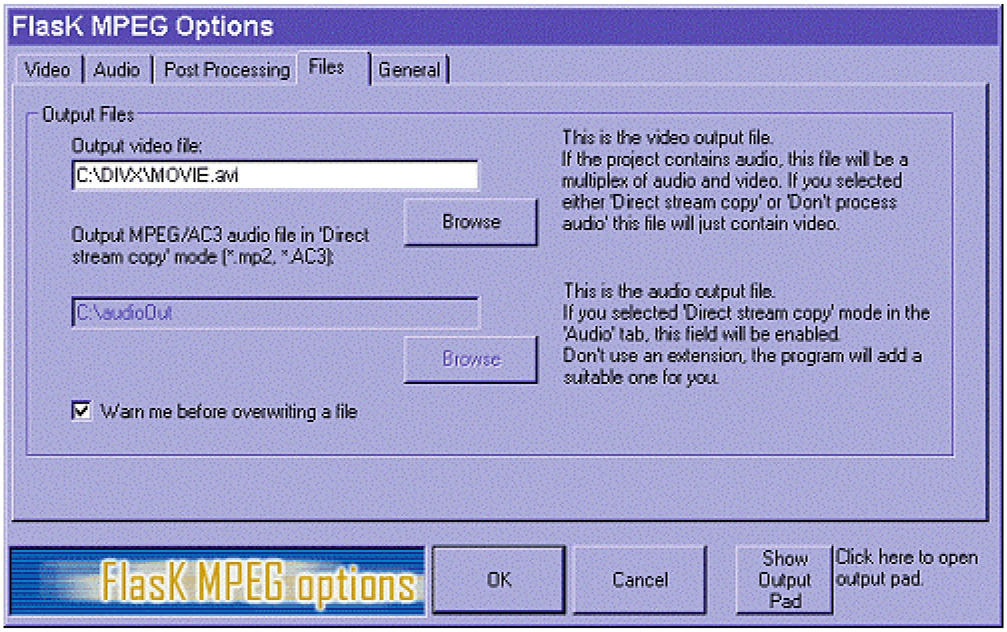
Correct aspect ratio if necessary.

Click here to open output pad.

Keep aspect ratio: PAL users should always tick this box unless you know you don't need it. This is even more important with Widescreen DVD's. If you use NTSC DVD's the image may become stretched slightly wrongly. If you notice this uncheck the 'keep aspect ratio' and work out the ratio yourself. See the article *"Resizing DVD's with Correct Aspect Ratios"* in my appendix.

Crop & Letterboxing: All the settings for cropping and letterboxing the DVD can be entered here or the output pad can be used. For detailed information on how to resize a movie in Flask read section 2nd of this guide: *"resizing the video"*.

FILES TAB



Output Files

Output video file: C:\DIVX\MOVIE.avi

This is the video output file.
If the project contains audio, this file will be a multiplex of audio and video. If you selected either 'Direct stream copy' or 'Don't process audio' this file will just contain video.

Output MPEG/AC3 audio file in 'Direct stream copy' mode (*.mp2, *.AC3): C:\audioOut

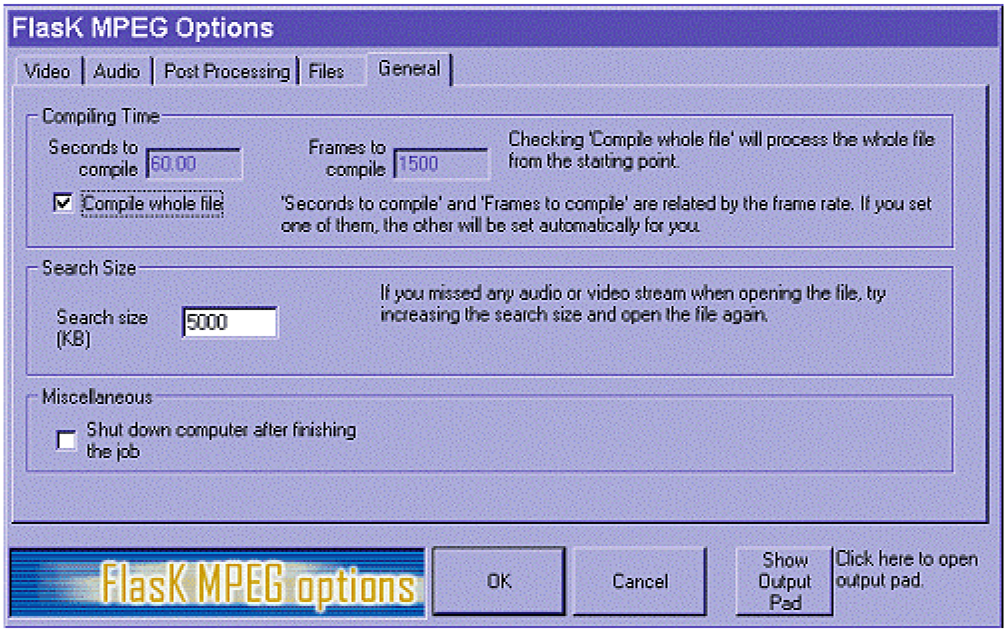
This is the audio output file.
If you selected 'Direct stream copy' mode in the 'Audio' tab, this field will be enabled.
Don't use an extension, the program will add a suitable one for you.

☑ Warn me before overwriting a file

Click here to open output pad.

Choose where you want to save your final movie. The audio save option is grayed out because you are encoding the video with audio in it. If you selected 'Direct Stream Copy' on the previous Audio Tab then you could say where you wanted it saved.

GENERAL TAB



Compiling Time

Seconds to compile: 60.00

Frames to compile: 1500

Checking 'Compile whole file' will process the whole file from the starting point.

☑ Compile whole file

'Seconds to compile' and 'Frames to compile' are related by the frame rate. If you set one of them, the other will be set automatically for you.

Search Size

Search size (KB): 5000

If you missed any audio or video stream when opening the file, try increasing the search size and open the file again.

Miscellaneous

☐ Shut down computer after finishing the job

Click here to open output pad.

Compiling Time: This speaks for itself. If you check the 'Compile whole file' box it will convert the whole DVD. If you uncheck it you can say how many seconds to encode.

Obviously 60 is one minuet and 120 is two mins. The frames to compile is basically the same thing. There are 25 frames to one second of video for PAL and either 23.976 or 29.970 for NTSC. Using these options you can encode a small clip to test the quality etc. I recommend you shut down and restart Flask just before you encode a long movie to reduce chances of it crashing.

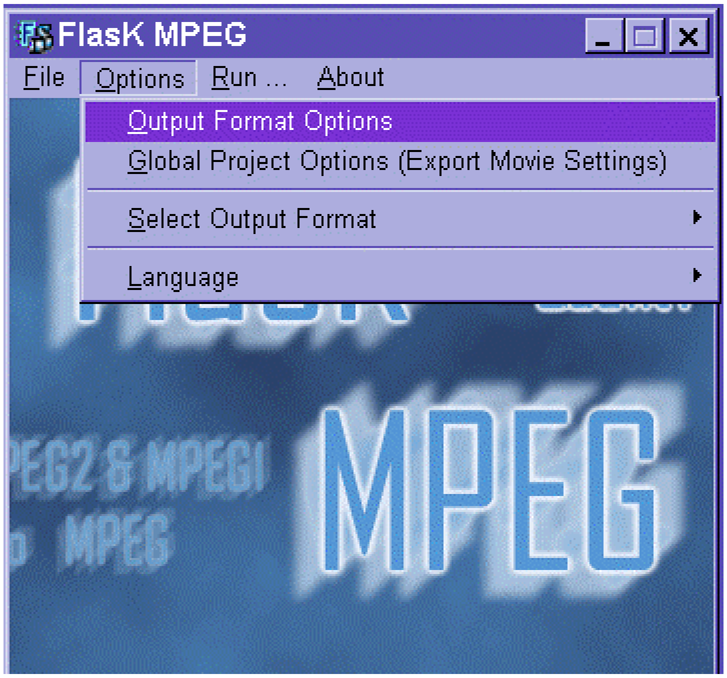
Search Size: This searches the DVD for the audio. As it says, if you have problems make the number bigger. If all else fails and you cannot find audio you can try one of the other methods I explain in the section:*"DVD Audio Extraction"*.

Miscellaneous: Like the thing says, it'll shut down the computer if it can when its done :)

That's it you're nearly ready to encode! Lets set the output options...

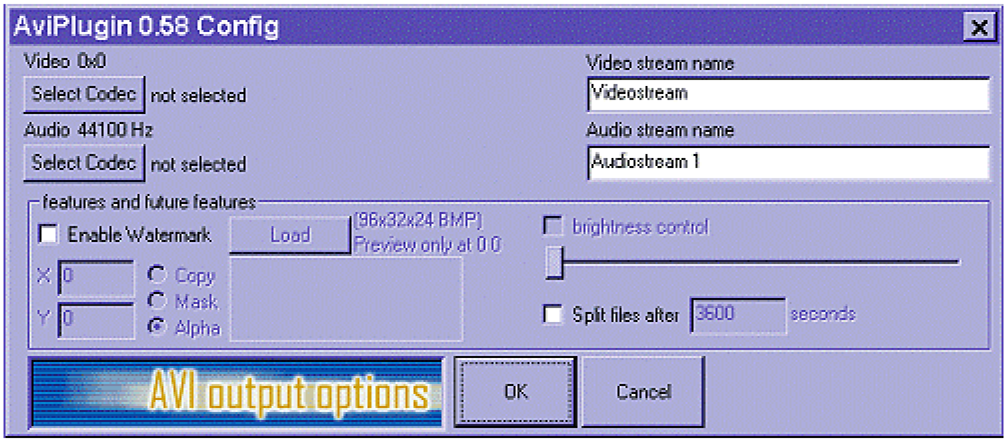
OUTPUT FORMAT OPTIONS

Go to Options > Output Format Options.



And up will pop the box shown below:

Features and Future Features: Flask has an option to 'Split files after' a certain amount of seconds. This is useful if you want a VCD to be split in half so that one part can be put on a single CD. I don't use it for Divx because I prefer to split them in VirtualDub. To use it, just work out how many seconds there are in the movie. If the movie is 131 minuets then multiply 131 by 60 and you have the seconds (7860 seconds) then just divide that in half and you have your split number (3930).



AviPlugin 0.58 Config

Video 0x0
not selected

Audio 44100 Hz
not selected

Video stream name: Videostream

Audio stream name: Audiostream 1

features and future features

□ Enable Watermark (96x32x24 BMP) Preview only at 0.0

X 0
Y 0

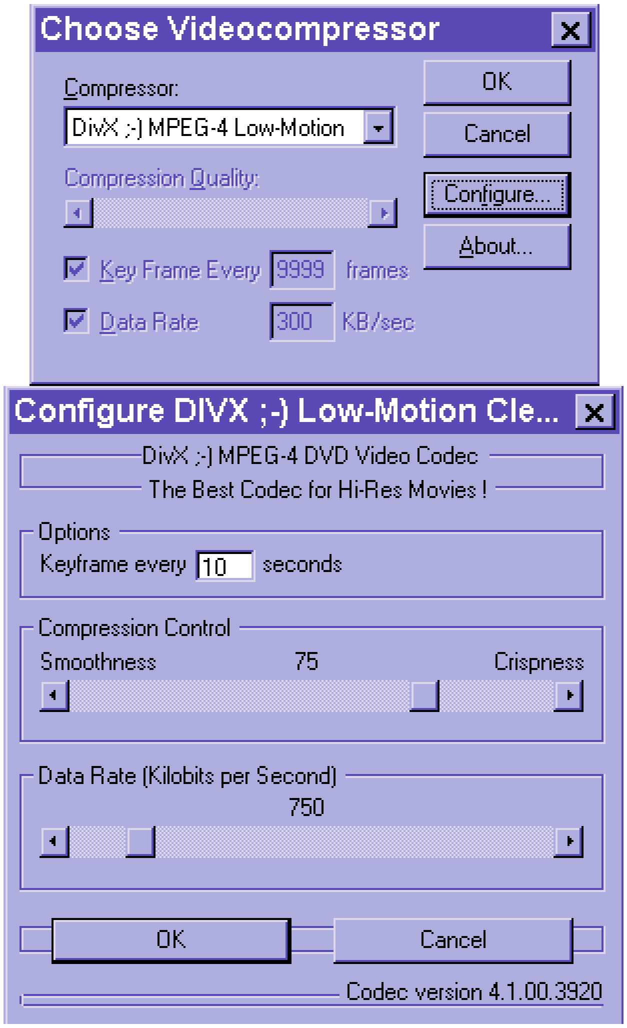
○ Copy
○ Mask
● Alpha

□ brightness control

□ Split files after 3600 seconds

Enable Watermark: This lets you use any bitmap picture you want as a logo. So you can make a little picture saying made by 'smiffy' or whatever you like. The X and Y boxes let you say where the logo will appear on the movie. Then you have three options on how it will look. It can "Copy", which just sticks the bitmap on top. It can "Mask" which lets part of the image become transparent. Or it can "Alpha" which makes the whole image transparent. I do not recommend using logos because they increase the filesize of the picture and degrade the quality.

Select Codec: These are the important buttons! Hit the top one and the following box appears:



Choose Videocompressor

Compressor: DivX ;-) MPEG-4 Low-Motion

Compression Quality:

☑ Key Frame Every 9999 frames

☑ Data Rate 300 KB/sec

Configure DIVX ;-) Low-Motion Cle...

DivX ;-) MPEG-4 DVD Video Codec

The Best Codec for Hi-Res Movies !

Options

Keyframe every 10 seconds

Compression Control

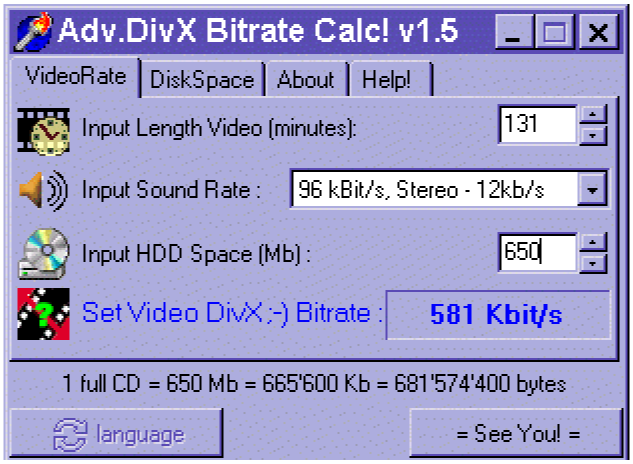
Smoothness 75 Crispness

Data Rate (Kilobits per Second)

750

Codec version 4.1.00.3920

Choose the Divx ;-) MPEG-4' Low or High Motion codec from the drop down list depending on what you want. For getting the best setting you can look at my article: "Best Divx Quality & Bitrate Guide" in the appendix. But in short, use a bitrate calculator to determine the best bitrate to put in here. I'm using the 'Advanced Divx Bitrate Calc! Version 1.5'. It seems to give a more reliable amount and will not make a movie larger than you intend.



Input Length Video (minutes): 131

Input Sound Rate : 96 kBit/s, Stereo - 12kb/s

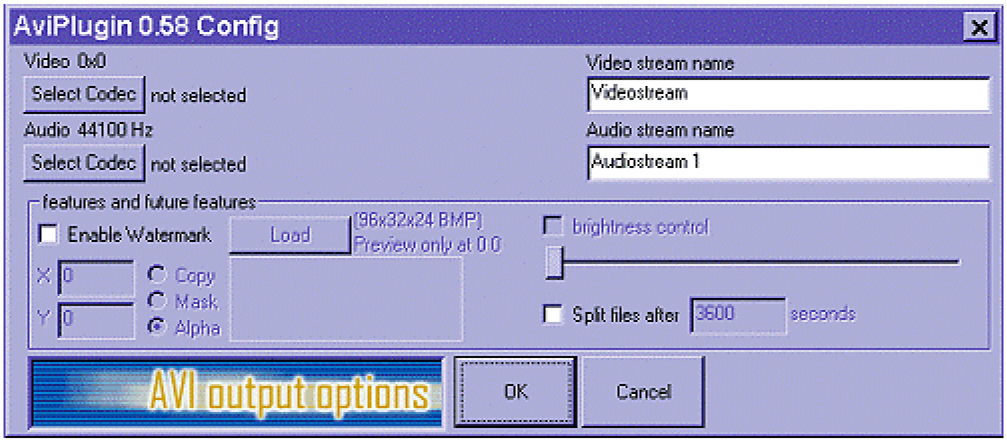
Input HDD Space (Mb) : 650

Set Video DivX ;-) Bitrate : 581 Kbit/s

1 full CD = 650 Mb = 665'600 Kb = 681'574'400 bytes

Just put the DVD movie length in minuets in the top box. Select the audio bitrate you want to use (for Flask this will almost always be 96 kBit/s Stereo). Then it will tell you the amount to use! In the above picture the calculator says 581 so you would move the 'Data Rate slider bar to 581 instead of the 750 it is in the above picture. As for keyframes, I tend to use a keyframe every 1 second, but you can use one every 10 seconds to saves more memory. Then set the 'Crispiness to 75% and you are done!

Ok. Back to our previous dialogue box. Press the bottom 'Select Codec' button to select the audio compression.



Video 0x0

Select Codec not selected

Audio 44100 Hz

Select Codec not selected

Video stream name: Videostream

Audio stream name: Audiostream 1

□ Enable Watermark (96x32x24 BMP) Preview only at 0.0

X 0

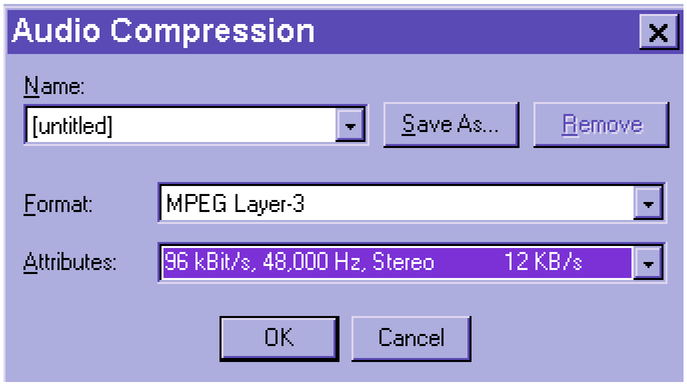
Y 0

○ Copy ○ Mask ● Alpha

□ brightness control

□ Split files after 3600 seconds

Up pops this box:



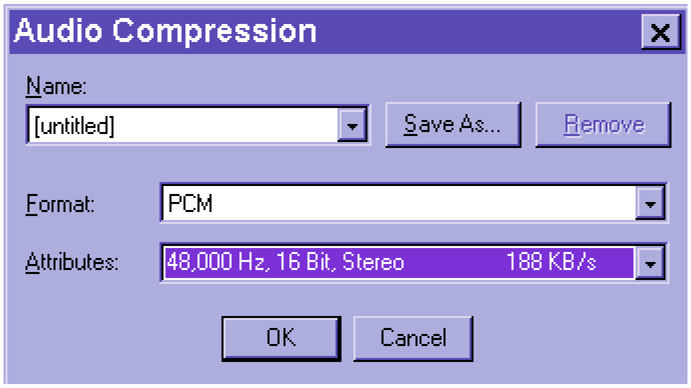
Name: [untitled]

Format: MPEG Layer-3

Attributes: 96 kBit/s, 48,000 Hz, Stereo 12 KB/s

Choose 'MPEG Layer-3' at 96 kBit/s 48,000Hz. You can choose what you like here, 128 or 190 etc., if you prefer, but there is no need it will just make the file size larger.

There are two problems with converting the audio to Mp3. First, the audio must be 48,000Hz. Some sound cards cannot handle the format and so it may play the audio back dull and buzzing. Secondly, because of the dynamic range of a DVD the audio level will sound very low and may not be loud enough for you. Check out the section *"Dull Audio, Normalization & Samplerates"* for a detailed explanation and a solution to all this. In short, you can choose PCM audio instead of mp3 and add it later. So here is where you would do that. Instead of choosing Mp3 choose PCM 48,000Hz 16bit, Stereo. *No* other audio format will work in Flask without potential problems!



Name: [untitled]

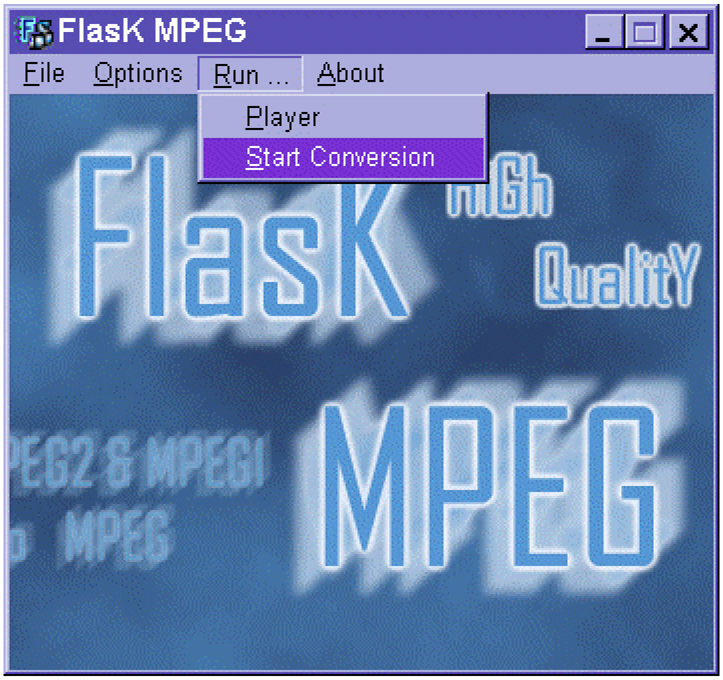
Format: PCM

Attributes: 48,000 Hz, 16 Bit, Stereo 188 KB/s

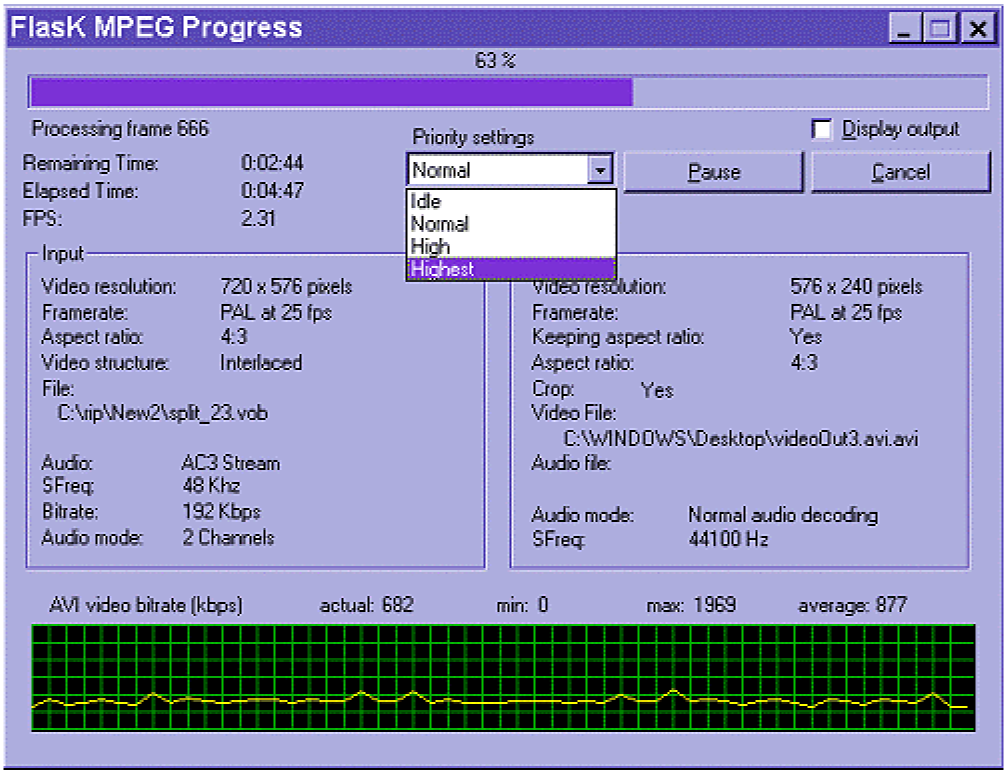
Warning! PCM Wav audio is uncompressed and creates a seriously large movie file 1-2GB! If you are making a single CD rip you will probably be okay using PCM audio. If you are making a 2 CD rip, your file may be so large that VirtualDub will not open it and Windows will not play it! This is annoying because Divx Mpeg4 can be larger than 2GB and would normally probably work okay at anything upto 4GB. But Flask seems to create a file that is corrupted at these sizes! So, the PCM Wav is an advanced option that should be treated with great respect! If you are a beginner just use Mp3!

PRESS GO!

Once you have finished setting up your movie just go to Run > Start Conversion.



Up will pop the following screen.



63 %

Processing frame 666

Remaining Time: 0:02:44
Elapsed Time: 0:04:47
FPS: 2.31

Priority settings: Normal

Input

Video resolution: 720 x 576 pixels
Framerate: PAL at 25 fps
Aspect ratio: 4:3
Video structure: Interlaced
File:
C:\rip\New2\split_23.vob

Audio: AC3 Stream
SFreq: 48 Khz
Bitrate: 192 Kbps
Audio mode: 2 Channels

Video resolution: 576 x 240 pixels
Framerate: PAL at 25 fps
Keeping aspect ratio: Yes
Aspect ratio: 4:3
Crop: Yes
Video File:
C:\WINDOWS\Desktop\videoOut3.avi.avi
Audio file:

Audio mode: Normal audio decoding
SFreq: 44100 Hz

AVI video bitrate (kbps) actual: 682 min: 0 max: 1969 average: 877

If you are going to leave your computer alone to compress then select Highest in the 'Priority settings'. This will hog ALL CPU power to make compression faster. This CPU option only really works if other programs are running when Flask is though! When using idle or normal, you will be able to do basic tasks while it is compressing, such as checking Email etc. Be careful not to do too much because if your computer crashes because of another program then you'll need to start over again. Unchecking the 'Display output' box will also help speed things up a little. But if you are using a Dual Processor computer system just leave it on normal otherwise it slows it down.

That's it! In 15-20 hours on a 500Mhz machine you will have a perfect DivX movie ;).

Extracting the DVD to Hard Disk

As soon as you begin to think they cannot make a better ripper suddenly one turns up! SmartRipper is now my no.1 ripper. And, after you try it, I'm sure you will agree with me =). What makes SmartRipper so good is it can actually read a DVD just like a DVD player! It lets you choose which language you wish to rip, and it looks at the information files (.ifo) on the DVD so it can handle mult-angle movies in the same way Flask Mpeg does in DVD mode! In fact, if it wasn't for the fact that Flask Mpeg lets you add subtitles, there would be no need to bother with Flask's DVD mode at all!

***Note:** If SmartRipper says "can't unlock drive" you will need to start your DVD playing in your PC DVD player (such as PowerDVD, WinDVD etc). Start it, press pause, and then launch SmartRipper.*

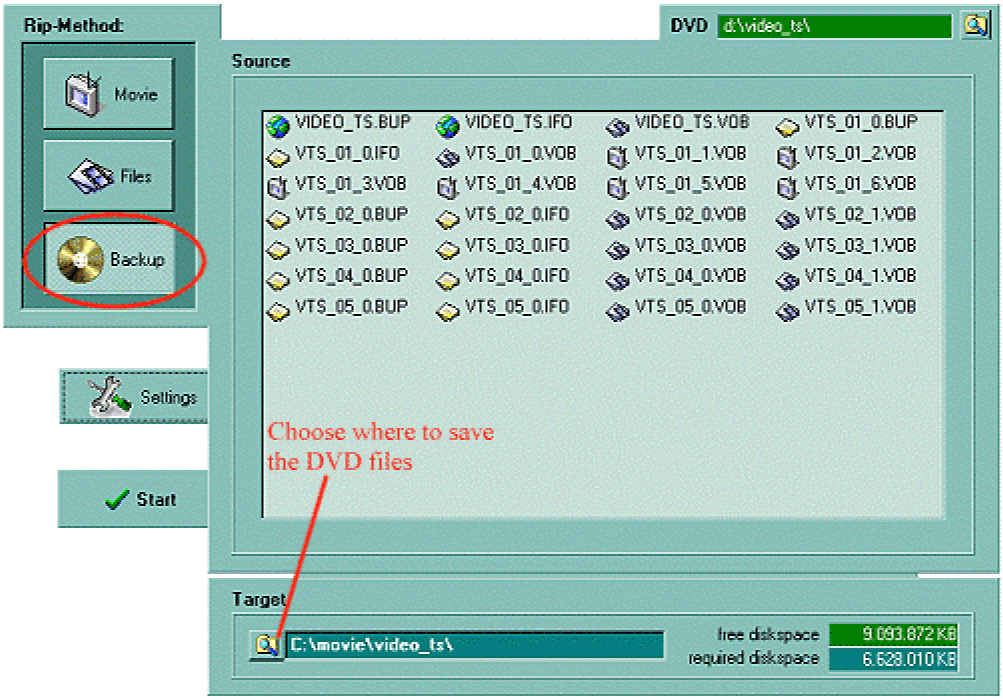
Before I start you will need:

Smartripper 2.02

SmartRipper has three basic modes:

BACKUP MODE

This mode backs up the entire DVD. It doesn't let you choose what language you want to use or what angle of the movie to rip. It is used solely for the purpose of copying exactly the whole DVD unchanged to your hard disk. It does, of course, decrypt the DVD but all files remain the same.



DVD d:\video_ts\

VIDEO_TS.BUP VIDEO_TS.IFO VIDEO_TS.VOB VTS_01_0.BUP
VTS_01_0.IFO VTS_01_0.VOB VTS_01_1.VOB VTS_01_2.VOB
VTS_01_3.VOB VTS_01_4.VOB VTS_01_5.VOB VTS_01_6.VOB
VTS_02_0.BUP VTS_02_0.IFO VTS_02_0.VOB VTS_02_1.VOB
VTS_03_0.BUP VTS_03_0.IFO VTS_03_0.VOB VTS_03_1.VOB
VTS_04_0.BUP VTS_04_0.IFO VTS_04_0.VOB VTS_04_1.VOB
VTS_05_0.BUP VTS_05_0.IFO VTS_05_0.VOB VTS_05_1.VOB

Choose where to save the DVD files

C:\movie\video_ts\

free diskspace 9.093.872 KB
required diskspace 6.528.010 KB

Backing up a whole DVD in one go takes a ***lot*** of space, something like 6-9GB. For this reason we will *not* be using backup mode for this guide. But either way its laughingly easy to use: simply choose where you want to save your DVD file by clicking on the 'Target' button. Then click on the large Start button. That's it!

FILE MODE

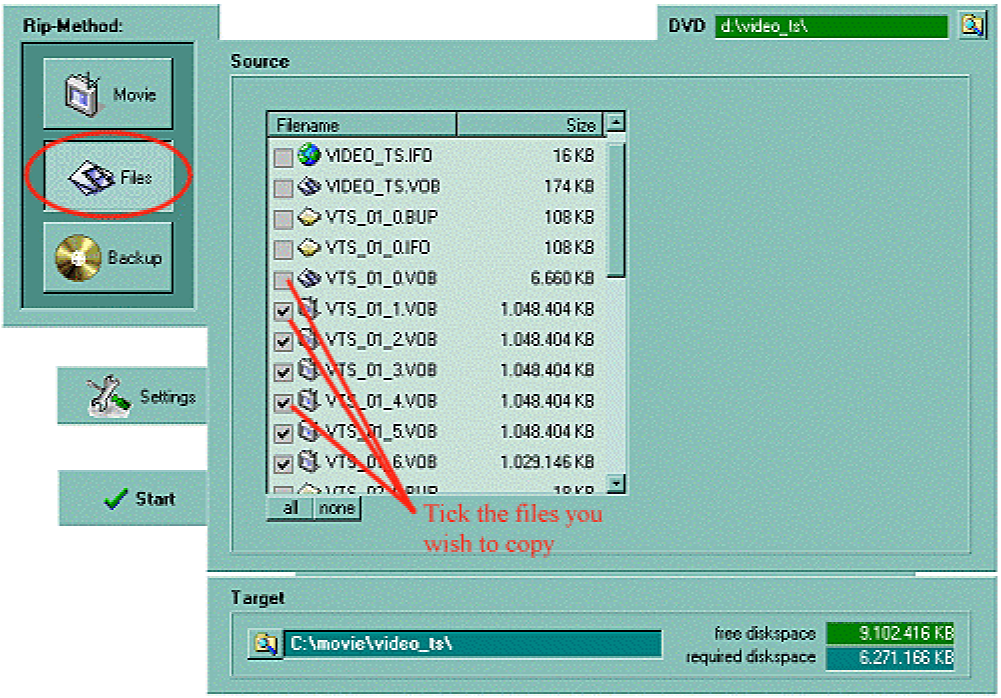
File Mode is almost the same as Backup Mode except it lets you choose which files to copy. Again, it does not read multi-angle's or let you choose which language it will rip, it literally just copies and decrypts any file you point to.

Note for Flask's DVD mode:

If you are going to use Flask Mpeg in DVD mode, then SmartRippers File Mode is the best way to rip the files for Flask to use. In File mode SmartRipper automatically selects the correct files for the main movie for you - they will have a TV screen picture by them. But don't forget to copy the IFO file that belongs to the movie as well. This will be called by the same name as the first movie file. For example, if the first movie file is called **VTS_01_0.VOB** the IFO file to copy with it will be **VTS_01_0.IFO**.

Remember, the main movie will not always have the same name. So if first file of the main movie were called **VTS_02_0.VOB** then the IFO file to copy with it will obviously also be **VTS_02_0.IFO**

For more details on this subject read the article called *"Introduction to the Structure of a DVD"* in the appendix section.



DVD d:\video_ts\

	Filename	Size
□	VIDEO_TS.IFO	16 KB
□	VIDEO_TS.VOB	174 KB
□	VTS_01_0.BUP	108 KB
□	VTS_01_0.IFO	108 KB
□	VTS_01_0.VOB	6.660 KB
☑	VTS_01_1.VOB	1.048.404 KB
☑	VTS_01_2.VOB	1.048.404 KB
☑	VTS_01_3.VOB	1.048.404 KB
☑	VTS_01_4.VOB	1.048.404 KB
☑	VTS_01_5.VOB	1.048.404 KB
☑	VTS_01_6.VOB	1.029.146 KB
□	VTS_02_0.BUP	18 KB

Tick the files you wish to copy

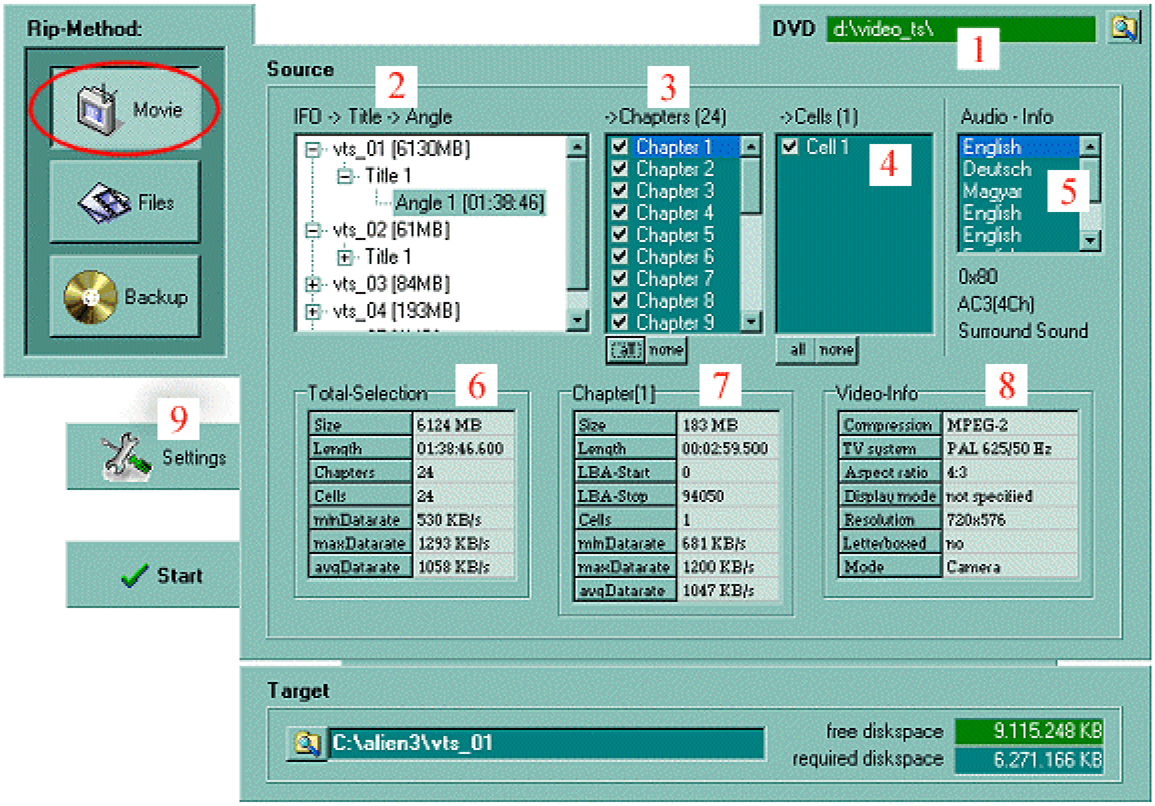
Target: C:\movie\video_ts\

free diskspace 9.102.416 KB
required diskspace 6.271.166 KB

File mode is easy to use too; select where you wish to save your files in the 'Target' section. Tick the boxes next to the files you wish to backup. Then click on the large Start button. Your done!

MOVIE MODE

Movie mode is SmartRippers default mode and is by far the most powerful. Put in your CD and start SmartRipper and up pops the picture below. It looks complicated but I'll explain the features one by one.



DVD d:\video_ts\

IFO -> Title -> Angle

- vts_01 [6130MB]
  - Title 1
    - Angle 1 [01:38:46]
- vts_02 [61MB]
  - Title 1
- vts_03 [84MB]
- vts_04 [193MB]

->Chapters (24)

☑ Chapter 1
☑ Chapter 2
☑ Chapter 3
☑ Chapter 4
☑ Chapter 5
☑ Chapter 6
☑ Chapter 7
☑ Chapter 8
☑ Chapter 9

->Cells (1)

☑ Cell 1

Audio - Info

English
Deutsch
Magyar
English
English

0x80
AC3[4Ch]
Surround Sound

Total-Selection

Size	6124 MB
Length	01:38:46.600
Chapters	24
Cells	24
minDatarate	530 KB/s
maxDatarate	1293 KB/s
avgDatarate	1058 KB/s

Chapter[1]

Size	183 MB
Length	00:02:59.500
LBA-Start	0
LBA-Stop	94050
Cells	1
minDatarate	681 KB/s
maxDatarate	1200 KB/s
avgDatarate	1047 KB/s

Video-Info

Compression	MPEG-2
TV system	PAL 625/50 Hz
Aspect ratio	4:3
Display mode	not specified
Resolution	720x576
Letterboxed	no
Mode	Camera

Target

C:\alien3\vts_01

free diskspace 9.115.248 KB
required diskspace 6.271.166 KB

1. DVD Selection Box: This lets you change the CD drive that SmartRipper is reading.

2. Movie Selection Box: This shows a list of movie titles SmartRipper has found. You can choose which movie title to rip. Since SmartRipper is reading the DVD like a DVD player, it will not rip half a movie or parts of the movie in wrong order. Each Title and Angle represents a *whole* DVD feature, be it a trailer, special feature, main movie, or even the menu selection! Each feature is identified by its DVD Vob name, such as: vts_something. The main movie is automatically chosen by SmartRipper, so you don't need to think much, just press start =). If more than one 'angle' appear for a single title, the correct one will usually be the longest in length. If they are the same length its probably a true multi-angle movie so choose the angle you prefer or just the first one.

3. Chapter Selection: This lets you extract individual chapters from each title. It is useful if you want to extract small clips for testing purposes before you spend the time converting a whole movie.

4. Cell: This doesn't seem to do much. Vob files can be divided by Cell or Vob ID, but so far I haven't had a DVD that needed this feature.

5. Audio Selection: Lets you select which audio track you wish to rip with the video.

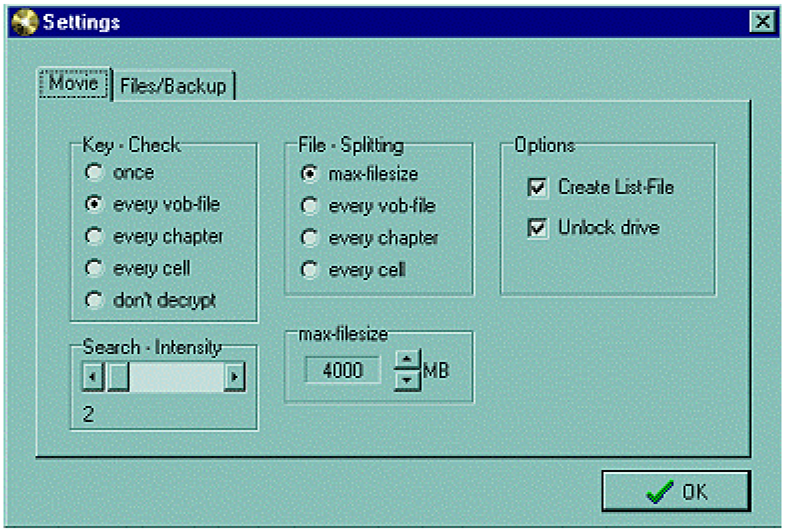
6. Total Selection: This is a display to tell you exactly what will rip when you press the start button. It offers useful info on data rate, cells, filesize and the length of the title.

7. Chapter: This offers the same information as **6** but only for the chapter you have selected in box **3**.

8. Video Info: This offers general information on the DVD. The aspect ratio is not usually correct, but such information is not needed anyway.

9. Settings: If you want to get even more technical, you can play about with the rip settings too. But you can quite happily leave this section alone if you prefer - the movie will rip perfectly without you touching it!

Finally, there is one option in the settings (**9**) that you may want to know about. SmartRipper is automatically set to 'max-filesize' in the 'File-Splitting' section. Instead of ripping every Vob separately, it will merge all the Vob files into one. *But* it will split them into separate files when it reaches the 'max-filesize' shown at the bottom. In the picture below this is set to 4000 MB (4 Gigabytes).



Settings

Movie | Files/Backup

Key - Check
○ once
● every vob-file
○ every chapter
○ every cell
○ don't decrypt

File - Splitting
● max-filesize
○ every vob-file
○ every chapter
○ every cell

Options
☑ Create List-File
☑ Unlock drive

Search - Intensity: 2

max-filesize: 4000 MB

There are two reasons for this feature. Firstly, Mpeg2avi and Flask Mpeg will only convert about 7 Vob files at a time. This means if the DVD has more than that, you will not be able to convert it without problems. To avoid this, Smartripper will merge the files into as few as possible. Secondly, in Windows 95 / 98 there is a storage limit of 4GB per file. So, instead of trying to create a file larger than this, SmartRipper has the 'max-filesize' amount set to split them at 4GB. For more information on this read the '*The AVI Four Gigabyte Limit Explained*'' section in my appendix.

That's it! Because SmartRipper can handle most multi-angle movies you will be able to rip difficult movies like The Matrix without problems. There are still some very rare movies that neither Flask Mpeg nor SmartRipper are able to rip completely! One example is Terminator 2. This is a multi-branching movie and can currently only be done perfectly with vstrip. Check out my ''Repeated Sequences and multi-angle'' guide if you come across any of these movies.

Whenever you rip anything to hard drive its always a good idea to check it has been decoded correctly. The best way to do this is to play it in your computer DVD player. DVD Station, Power DVD, Win DVD ect., should all play them perfectly. They shouldn't have green or pink blocks and they shouldn't have any repeated scenes - although its quite normal for it to have a slightly squashed looking image. Check out the aspect ratio section of my appendix for an explanation of this.

DVD to DivX with Mpeg2avi

The video encoding speed of Mpeg2avi is much much faster than that of Flask Mpeg. It also gives a slightly smoother playback and picture quality. There are a few downsides of Mpeg2avi, mainly that we have to get the audio and add it to the video separately. This may sound hard but it not too difficult, and, if done correctly, will have just as few synchronization problems as Flask Mpeg. Of course, getting audio separately with Graphedit gives better quality audio. And for reasons of speed and quality Mpeg2avi is my preferred choice. But before you switch you should realize that Mpeg2avi has no DVD .IFO reading ability and almost no way of fixing interlace problems!

Before I start here are the things you will need:

DivX ;-) Mpeg-4 Codec 3.1alpha (or greater)

Fhg Radium MP3 codec

DanniDin's GUI v0.19

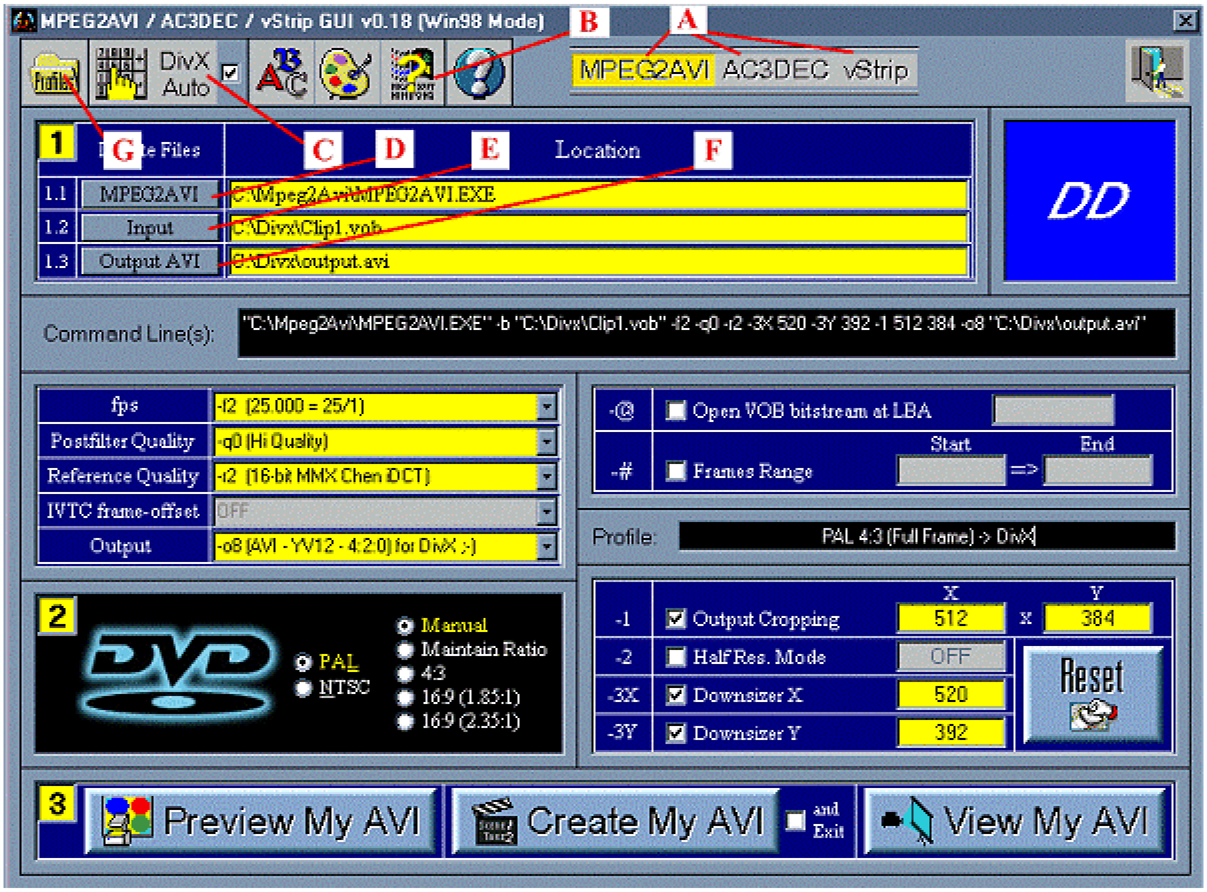
Mpeg2avi 0.16B35

Advanced Bitrate Calculator 1.8

If you haven't installed the Radium MP3 and the Divx codec's you may as well do so now. Remember to run the "run me first" option on the Divx install too or it will not be installed correctly. Don't use the crappy AngelPotion codec, its a bad Mpeg-4 hack, yes, that's right! A hack and not a self made Mpeg-4 codec from the specifications!

Open the GUI (graphical User Interface) and you are faced with this scary looking beast! Don't worry, I'll take you through everything =). If you look at (**A**) you will notice there are three buttons. Clicking these swaps between the Mpeg2avi, Ac3Dec and VStrip programs. But first thing we should do is make sure the windows version button (**B**) is selected. Hopefully this should make it more compatible with Windows long filenames (i.e. by adding quotation marks).

Next check the box where it says Divx Auto (**C**). This automatically sets the bitrate you want to encode your Divx with so you don't have to keep putting in the same values for each vob file ;). Set it and press save.



1	Locate Files	Location
1.1	MPEG2AVI	C:\Mpeg2Avi\MPEG2AVI.EXE
1.2	Input	C:\Divx\Clip1.vob
1.3	Output AVI	C:\Divx\output.avi

Command Line(s): "C:\Mpeg2Avi\MPEG2AVI.EXE" -b "C:\Divx\Clip1.vob" -f2 -q0 -r2 -3X 520 -3Y 392 -1 512 384 -o8 "C:\Divx\output.avi"

fps	-f2 (25.000 = 25/1)
Postfilter Quality	-q0 (Hi Quality)
Reference Quality	-r2 (16-bit MMX Chen iDCT)
IVTC frame-offset	OFF
Output	-o8 (AVI - YV12 - 4:2:0) for DivX ;-)

		Start		End
-@	□ Open VOB bitstream at LBA			
-#	□ Frames Range		=>	

Profile: PAL 4:3 (Full Frame) -> DivX

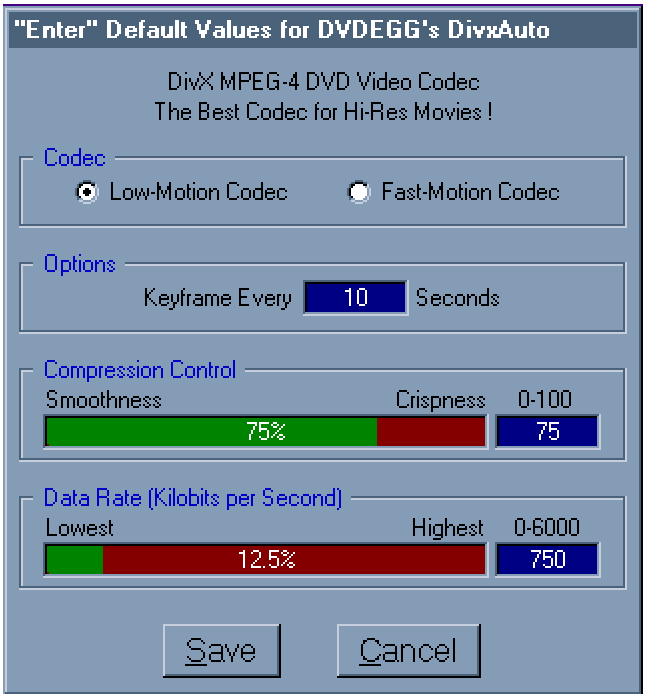
● PAL
○ NTSC

● Manual
○ Maintain Ratio
○ 4:3
○ 16:9 (1.85:1)
○ 16:9 (2.35:1)

		X		Y
-1	☑ Output Cropping	512	x	384
-2	□ Half Res. Mode	OFF		
-3X	☑ Downsizer X	520		
-3Y	☑ Downsizer Y	392		

☑ DivX Auto

□ and Exit



"Enter" Default Values for DVDEGG's DivxAuto

DivX MPEG-4 DVD Video Codec
The Best Codec for Hi-Res Movies !

Codec

● Low-Motion Codec ○ Fast-Motion Codec

Options

Keyframe Every 10 Seconds

Compression Control

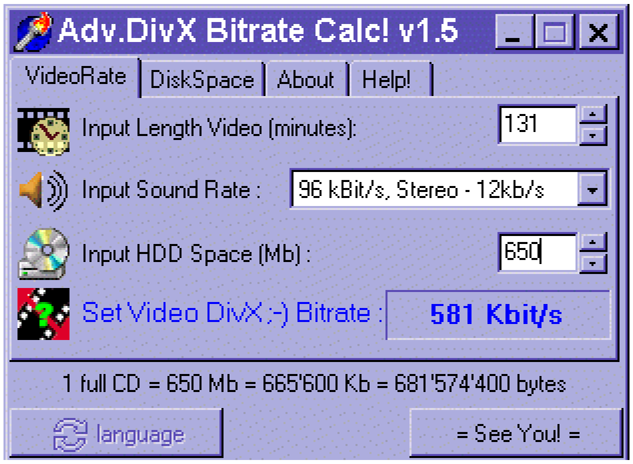
Smoothness 75% Crispness 0-100 75

Data Rate (Kilobits per Second)

Lowest 12.5% Highest 0-6000 750

For getting the best setting you can look at my article: "Best Divx Quality & Bitrate Guide" in the appendix. But in short, use a bitrate calculator to determine the best bitrate to put in here. I'm using the 'Advanced Divx Bitrate Calc! Version 1.5'. It seems to give a more reliable amount and will not usually make a movie larger than you intend.

Just put the DVD movie length in minuets in the top box. Select the audio bitrate you want to use. Then it will tell you the amount to use for the Low Motion codec! In the picture below the calculator says 581 so you would move the 'Data Rate slider bar to 581 instead of the 750 it is in the above picture. As for keyframes, I tend to use a keyframe every 1 second, but you can use one every 10 seconds to save more memory. Then set the 'Crispiness to 75% and you are done!



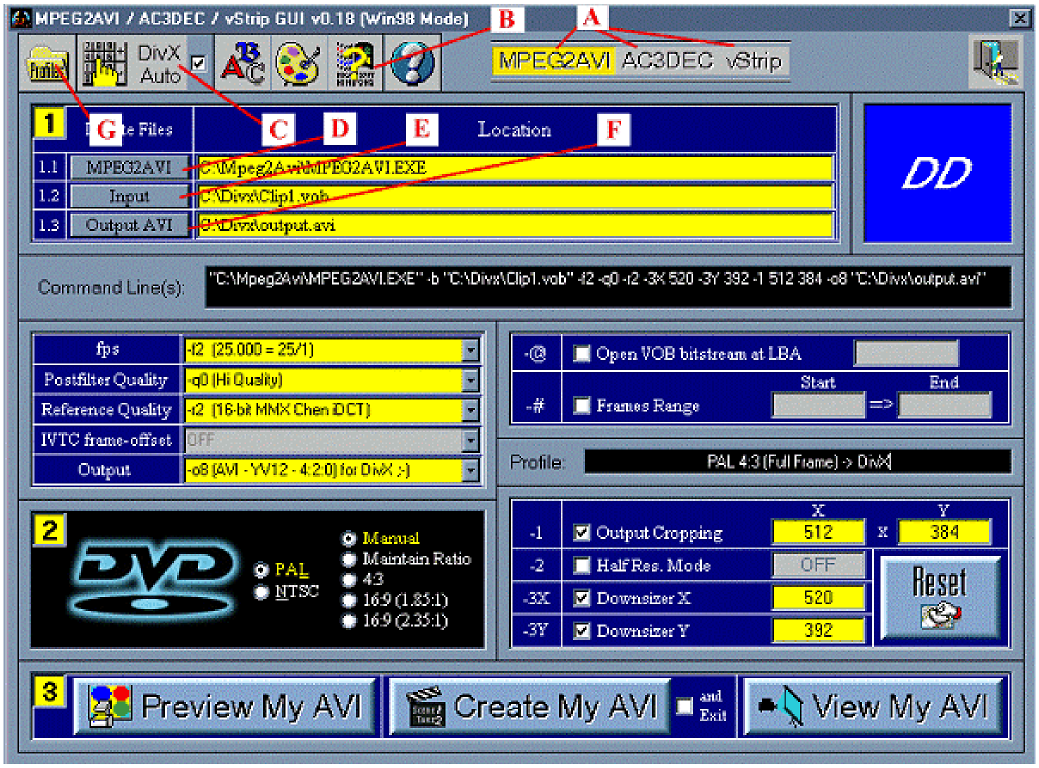
Input Length Video (minutes): 131

Input Sound Rate : 96 kBit/s, Stereo - 12kb/s

Input HDD Space (Mb) : 650

Set Video DivX ;-) Bitrate : 581 Kbit/s

1 full CD = 650 Mb = 665'600 Kb = 681'574'400 bytes



	[illegible]e Files	Location
1.1	MPEG2AVI	C:\Mpeg2Avi\MPEG2AVI.EXE
1.2	Input	C:\Divx\Clip1.vob
1.3	Output AVI	C:\Divx\output.avi

Command Line(s):

```
"C:\Mpeg2Avi\MPEG2AVI.EXE" -b "C:\Divx\Clip1.vob" -f2 -q0 -r2 -3X 520 -3Y 392 -1 512 384 -o8 "C:\Divx\output.avi"
```

fps	-f2 (25.000 = 25/1)
Postfilter Quality	-q0 (Hi Quality)
Reference Quality	-r2 (16-bit MMX Chen iDCT)
IVTC frame-offset	OFF
Output	-o8 (AVI - YV12 - 4:2:0) for DivX ;-)

-@	☐ Open VOB bitstream at LBA		
		Start	End
-#	☐ Frames Range		=>

Profile: PAL 4:3 (Full Frame) -> DivX

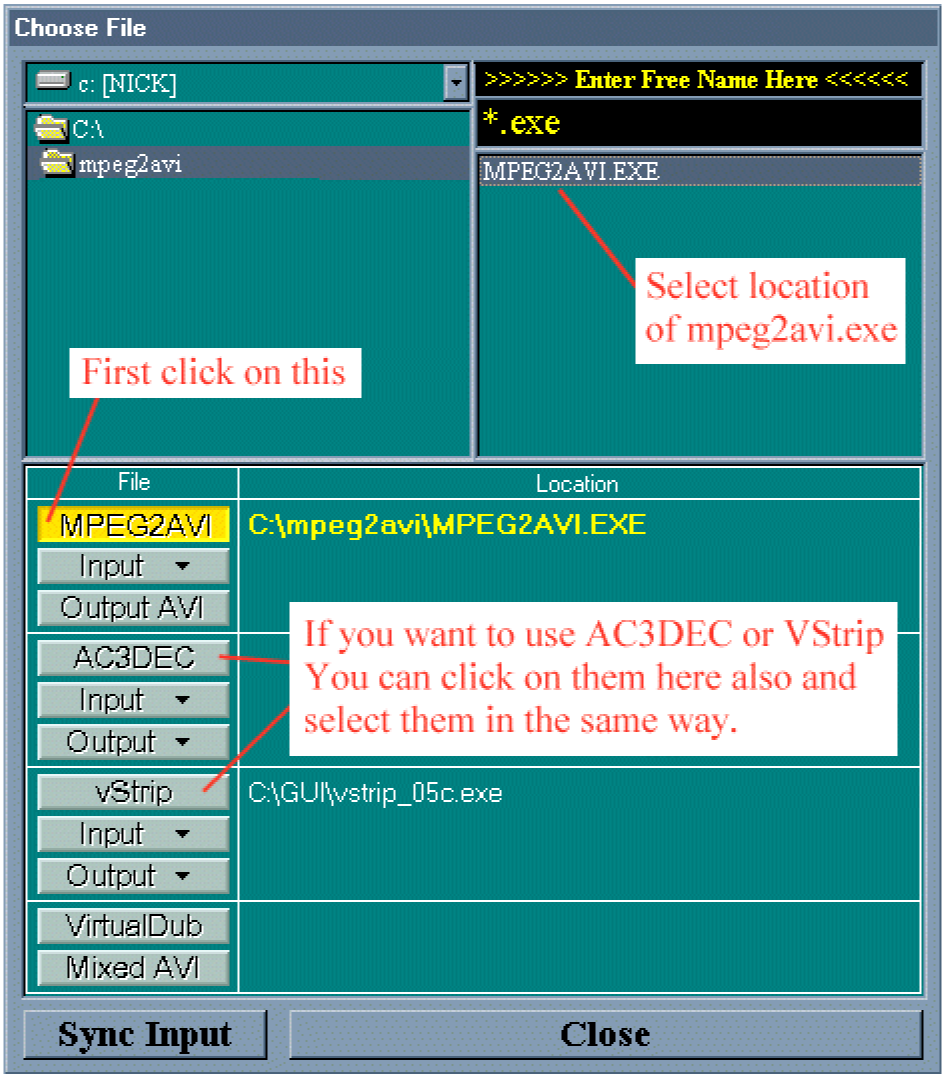
● PAL
○ NTSC

● Manual
○ Maintain Ratio
○ 4:3
○ 16:9 (1.85:1)
○ 16:9 (2.35:1)

		X		Y
-1	☑ Output Cropping	512	x	384
-2	☐ Half Res. Mode	OFF		
-3X	☑ Downsizer X	520		
-3Y	☑ Downsizer Y	392		

☐ and Exit

Back to the GUI. The GUI just tells Mpeg2avi what to do so it must know where you put the Mpeg2avi program on your computer. So browse for the location of the actual Mpeg2avi.exe by clicking button (**D**).



*.exe

MPEG2AVI.EXE

C:\mpeg2avi\MPEG2AVI.EXE

C:\GUI\vstrip_05c.exe

Select location of mpeg2avi.exe

First click on this

If you want to use AC3DEC or VStrip You can click on them here also and select them in the same way.

LIST FILES

Mpeg2avi can encode your ripped DVD Vob files in two ways. Firstly, it can select an individual Vob file and encode it. But there are problem with encoding individual Vob files if they are parts of a whole movie. For a detailed explanation of this read my "Key Frames & Delta Frames Explained" section of the appendix.

Unless you only have one Vob file to encode, you must make a list file. This is easy to do and, in fact, if you used SmartRipper in "movie" mode, then one would have been created for you. To make a list file, open notepad and put the names of the files you want to encode all together in order like this:

C:\folder\vts_03_1.vob

C:\folder\vts_03_2.vob

C:\folder\vts_03_3.vob

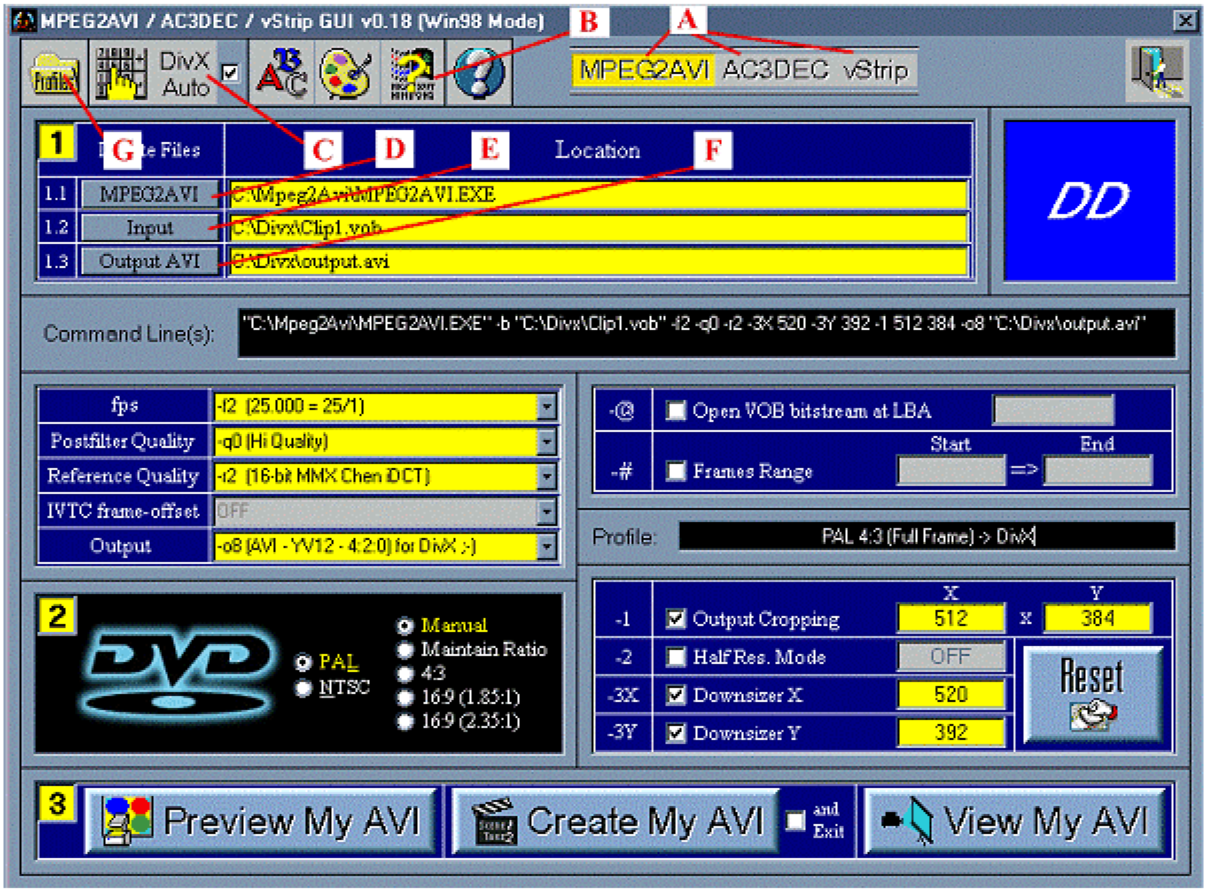
C:\folder\vts_03_4.vob

C:\folder\vts_03_5.vob

The colours are mine, of course, and the blue parts represent the location of the Vob files and the red parts represents the names of the Vobs you need to encode. You must save this text file with a .lst extension. To do this, just add quotation marks. So if you want to save it as movie.lst you choose > save and call it "movie.lst". Make sure it isn't saved as a something like movie.lst.txt or something like that though.

SELECTING THE VOBS

Back to the GUI. We can now browse to find the Vob or List file you want to encode from by pressing (**E**).



MPEG2AVI / AC3DEC / vStrip GUI v0.18 (Win98 Mode)

1.1	MPEG2AVI	C:\Mpeg2Avi\MPEG2AVI.EXE
1.2	Input	C:\Divx\Clip1.vob
1.3	Output AVI	C:\Divx\output.avi

Command Line(s):

```
"C:\Mpeg2Avi\MPEG2AVI.EXE" -b "C:\Divx\Clip1.vob" -f2 -q0 -r2 -3X 520 -3Y 392 -1 512 384 -o8 "C:\Divx\output.avi"
```

fps	-f2 (25.000 = 25/1)
Postfilter Quality	-q0 (Hi Quality)
Reference Quality	-r2 (16-bit MMX Chen IDCT)
IVTC frame-offset	OFF
Output	-o8 (AVI - YV12 - 4:2:0) for DivX ;-)

-@ □ Open VOB bitstream at LBA

-# □ Frames Range Start => End

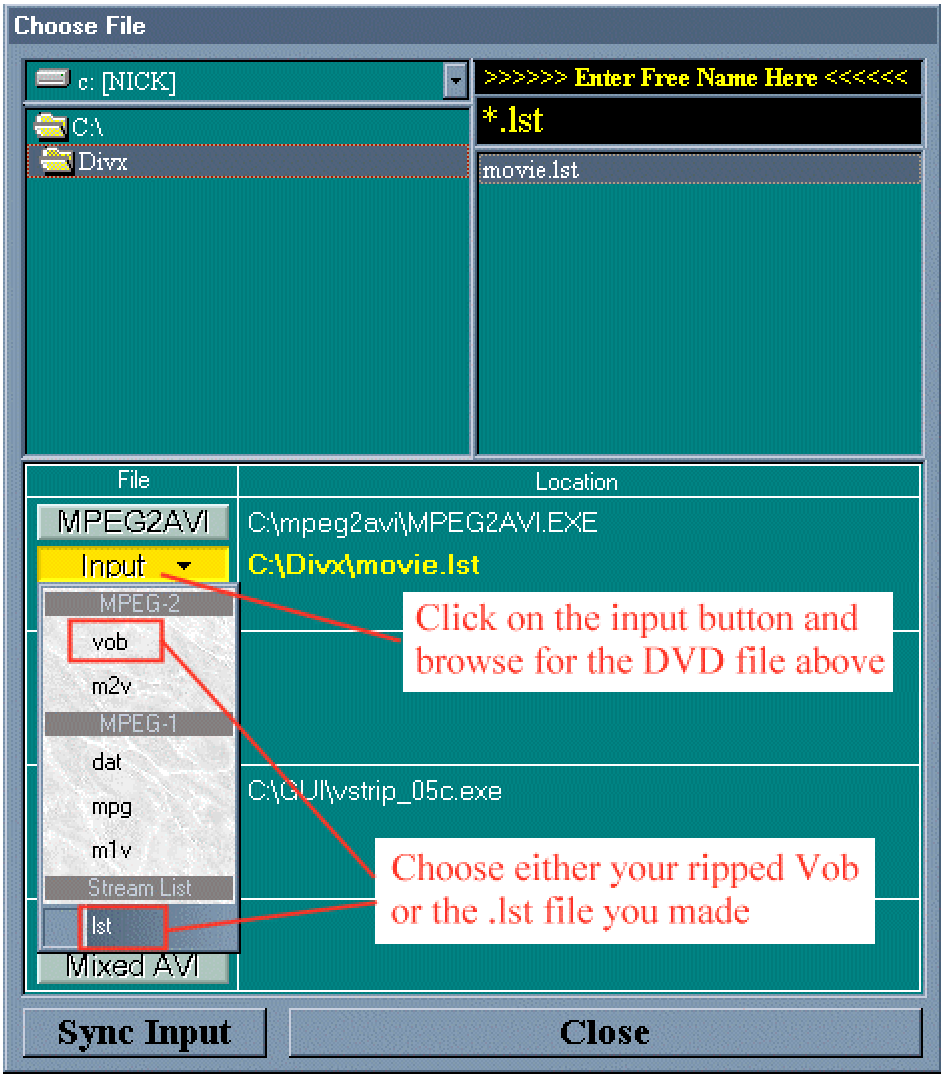
Profile: PAL 4:3 (Full Frame) -> DivX

● PAL
○ NTSC

● Manual
○ Maintain Ratio
○ 4:3
○ 16:9 (1.85:1)
○ 16:9 (2.35:1)

		X		Y
-1	☑ Output Cropping	512	x	384
-2	□ Half Res. Mode	OFF		
-3X	☑ Downsizer X	520		
-3Y	☑ Downsizer Y	392		

□ and Exit



*.lst

movie.lst

C:\mpeg2avi\MPEG2AVI.EXE

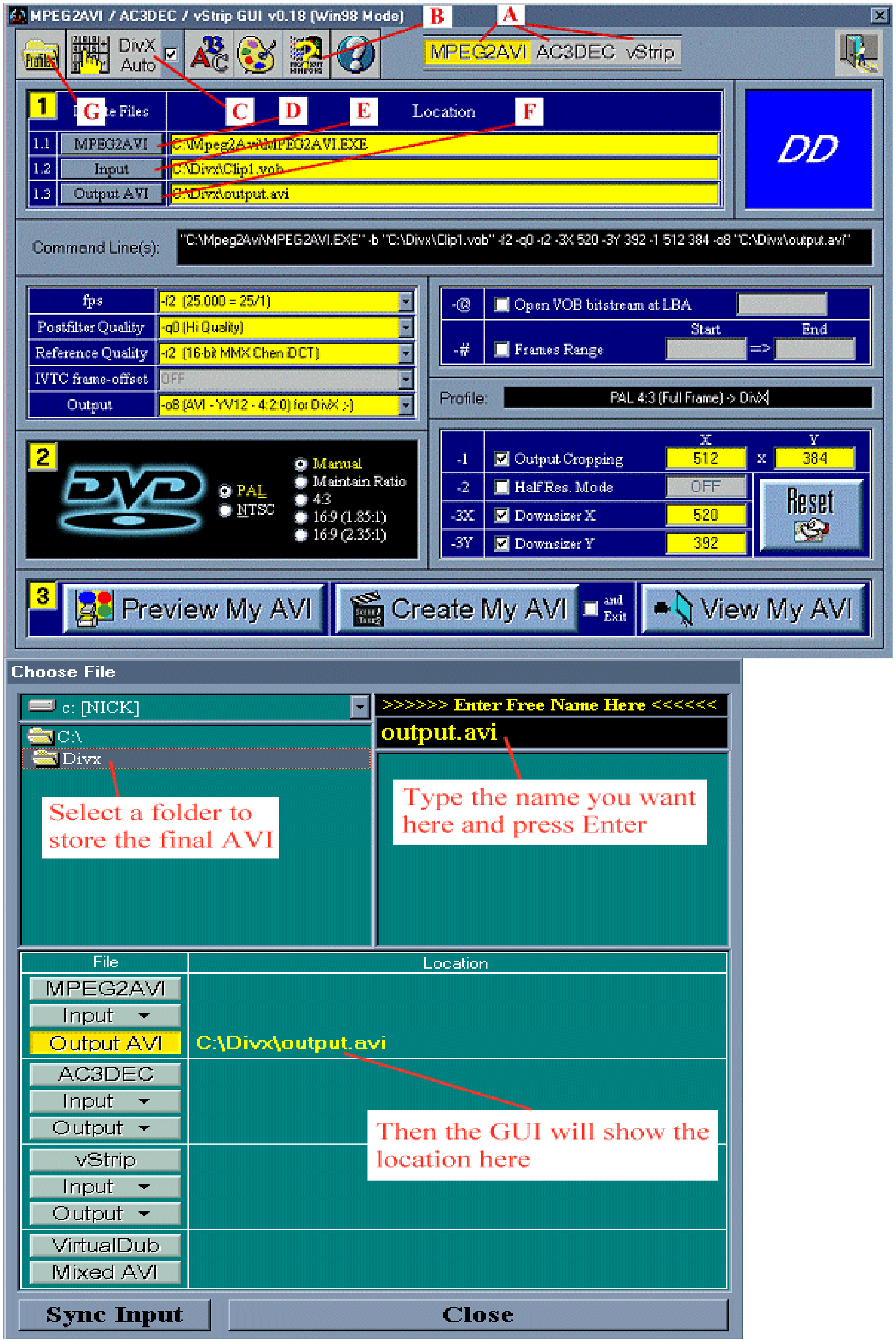
C:\Divx\movie.lst

Click on the input button and browse for the DVD file above

C:\GUI\vstrip_05c.exe

Choose either your ripped Vob or the .lst file you made

Finally we choose the output location and filename by clicking (**F**).



	File	Location
1.1	MPEG2AVI	C:\Mpeg2Avi\MPEG2AVI.EXE
1.2	Input	C:\Divx\Clip1.vob
1.3	Output AVI	C:\Divx\output.avi

Command Line(s):

```
"C:\Mpeg2Avi\MPEG2AVI.EXE" -b "C:\Divx\Clip1.vob" -I2 -q0 -r2 -3X 520 -3Y 392 -1 512 384 -o8 "C:\Divx\output.avi"
```

fps	-I2 (25.000 = 25/1)
Postfilter Quality	-q0 (Hi Quality)
Reference Quality	-r2 (16-bit MMX Chen iDCT)
IVTC frame-offset	OFF
Output	-o8 (AVI - YV12 - 4:2:0) for DivX ;-)

-@	□ Open VOB bitstream at LBA		
		Start	End
-#	□ Frames Range		=>

Profile: PAL 4:3 (Full Frame) -> DivX

◉ PAL
○ NTSC

◉ Manual
○ Maintain Ratio
○ 4:3
○ 16:9 (1.85:1)
○ 16:9 (2.35:1)

		X		Y
-1	☑ Output Cropping	512	x	384
-2	□ Half Res. Mode	OFF		
-3X	☑ Downsizer X	520		
-3Y	☑ Downsizer Y	392		

c: [NICK]

C:\
Divx

>>>>>> Enter Free Name Here <<<<<<

output.avi

Select a folder to store the final AVI

Type the name you want here and press Enter

File	Location
MPEG2AVI	
Input	
Output AVI	C:\Divx\output.avi
AC3DEC	
Input	
Output	
vStrip	
Input	
Output	
VirtualDub	
Mixed AVI	

Then the GUI will show the location here

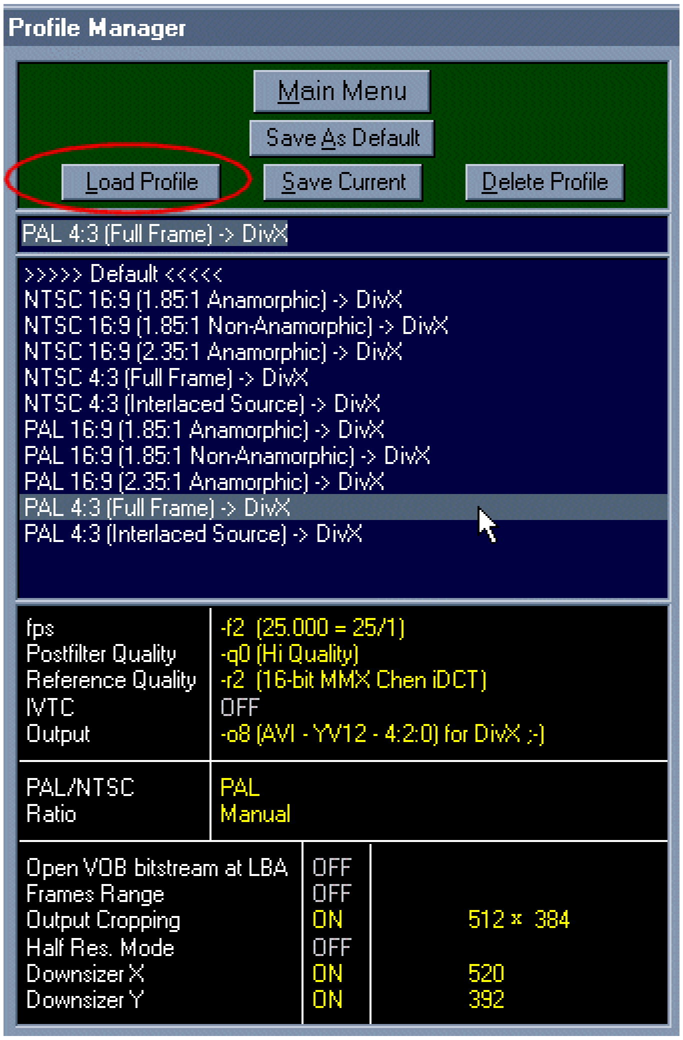
THE DECODING SETTING & PROFILES

The GUI has an option to load in and save encoding settings - called 'profiles. Many of you will probably be happy to just download a profile and use it. For this purpose I have added a link for you to download them here:

Popular Profiles

And here is how you'd load a profile:

Click on the Profiles icon (**G**) and up pops some pre-made and tested setting for Divx Choose the profile that you want to use i.e. NTSC or PAL; Anamorphic for Wide Screen 16:9 ratio or 4:3 for Normal TV screen ratio etc., then press Load Profile.



PAL 4:3 (Full Frame) -> DivX

>>>>> Default <<<<<
NTSC 16:9 (1.85:1 Anamorphic) -> DivX
NTSC 16:9 (1.85:1 Non-Anamorphic) -> DivX
NTSC 16:9 (2.35:1 Anamorphic) -> DivX
NTSC 4:3 (Full Frame) -> DivX
NTSC 4:3 (Interlaced Source) -> DivX
PAL 16:9 (1.85:1 Anamorphic) -> DivX
PAL 16:9 (1.85:1 Non-Anamorphic) -> DivX
PAL 16:9 (2.35:1 Anamorphic) -> DivX
PAL 4:3 (Full Frame) -> DivX
PAL 4:3 (Interlaced Source) -> DivX

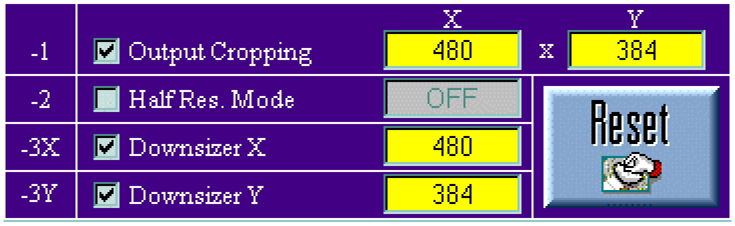
fps	-f2 (25.000 = 25/1)
Postfilter Quality	-q0 (Hi Quality)
Reference Quality	-r2 (16-bit MMX Chen iDCT)
IVTC	OFF
Output	-o8 (AVI - YV12 - 4:2:0) for DivX ;-)
PAL/NTSC	PAL
Ratio	Manual

Open VOB bitstream at LBA	OFF	
Frames Range	OFF	
Output Cropping	ON	512 x 384
Half Res. Mode	OFF	
Downsizer X	ON	520
Downsizer Y	ON	392

But I never use profiles and I don't think you need to either. Throughout my guides I have shown you how to configure everything yourselves. The final few settings on Mpeg2avi are no more difficult than anything else, and other peoples profiles do not always give the best settings for your movie anyway.

RESIZE

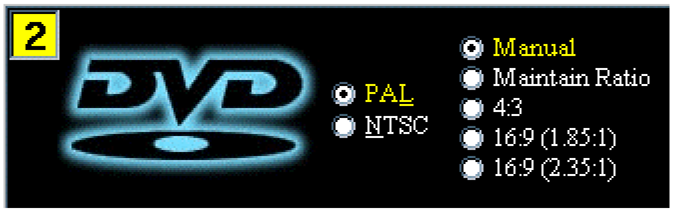
This section lets you say how you want the movie resized or cropped. Instead of explaining this here, I have done so in much greater detail in the article *"Resizing the Video in Mpeg2avi".*



		X		Y
-1	☑ Output Cropping	480	x	384
-2	□ Half Res. Mode	OFF		
-3X	☑ Downsizer X	480		
-3Y	☑ Downsizer Y	384		

BITS 'n' BOBS

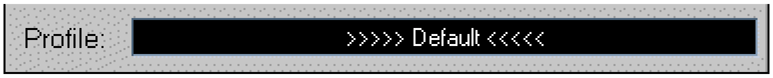
This bit below has some pre-defined crop and resizing settings. Just click on them and see what you think - I don't use them.



● PAL
○ NTSC

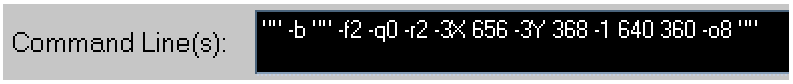
● Manual
○ Maintain Ratio
○ 4:3
○ 16:9 (1.85:1)
○ 16:9 (2.35:1)

The box below just tells you which profile you are using.



Profile: >>>>> Default <<<<<

The command line section below just displays what commands the GUI is sending to Mpeg2avi.

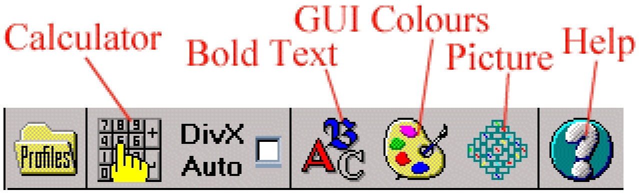


Command Line(s):

```
"" -b "" -f2 -q0 -r2 -3X 656 -3Y 368 -1 640 360 -o8 ""
```

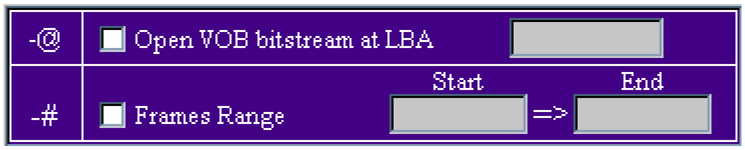
Most of the options below are cosmetic and useless for anything but making the GUI look nice. The Calculator is ok, but confusing so I use another. The Bold text option just turns text bold. You can change the GUI Colours or use a background Picture. The

Picture option lets you choose a background picture for the GUI. And the Help doesn't help at all yet =).



The "open VOB bitstream at LBA" option won't be needed unless there is lots of green or pink garbage at the start of your Vob file. Choosing a new Logical Block Address (LBA) basically jumps 2048 bytes forward to allows you to skip by the garbage. If Mpeg2avi cannot find a system header it will automatically scan to the next LBA until one is found.

Below the LBA option is the Frames Range option. This will allow you skip to a frame in the movie. To do this put the frame number in the 'Start' box. (25 frames make 1 second for PAL and 23.976 or 29.970 make 1 second for NTSC). Putting a number in the 'End' box tells Mpeg2avi what frame to stop encoding.

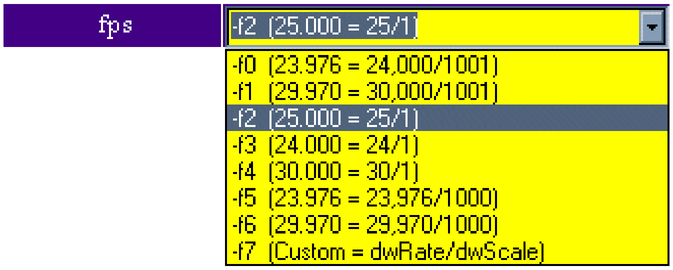


-@ □ Open VOB bitstream at LBA

-# □ Frames Range Start => End

FINAL SETTINGS

Set the framerate you wish to encode your movie with. PAL will almost always use - f2 25.000 fps except some DVD specials. NTSC will use either -f5 23.976 or -f6 29.970 or even -f3 24.000 fps depending on the movie; but most of the time, I am told, it will be -f5 23.976 fps for NTSC.

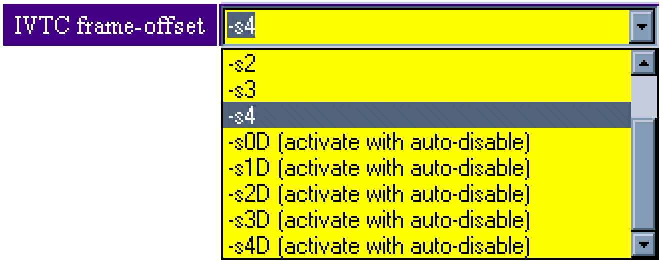


fps -f2 (25.000 = 25/1)

-f0 (23.976 = 24,000/1001)
-f1 (29.970 = 30,000/1001)
-f2 (25.000 = 25/1)
-f3 (24.000 = 24/1)
-f4 (30.000 = 30/1)
-f5 (23.976 = 23,976/1000)
-f6 (29.970 = 29,970/1000)
-f7 (Custom = dwRate/dwScale)

IVTC SETTINGS

This option is designed to handle interlace problems. For a detailed explanation of this problem read my article: "Video Formats: NTSC & PAL / Telecine". Most PAL DVDs don't have interlace problems except in the specials section. So far I haven't managed to solve this interlace problem. The Mpeg2avi ivtc text files recomends using the -s4 setting to telecine NTSC movies, so try that and see what happens. If thats no good you'll have to Flask Mpeg the movie. If you don't have interlace problems at all just turn this option OFF.

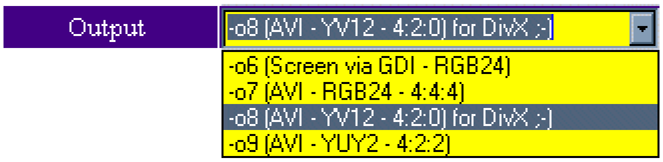


IVTC frame-offset: -s4

-s2
-s3
-s4
-s0D (activate with auto-disable)
-s1D (activate with auto-disable)
-s2D (activate with auto-disable)
-s3D (activate with auto-disable)
-s4D (activate with auto-disable)

OUTPUT SETTINGS

Because Mpeg2avi is designed to output to all kinds of codec's it allows options to tell it what format to give them. For Divx it should always be -o8 YV12.

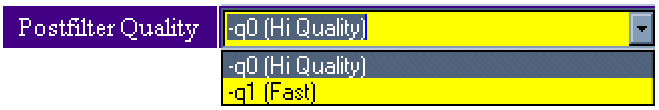


Output: -o8 (AVI - YV12 - 4:2:0) for DivX ;-)

-o6 (Screen via GDI - RGB24)
-o7 (AVI - RGB24 - 4:4:4)
-o8 (AVI - YV12 - 4:2:0) for DivX ;-)
-o9 (AVI - YUY2 - 4:2:2)

POSTFILTERING

Just choose Hi Quality.

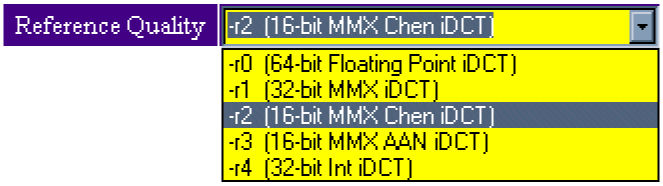


Postfilter Quality: -q0 (Hi Quality)

-q0 (Hi Quality)
-q1 (Fast)

REFERENCE QUALITY

Inverse Discrete Cosine Transform (iDCT) determines the accuracy of how certain aspects of an mpeg file are converted back into its original image. Since we are going to recompress and probably shrink the Divx anyway, it is probably the best compromise between speed and image quality to use -r2 (16-bit MMX Chen iDCT). Anything else is overkill and in fact this is the most bug free option anyway.



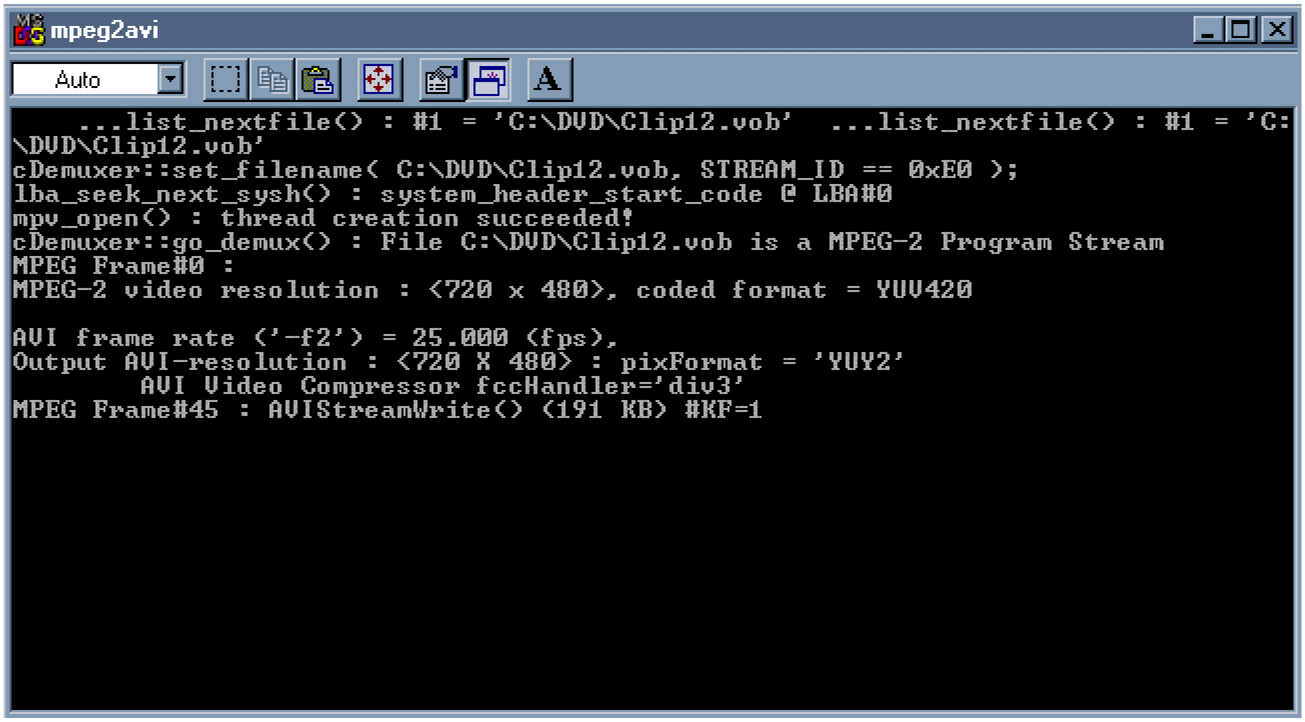
Reference Quality: -r2 (16-bit MMX Chen iDCT)

-r0 (64-bit Floating Point iDCT)
-r1 (32-bit MMX iDCT)
-r2 (16-bit MMX Chen iDCT)
-r3 (16-bit MMX AAN iDCT)
-r4 (32-bit Int iDCT)

THE FINAL VIDEO

Finally press the Create my AVI button!

A window like the one below will appear and the movie will encode.



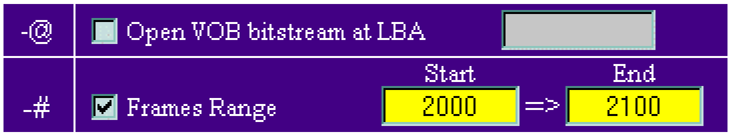
```
    ...list_nextfile() : #1 = 'C:\DVD\Clip12.vob'   ...list_nextfile() : #1 = 'C:
\DVD\Clip12.vob'
cDemuxer::set_filename( C:\DVD\Clip12.vob, STREAM_ID == 0xE0 );
lba_seek_next_sysh() : system_header_start_code @ LBA#0
mpv_open() : thread creation succeeded!
cDemuxer::go_demux() : File C:\DVD\Clip12.vob is a MPEG-2 Program Stream
MPEG Frame#0 :
MPEG-2 video resolution : <720 x 480>, coded format = YUV420

AVI frame rate ('-f2') = 25.000 (fps),
Output AVI-resolution : <720 X 480> : pixFormat = 'YUY2'
        AVI Video Compressor fccHandler='div3'
MPEG Frame#45 : AVIStreamWrite() (191 KB) #KF=1
```

That's it! =) The next step is to extract the audio from the DVD and multiplex to your DivX.

Resizing and Cropping a DivX in Mpeg2avi

If you want to fit a movie into the smallest possible size, cropping can do a lot to help out. The most vital thing to remember for optimal compression is to always crop a few pixels *into* the image and delete *all* of the black bars. Mpeg compression works best on blurry images; so if a hard black line is seen at the bottom or top of your cropped movie it will not compress as effectively. In fact, if you cannot crop into the image then I'd say don't bother cropping at all.

Because Mpeg2avi doesn't have a preview box we will have to do a few quick tests to see what we will get. The best way to do this is to set it up to encode about 100 frames. Mpeg2avi has a start and end from such and such a frame option. So we can put in where we want to test the movie and how many frames to encode. This is done like this:



-@ □ Open VOB bitstream at LBA

-# ☑ Frames Range Start 2000 => End 2100

This will make it jump 2000 frames into the movie and encode 100 frames of it. Then we can see what the picture will look like. It is also a good idea to start from the second Vob so we don't get the start credits. Now, every time we want to see what the image looks like we simply hit the 'Create my AVI' button and in a few seconds we can check it.

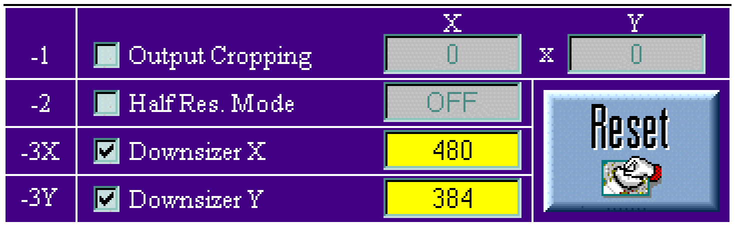
RESIZING A 4:3 MOVIE

Many get confused by Mpeg2avi's crop and resize settings, but really they are very easy enough to use. The only down side is you don't have an instant preview to work from like Flask Mpeg. As always we are using Danni's amazingly cool GUI to do this, so you will not need to mess around with the DOS prompt =).

Okay, lets do this: assume we are resizing a 720 x 576 PAL DVD to 480 x 384 this is a 4:3 aspect ratio of 1.25:1. Mpeg2avi's resizing options are:

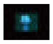
> **Downsizer X:** This option shrinks the width smaller according to the size you set it.
>
> **Downsizer Y:** This shrinks the height according towat you set it.

It's important to remember that the downsizer options *only* shrink the picture itself. So, for example, in the picture below the original movie was 720 x 576 and I used a downsizer of:



		X		Y
-1	□ Output Cropping	0	x	0
-2	□ Half Res. Mode	OFF		
-3X	☑ Downsizer X	480		
-3Y	☑ Downsizer Y	384		

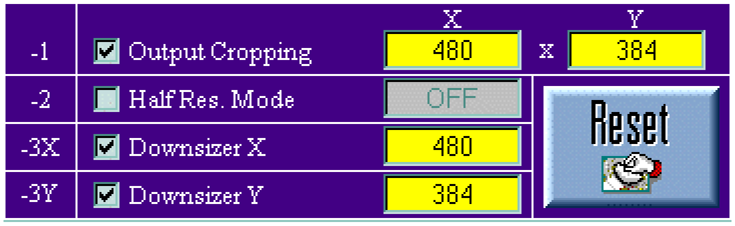
Now, the movie below is *still* 720 x 576 but the picture *inside* it is 480 x 384!



To get the movie perfect we must crop out the black areas too. The Mpeg2avi crop functions are:

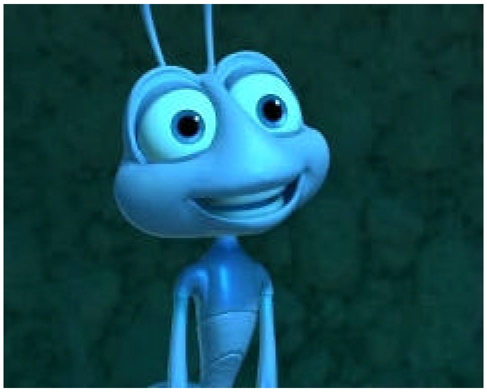
> **Output Cropping X:** crops the width of the whole window. Note that its doesn't just take a bit from one size or the other but takes it evenly.
>
> **Output Cropping Y:** same again, but it crops the top and bottom.

So to crop out the black areas we just put the same resize values in the cropping sections as are in the resize, like this:



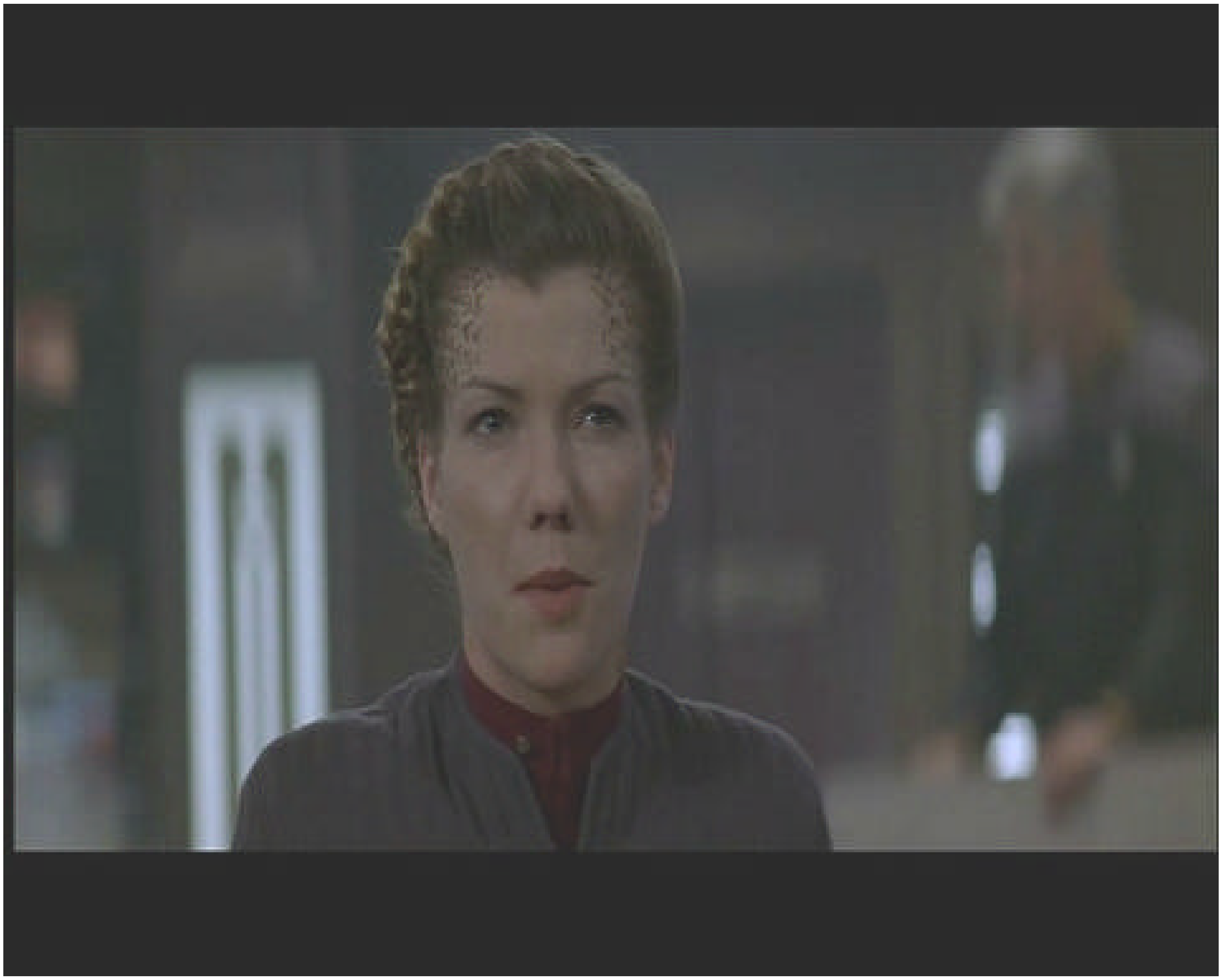
		X		Y
-1	☑ Output Cropping	480	x	384
-2	□ Half Res. Mode	OFF		
-3X	☑ Downsizer X	480		
-3Y	☑ Downsizer Y	384		

The final picture will look like this:

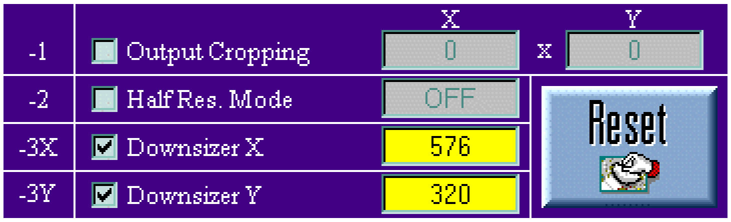


CROPPING WIDESCREEN MOVIES

The same process is used again, but because Widescreen movies don't always use the same sizes we may have to "tweek" the measurements a little to get the best posible cropping. Lets say we have a 2.35:1 Anamorphic movie. Its 720 x 576 and looks like this:

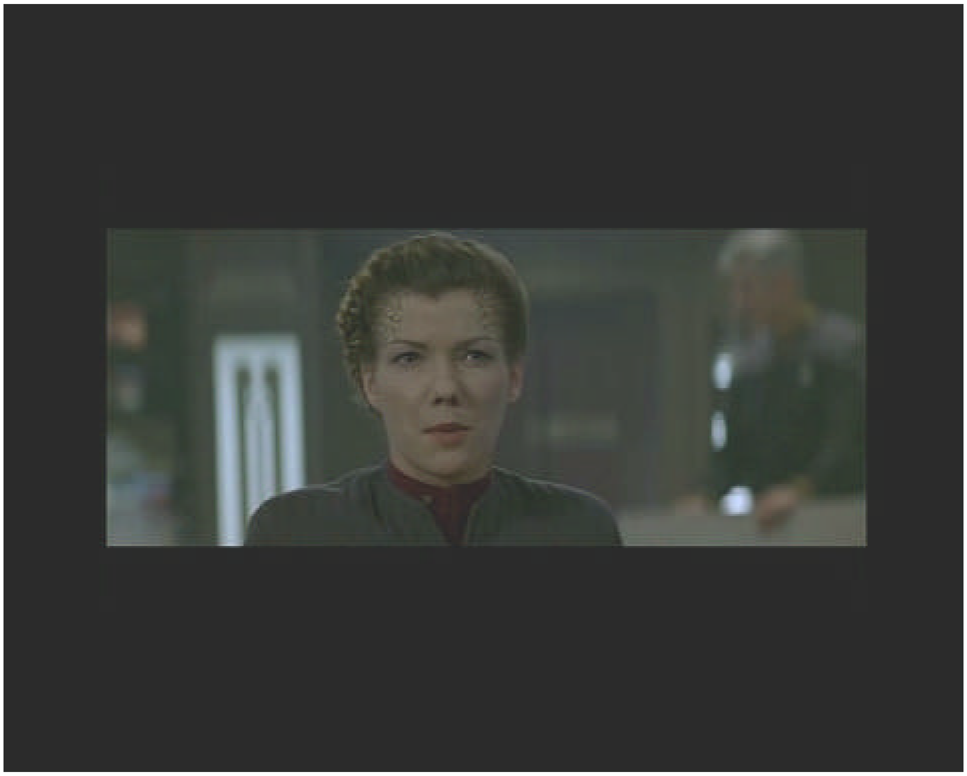


As you can see anamorphic DVD's have squashed pictures. For an explanation of this subject be sure to read the section called *"Anamorphic DVD's & Aspect Ratios"* in the my appendix. But to cut a long story short we will resize the image to an aspect ratio of 1.80:1 and I want to resize my movie to about 576 x 320. So I put:

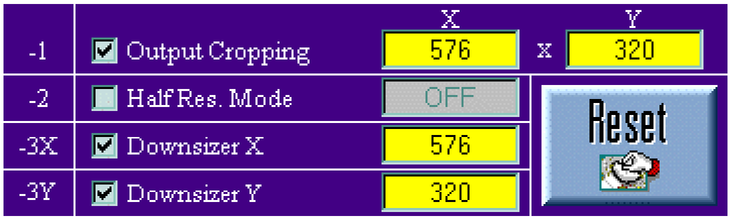


		X		Y
-1	□ Output Cropping	0	x	0
-2	□ Half Res. Mode	OFF		
-3X	☑ Downsizer X	576		
-3Y	☑ Downsizer Y	320		

When I check the move it is now in perfect aspect ratio:

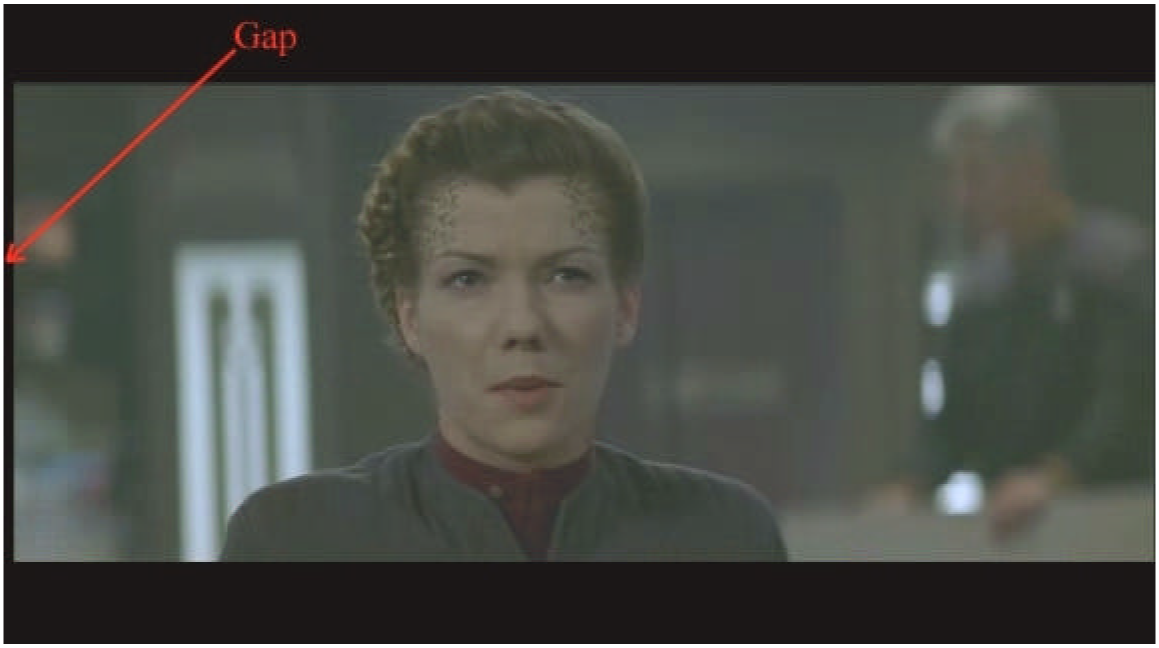


Next is to crop out the picture. Since we know the correct width is 576 across we can set the cropping to 576. We also know that 320 will be higher than the picture because we have resized the picture to Y=320. So lets put that in too like this:

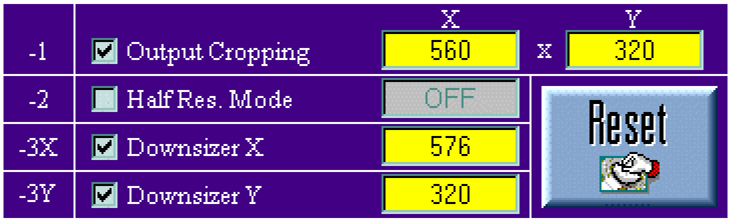


		X		Y
-1	☑ Output Cropping	576	x	320
-2	□ Half Res. Mode	OFF		
-3X	☑ Downsizer X	576		
-3Y	☑ Downsizer Y	320		

And we get this:

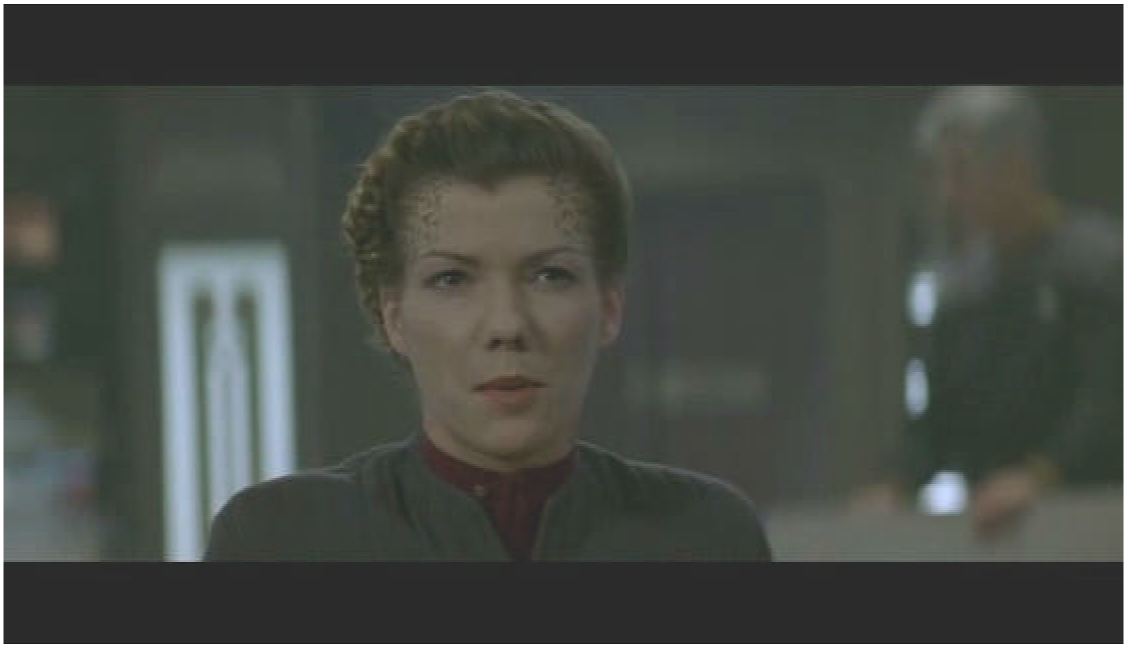


But because the DVD had a gap on the left, this is going to detract from our otherwise perfect cropping. All it needs is to bring the sides in 16 pixels and it will look nice. So instead of 576 we can crop it to 560 or we could resize the image 16 pixels to so it was 592 x 336. So we:

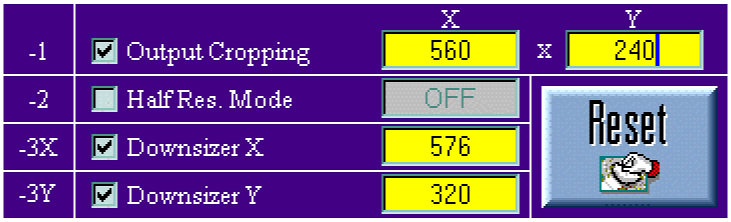


		X		Y
-1	☑ Output Cropping	560	x	320
-2	□ Half Res. Mode	OFF		
-3X	☑ Downsizer X	576		
-3Y	☑ Downsizer Y	320		

And we get the following:

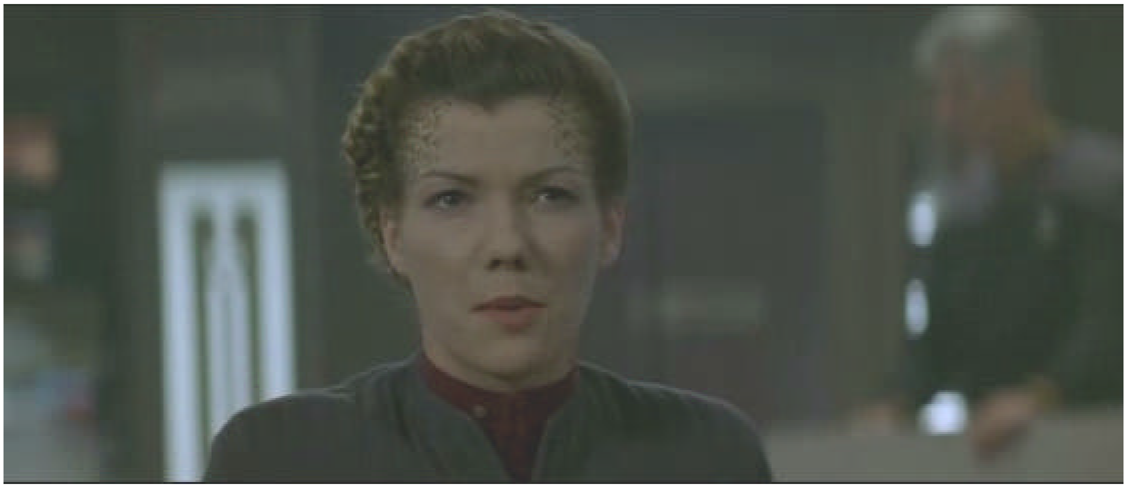


Finally we keep cropping a little at a time until we get the bottom and top right. At the moment the cropping is Y=320. We want that to be smaller so we take away 16 pixel to make 304 (always resize by multiples of 16 pixels). Thats no good, so we take off 16 more until its 288 and try that. Then we try 272, then 256 and finally we find that 240 is perfect! So we put:



		X		Y
-1	☑ Output Cropping	560	x	240
-2	□ Half Res. Mode	OFF		
-3X	☑ Downsizer X	576		
-3Y	☑ Downsizer Y	320		

And the movie is cropped perfectly and has perfect aspect ratio!



I'm not joking when I say that cropping like this really does improve the picture quality. More bitrate can be assigned to the image and less to keeping those sharp black lines! Forget what all those morons say about how a DivX can rip full DVD resolution! Even if you want to use 2 or 3 CD's to make them! Video CD's are 352 x 240 pixels and look almost as good as VHS videos when played on a TV. Our cropped DivX is exactly the same height and almost double the width, and, when done right can achive as good or better quality than a VCD and yet still fit onto a single CD. Don't use crazy large sizes - do it right!

Final Warning

Finally, its only fair to warn you that, athough in my opinion cropping improves the image quality of the DivX rip by a lot, before you decide that cropping is the best way to go, you must consider these four facts:

1. A cropped movie is sometimes harder to convert into another format. This is because you may need to re-add the black bars to the top and bottom of the movie first or the movie may be stretched out of shape.

2. A cropped DivX movie will play at the wrong aspect ratio in PowerDVD and some other Video players. On the other hand, Media Player, MicroDVD and many other players will play back a cropped movie perfectly.

3. VCD's and SVCD's cannot be cropped if they are to be played in a standalone DVD player because it will not accept them.

4. Finally, Mpeg-4 files (and DVD files for that matter) have problems playing back on some hardware if they are not encoded in sizes that can be divided by 32. This means the Matrox G400 or the Nvidia GeForce would probably have problems outputing it to TV.

For example:
A 528 pixels wide size divided by 32 = 16.5. This is not a multiple of 32 and so may have trouble.
But a 576 pixel wide size divided by 32 = 18. This is a multiple of 32 and will play back perfectly.

DVD Audio Extraction with GraphEdit

Graphedit is by far the best method for extracting audio from a DVD. The quality is perfect, the sound doesn't need normalizing (i.e. increasing the volume). It rips 10 times faster than ac3dec and there is almost never any synch problems! As always the DVD Vob files need to be decrypted to your hard disk or GraphEdit will not be able to read them.

Before I start here are the things you will need:

GraphEdit Build 980303

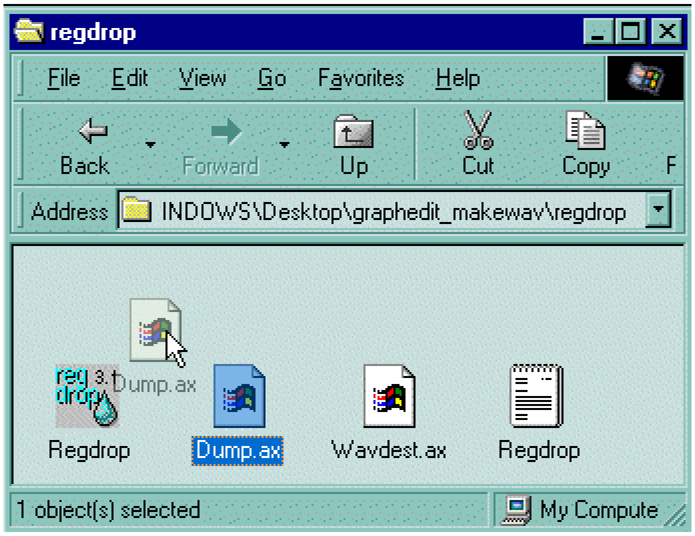
Ligos Mpeg-2 Filter

WinDVD Audio Filter

I-Media Multiple Audio Filter

RegDrop

Put GraphEdit in a folder somewhere and register all the filters with RegDrop. This is easy, you simply drag the .ax file over the RegDrop program and your files are registered. Like this:


regdrop
File Edit View Go Favorites Help
Back Forward Up Cut Copy F
Address INDOWS\Desktop\graphedit_makewav\regdrop
Dump.ax
Regdrop
Dump.ax
Wavdest.ax
Regdrop
1 object(s) selected
My Compute

Or you can just use the DOS line command Regsvr32 (ie. regsvr32 "C:\folder\filename.ax").

Install all of the following files that you downloaded:

Wavedest.ax

Dump.ax

Mpeg2Decoder.ax

Iviaudio.ax

MAKING A LIST FILE (*.LST)

Next we need to have a text file with a list of all the DVD movie files that we wish to extract the sound from. If you used SmartRipper to extract the movie files in its 'movie mode', it would have created a list file for you called something like vts_0x.lst. If you used Mpeg2avi to encode your movie you should still have one.

If not, open notepad or wordpad or whatever text editor you like and put the names of the vob files you want to extract the audio from on it, like this:

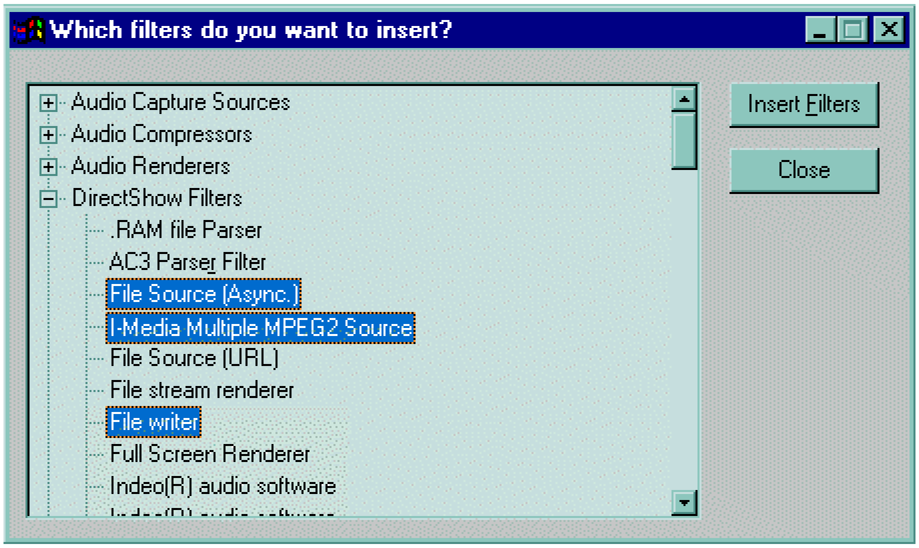
```
C:\folder\vts_03_1.vob
C:\folder\vts_03_2.vob
C:\folder\vts_03_3.vob
C:\folder\vts_03_4.vob
C:\folder\vts_03_5.vob
```

Save the file as whatever name you like and put a .lst extension on it. For example, you can save the file as "dvd.lst", but please remember to add the quotation marks to force the text editor to save the file as a .lst extension, otherwise it will save it as dvd.lst.txt or dvd.lst.doc or whatever and it wont work! We do this so that, instead of choosing just one individual .Vob file at a time, we can just select the dvd.lst file instead! Then all the files written in it will be encoded and joined together into one whole file!

GETTING THE AUDIO

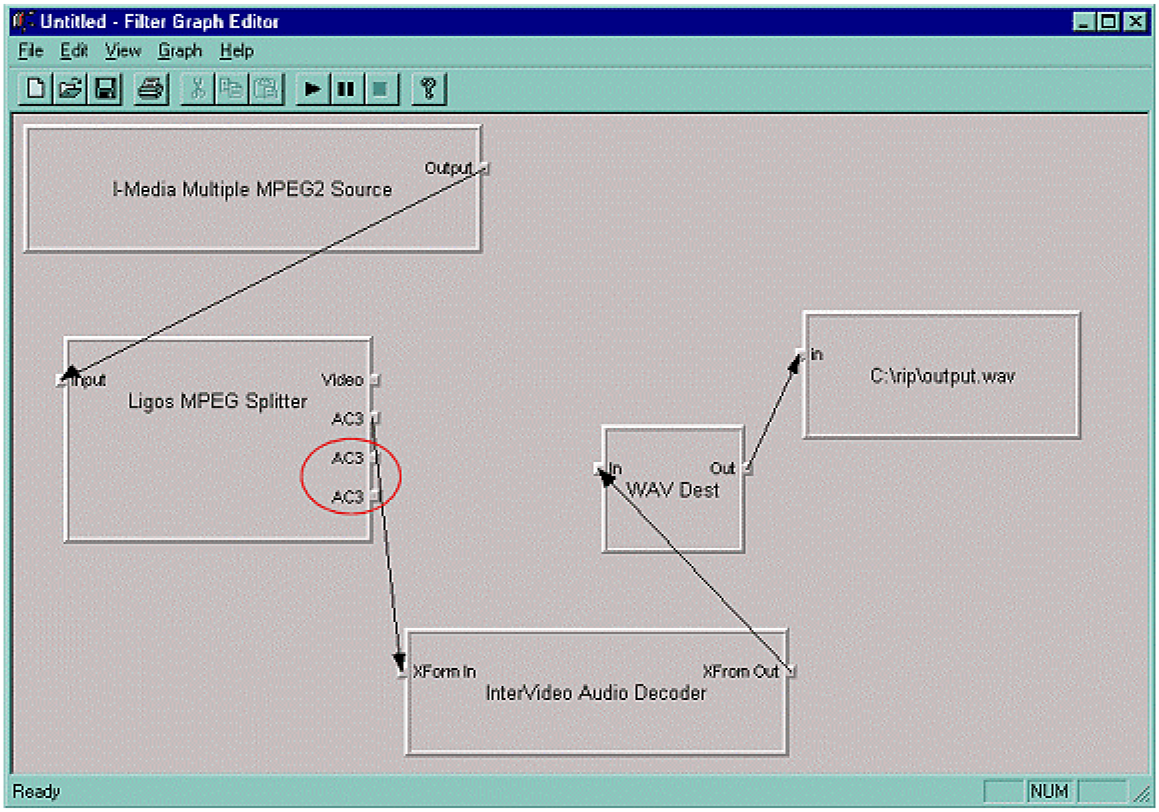
GraphEdit looks baffling to the newbie because its vague about what it says it does! Basically, all windows media files need codec's to play them; that is why you need to install the DivX codecs so you can play or encode to DivX. GraphEdit lets you connect one decoder directly to another by dragging a line between them. Each codec is represented by a box with the codec's name writen on it. By connecting the correct encoder and decoder boxes between your Vob file and the output 'File Writer' filer you can convert any Vob audio to a Wave.

So, open up Graphedit. Go to Graph > insert filters and up will pop the following box. All of the files you need will come under the category of Directshow Filters because they are designed to play Audio and Video.

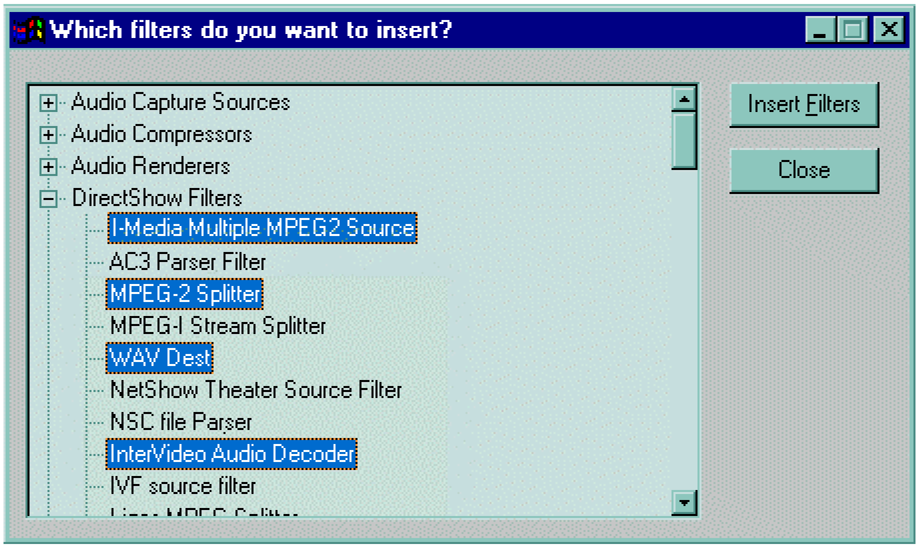


First, insert the 'I-Media Multiple MPEG2 Source' filter. It will ask you for the location of the .lst file we made and called dvd.lst. Select it and press open. Then the first box (top left of the picture below) will appear. If you only have one Vob file, you can use 'File Source (Async.)' instead of a dvd.lst file.

Next, tell it the output location by inserting the 'File Writer' filter. Again another browser window appears and ask you where you want the audio saved. You can see this box at the far right of the picture below saying C:\rip\output.wav.



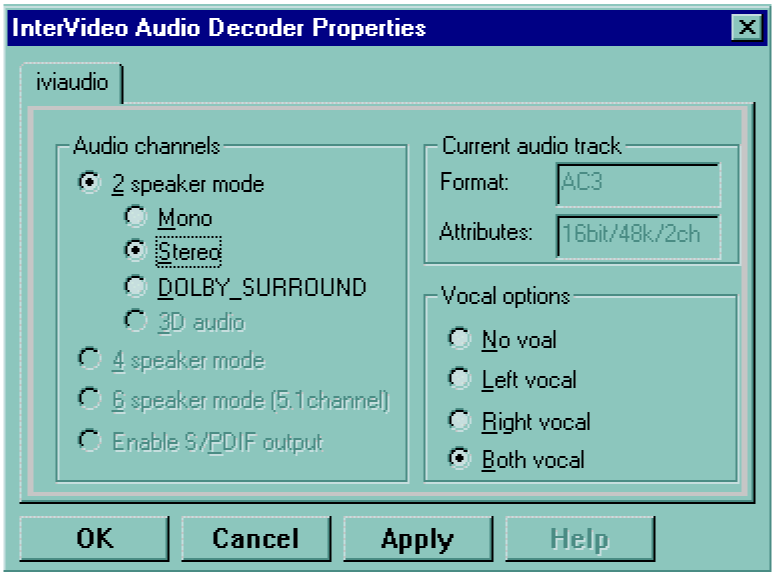
Next we must choose all those conversion filters placed in-between the in and out. Choose the following: MPEG-2 Splitter (or Ligos Mpeg Splitter); WAV Dest and InterVideo Audio Decoder. Finally, you must drag the arrow from the first 'Output' box to the Mpeg-2 box; then drag from the AC3 peg to the Intervideo Audio Decoder; then finally again to the in box where you save your file. Well, as long as it looks like the image above you are fine.


Which filters do you want to insert?

- Audio Capture Sources
- Audio Compressors
- Audio Renderers
- DirectShow Filters
  - I-Media Multiple MPEG2 Source
  - AC3 Parser Filter
  - MPEG-2 Splitter
  - MPEG-I Stream Splitter
  - WAV Dest
  - NetShow Theater Source Filter
  - NSC file Parser
  - InterVideo Audio Decoder
  - IVF source filter
  - Ligos MPEG Splitter

Notice also the AC3 points (encircled in red). These represent alternative audio tracks (ie. German, French etc.,) if you don't get the audio track you want, try the next one in the list until you find it. English is usually the first one, though, and I have chosen that in the above picture.

If you want to set the WinDVD to stereo just right click on the InterVideo box and choose properties, like so:


InterVideo Audio Decoder Properties

iviaudio

Audio channels
- ● 2 speaker mode
  - ○ Mono
  - ● Stereo
  - ○ DOLBY_SURROUND
  - ○ 3D audio
- ○ 4 speaker mode
- ○ 6 speaker mode (5.1channel)
- ○ Enable S/PDIF output

Current audio track
Format: AC3
Attributes: 16bit/48k/2ch

Vocal options
- ○ No voal
- ○ Left vocal
- ○ Right vocal
- ● Both vocal

Okay, once you are finished setting it up press the play button and wait for your finished Wave audio. Remember this could end up about a Gigabyte or two in size so make sure you have the space free. There are many other ways to use GraphEdit to get sound and even video, but this is the most common and most effective method I know.

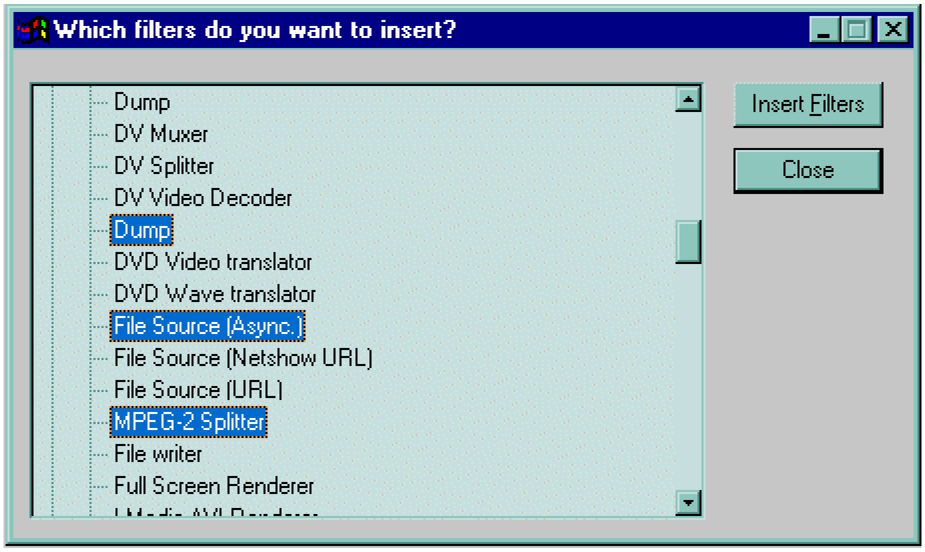
Graphedit may appear like its not doing anything because there is no progess bar. But as long as the 'play' button is grayed out it will be processing. You will know when its finished when it turns dark again.

OTHER GRAPHEDIT TRICKS

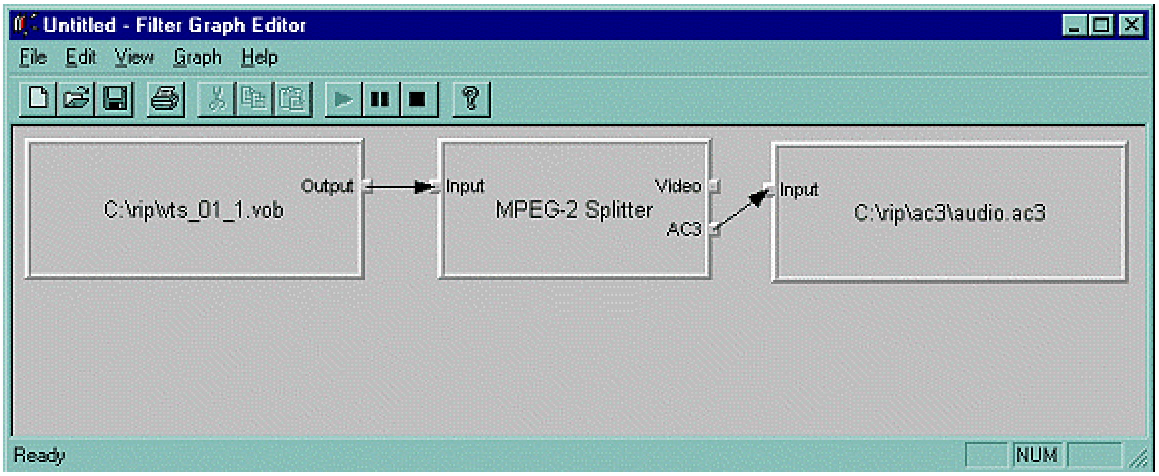
Note: The following additional information is for novelty value and come with no guarantee's. Connecting graphs can be tricky, so please don't email me if the following don't work for you.

EXTRACTING AN AC3 AUDIO FROM A DVD

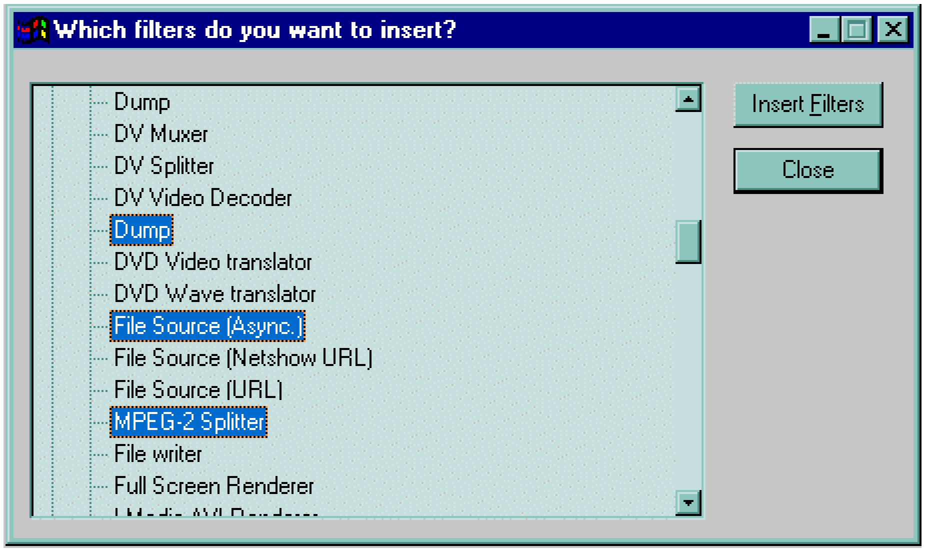
Now that you understand how Graphedit works it will be easier to show you how to do lots of other things. For example, to extract the AC3 audio from a Vob file you can do this.



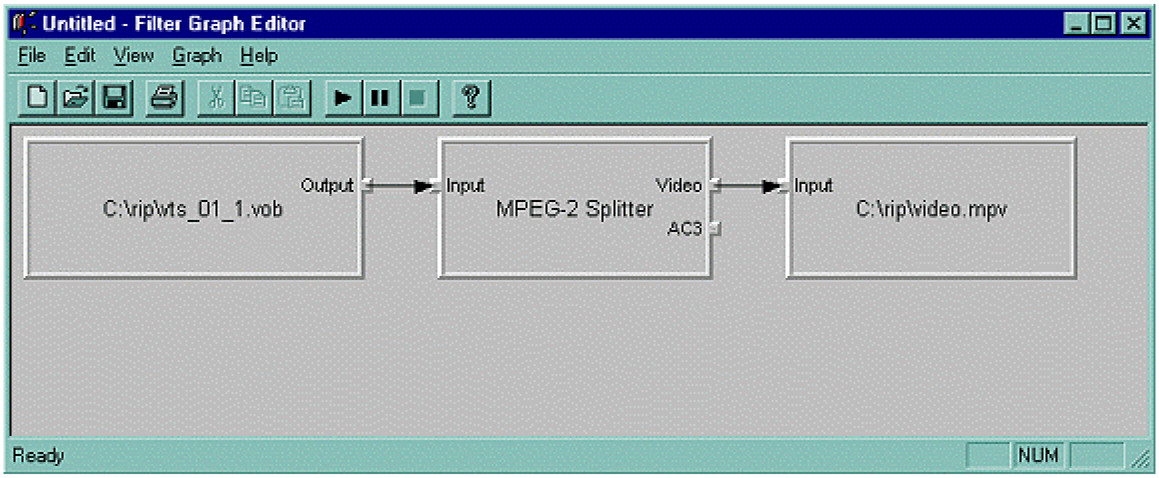
First use 'File Source (Async.)' and locate the vob file you wish to get the audio from. Then insert the 'Mpeg-2 Splitter'. And finally insert the 'Dump' filter and locate where you want to save the ac3 audio. Name it something.ac3 and press ok. Then Press play and wait.



EXTRACTING THE VIDEO FROM A DVD

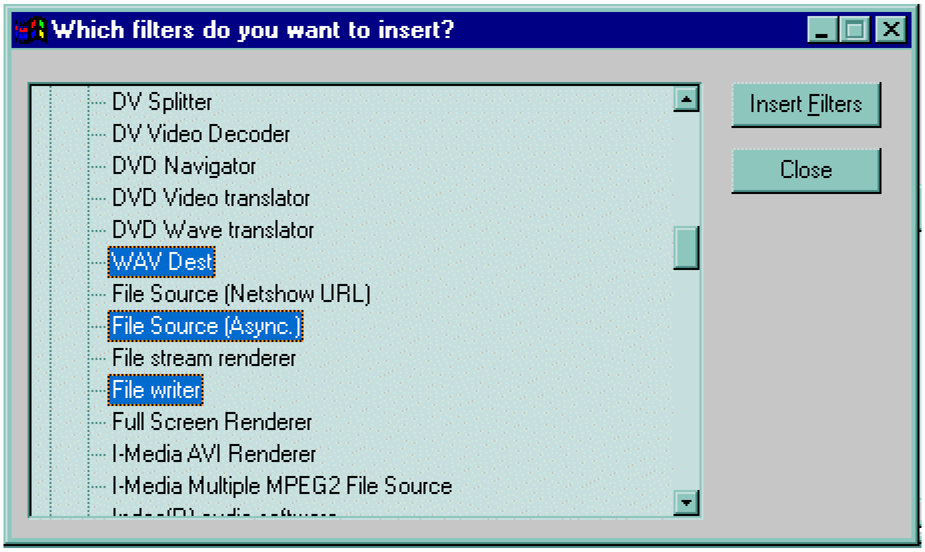


Same again, only instead of connecting the 'Dump' filter to the AC3 pin we connect it to the Video pin. We then name the video something.mpv (which is an Mpeg-2 file extention).

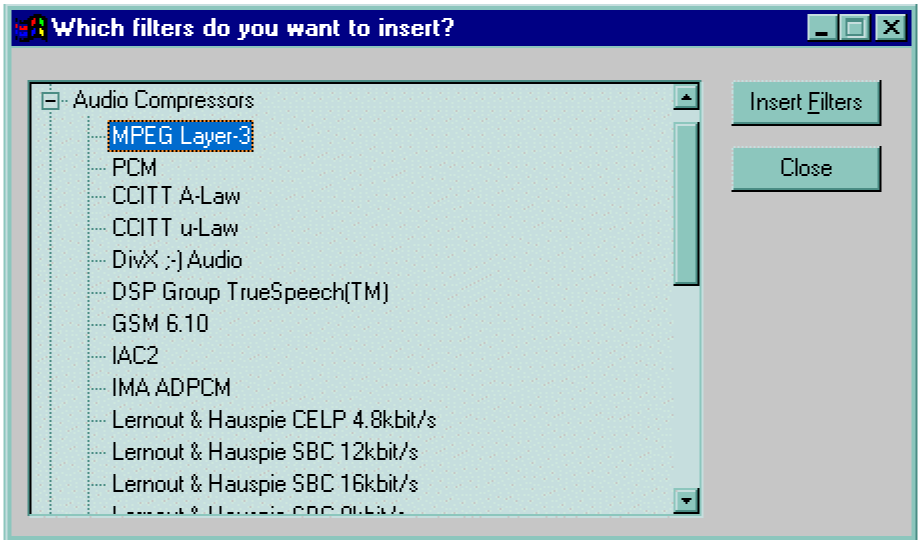


COMPRESSING AUDIO WITH GRAPHEDIT

While I'm at it, it may be interesting to know that you can also compress most audio types if you use the 'Wav Dest' and 'File Writer' filters. This can be useful if you are running short on Hard Disk space. Lets say you want to extract an AC3 file but its too large as a PCM, you could convert it to Mp3. But before I show you that, lets see what we need to change a PCM Wav into Mp3, select:



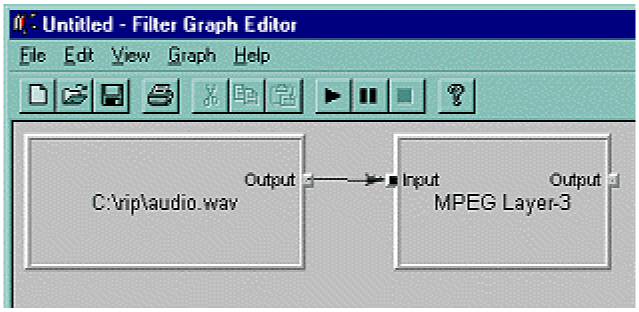
Then go into the Audio Compressor's section and choose MPEG Layer-3.



Which filters do you want to insert?

- Audio Compressors
  - MPEG Layer-3
  - PCM
  - CCITT A-Law
  - CCITT u-Law
  - DivX ;-) Audio
  - DSP Group TrueSpeech(TM)
  - GSM 6.10
  - IAC2
  - IMA ADPCM
  - Lernout & Hauspie CELP 4.8kbit/s
  - Lernout & Hauspie SBC 12kbit/s
  - Lernout & Hauspie SBC 16kbit/s

Select either an wave audio file or an .AVI file using the 'File Source (Async)' Filter. Then drag it to the 'MPEG Layer-3'.

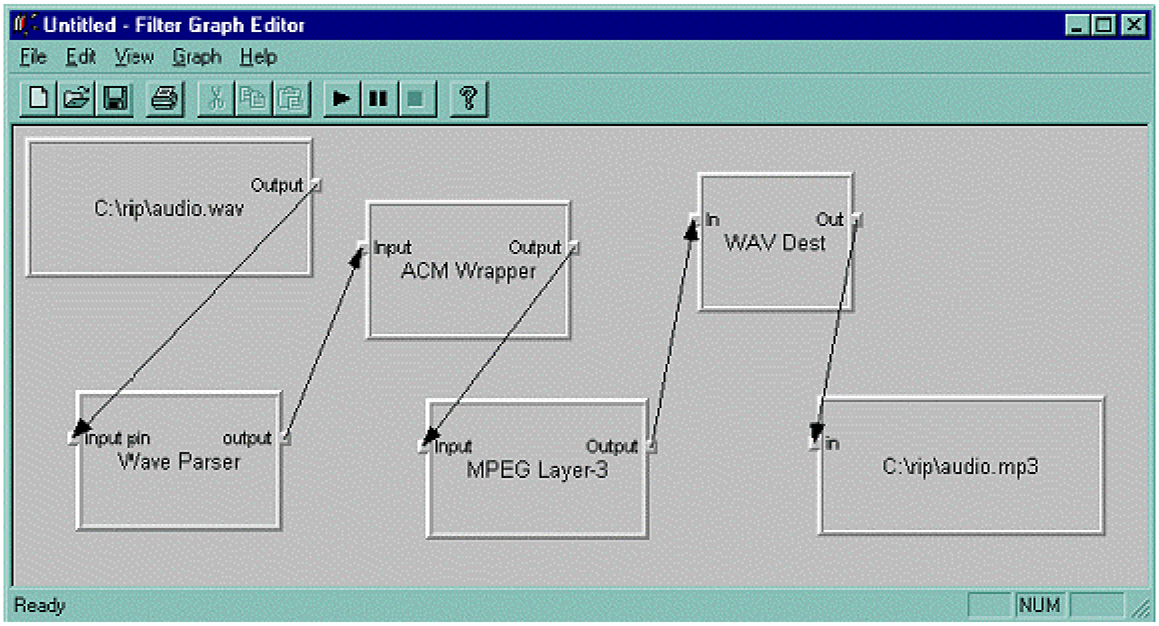


Untitled - Filter Graph Editor

C:\rip\audio.wav

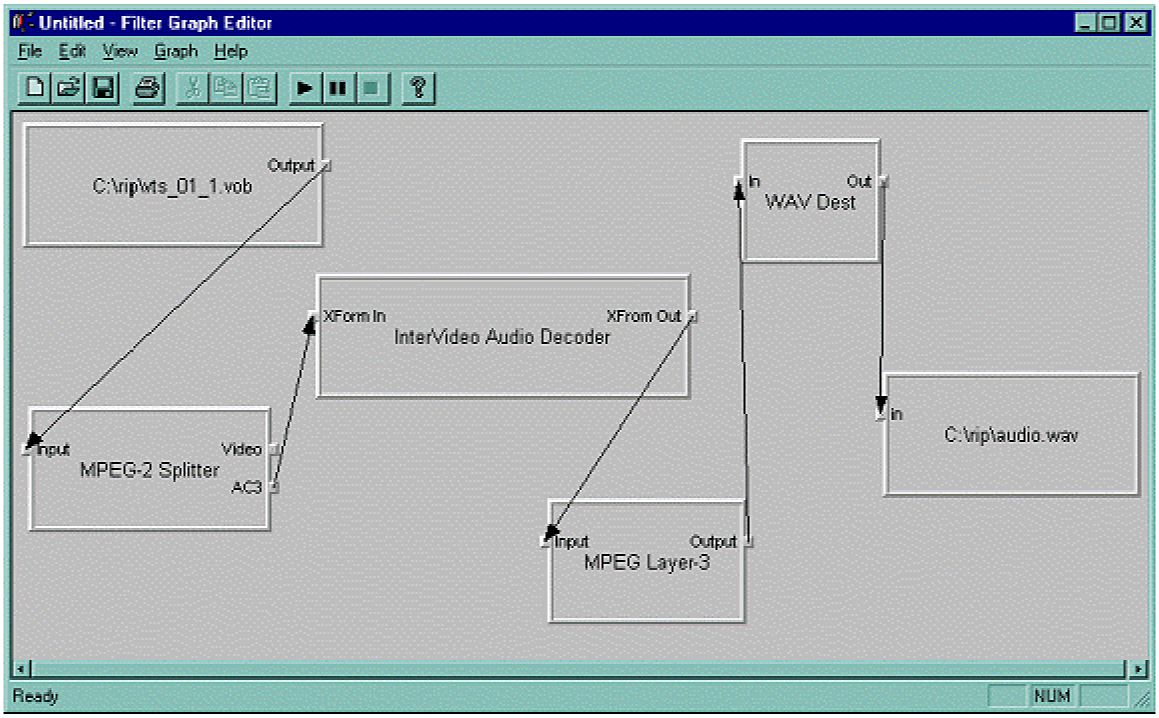
MPEG Layer-3

Bingo! Two more filters will appear between them, namely, the 'Wave Parser' and the 'ACM Wrapper'.

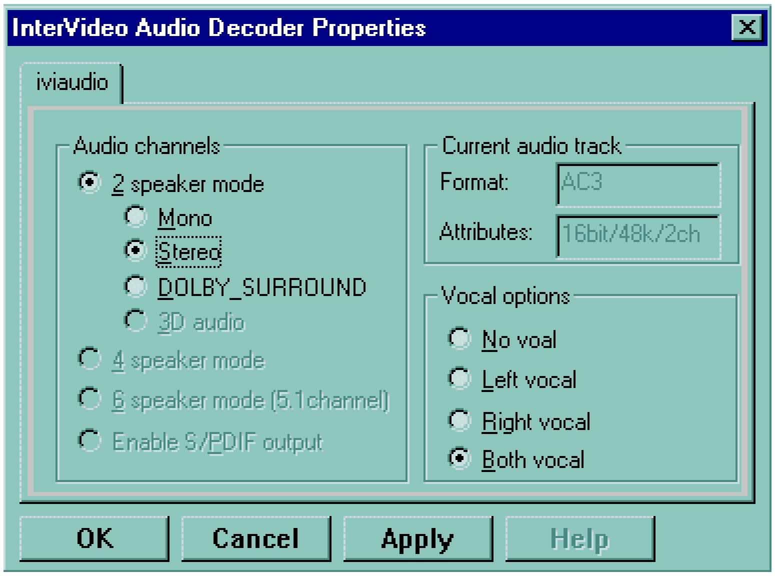


Finish the connections, press play and you will have mp3! But this is not the ideal way to compress audio o mp3! There is little no control. It will defaut to the first setting (i.e. 320kbit/s mp3) and the mp3 will also only take standard samplerates! =(

NOW FROM AC3 to MP3



The only differnce with this one is that InterVideo does all the Wav Parsing and so forth for us. We must set it to stereo too to make sure it converts correctly. This is done by right-clicking on the box and selecting it as in the picture below.



InterVideo Audio Decoder Properties

iviaudio

Audio channels

- ● 2 speaker mode
  - ○ Mono
  - ● Stereo
  - ○ DOLBY_SURROUND
  - ○ 3D audio
- ○ 4 speaker mode
- ○ 6 speaker mode (5.1channel)
- ○ Enable S/PDIF output

Current audio track

Format: AC3

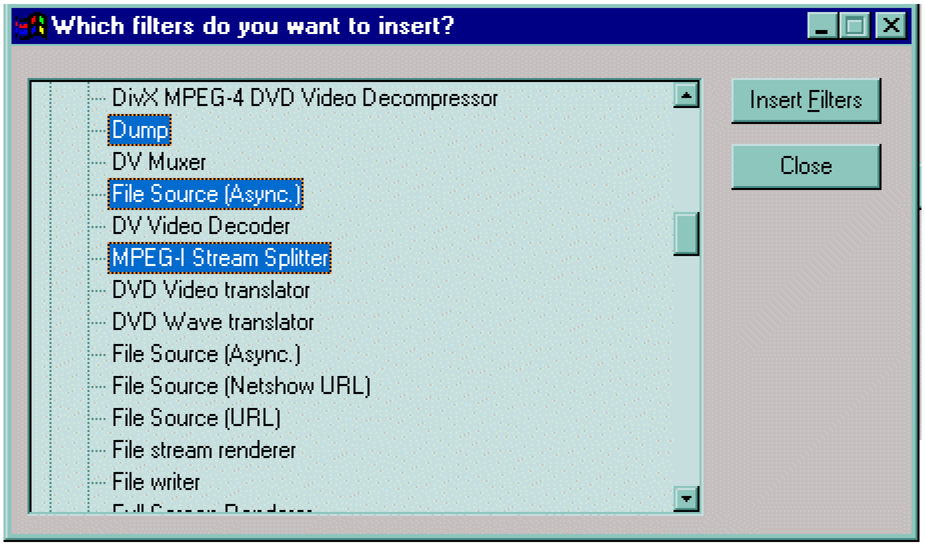
Attributes: 16bit/48k/2ch

Vocal options

- ○ No voal
- ○ Left vocal
- ○ Right vocal
- ● Both vocal

EXTRACTING THE AUDIO & VIDEO FROM AN MPEG-1 FILE

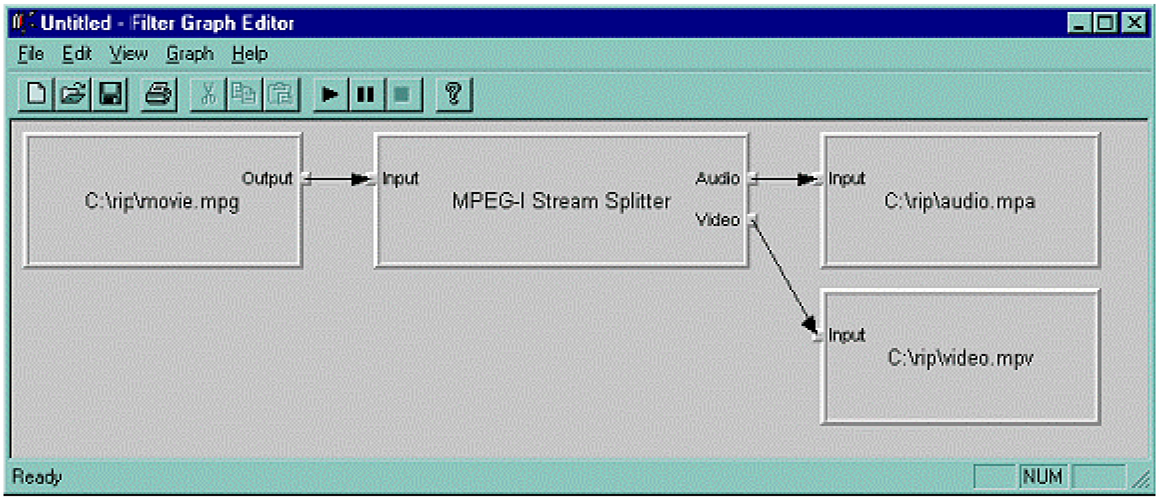
Graphedit can also be used to extract audio or video from other file formats, not just DVD. Lets take Mpeg-1 as a final example, we use:



Which filters do you want to insert?

- DivX MPEG-4 DVD Video Decompressor
- Dump
- DV Muxer
- File Source (Async.)
- DV Video Decoder
- MPEG-I Stream Splitter
- DVD Video translator
- DVD Wave translator
- File Source (Async.)
- File Source (Netshow URL)
- File Source (URL)
- File stream renderer
- File writer
- Full Screen Renderer

Mpeg-1 audio and video extraction is just as easy. We select the input file 'File Source (Async.)'. Split it with Microsofts 'MPEG-I Stream Splitter'. And use the 'Dump' filter twice - once to save the file: audio.mpa and the other to save: video.mpv.



I have highlighted some of the most useful things you can do with graphedit, but I'm sure now you have the idea you will find many more ways to do things.

DVD Audio Extraction with Ac3Dec

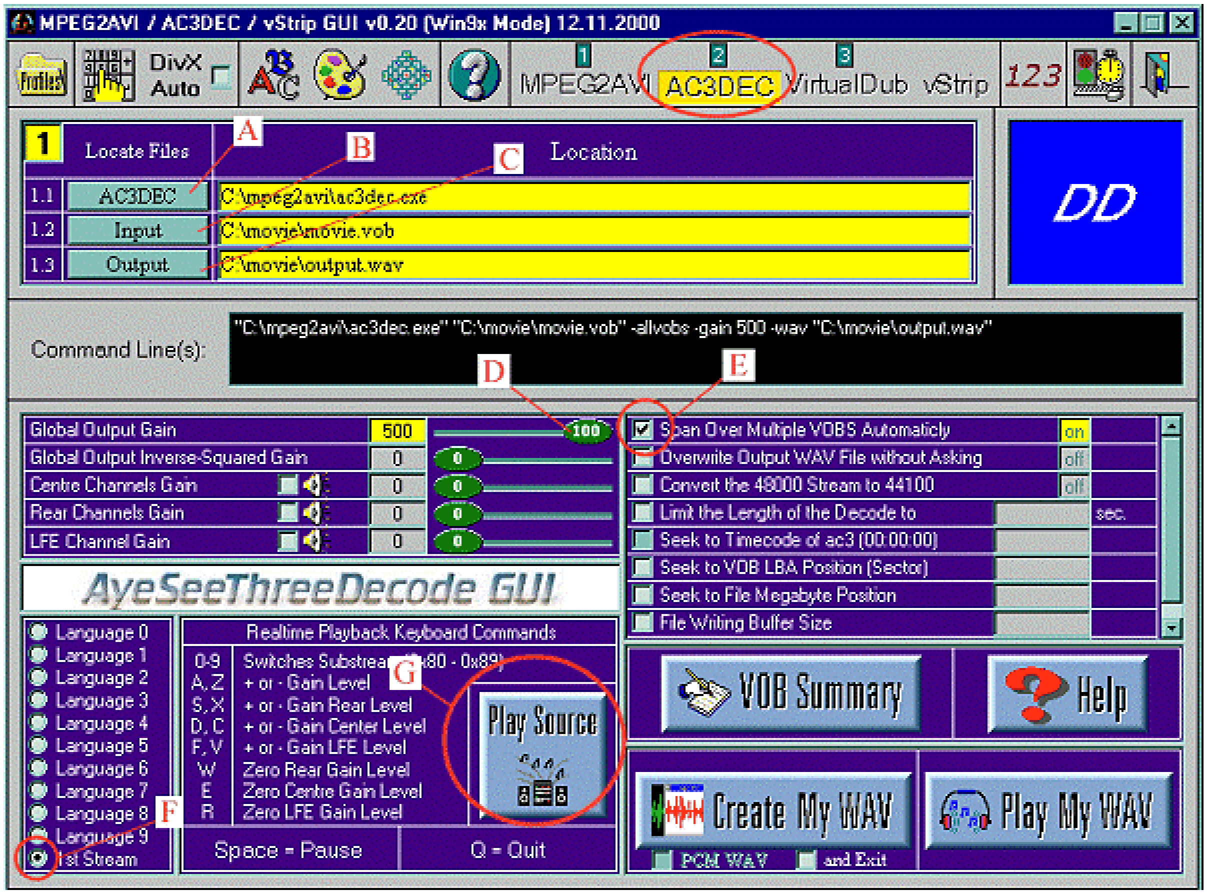
Ac3Dec is an excellent program for extracting AC3 audio and converting it to PCM Audio. It can handle almost any Vob or AC3 file with ease. The quality is not quite as good as GraphEdit, but perhaps that will be fixed in future updates. As always the DVD Vob files need to be decrypted to your hard disk or Ac3Dec will not be able to read them.

Before I start here are the things you will need:

Ac3Dec.exe *(AyeSeeThreeDecode)*

DanniDin's GUI v0.20

Put the Ac3Dec.exe file in a folder on your main hard drive (usually C:) and start up Danni's GUI. Click on the AC3DEC button at the top (encircled in red) to bring up the Ac3Dec page we see below us. There are lot of little useful options in Ac3Dec but these are not necessary to explain here. I will just concentrate on the important task of getting the audio =).



1	Locate Files	Location
1.1	AC3DEC	C:\mpeg2avi\ac3dec.exe
1.2	Input	C:\movie\movie.vob
1.3	Output	C:\movie\output.wav

Command Line(s): "C:\mpeg2avi\ac3dec.exe" "C:\movie\movie.vob" -allvobs -gain 500 -wav "C:\movie\output.wav"

Global Output Gain	500	100
Global Output Inverse-Squared Gain	0	0
Centre Channels Gain	0	0
Rear Channels Gain	0	0
LFE Channel Gain	0	0

☑ Span Over Multiple VOBS Automaticly	on
□ Overwrite Output WAV File without Asking	off
□ Convert the 48000 Stream to 44100	off
□ Limit the Length of the Decode to	sec.
□ Seek to Timecode of ac3 (00:00:00)	
□ Seek to VOB LBA Position (Sector)	
□ Seek to File Megabyte Position	
□ File Writing Buffer Size	

○ Language 0
○ Language 1
○ Language 2
○ Language 3
○ Language 4
○ Language 5
○ Language 6
○ Language 7
○ Language 8
○ Language 9
● 1st Stream

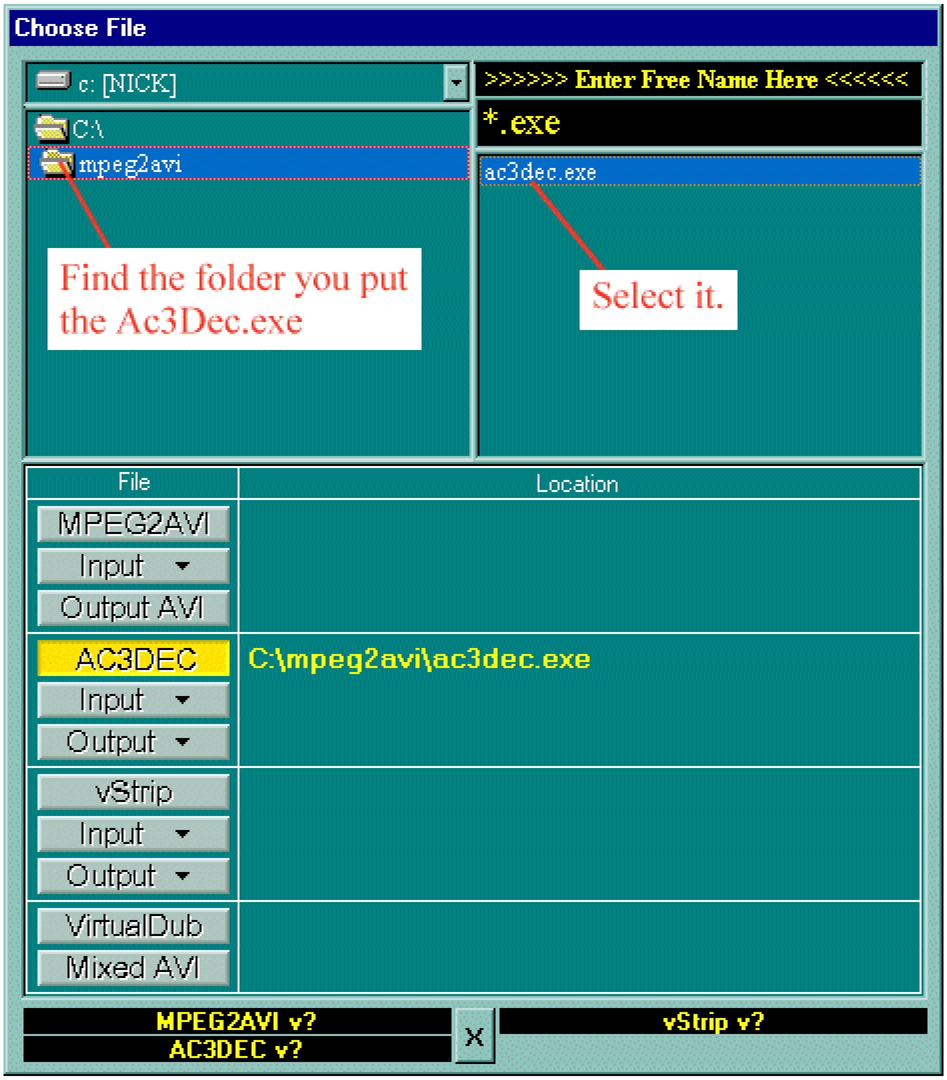
Realtime Playback Keyboard Commands

0-9	Switches Substream (0x80 - 0x89)
A, Z	+ or - Gain Level
S, X	+ or - Gain Rear Level
D, C	+ or - Gain Center Level
F, V	+ or - Gain LFE Level
W	Zero Rear Gain Level
E	Zero Centre Gain Level
R	Zero LFE Gain Level

Space = Pause Q = Quit

□ PCM WAV □ and Exit

Press the (**A**) button and select the location of the Ac3Dec.exe file.



Choose File

c: [NICK]

C:\
mpeg2avi

>>>>>> Enter Free Name Here <<<<<<

*.exe

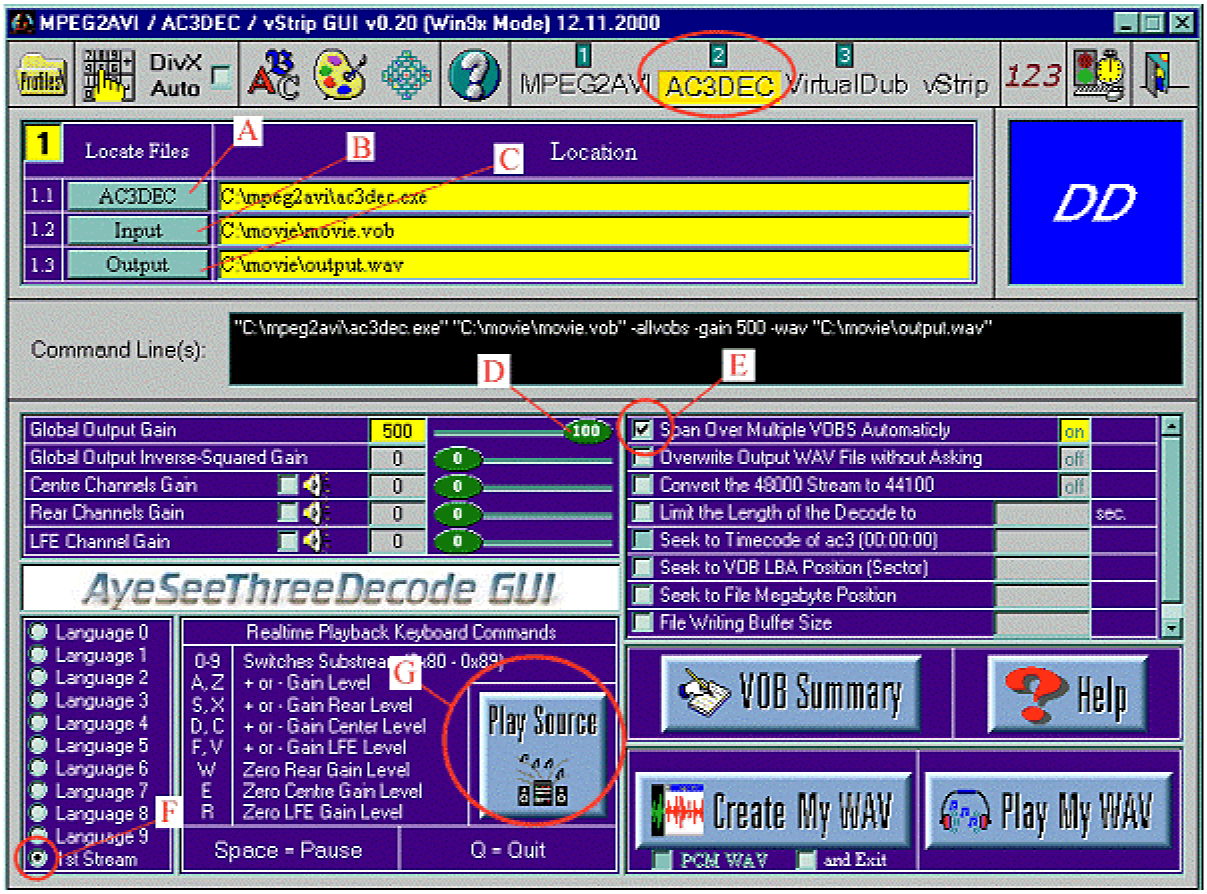
ac3dec.exe

Find the folder you put the Ac3Dec.exe

Select it.

File	Location
MPEG2AVI Input Output AVI	
AC3DEC Input Output	C:\mpeg2avi\ac3dec.exe
vStrip Input Output	
VirtualDub Mixed AVI	

Then tell it where the first Vob of your movie is by pressing (**B**) and selecting it.



MPEG2AVI / AC3DEC / vStrip GUI v0.20 (Win9x Mode) 12.11.2000

1	Locate Files	Location
1.1	AC3DEC	C:\mpeg2avi\ac3dec.exe
1.2	Input	C:\movie\movie.vob
1.3	Output	C:\movie\output.wav

Command Line(s):

```
"C:\mpeg2avi\ac3dec.exe" "C:\movie\movie.vob" -allvobs -gain 500 -wav "C:\movie\output.wav"
```

Global Output Gain	500	100
Global Output Inverse-Squared Gain	0	0
Centre Channels Gain □	0	0
Rear Channels Gain □	0	0
LFE Channel Gain □	0	0

☑ Span Over Multiple VOBS Automaticly	on	
□ Overwrite Output WAV File without Asking	off	
□ Convert the 48000 Stream to 44100	off	
□ Limit the Length of the Decode to		sec.
□ Seek to Timecode of ac3 (00:00:00)		
□ Seek to VOB LBA Position (Sector)		
□ Seek to File Megabyte Position		
□ File Writing Buffer Size		

○ Language 0
○ Language 1
○ Language 2
○ Language 3
○ Language 4
○ Language 5
○ Language 6
○ Language 7
○ Language 8
○ Language 9
● 1st Stream

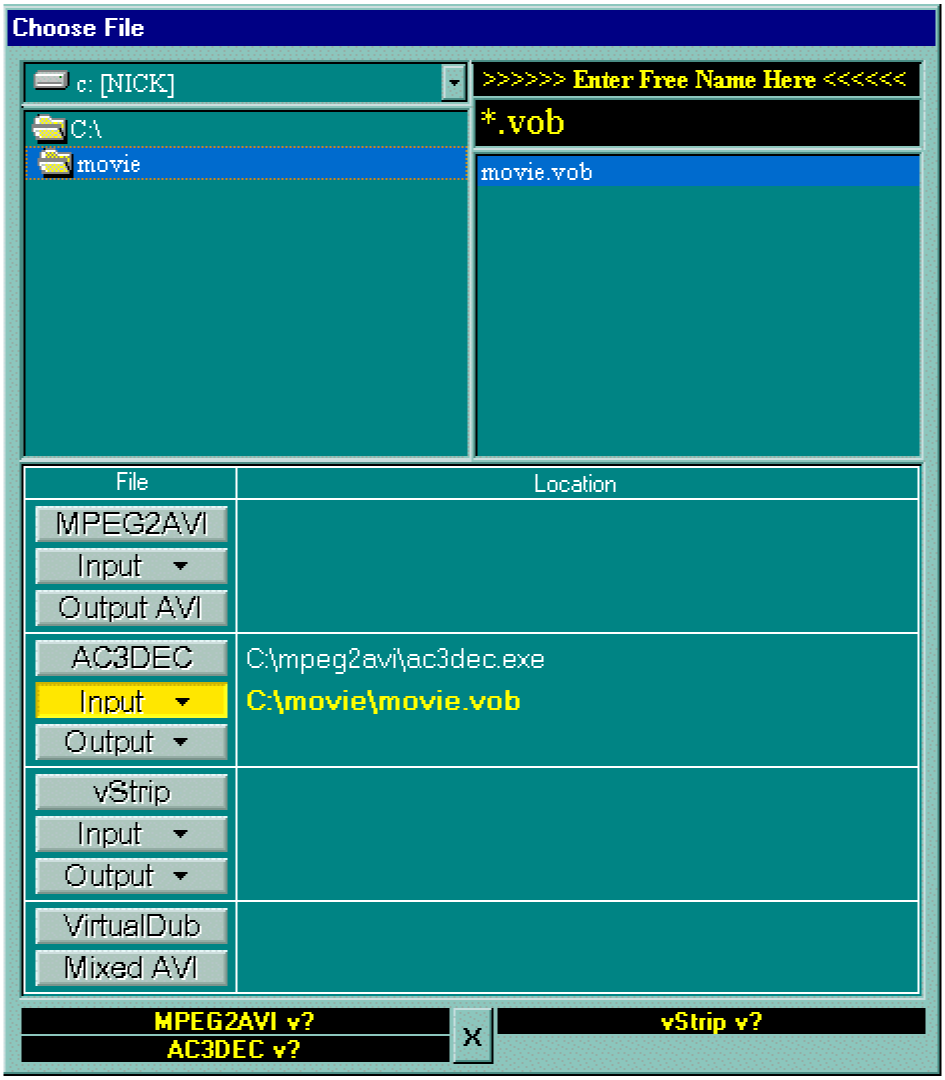
Realtime Playback Keyboard Commands

0-9	Switches Substream (0x80 - 0x89)
A, Z	+ or - Gain Level
S, X	+ or - Gain Rear Level
D, C	+ or - Gain Center Level
F, V	+ or - Gain LFE Level
W	Zero Rear Gain Level
E	Zero Centre Gain Level
R	Zero LFE Gain Level

Space = Pause   Q = Quit

□ PCM WAV   □ and Exit

You can choose either a Vob or an AC3 file if you wish.



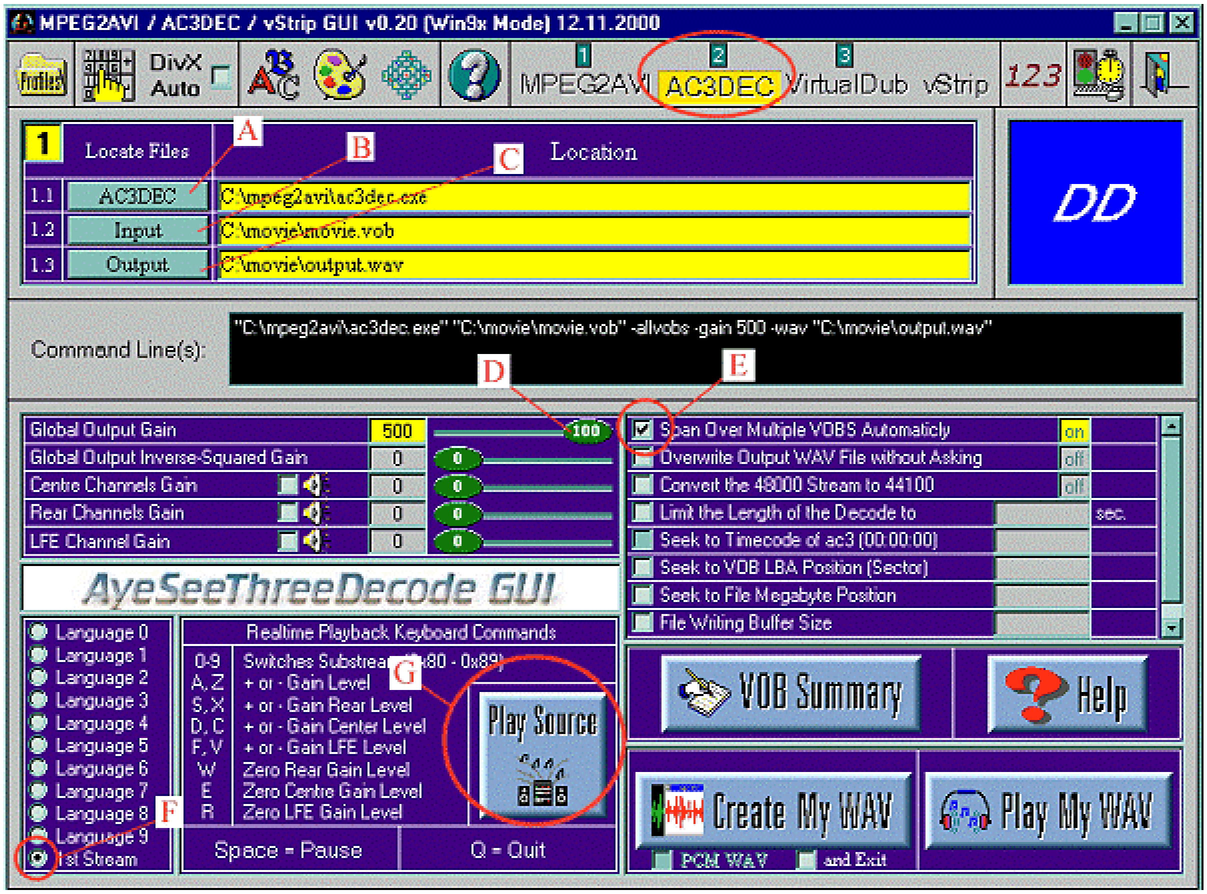
*.vob

movie.vob

C:\mpeg2avi\ac3dec.exe

C:\movie\movie.vob

And then tell it where you want to save your final Wave by selecting (**C**)



1	Locate Files	Location
1.1	AC3DEC	C:\mpeg2avi\ac3dec.exe
1.2	Input	C:\movie\movie.vob
1.3	Output	C:\movie\output.wav

Command Line(s):

```
"C:\mpeg2avi\ac3dec.exe" "C:\movie\movie.vob" -allvobs -gain 500 -wav "C:\movie\output.wav"
```

Global Output Gain	500	100
Global Output Inverse-Squared Gain	0	0
Centre Channels Gain □	0	0
Rear Channels Gain □	0	0
LFE Channel Gain □	0	0

☑ Span Over Multiple VOBS Automaticly	on	
□ Overwrite Output WAV File without Asking	off	
□ Convert the 48000 Stream to 44100	off	
□ Limit the Length of the Decode to		sec.
□ Seek to Timecode of ac3 (00:00:00)		
□ Seek to VOB LBA Position (Sector)		
□ Seek to File Megabyte Position		
□ File Writing Buffer Size		

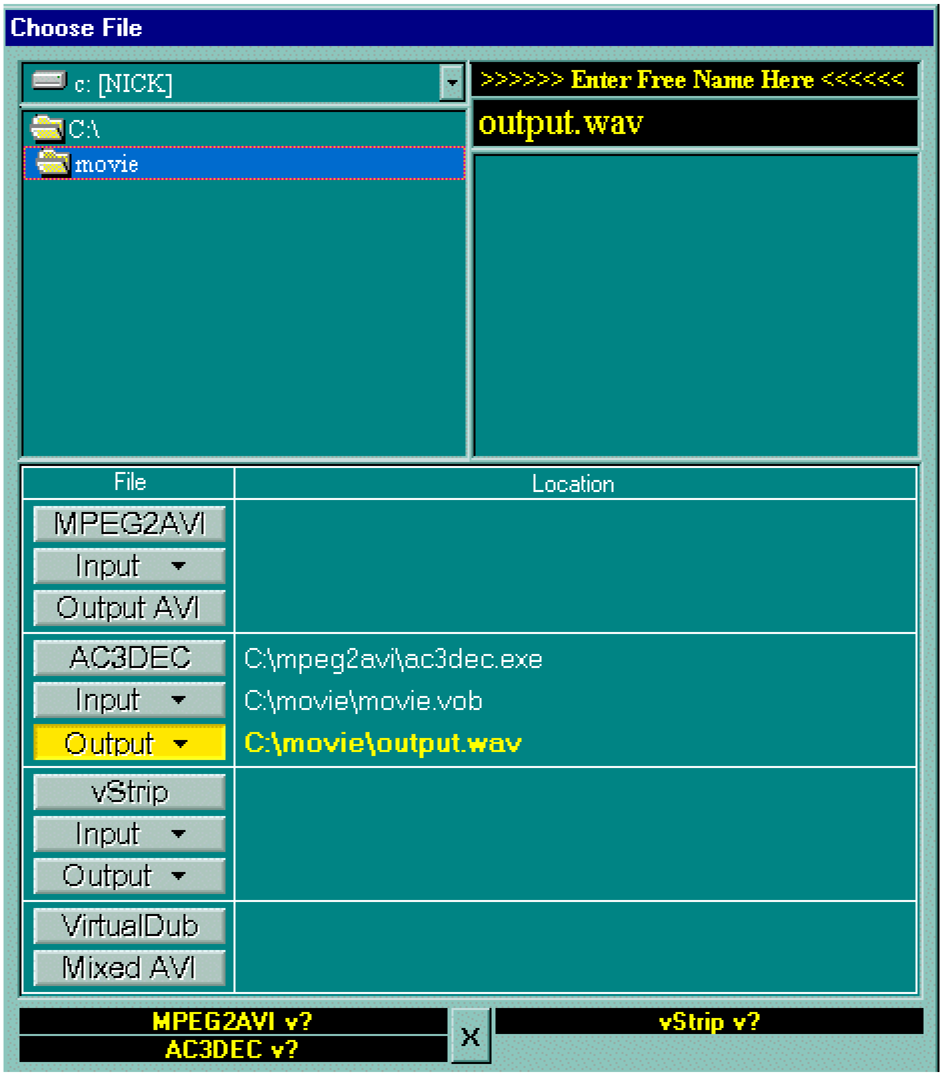
○ Language 0
○ Language 1
○ Language 2
○ Language 3
○ Language 4
○ Language 5
○ Language 6
○ Language 7
○ Language 8
○ Language 9
● 1st Stream

Realtime Playback Keyboard Commands

0-9	Switches Substream (0x80 - 0x89)
A, Z	+ or - Gain Level
S, X	+ or - Gain Rear Level
D, C	+ or - Gain Center Level
F, V	+ or - Gain LFE Level
W	Zero Rear Gain Level
E	Zero Centre Gain Level
R	Zero LFE Gain Level

Space = Pause   Q = Quit

□ PCM WAV □ and Exit



Choose File

c: [NICK]

C:\
movie

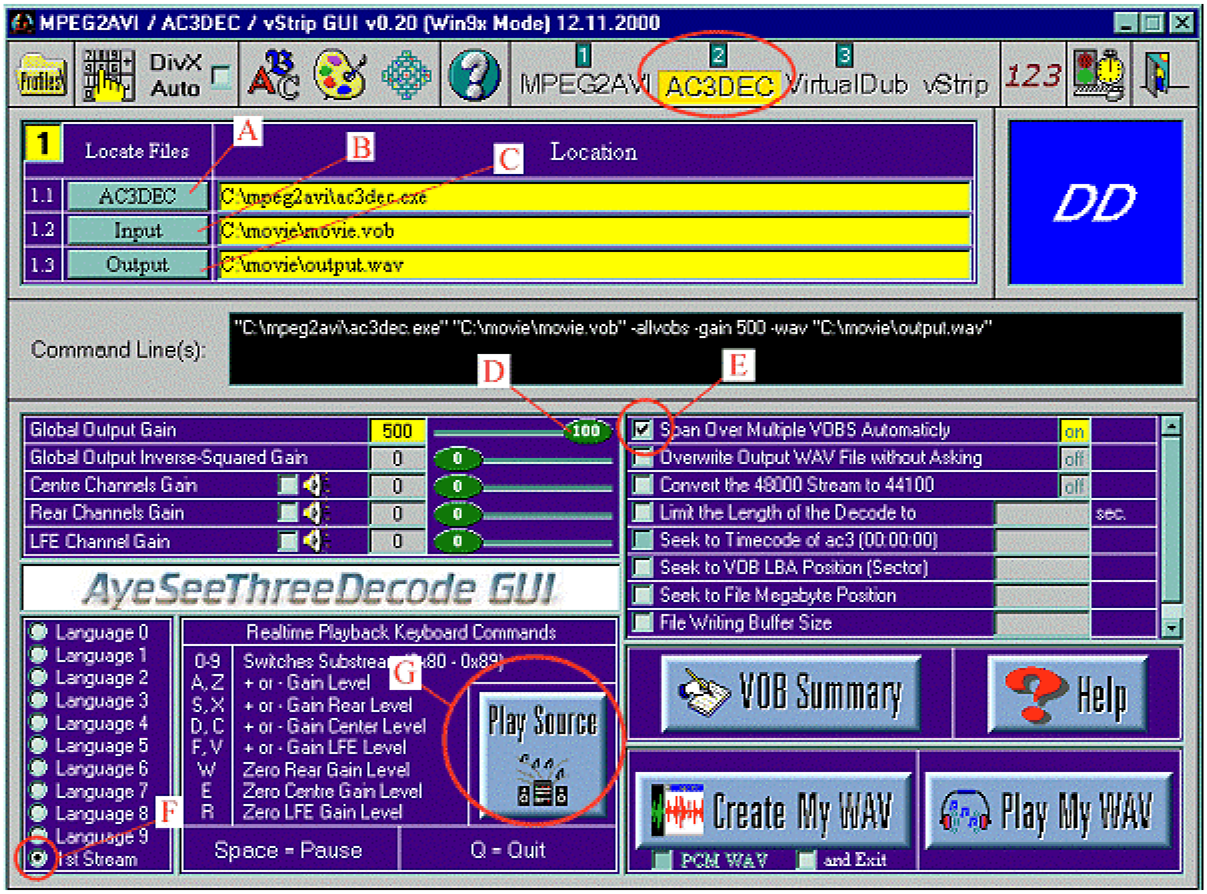
>>>>>> Enter Free Name Here <<<<<<

output.wav

File	Location
MPEG2AVI	
Input	
Output AVI	
AC3DEC	C:\mpeg2avi\ac3dec.exe
Input	C:\movie\movie.vob
Output	C:\movie\output.wav
vStrip	
Input	
Output	
VirtualDub	
Mixed AVI	

DECODING SETTINGS

I always set the Global Output Gain (**D**) to 500 to help boost volume. You can also turn up the 'inverse-squared gain' if it is still a bit low. If you turn too many channels up you will generally get distorted audio, so be careful. But play about with these settings until you are happy with the output. To test the volume etc., press the 'Play Source' button (**G**).



MPEG2AVI / AC3DEC / vStrip GUI v0.20 (Win9x Mode) 12.11.2000

1	Locate Files	Location
1.1	AC3DEC	C:\mpeg2avi\ac3dec.exe
1.2	Input	C:\movie\movie.vob
1.3	Output	C:\movie\output.wav

Command Line(s): "C:\mpeg2avi\ac3dec.exe" "C:\movie\movie.vob" -allvobs -gain 500 -wav "C:\movie\output.wav"

Global Output Gain	500
Global Output Inverse-Squared Gain	0
Centre Channels Gain	0
Rear Channels Gain	0
LFE Channel Gain	0

☑ Span Over Multiple VOBS Automaticly	on
□ Overwrite Output WAV File without Asking	off
□ Convert the 48000 Stream to 44100	off
□ Limit the Length of the Decode to	sec.
□ Seek to Timecode of ac3 (00:00:00)	
□ Seek to VOB LBA Position (Sector)	
□ Seek to File Megabyte Position	
□ File Writing Buffer Size	

Realtime Playback Keyboard Commands

0-9	Switches Substream 0x80 - 0x89
A, Z	+ or - Gain Level
S, X	+ or - Gain Rear Level
D, C	+ or - Gain Center Level
F, V	+ or - Gain LFE Level
W	Zero Rear Gain Level
E	Zero Centre Gain Level
R	Zero LFE Gain Level

Space = Pause | Q = Quit

○ Language 0
○ Language 1
○ Language 2
○ Language 3
○ Language 4
○ Language 5
○ Language 6
○ Language 7
○ Language 8
○ Language 9
● 1st Stream

□ PCM WAV □ and Exit

If you have more than one Vob file (as is almost always the case), make sure the 'Span Over Multiple VOBS Automatically' (**E**) is checked. It is important that your Vob files are numbered in order for this to work. So you should have something like:

vts_01_**1**.vob

vts_01_**2**.vob

vts_01_**3**.vob

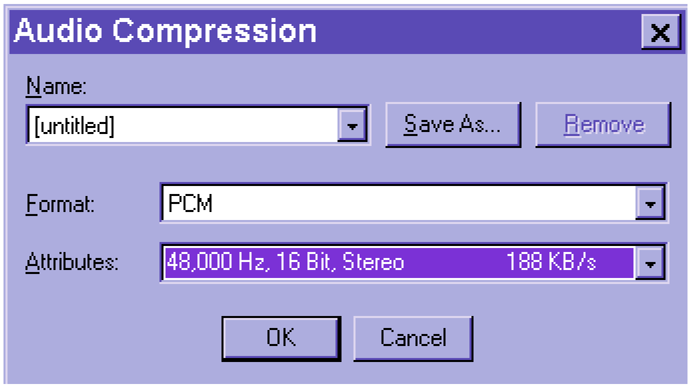
vts_01_**4**.vob

This way, when it comes to the end of the first Vob, it will look for the next one in the list.

One last buy *very* important option is (**F**). Since a DVD can have many language tracks we have to select the correct one. Usually the default '1st stream' option is English. But test it by hitting the 'Play Source' button again (**G**). Listen to see what language is played, if it is the wrong one, select the next one above it. If that is wrong try the next one above that and the next and so on, until you find the correct audio track you want.

When everything is done press the large 'Create my Wav' button.

Up will pop a selection box asking for what audio format you want the DVD AC3 converted into. Choose PCM 48,000Hz 16 Bit, Stereo. You could also choose MP3 if you wanted, but for best results you will need to use PCM. This way you can multiplex it to the video file and compress it using VirtualDub.

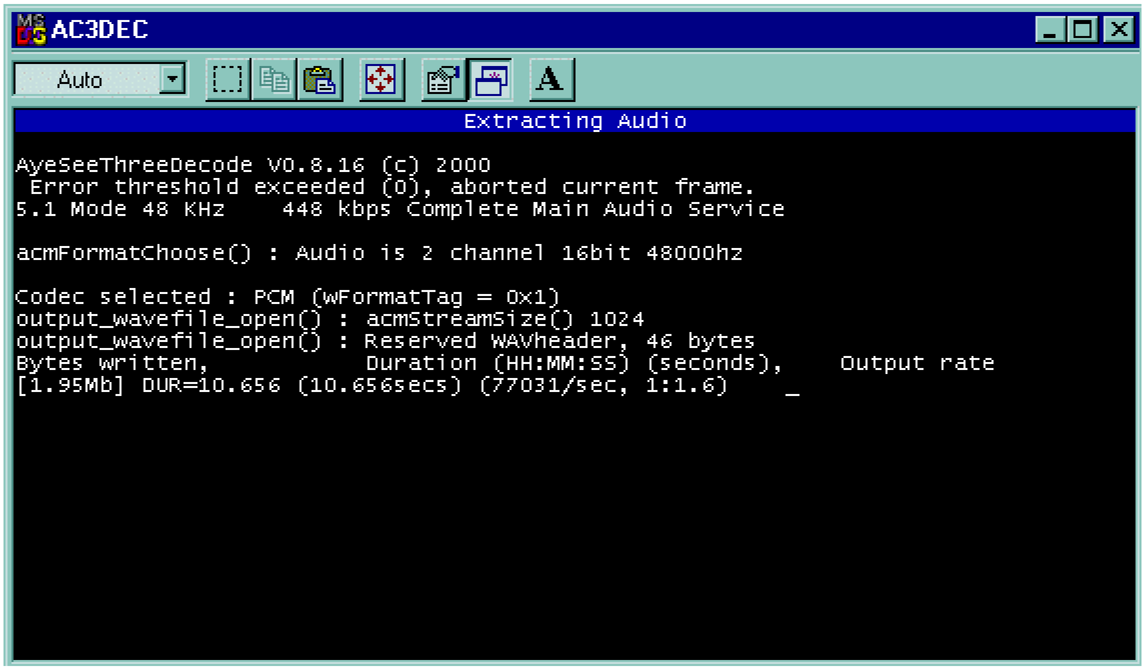


Name: [untitled]

Format: PCM

Attributes: 48,000 Hz, 16 Bit, Stereo 188 KB/s

Thats it! A window like the one below will appear telling you the progress of your conversion. Ac3Dec converts to PCM audio in real time, so it will take as long as watching the movie to convert it into a Wave file.



```
                              Extracting Audio

AyeSeeThreeDecode V0.8.16 (c) 2000
 Error threshold exceeded (0), aborted current frame.
5.1 Mode 48 KHz    448 kbps Complete Main Audio Service

acmFormatChoose() : Audio is 2 channel 16bit 48000hz

Codec selected : PCM (wFormatTag = 0x1)
output_wavefile_open() : acmStreamSize() 1024
output_wavefile_open() : Reserved WAVheader, 46 bytes
Bytes written,          Duration (HH:MM:SS) (seconds),    Output rate
[1.95Mb] DUR=10.656 (10.656secs) (77031/sec, 1:1.6)    _
```

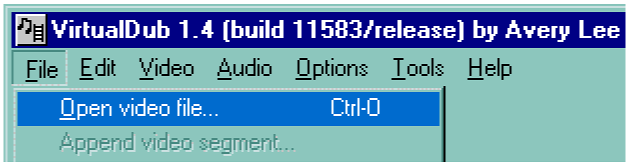
DivX Audio Multiplexing with VirtualDub

To multiplex is to merge (interleave) an audio file with a video file. All our AVI multiplexing is done with VirtualDub. VirtualDub is the most useful and powerful video editing application you will ever download! It is more useful than other video editing applications that would cost hundreds of pounds to buy - it is also the easiest of them to learn! And what's more, its free! For more details on using this amazing application check out my Advanced VirtualDub section.

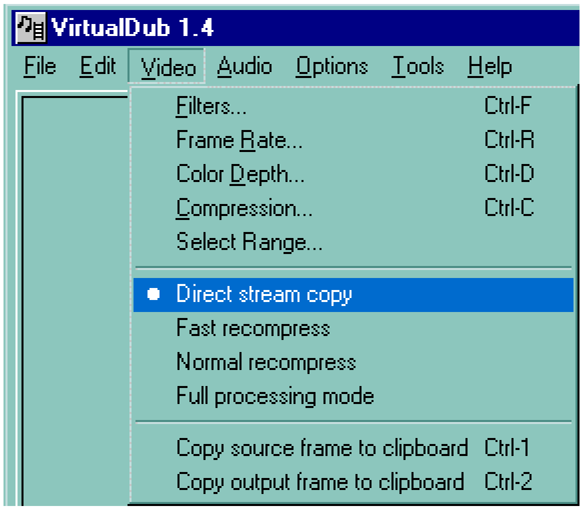
Before I start you will need:

VirtualDub 1.4c

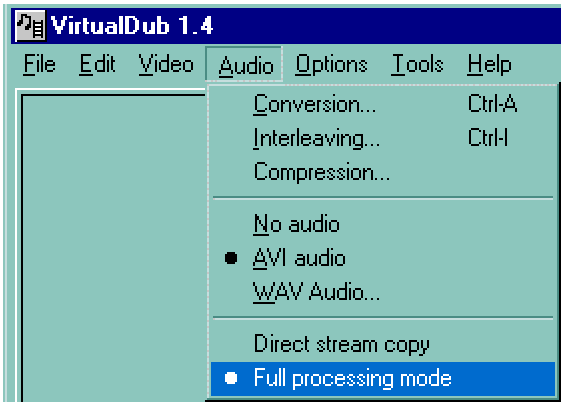
Whatever method you used to extract your audio to PCM Wav the method for multiplexing is the same. Open your movie file in VirtualDub by going to File > Open video file...


VirtualDub 1.4 (build 11583/release) by Avery Lee
File
Edit
Video
Audio
Options
Tools
Help
Open video file... Ctrl-O
Append video segment...

We don't want to effect the video itself so we use Video > Direct stream copy. Whenever we select an option, VirtualDub adds a black dot by it to tell us it is selected.



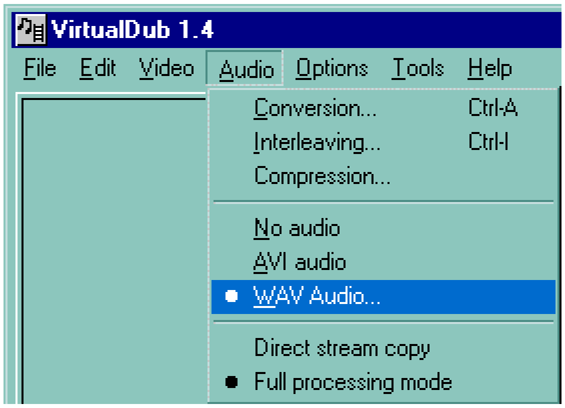
Then select Audio > Full processing mode.



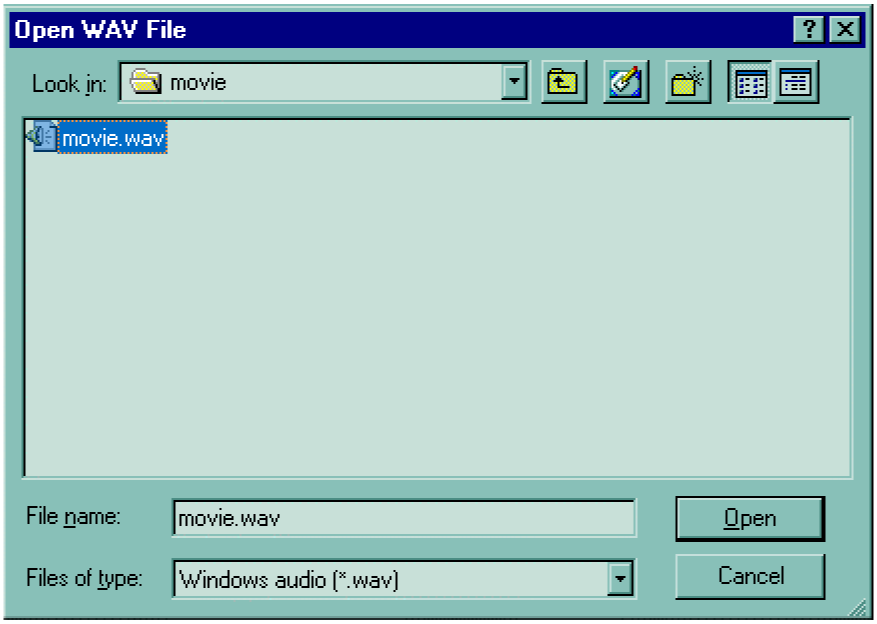
SELECT THE WAVE

If you used PCM audio in Flask Mpeg to create the audio for your AVI then you can skip this bit and move on to 'Converting the Samplerate'. Flask mixes the PCM audio with the video, so all we would need to do is compress the audio. But for those who exracted the audio with Graphedit or Ac3Dec or Normalized the audio seperately, we select it like this:

Choose Audio > WAV Audio.



When we do this VirtualDub will ask us to locate the wave file we wish to use. Do so and select Open.



Look in: movie

movie.wav

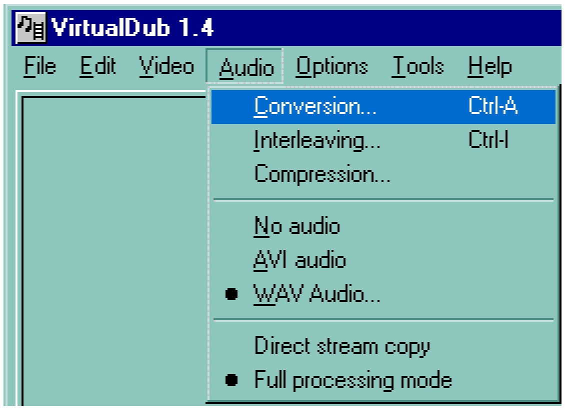
File name: movie.wav

Files of type: Windows audio (*.wav)

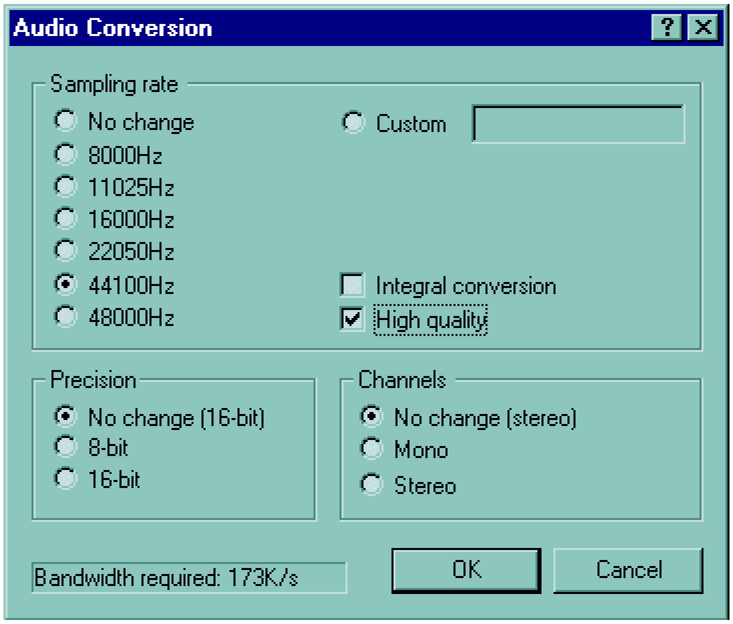
CONVERTING THE SAMPLERATE

If your audio file was sampled at 48,000Hz, as is usually the case with DVD audio, then it is better we convert it to 44,100Hz. For more details on this subject read my article called *"Fixing Bad Audio"*.

Choose Audio > Conversion...



Choose 44100Hz and make sure the 'High quality' is selected, and press OK.



Audio Conversion

Sampling rate

○ No change
○ 8000Hz
○ 11025Hz
○ 16000Hz
○ 22050Hz
● 44100Hz
○ 48000Hz
○ Custom
□ Integral conversion
☑ High quality

Precision

● No change (16-bit)
○ 8-bit
○ 16-bit

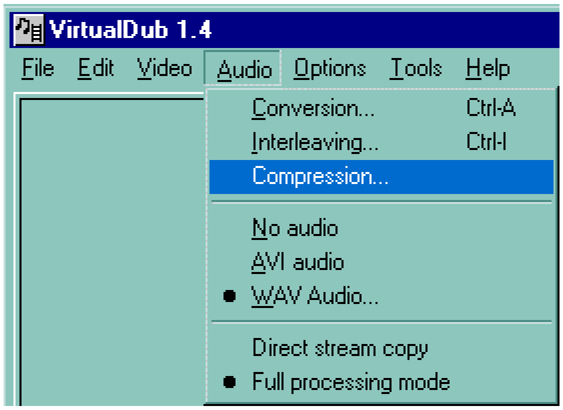
Channels

● No change (stereo)
○ Mono
○ Stereo

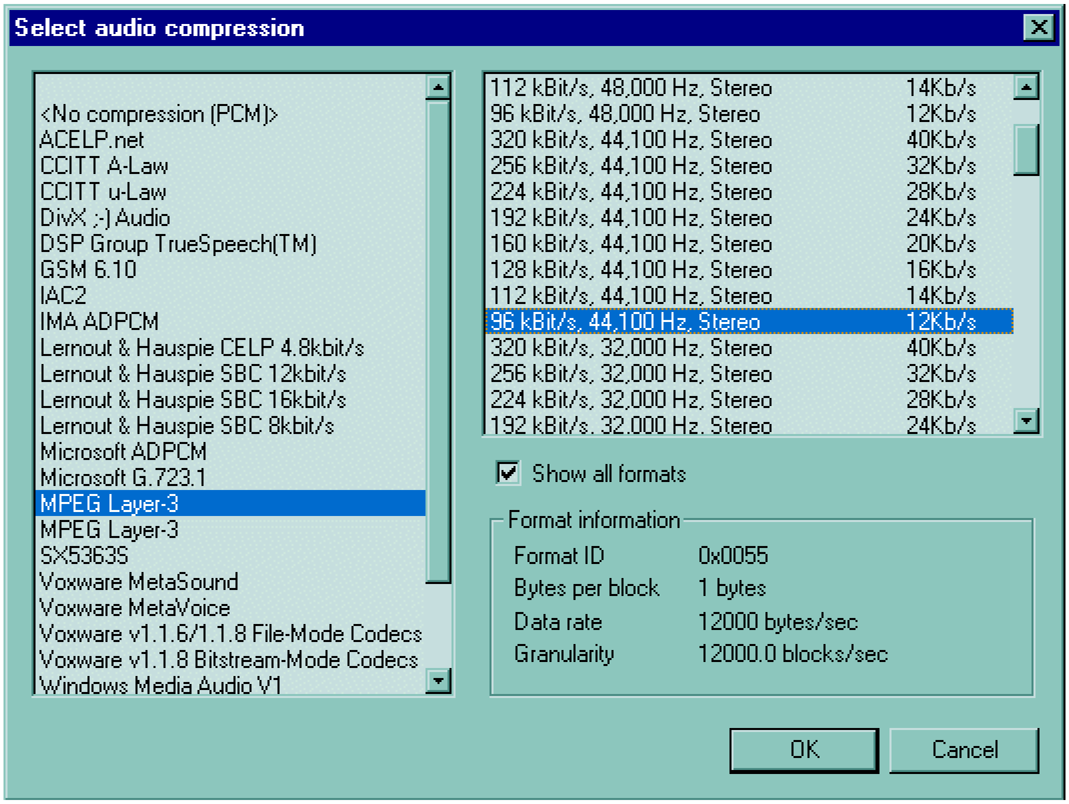
Bandwidth required: 173K/s

COMPRESSING THE AUDIO

Finally, because a PCM wave file for the average movie is something like 1 to 2 gigabytes in size, we will need to compress it to MP3 or WMA audio. To do this choose Audio > Compression...



Choose MPEG Layer-3. Hit the 'show all formats box' and then select 96kBit/s, 44100Hz, Stereo. If you want even higher quality audio (i.e. perhaps this is a music video), then you can choose 128 kBits/s 44100Hz stereo. This is said to be equivalent to CD quality audio! On the other hand, if you really want loads more audio compression, you can choose radio quality 64 kBit/s 44100Hz mono.



Select audio compression

<No compression (PCM)>
ACELP.net
CCITT A-Law
CCITT u-Law
DivX ;-) Audio
DSP Group TrueSpeech(TM)
GSM 6.10
IAC2
IMA ADPCM
Lernout & Hauspie CELP 4.8kbit/s
Lernout & Hauspie SBC 12kbit/s
Lernout & Hauspie SBC 16kbit/s
Lernout & Hauspie SBC 8kbit/s
Microsoft ADPCM
Microsoft G.723.1
MPEG Layer-3
MPEG Layer-3
SX5363S
Voxware MetaSound
Voxware MetaVoice
Voxware v1.1.6/1.1.8 File-Mode Codecs
Voxware v1.1.8 Bitstream-Mode Codecs
Windows Media Audio V1

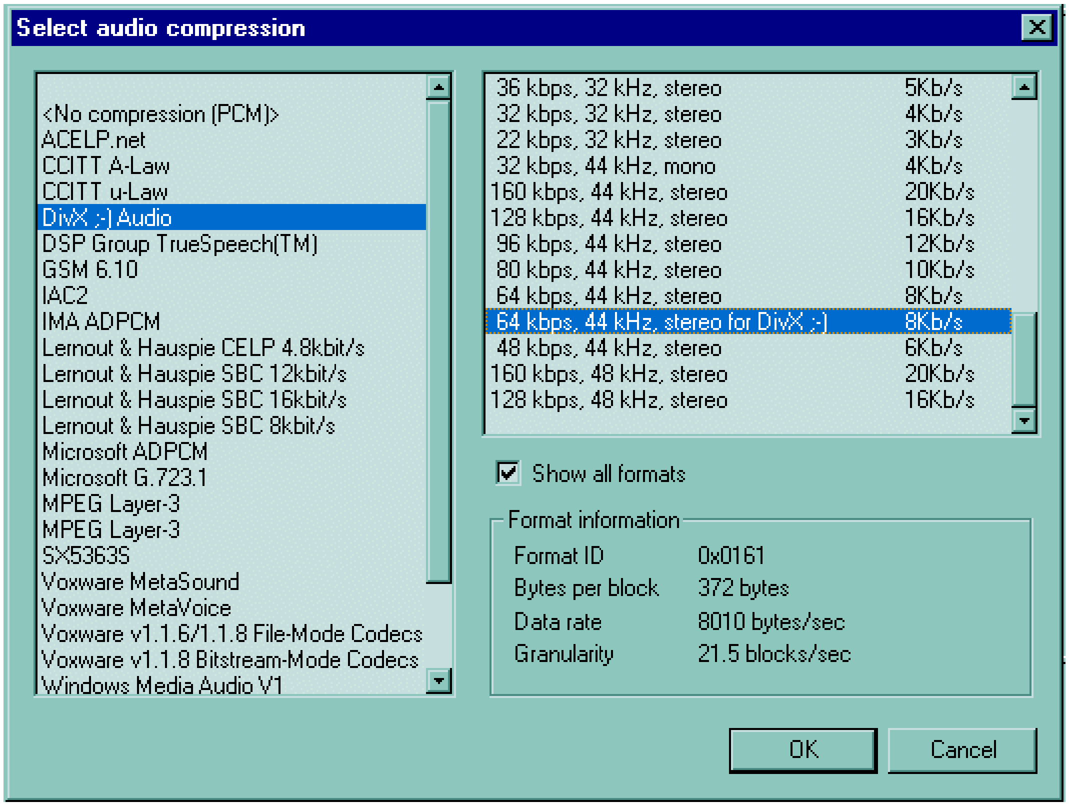
112 kBit/s, 48,000 Hz, Stereo	14Kb/s
96 kBit/s, 48,000 Hz, Stereo	12Kb/s
320 kBit/s, 44,100 Hz, Stereo	40Kb/s
256 kBit/s, 44,100 Hz, Stereo	32Kb/s
224 kBit/s, 44,100 Hz, Stereo	28Kb/s
192 kBit/s, 44,100 Hz, Stereo	24Kb/s
160 kBit/s, 44,100 Hz, Stereo	20Kb/s
128 kBit/s, 44,100 Hz, Stereo	16Kb/s
112 kBit/s, 44,100 Hz, Stereo	14Kb/s
96 kBit/s, 44,100 Hz, Stereo	12Kb/s
320 kBit/s, 32,000 Hz, Stereo	40Kb/s
256 kBit/s, 32,000 Hz, Stereo	32Kb/s
224 kBit/s, 32,000 Hz, Stereo	28Kb/s
192 kBit/s, 32,000 Hz, Stereo	24Kb/s

☑ Show all formats

Format information

Format ID	0x0055
Bytes per block	1 bytes
Data rate	12000 bytes/sec
Granularity	12000.0 blocks/sec

Alternatively you can use WMA audio (which is called Divx Audio because it comes hacked with the DivX codec). Choose DivX ;-) Audio. Then choose 64 kbps, 44 kHz, stereo for Divx ;-). This is the most reliable WMA format that keeps best synching.



Select audio compression

<No compression (PCM)>
ACELP.net
CCITT A-Law
CCITT u-Law
DivX ;-) Audio
DSP Group TrueSpeech(TM)
GSM 6.10
IAC2
IMA ADPCM
Lernout & Hauspie CELP 4.8kbit/s
Lernout & Hauspie SBC 12kbit/s
Lernout & Hauspie SBC 16kbit/s
Lernout & Hauspie SBC 8kbit/s
Microsoft ADPCM
Microsoft G.723.1
MPEG Layer-3
MPEG Layer-3
SX5363S
Voxware MetaSound
Voxware MetaVoice
Voxware v1.1.6/1.1.8 File-Mode Codecs
Voxware v1.1.8 Bitstream-Mode Codecs
Windows Media Audio V1

36 kbps, 32 kHz, stereo	5Kb/s
32 kbps, 32 kHz, stereo	4Kb/s
22 kbps, 32 kHz, stereo	3Kb/s
32 kbps, 44 kHz, mono	4Kb/s
160 kbps, 44 kHz, stereo	20Kb/s
128 kbps, 44 kHz, stereo	16Kb/s
96 kbps, 44 kHz, stereo	12Kb/s
80 kbps, 44 kHz, stereo	10Kb/s
64 kbps, 44 kHz, stereo	8Kb/s
64 kbps, 44 kHz, stereo for DivX ;-)	8Kb/s
48 kbps, 44 kHz, stereo	6Kb/s
160 kbps, 48 kHz, stereo	20Kb/s
128 kbps, 48 kHz, stereo	16Kb/s

☑ Show all formats

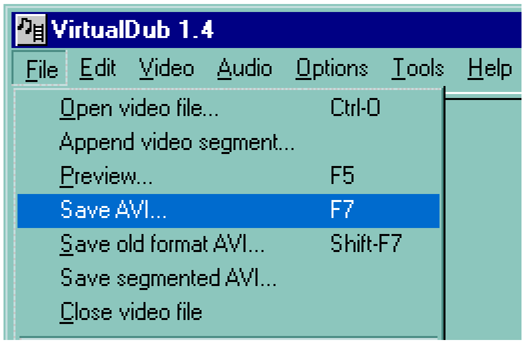
Format information

Format ID	0x0161
Bytes per block	372 bytes
Data rate	8010 bytes/sec
Granularity	21.5 blocks/sec

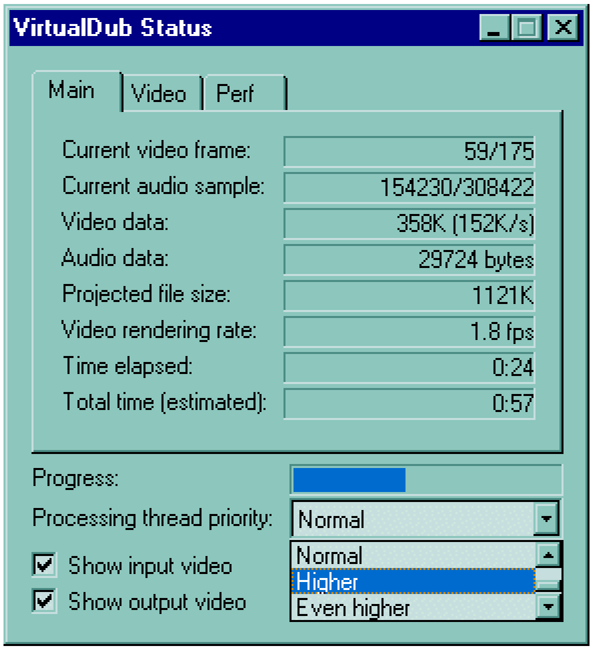
WMA audio compresses about 5 times faster than MP3 and the quality is slightly better at lower bitrates! Nevertheless, MP3 at 96 kBit/s is superior quality to 64 kbps WMA audio, so the choice is yours. One final warning about WMA audio. It doesn't always keep perfect timing. For example, if the move is long, then it may end up a quarter of a second or so shorter than the video length. I find this easy to correct by using my synching techniques. But if you want to more or less guarantee a perfect synch, then choose MP3.

PROCESS IT

Okay, all we need to do is select the Save AVI... option to save our final movie =)



Up will pop the progress box. You can uncheck the show 'input' and 'output' video options to speed it up a bit. You can also choose a higher priority setting to help VirtualDub do the job faster by hogging more CPU power.



Current video frame: 59/175
Current audio sample: 154230/308422
Video data: 358K (152K/s)
Audio data: 29724 bytes
Projected file size: 1121K
Video rendering rate: 1.8 fps
Time elapsed: 0:24
Total time (estimated): 0:57

Progress:
Processing thread priority: Normal

☑ Show input video
☑ Show output video

That's it! In less than 3 hours you should have perfect audio with your DivX video!

Open Any File with AVISynth

AVISynth is a really cool utility, yet it seems very few people are using it! The reason many haven't started is probably not because AVISynth is hard to use, but basically because they downloaded it, double-clicked on each file to try and get it working only to find that it doesn't seem to do anything!? So perhaps it may be a good idea if I explained what AVISynth does and how it can be used!

What can we do with AVISynth?

Lets say you downloaded a weird format video file and wanted to edit it with VirtualDub or convert it to Mpeg with Panasonic Mpeg encoder or with some other Mpeg / AVI etc., encoder. It would normally be impossible to do this because neither VirtualDub nor Panasonic Mpeg Encoder supports this format. Windows Media Player, however, is usually able to play just about any file with no problems. If it cannot play it there will usually be a codec you can install that will allow Media Player to do so. Well, if Media Player can play it usually AVISynth will let you open it in whatever application you need! This even applies to Mpeg-2 files! Yep, that's right! If Media Player can play your Mpeg-2 file then AVISynth will let you open them in almost any video editing or converting utility!

But that is not all! Before it sends the video to be opened in VirtualDub (or whatever application you want) it gives you the option to do all sorts of things to the video! You can resize it, crop it, sharpen, blend, separate interlaced fields, delete frames, duplicate frames, add 3:2 pulldown, reweave interlaced fields, bob deinterlace, convert between YUY2 and RGB and a whole host of other interesting and useful features!

How does it work?

AVISynth acts as a go-between between your video file and any video application you wish to use. You install AVISynth by putting a small .dll program in the Windows System folder. This is almost like a Windows upgrade, it then intercepts all files opened that are called something.**avs** and then runs them through AVISynth. In turn AVISynth decodes them using Media Players codec's and sends them to the application you were opening them with - cool huh!

Basically to do this you need to write a simple text file saying what file you wish to open (just like making a .lst file for Mpeg2avi) and rename the text file with an *.avs extension. Then, when you open this file in VirtualDub or Panasonic Mpeg Encoder (or whatever application you are using) AVISynth will take over and covert it for that application.

Don't worry, it may not sound too clear to you right now, but its *not* hard and after you have followed my step-by-step examples you will be able to set it up and open any file in seconds. So without further delay lets get started.

Lesson 1: DVD to AVI with AVISynth!

Since we are Mpeg-2 obsessed at Digital Digest, what better than to show you how to open a DVD in VirtualDub. As you know VirtualDub doesn't support Mpeg-2 and cannot open or edit Vob files...we will soon change that!

Can you open DVD or Mpeg-2 in Media Player?

As I mentioned, AVISynth can open just about any file that Media Player can handle. But if you cannot open a Vob file in Media Player then you will need to install the Ligos and WinDVD codec's first so Media Player can. This is easy enough to do:

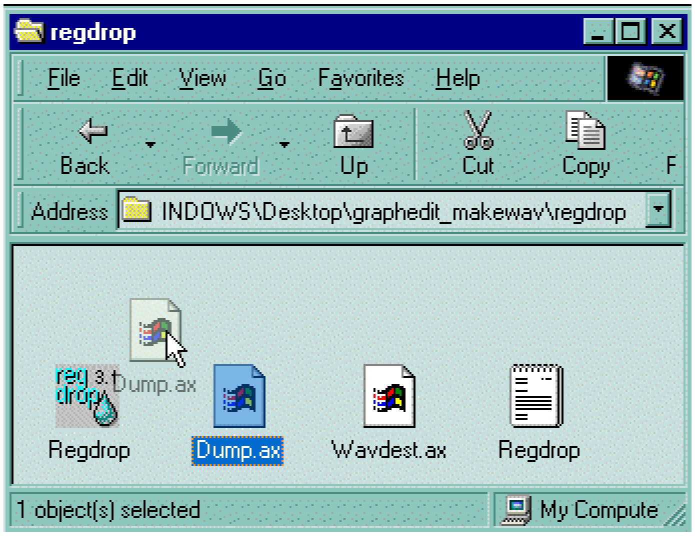
You will need:

RegDrop

Ligos Mpeg-2 Filter

WinDVD Audio Filter

Download the above files and drag the something *.ax file over the RegDrop program and your files are registered. Like this:



Note: *the picture above is only an example of what you do, you do not need any Dump.ax or Wavedest.ax filters.*

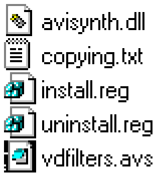
Installing AVISynth

<u>Before I start here are the things you will need:</u>

AVISynth 1.0 Beta 3

VirtualDub 1.4c

After you have downloaded and unzipped AVISynth you should end up with a folder containing the following files:



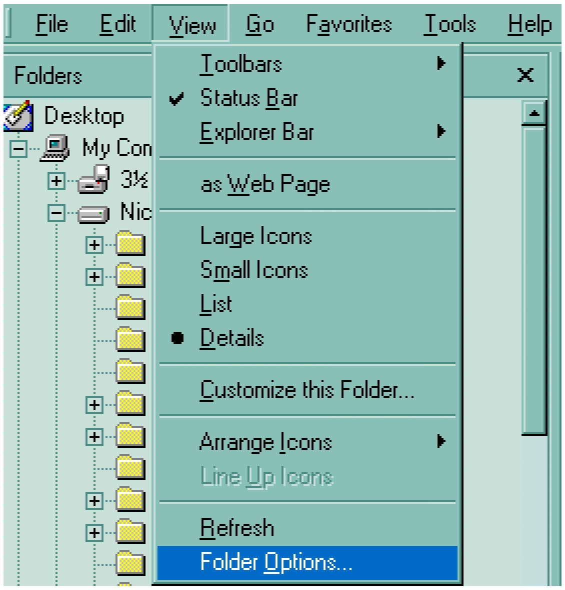
avisynth.dll
copying.txt
install.reg
uninstall.reg
vdfilters.avs

Cut and paste the first file called avisynth.dll into your C:\Windows\System folder. Then double-click on the install.reg file and the program will be installed. To uninstall it double-click on the uninstall.reg. To uninstall you can also delete the avisynth.dll file from your Windows\System folder but that's up to you.

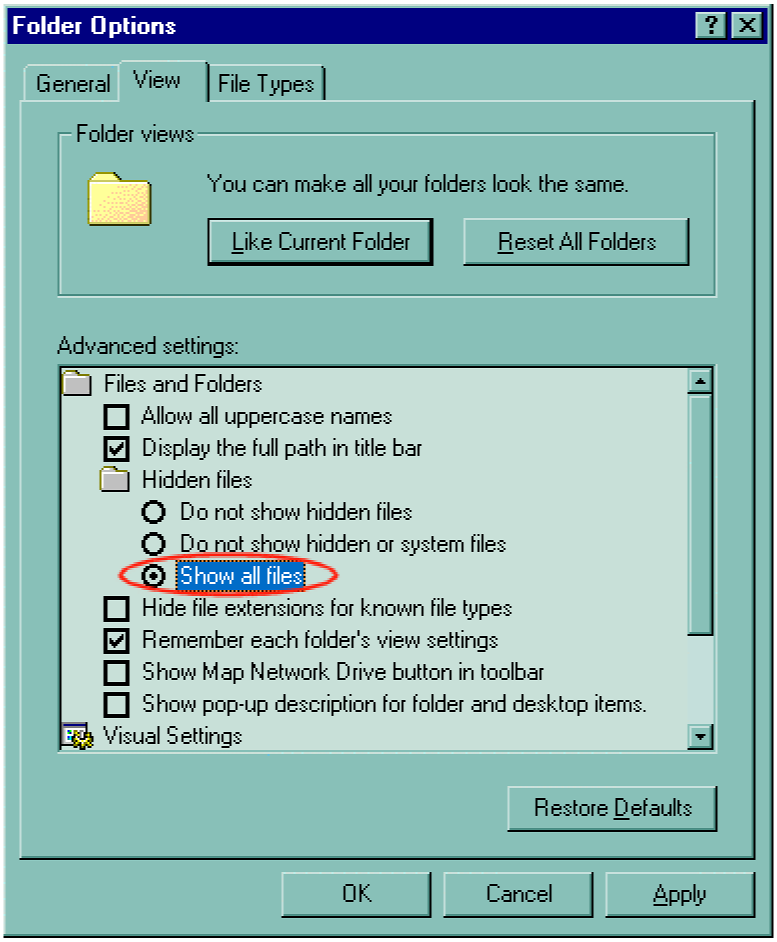
<u>Cannot find the DLL file?</u>

If you cannot find the avisynth.dll file then its very likely that windows is hiding it from you. Windows hides all files that have the DLL extension. in case you accidentally delete any of them and cause windows to become corrupted! To view *all* files you should open windows explorer and go to:

View > Folder Options...



Hit the View tab button and select the option: Show all files



Folder views

You can make all your folders look the same.

Advanced settings:

- Files and Folders
  - □ Allow all uppercase names
  - ☑ Display the full path in title bar
  - Hidden files
    - ○ Do not show hidden files
    - ○ Do not show hidden or system files
    - ● Show all files
  - □ Hide file extensions for known file types
  - ☑ Remember each folder's view settings
  - □ Show Map Network Drive button in toolbar
  - □ Show pop-up description for folder and desktop items.
- Visual Settings

This will allow you to see the avisynth.dll file and place it in the Windows\system folder :). If you are worried about people accidentally deleting DLL files you can turn the 'hide hidden or system files' option on.

Creating an AVS File

Okay, all the hard work is done! Make a folder on your main drive (usually C:) and put the Vob file you wish to open inside it. I have called my folder vobfiles and the file I'm going to open is called VTS_01_01.Vob.

So open windows notepad (or any text editor) and type the following into it:

```
DirectShowSource("C:\vobfiles\VTS_01_01.Vob")
```

As usual the colours are mine and here to help explain this line means. The DirectShowSource is a command that basically means open anything using Media Players codecs. Then, in brackets and quotation marks, we say what file we wish to open.

In this case I said look for the Vob file in the folder C:\vobfiles and called VTS_01_01.Vob.

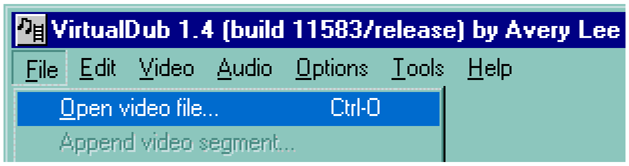
Save it!

Anyway, save the file as whatever you like but put the extension. *.avs on it instead of .txt or .doc etc. I'm calling my text file movie.avs. So when I choose Save in notepad I will type the filename "movie.avs" make sure you use quotation marks (" ") to force notepad to save it as avs instead of txt. Again, it will not work if the text file is called movie.avs.txt or movie.avs.doc or something like that.

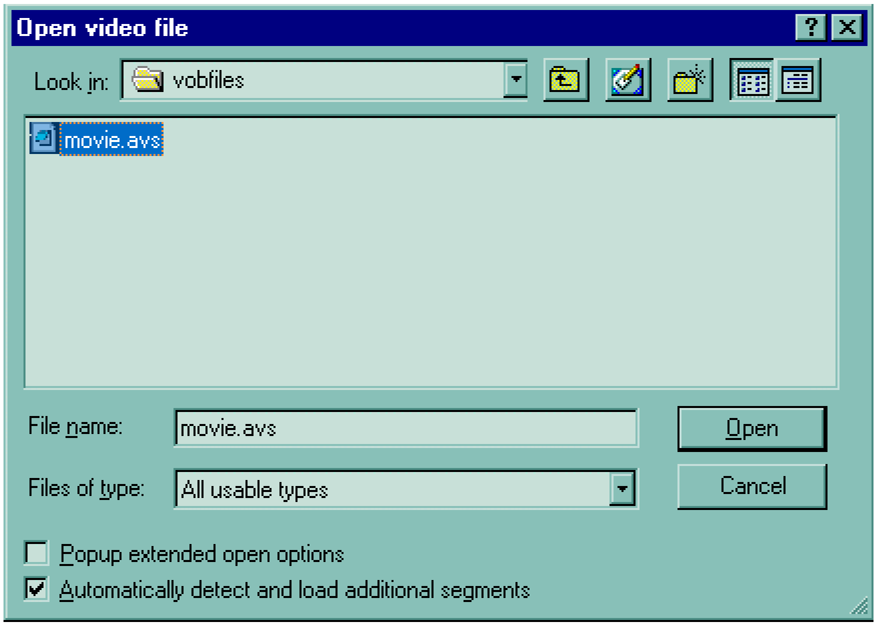
Opening the File in VirtualDub

Okay, time to open our Vob file. Were are going to use VirtualDub, but, as I said before almost any program will open the avs file. Load VirtualDub and go to:

File > Open


VirtualDub 1.4 (build 11583/release) by Avery Lee
File
Edit
Video
Audio
Options
Tools
Help
Open video file... Ctrl-O
Append video segment...

Find the avs file you made:



Look in: vobfiles

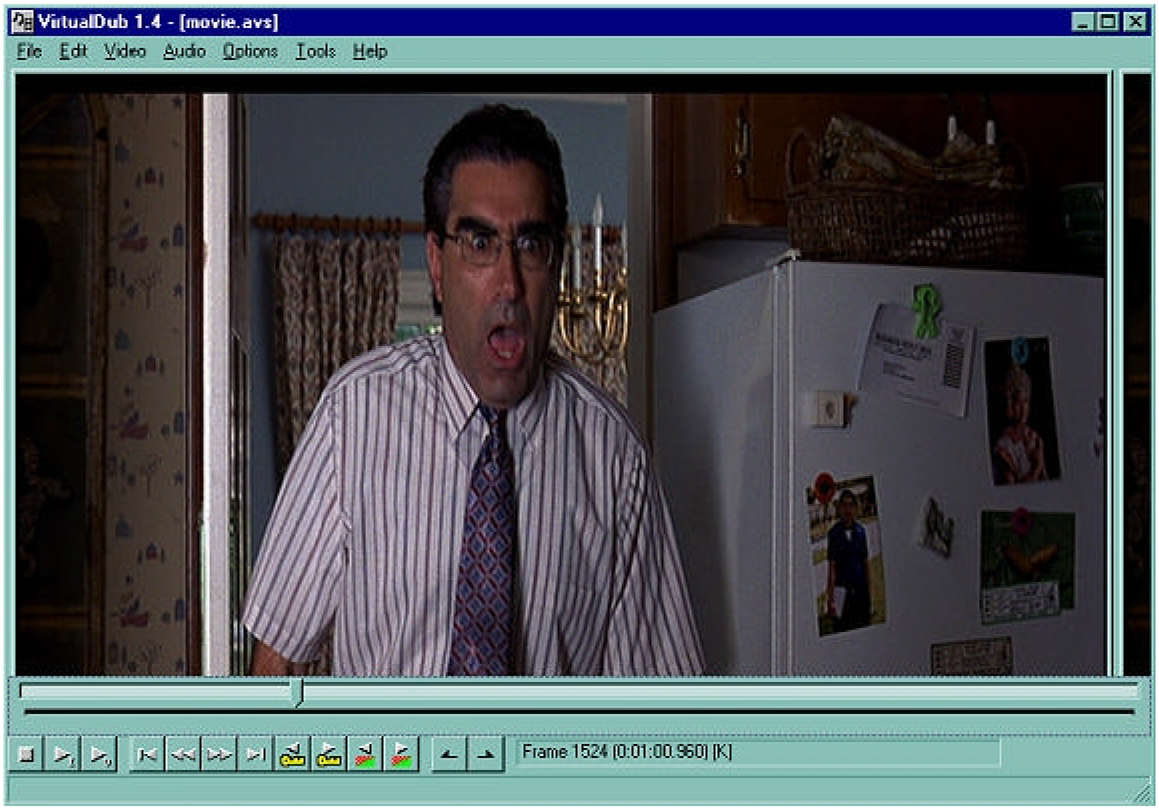
File name: movie.avs

Files of type: All usable types

□ Popup extended open options

☑ Automatically detect and load additional segments

And this is what you get below - VirtualDub reads it just like it was an AVI file!



That's right! Has he discovered how much his son likes apple pie, or has he just learnt about what AVISynth can do? You decide!

AVISynth doesn't offer the ultimate solution for DVD conversion. For one thing it cannot handle subtitles or multiangles directly like Flask Mpeg can. Also, if Media Player cannot open a Vob file or if it cannot play the sound you will not be able to get it in VirtualDub either! All in all VirtualDub may not be as fast or versatile at manipulating Vob files another dedicated program such as Mpeg2avi because it was designed solely for that purpose. Nevertheless, have a play and see what you can come up with :). But don't forget, AVISynth is not just for opening Vob files it can handle almost any file.

Advanced avs Files

That introduction tells us the basics, but there is so much more AVISynth can do with these files *before* they are given to VirtualDub or whatever application you may want to open them in. I will not explain every possibility because that's a mammoth task. Instead, I will explain how to use some of the basic commands and then you will know how to design your own better.

Opening ASF and MOV files

Its not an exact science but you can often open many MOV or ASF files. Both MicroSofts ASF and Apples QuickTime are generic codecs. This means that different codecs are used for both ASF and QuickTime files but they are still called ASF or MOV. What I'm trying to say is you can open some but not others. To open them you usually need to specify the Framerate. You do that like this:

DirectShowSource("C:\vobfiles\Video.ASF"**,fps=29.970**)

Opening Multiple Files

If you want to open lots of files so they open as one single file like making a mpeg2avi .lst file, all you need to do is add a plus (+) sign and tell it which files to join. For example, instead of this in out text file:

DirectShowSource("C:\vobfiles\VTS_01_01.Vob")

iles\VTS_01**_01.Vob+**DirectShowSource("VTS_01**_02.Vob**"**)+**DirectShowSource("VTS_01**_03.Vob**"**)+**DirectShowSource("

That would tell AVISynth to open vob files: 01, 02, 03 & 04 as though it were a single file.

Other File Opening Options

DirectShowSource("*filename&location***")**. As we have already seen, this command will open just about any file that Media Player can play. This includes audio files such as WAV, MP3 etc. For example, you could open just about any audio file in an audio editing package such as SoundForge even if SoundForge didn't support that format. Just make an avs text file like we did before and open it in SoundForge the way you did VirtualDub. Unfortunately ASF files do not work correctly and look upside down and ma not have any sound :(.

There is also an experimental Vob opening command called:

VOBVideoSource("*filename&location***")**

But this may or may not work for you so you may wish to swap DirectShowSource and VOBVideoSource.

AVISynth also offers other AVI opening commands. The reason for this is because handler may be able to open a certain file type better or process it faster, or may be supported better by the application AVISynth is opening it in.

```
AVISource("filename&location")
AVIFileSource("filename&location")
OpenDMLSource("filename&location")
```

So between this lot you should be able to open almost anything.

```
WAVSource("filename&location")
```

This command is designed for opening wave files.

Video Editing Commands

Once you've decided how you will open the file you can command AVISynth to do stuff with them beforeit sends them to the application you wish to open it with. This is very useful if you need to crop or resize or something like that but the application you want to open it in doesn't let you. All this crap may sound a little like we are about to learn a programming language, but really its just a matter of listing what things you want done. For example, say we want to open the file but make it half the size it was originally, we could put:

```
DirectShowSource("filename&location")
ReduceBy2
```

That's all we need to do! Save the avs text file as usual and open it in VirtualDub to see how it looks. You will find that it opens the video file but at half of its original size!

Okay, lets add to that. Lets say the application you wish to open the file in prefers the RGB AVI format but your original file is in YUY2 format we could put:

```
DirectShowSource("filename&location")
ReduceBy2
ConvertToRGB
```

This time it will open the file, half its size and change the format to RGB. Notice also how I have just added this new command to the last one like a shopping list :). Each command can usually be just stuck underneath the last and AVISynth will perform them in that order.

File Command Order

The order in which you ask AVISynth to perform commands can affect both what the final result looks like and how fast the option is completed. For example, say we opened a huge 720 x 576 pixel DVD file. We want to sharpen the image and resize it to 352 x 288 according to a bicubic resize. We could do it like this:

```
AVISource("filename&location")
Sharpen(1.0)
BicubicResize(352,240)
```

But that is a slow way to do it because this way the sharpen filter comes first in the list. Obviously it takes longer to sharpen a 720 x 576 image than it takes to sharpen a 352 x 288 image! So instead its better to do it in this order:

```
AVISource("filename&location")
BicubicResize(352,240)
Sharpen(1.0)
```

This way the image is shrunk first and then sharpened after.

We can also repeat effects. For example, if you wanted to make it twice as sharp you could do this:

```
AVISource("filename&location")
Sharpen(1.0)
Sharpen(1.0)
```

Note: Not all programs will be able to handle every option or some repeated sequences of options so its trial and error. For example, VirtualDub will often crash if you try and feed it a Vob file with multiple AVISynth alterations added it it.

Frames & Fields

I couldn't finish this article without saying a few words about AVISynth's Field and Framerate commands. So here are a few that should help you get started.

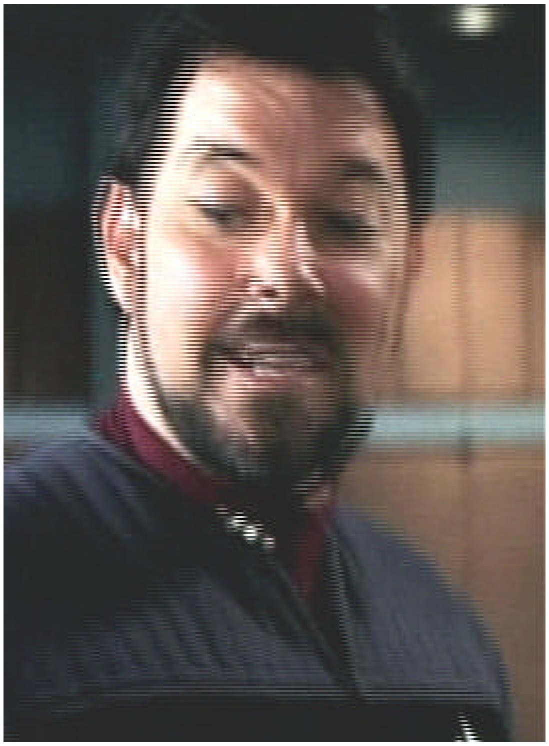
Bob

This is one of the easiest ways to solve those NTSC interlace problems when opening the file in an application that cannot deinterlace it. You could use:

DirectShowSource("*filename&location*")

Bob

Simple as that! You may be interested to know that Bob is the method that PowerDVD uses most often to solve those annoying interlace combing problems. This will fix all interlaced files that look corrupted like the following picture:



For more details on interlace problems you could check out my article on '*NTSC / PAL & Interlace*'.

SeparateFields

Another handy little feature that lets you separate the top and bottom frames of an interlaced picture. If for example you will be resizing an video file from 576 pixels high to 288 pixels high or lower an easy way to both speed things up and completely solve all interlace problems without any loss of detail s literally to separate fields. This can be done thusly:

DirectShowSource("*filename&location*")

SeparateFields

This will double the amount of frames in the movie because its now giving you frame 1 (top field) frame 2 (bottom field). So a 30fps movie becomes a 60 fps movie. So you may want to use the SelectEven or SelectOdd commands to fix this. Either command will select every other frame in the file. So select odd may just leave you with only bottom or top video fields. To use it just add SelectOdd to the list, like this:

```
DirectShowSource("filename&location")
SeparateFields
SelectOdd
```

This will make the 60 fps back into 30. All these options shouldn't take very much processing power because AVISynth is only taking the information from the file in a certain order, it doesn't need to do heavy calculations on it.

You may be wondering why your movie now looks like a pancake! This is because it only has half the lines it did originally. Like I said you intend on resizing the movie anyway and this solves both interlace problems and cuts down on CPU power because there is of a picture to resize :). To make the final image complete we simply use a BilinearResize option like this:

```
DirectShowSource("filename&location")
SeparateFields
SelectOdd
BilinearResize(384,208)
```

LEARNING MORE COMMANDS

Want to Learn More about AVISynth commands? Here they all are! Just click on each one and it will take you to the info found on the AVISynth's official Webpage:

- Filters that read or produce a video clip
  - AVISource / OpenDMLSource / AVIFileSource / WAVSource
  - SegmentedAVISource
  - DirectShowSource
  - IPCSource
  - Blackness
  - Version
- Filters for changing the frame size
  - HorizontalReduceBy2 / VerticalReduceBy2 / ReduceBy2
  - BilinearResize
  - BicubicResize

    - Crop / CropBottom
    - AddBorders
- Filters for image processing
    - Levels
    - Blur / Sharpen
    - SpatialSoften / TemporalSoften
    - FixLuminance
    - Letterbox
    - Subtitle
    - PeculiarBlend
- Filters for converting between color spaces
    - ConvertToYUY2 / ConvertToRGB
- Filters for dealing with interlaced video
    - AssumeFrameBased / AssumeFieldBased
    - ComplementParity
    - SeparateFields
    - Weave
    - DoubleWeave
    - Bob
    - SwapFields
- Filters for editing
    - Trim
    - AlignedSplice / UnalignedSplice
    - Dissolve
    - Reverse
    - FadeOut / FadeOut2
    - SelectEven / SelectOdd
    - Pulldown
    - Interleave
    - SelectEvery
- Filters for tweaking individual frames
    - FreezeFrame
    - DeleteFrame
    - DuplicateFrame
- Audio-related filters
    - AudioDub
    - DelayAudio
- Debugging
    - Subtract
    - ShowFrameNumber
    - StackHorizontal / StackVertical
    - ShowFiveVersions
- Meta-filters
    - Animate

AVISynth Frameserving with FlasKMpeg

Before we get started with this guide it would be helpful if you read through the AVISynth Guide first so you understand just what AVISynth is and what it can do. That guide also explains how to install AVISynth, which you will also need to do :).

What is Frameserving anyway? Frameserving is using one utility to decode or read a video file and then, after it has been decoded, sending it straight to any other video editing or encoding application. For example, we could decode a DVD Vob file using Flask Mpeg and then use AVISynth to 'frameserve' it to another utility such as TMPGEnc and convert it to a SVCD or VCD. Or we could frameserve it to VirtualDub and add filters and effects before we encoded it into another format such as DivX and so on. Frameserving is the holy grail of the video converting scene because it allows us to pick and choose what encoders we want to use without the need to convert our video file into a huge uncompressed AVI format first!

Of course some formats and some utilities just cannot support frameserving, so we still are limited to what we can do. For now I will describe how to use the AVISynth plugin for Adobe Premiere to frameserve from FlaskMpeg into almost any other application.

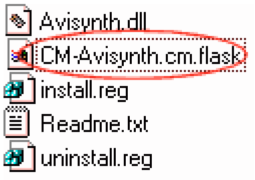
Before I start here are the things you will need:

AVISynth

AVISynth Premiere Plugin

Note: *the plugin I'm using is the one found on http://www.videotools.net/ but the official AVISynth one should work just as well in most cases and can be obtained at http://www.math.berkeley.edu/~benrg/avisynth.html.*

Installing the Plugin



Avisynth.dll
CM-Avisynth.cm.flask
install.reg
Readme.txt
uninstall.reg

After you have installed AVISynth all you need to do to make Flask Mpeg Frameserve is to copy and paste the CM-Avisynth.cm.flask plugin (encircled in red above) into the same folder as Flask Mpeg. This plugin was designed for Adobe Premiere so if you want to use it with that too just rename it to cm-avisynth.cm.prm and copy it into your Adobe Premiere's plugin's folder.

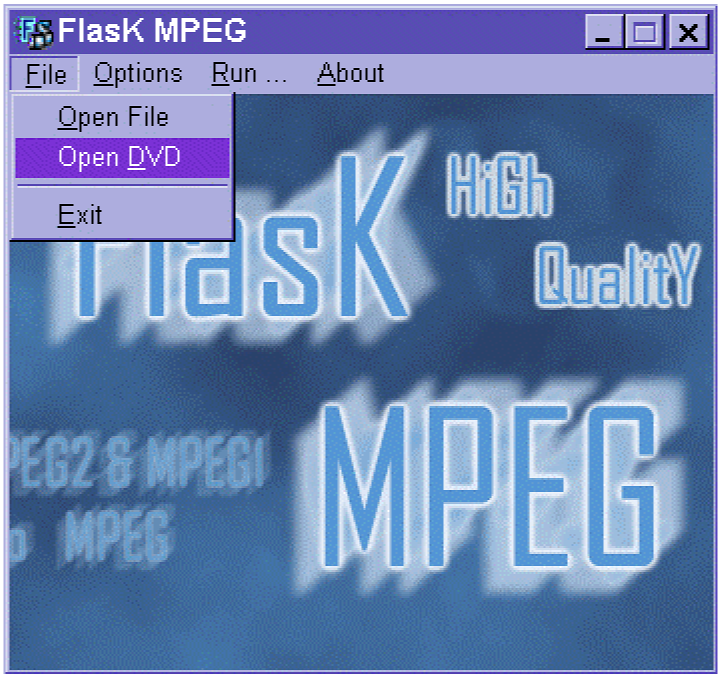
Frameserving from Flask Mpeg

It is technically possible to use AVISynth and an application such as VirtualDub to convert a Vob into DivX or something like that. But I think to try is a bit crazy, because it wasn't really designed for that. You will almost always run into serious problems if you try. Flask Mpeg, however, is another story. Flask has a tried and tested with DVD reading and can convert almost any DVD video including audio, multiangle titles, subtitles, keeping correct aspect ratios, cropping and so on. In short, Flask Mpeg is probably our best bet for decoding a DVD if we wish to frameserve it to be used by another application such as TMPGEnc etc.

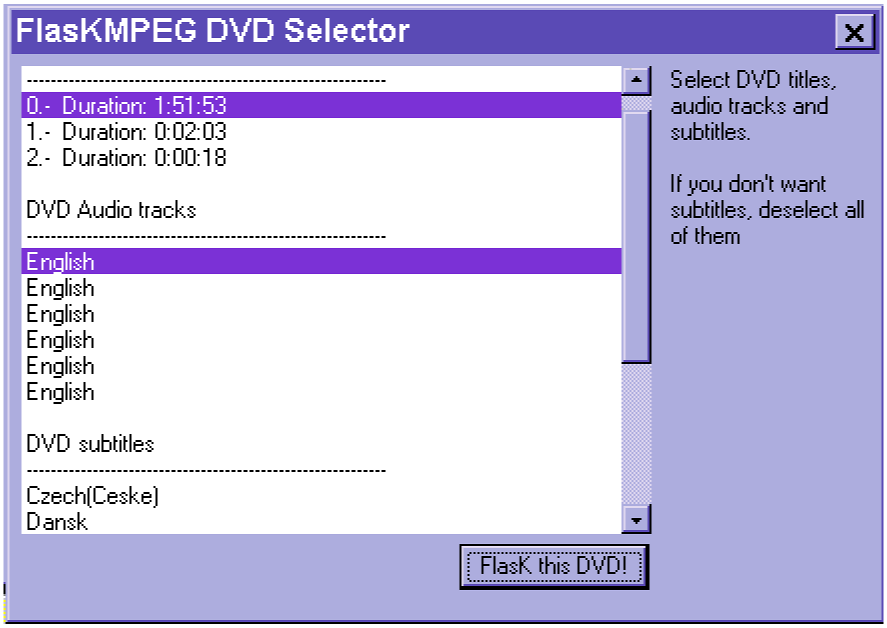
Ok, we have put the AVISynth Plugin into the Flask folder. I'm assuming you have ripped and decoded the Vob files and IFO files to your hard disk like have explained in my Flask Mpeg guide. So imagine you were going to convert a DVD using Flask Mpeg as normal. I have repeated the process breifly here so just skim through it:

OPEN THE FILE WITH 'Open DVD'

Select 'Open DVD' and find the .IFO file for your movie.



Up will pop something like the picture below:


FlasKMPEG DVD Selector

0.- Duration: 1:51:53
1.- Duration: 0:02:03
2.- Duration: 0:00:18

DVD Audio tracks

English
English
English
English
English
English

DVD subtitles

Czech(Ceske)
Dansk

Select DVD titles, audio tracks and subtitles.

If you don't want subtitles, deselect all of them

Select the movie Duration, in this case 1.51.53. This will usually be the first one in the list, but you can usually see from the length which one to choose.

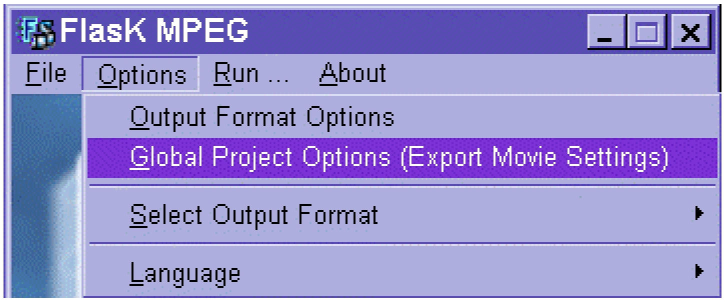
Next choose the language. Obviously they cannot ALL be English so choose the first and encode a minuet of the film and listen to it. If its not English choose the next in the list, and the next and the next etc., until you find the correct one. Or open the DVD in a DVD player and see the order they are in, usually it will be the same.

Lastly we have subtitles. Flask is not always able to do subtitles correctly, so again, you will have to try it and see. There are other ways to get subtitles, though; why not check out my subtitle ripping section for this.

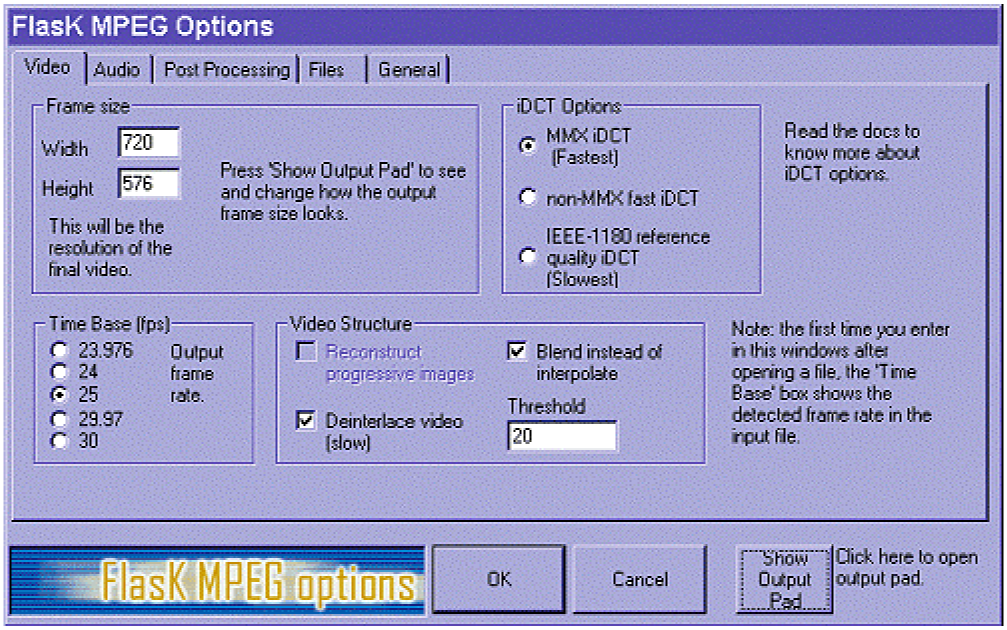
Now Press FlasK this DVD!

GLOBAL PROJECT OPTIONS

Select Global Project options.



VIDEO TAB



Frame size

Width 720

Height 576

This will be the resolution of the final video.

Press 'Show Output Pad' to see and change how the output frame size looks.

iDCT Options

● MMX iDCT (Fastest)

○ non-MMX fast iDCT

○ IEEE-1180 reference quality iDCT (Slowest)

Read the docs to know more about iDCT options.

Time Base (fps)

○ 23.976
○ 24
● 25
○ 29.97
○ 30

Output frame rate.

Video Structure

□ Reconstruct progressive images

☑ Deinterlace video (slow)

☑ Blend instead of interpolate

Threshold 20

Note: the first time you enter in this windows after opening a file, the 'Time Base' box shows the detected frame rate in the input file.

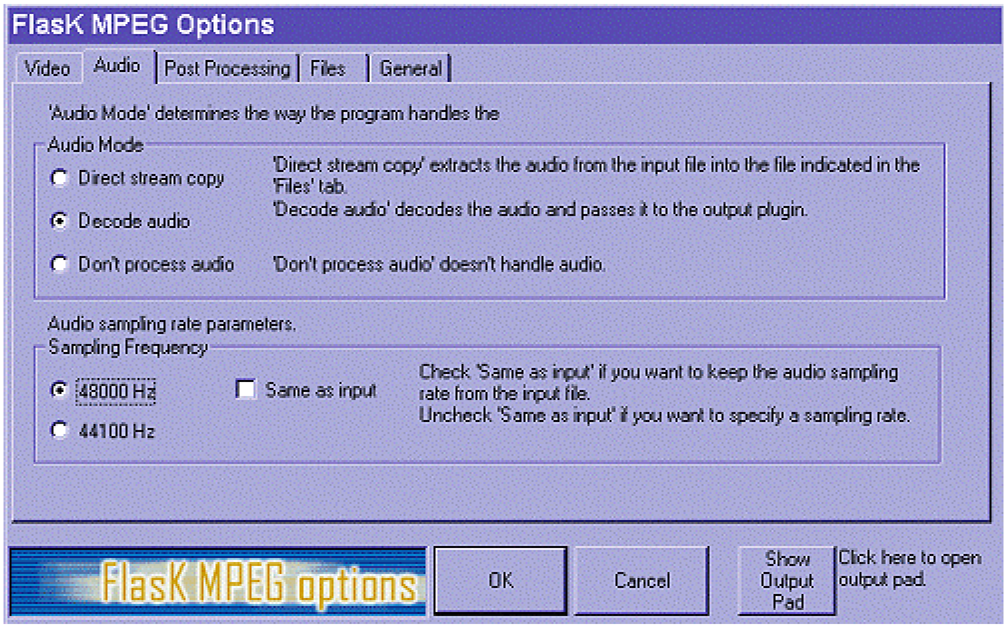
Click here to open output pad.

Frame Size: set the Width and Height.

Time Base (fps): Flask will normally choose the best framerate for your movie. All PAL movies (European) are 25 frames per second (fps). So if you know your movie is PAL make sure your movie is set to 25. All North American movies are NTSC which means they are 29.97 fps. BUT because of the way they are encoded to DVD most will appear Flask will choose 23.976 fps. This is usually correct so don't change it. As always test a short clip before you do a whole movie to make sure its ok.

AUDIO TAB

On to the Audio tab always select 'Decode audio' if you want sound. For DVD's un-check the 'same as input box' and select 48000 Hz (just to make sure you have the right setting). Some utilities require you use 44100Hz so its tough noogie, you may have audio synchronization problems because of this depending on the application you are frameserving to.



'Audio Mode' determines the way the program handles the

Audio Mode

○ Direct stream copy — 'Direct stream copy' extracts the audio from the input file into the file indicated in the 'Files' tab.

● Decode audio — 'Decode audio' decodes the audio and passes it to the output plugin.

○ Don't process audio — 'Don't process audio' doesn't handle audio.

Audio sampling rate parameters.

Sampling Frequency

● 48000 Hz □ Same as input

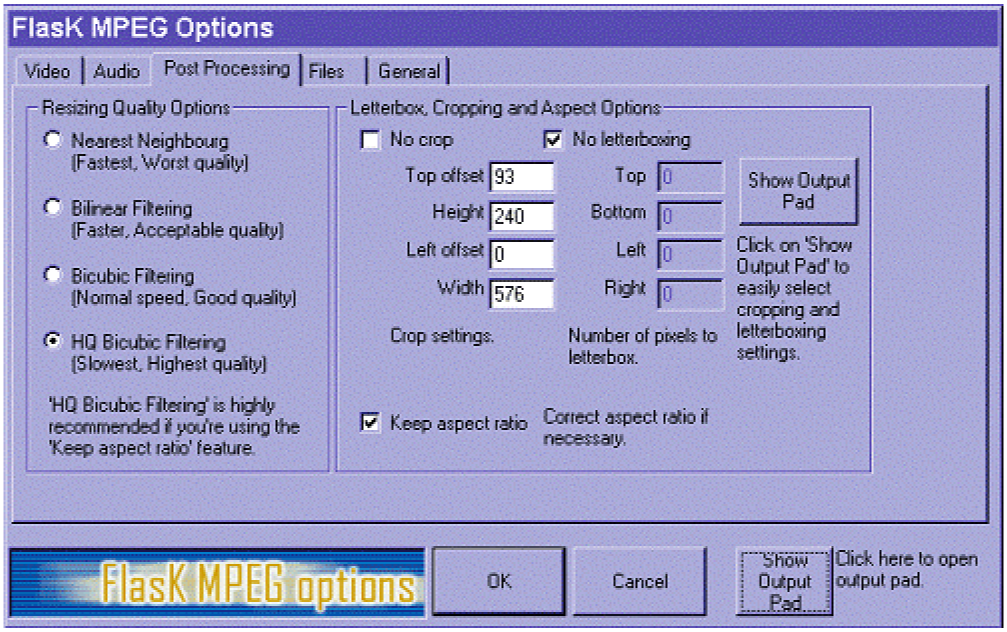
○ 44100 Hz

Check 'Same as input' if you want to keep the audio sampling rate from the input file.
Uncheck 'Same as input' if you want to specify a sampling rate.

Click here to open output pad.

POST PROCESSING TAB

This section deals almost exclusively with resizing. Never use 'Nearest Neighbouring' unless you are not resizing the picture because the quality is crud.



Resizing Quality Options

○ Nearest Neighbourg (Fastest, Worst quality)

○ Bilinear Filtering (Faster, Acceptable quality)

○ Bicubic Filtering (Normal speed, Good quality)

● HQ Bicubic Filtering (Slowest, Highest quality)

'HQ Bicubic Filtering' is highly recommended if you're using the 'Keep aspect ratio' feature.

Letterbox, Cropping and Aspect Options

□ No crop ☑ No letterboxing

Top offset 93 Top 0
Height 240 Bottom 0
Left offset 0 Left 0
Width 576 Right 0

Crop settings. Number of pixels to letterbox.

Click on 'Show Output Pad' to easily select cropping and letterboxing settings.

☑ Keep aspect ratio — Correct aspect ratio if necessary.

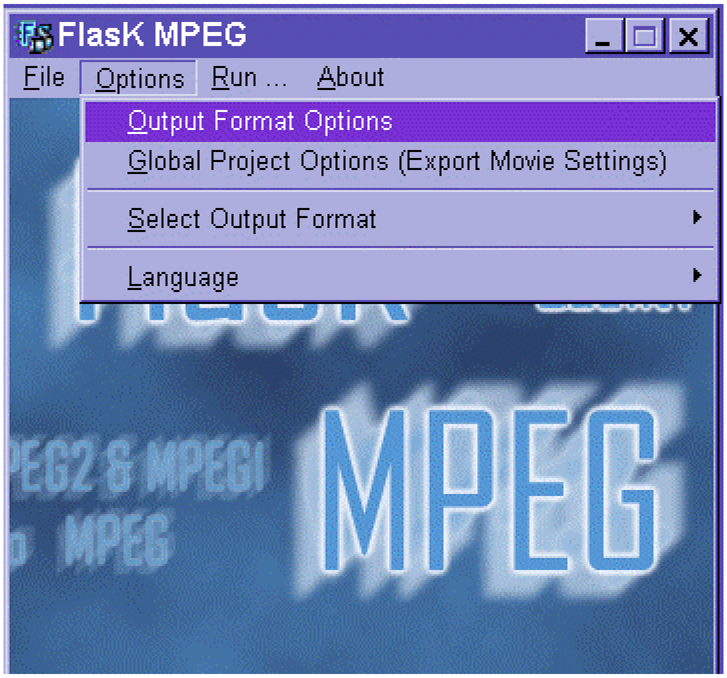
Click here to open output pad.

Keep aspect ratio: PAL users should always tick this box unless you know you don't need it. This is even more important with Widescreen DVD's. If you use NTSC DVD's the image may become stretched slightly wrongly. If you notice this uncheck the 'keep aspect ratio' and work out the ratio yourself. See the article *"Resizing DVD's with Correct Aspect Ratios"* in my appendix.

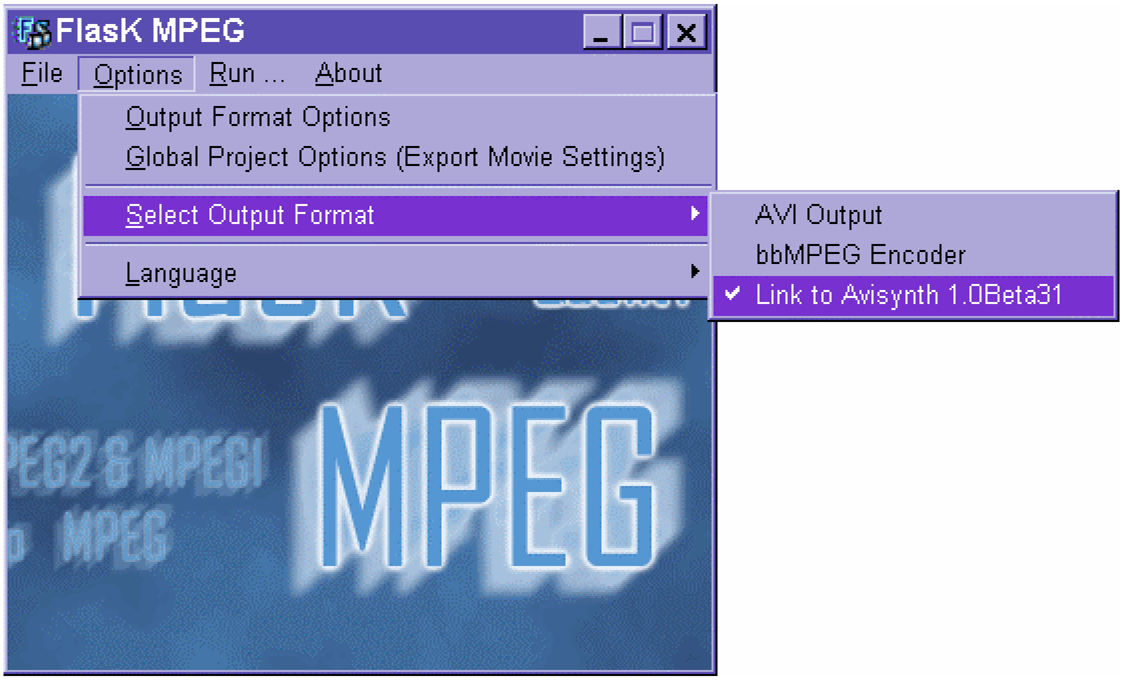
Crop & Letterboxing: All the settings for cropping and letterboxing the DVD can be entered here or the output pad can be used. For detailed information on how to resize a movie in Flask read section 2nd of this guide: *"resizing the video"*.

OUTPUT FORMAT OPTIONS

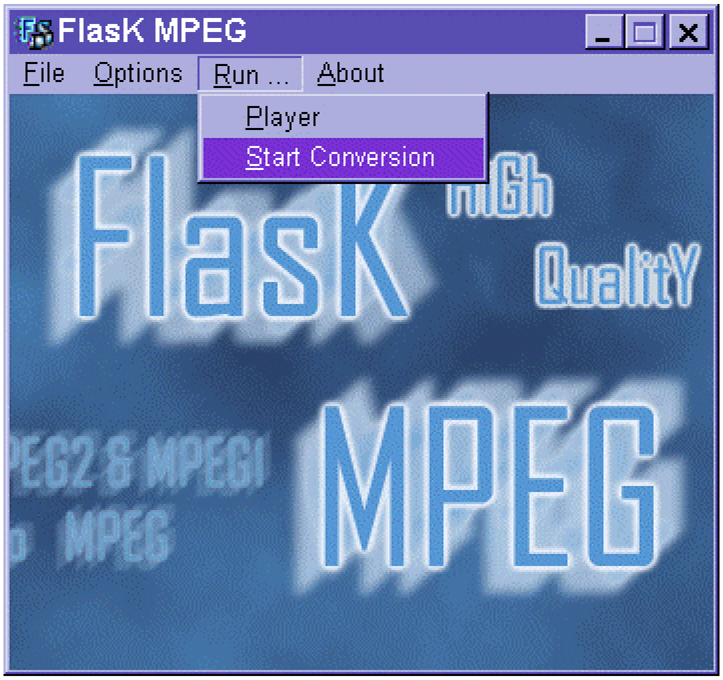
Normally you would now be setting your output format options.



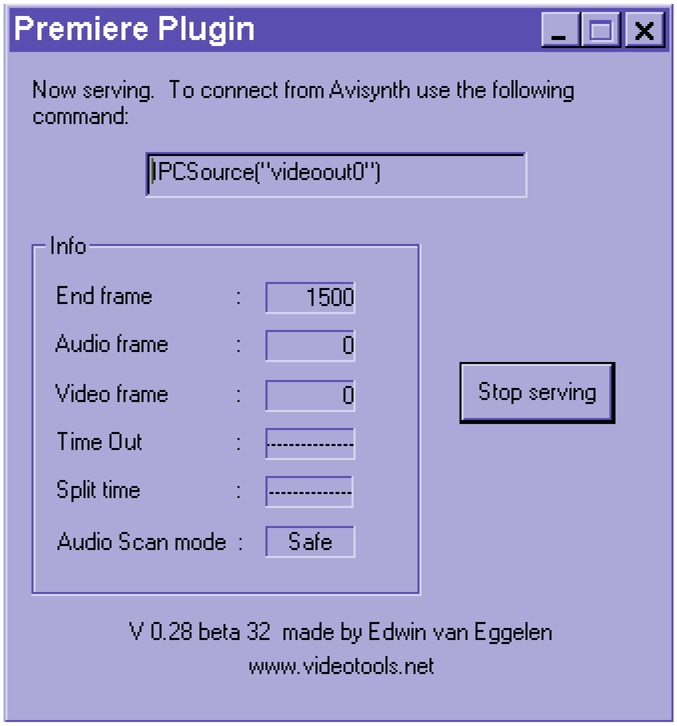
But because we are frameserving we do not need to do this so ignore that stuff. Instead go to 'Select Output Format' and choose 'Link to AVISynth':



And then start converting as usual:



Then up will pop the following box telling you what the file that Flask is frameserving out is called:



Now serving. To connect from Avisynth use the following command:

IPCSource("videoout0")

Info

End frame : 1500
Audio frame : 0
Video frame : 0
Time Out : --------------
Split time : -------------
Audio Scan mode : Safe

V 0.28 beta 32 made by Edwin van Eggelen
www.videotools.net

MAKING YOUR AVS FILE

If you recall my previous AVISynth guide you will know what I'm talking about next. Open notepad and paste exactly what AVISynth has just told us into it, namely:

```
IPCSource("videoout0")
```

This is the name of the video file that is being frameserved from Flask Mpeg. The name videoout0 will be whatever you named your output in Flask, but thats not important because the pop-up box has told us exactly what to write :).

Save it!

Okay, save the file as whatever name you like but put the extension. *.avs on it instead of .txt or .doc. I'm calling my text file video.avs. So when I choose Save in notepad I will type the filename "video.avs" make sure you use quotation marks (" ") to force notepad to save it as avs instead of txt. Again, it will not work if the text file is called video.avs.txt or video.avs.doc or something like that.

Thats it! Open the file we just made called: video.avs in any encoder or utility that can open AVI files! For example, open it in TMPGEnc and create a SVCD or VCD. TMPGEnc gives the highest quality and most configurable settings of any SVCD or VCD encoder I have ever seen - and its free! Just remember that instead of opening the AVI just open the *.avs file you made instead :).

MakeFilm TNG™

The Next Generation in Single CD DivX Creation!

For those of you who used the old MakeFilm method, read on...this way is a much easier and even better quality

Notes

Things have been moved fast since Doom and me invented this DivX joining idea. There are currently a few contenders for DivX joining and I thought it only fair to let you try the competition out yourselves. This MakeFilm guide is here because at the time I wrote this it was and is in my opinion, the best method. But since I'm getting emailed updates of all kinds of programs almost on a daily basis at the moment I will merely add links to all the joining programs so you can try them. That way I don't have to keep updating this thing every day.

MakeFilm TNG 1.2

Project DivX 1.5 *(Read Doom9's Guide on this one* ***Here****)*

AVI Revolution 2.0

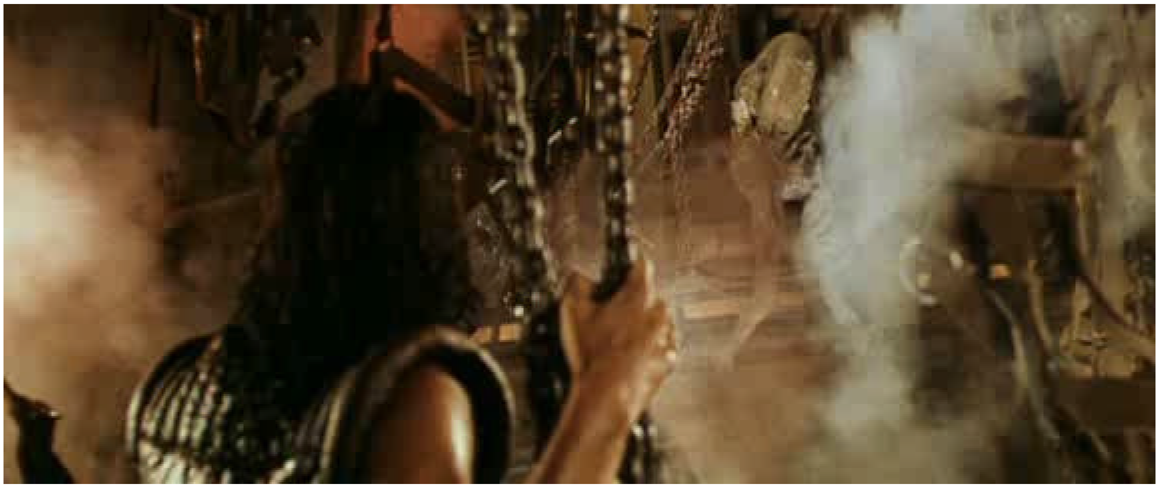
Compress AVI 1.04

Why should I use MakeFilm - TNG?

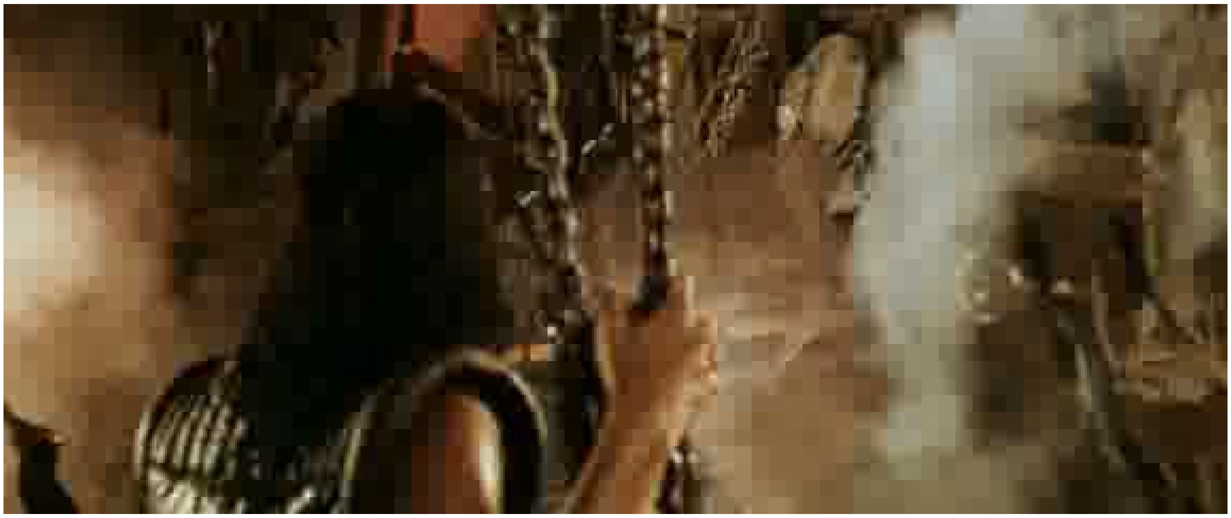
Because it's FASTER, SIMPLER and MORE ACCURATE than any competitive product! Now with *Full* NTSC support!!

What is MakeFilm TNG? Everyone is looking for the best settings to make the perfect DivX movie from their DVD. This was what urged me to write the articles: Divx quality check' and 'How to Join Divx Avi's and Double the Quality of Your Rips' and a whole bunch of other guides. Ideally the Microsoft Mpeg-4 codec should have been designed like Mpeg-2, which allows you to set both a minimum bitrate (so the video quality doesn't go below your needs) and a maximum bitrate (so it doesn't go so high that it doesn't fit into the CD you wish to put the final movie).

We have two DivX codecs: the Fast Motion, which, as its name suggests, is optimized for fast action scenes. It does normal scenes really bad quality but action scenes it does really really great quality!! With the Fast Motion codec you set the maximum bitrate. For example, here is a scene done with the Fast Motion codec 6000kbps:

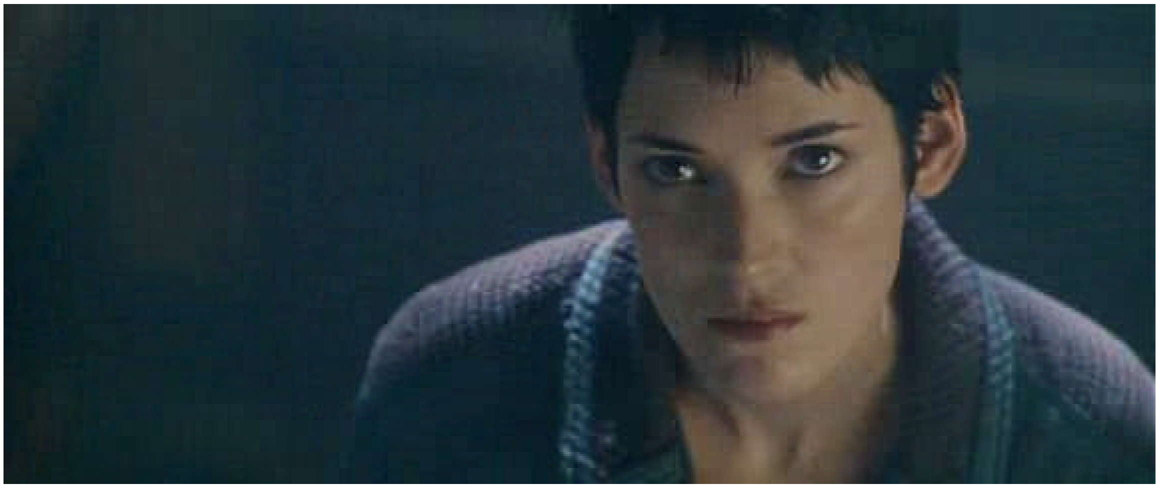


And here is the same scene with the Low Motion codec same size but at 600kbps:



Conversely, the Low Motion Codec does really really great quality, but only on low action scenes. For this codec you set the maximum bitrate for your movie. For example, here is a non-action scene using the Low Motion Codec at 600kbps:

And again, here it is with the Fast Motion codec 6000kbps:



As you can see, even at 6000kbps the Fast Motion codec cannot equal the Low Motion codec set at 600kbps. But then again, for the Low Motion codec to beat the Fast Motion on Fast action scenes it needs to be set at 1500kbps or above. This results in very large file sizes!

I'm sure its quite obvious to you that if we could split a movie up into fast action scenes and low action scenes, and then encode one lot with the Fast Motion codec and the other with the Low Motion Codec and put them together we would end up with a movie that gives us The Best of Both Worlds! And this was mainly what my how to join Mpeg-4 avi articles was about. Of course this article no longer exists because of MakeFilm.

The real problem with doing this was the difficulty in making two or more separate movies and joining the best quality bits together to make a finished movie with no noticeable transitions between scenes. My previous article was aimed at fixing a few very bad scenes rather than a suggestion to take on a task comparable to the labors of Hercules!

But now, thanks to a great fellow who goes by the name of MI-CHI, there is a way to make this process almost as easy as making the original movie itself! Let me introduce to you MakeFilm!

What Does MakeFilm TNG Do?

It allows us to automatically join the best parts of one Fast Motion and one Low Motion movie into one single super high quality movie. It has the most accurate bitrate calculation method of any other program and should be able to fit your Divx movies almost exactly to 650MB! You no longer even need to choose where it cuts the movie because it has built in quality algorithms that can decide which are the best parts from each movie. It even allows you to save an extra 50MB or more of space by re-encodeing the end credits! Those of you who who have tried the other joining programs, will be pleased to know that MakeFilm beats Compress AVI and Divx Project hands down, both in speed and in quality. MakeFilm TNG can combine combine two Divx movies in literally a few minuets! Anyway, enough of the blurb, I'll take you through what you need to do bit by bit.

Before I start you will need:

MakeFilm TNG 1.2

VirtualDub 1.4c

Note: *As one final note MakeFilm 1.2 now has the ability to make a text file explaining the bitrates used on your movies and also allows you to save your favorite settings.*

Step 1: Making the Fast Motion Movie

The Fast Motion movie should *always* be made the first movie you encode if you want to make the best use of MakeFilm! This is because the bitrate you should use for the second movie is based on the first - rather like 2 pass VBR encoding for Mpeg-2.

Single CD Divx

If you are making a single Divx movie I see no reason to use any less than 2000 kbps, unless your movie is very very long! So use 2000 kbps or a little more.

Double CD Divx

Two CD rips are a little more tricky. If your movie is very long and will be hard to fit on two CD's with any good quality, then use about 6000 kbps Fast Motion.

But MakeFilm was designed mainly for single CD rips! Here is the reason we have a problem with a double CD rip: the Low Motion codec tends to give better results than the Fast motion codec when it is set at a bitrate of somewhere between 1000 to 1200 kbps or more! Check out my quality and bitrate guide for detailed examples of this. So, even if we use the Fast Motion codec set at 6000 kbps it is possible that the Fast Motion movie will not look any better than the Low Motion movie!!

There is a possible solution for this, but it is not always easy to guess the best bitrate to use. Basically we can try to make two Low Motion movies. One at a very low bitrate, something like 700 kbps, and one at a high bitrate like 1500-2000 kbps. MakeFilm's calculator will not help us now, because as far as its concerned the high bitrate movie is already larger than the size you want to fit it into!

Setting Keyframes

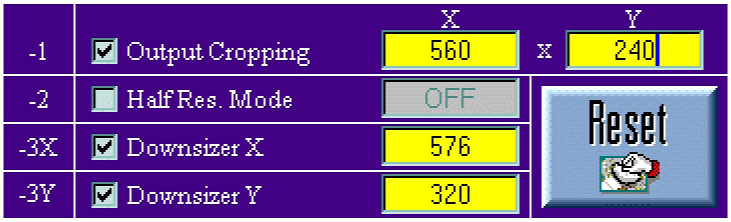
You should *always* set keyframes to 1 every second, unless your computer cannot play them back at 100% with this setting. Doing this will make scene selection much more accurate.

Setting Smoothness / Crispiness

The Divx codec likes to drop frames at low bitrates to save space. This sometimes causes bad frames, (i.e. your video stops but the audio continues). And it can also cause MakeFilm (or any joining program) not to work because both movies must have the same amount of frames in the same places. If you must use a quite low bitrate, such as 600 kbps or less then you may have problems unless you set the smoothness quite low. It is possible to have frame drops as low as 30% smoothness but usually you will be okay at 60-75%. So far I've been OK with these settings but you may not be so lucky.

Making the movie in Mpeg2avi

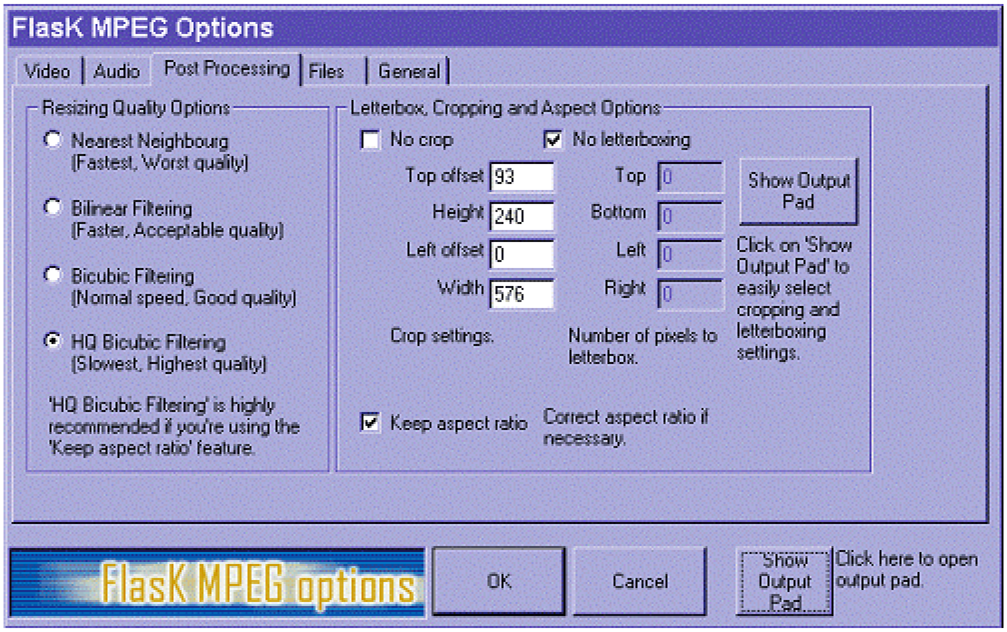
Write down or save the *resizing* settings you use to make the Fast Motion movie. So that when you make the Low Motion movie after it will look exactly the same. If the movie is cropped or resized wrongly you will have a corrupted file because MakeFilm cannot join two different resolutions!



		X		Y
-1	☑ Output Cropping	560	x	240
-2	□ Half Res. Mode	OFF		
-3X	☑ Downsizer X	576		
-3Y	☑ Downsizer Y	320		

Making the movie in Flask Mpeg

Again you must make take note of he settings you used to make your Fast Motion movie. The most important thing is resizing and cropping. Use the normal output pad as you normally would to crop and resize your Fast Motion movie. But, because its hard to get the same settings twice in Flask by hand, you must write down the numbers too - look below:



Resizing Quality Options

○ Nearest Neighbourg (Fastest, Worst quality)

○ Bilinear Filtering (Faster, Acceptable quality)

○ Bicubic Filtering (Normal speed, Good quality)

● HQ Bicubic Filtering (Slowest, Highest quality)

'HQ Bicubic Filtering' is highly recommended if you're using the 'Keep aspect ratio' feature.

Letterbox, Cropping and Aspect Options

□ No crop

☑ No letterboxing

Top offset 93

Height 240

Left offset 0

Width 576

Crop settings.

Top 0

Bottom 0

Left 0

Right 0

Number of pixels to letterbox.

Click on 'Show Output Pad' to easily select cropping and letterboxing settings.

☑ Keep aspect ratio

Correct aspect ratio if necessary.

Click here to open output pad.

See on the right side of this picture where it says things like:

top offset

height

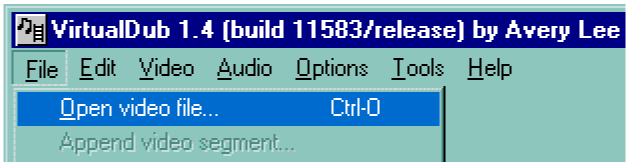
left offset

width

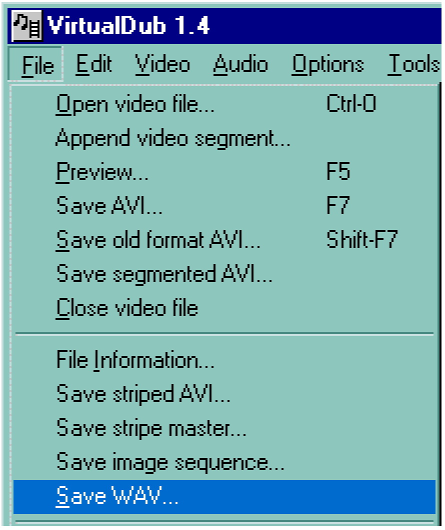
Copy every setting down that is shown on the right. Then when you are ready to make your Low Motion movie you should be able to type the same settings in these boxes (don't use the output pad for the second movie) and everything should be the same!

Taking out the Audio

If you used Flask Mpeg to make your movie you should extract the audio before you continue. Neither movie should contain audio when we put it through MakeFilm. First save the audio somewhere where you will not loose it :) Do this by opening VirtualDub i.e.:


VirtualDub 1.4 (build 11583/release) by Avery Lee
File Edit Video Audio Options Tools Help
Open video file... Ctrl-O
Append video segment...

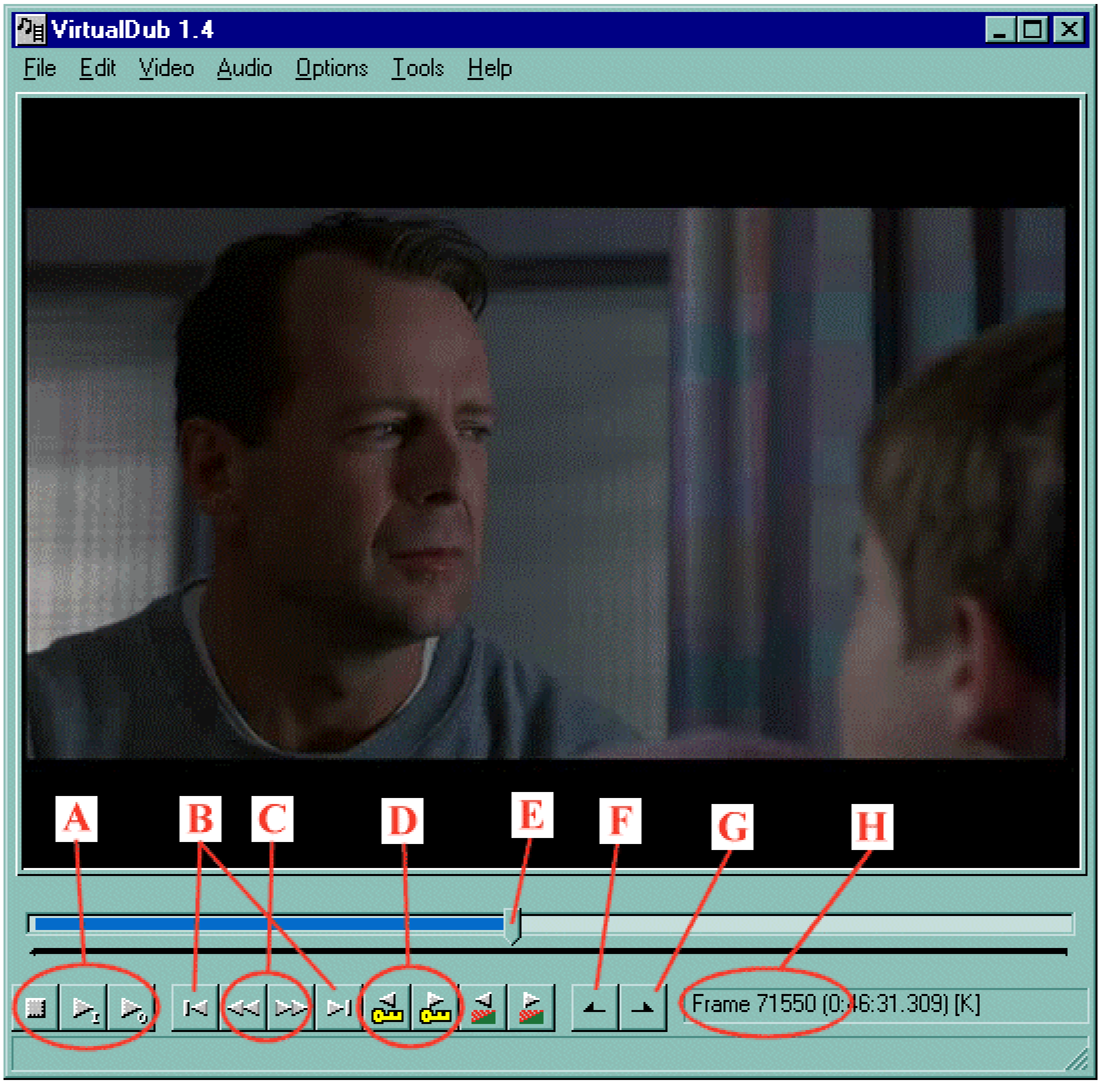
Choose File > Save Wav...



And don't worry it shouldn't be larger than about 100MB; in fact, its usually only about 60MB. VirtualDub doesn't change the audio to uncompressed PCM audio, it literally takes out whatever is in there. So if you used Mp3 in Flask that is exactly what it will extract...although it will still call it a something.wav.

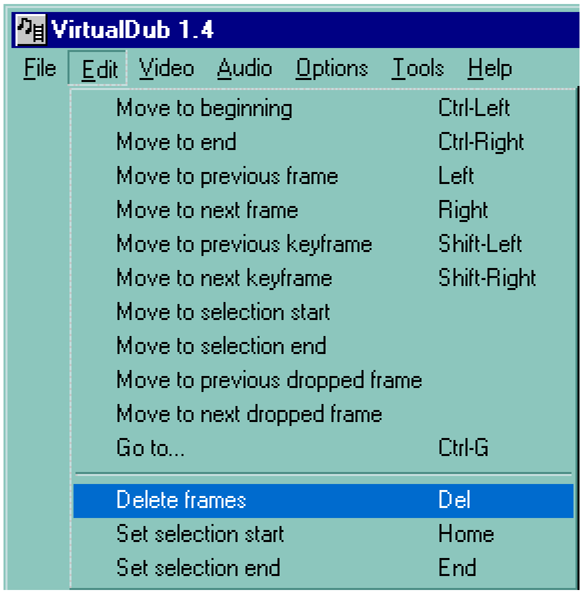
Step 2: Re-encoding the End Credits (staff roll).

This is an optional feature, but since it almost always saves anything upto and above 50MB it will help the quality of the final movie even more. Besides, there is no reason not to do this if we are gonna mess around cutting do all this stuff anyway! Okay, I hope you've read my cutting and joining Divx guide because it will help you to understand what I am going to say. Open the Fast Motion movie in VirtualDub.

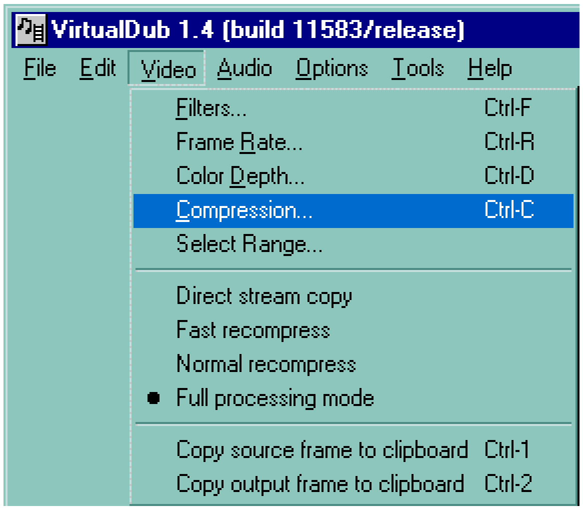

VirtualDub 1.4
File Edit Video Audio Options Tools Help
A
B
C
D
E
F
G
H
Frame 71550 (0:46:31.309) [K]

Move the slider bar (**E**) to get close to where the end credits start. Then use the keyframe buttons (**D**) to get to almost exactly where they start :). Don't worry if you are a little into them this way it won't matter. Now, write down the number shown at (**H**). Then, without moving the silder bar position, press the mark out button (**G**) once! And press the move to start of movie button (**B** the one on the left). Then press the "mark in" button (**F**). This will select the parts of the movie that we don't need to re-encode.

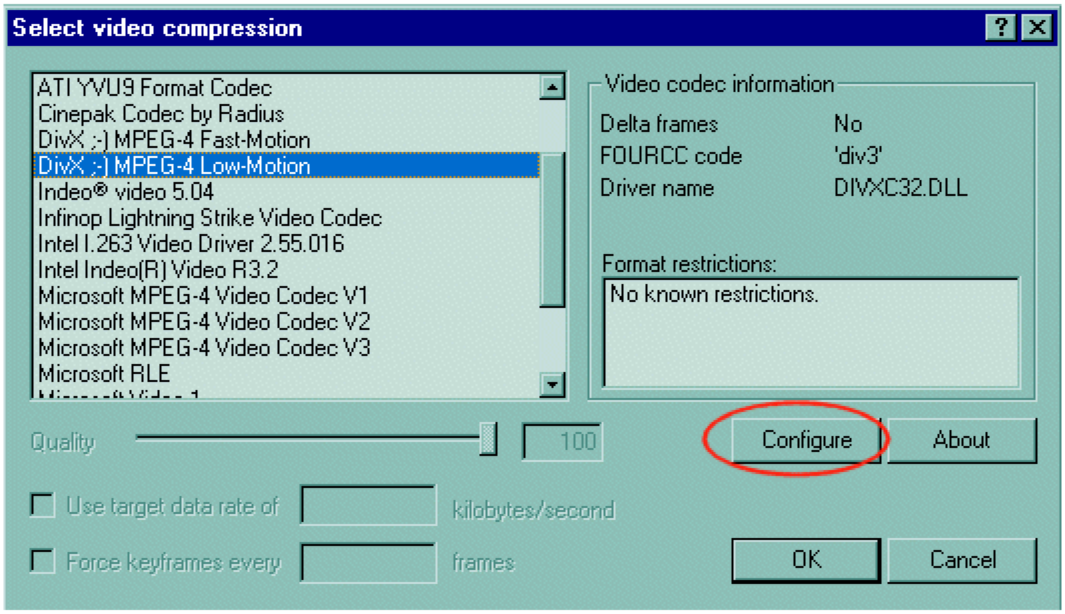
Finally, go to Edit > Delete Frames


VirtualDub 1.4
File Edit Video Audio Options Tools Help
Move to beginning Ctrl-Left
Move to end Ctrl-Right
Move to previous frame Left
Move to next frame Right
Move to previous keyframe Shift-Left
Move to next keyframe Shift-Right
Move to selection start
Move to selection end
Move to previous dropped frame
Move to next dropped frame
Go to... Ctrl-G
Delete frames Del
Set selection start Home
Set selection end End

Select Video > Compression



Choose the Fast Motion codec *(not the one shown in this picture below).*


Select video compression
ATI YVU9 Format Codec
Cinepak Codec by Radius
DivX ;-) MPEG-4 Fast-Motion
DivX ;-) MPEG-4 Low-Motion
Indeo® video 5.04
Infinop Lightning Strike Video Codec
Intel I.263 Video Driver 2.55.016
Intel Indeo(R) Video R3.2
Microsoft MPEG-4 Video Codec V1
Microsoft MPEG-4 Video Codec V2
Microsoft MPEG-4 Video Codec V3
Microsoft RLE
Quality
100
Configure
About
Use target data rate of
kilobytes/second
Force keyframes every
frames
OK
Cancel

Video codec information

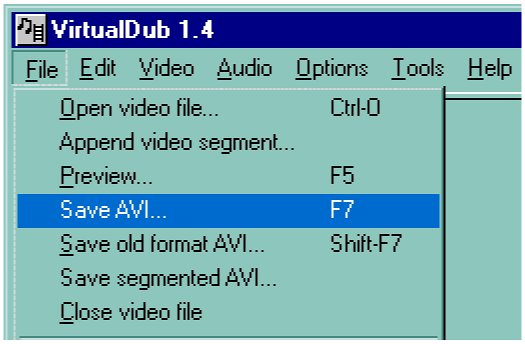
Delta frames	No
FOURCC code	'div3'
Driver name	DIVXC32.DLL

Format restrictions:

No known restrictions.

Hit the configure button and set bitrate to 128 kbps. Keyframes to one every 10 seconds and crispiness to 5%. You can play about with the settings if you prefer better or worse end credit quality. But this should be good enough to turn 60MB credits into a 2MB credits.

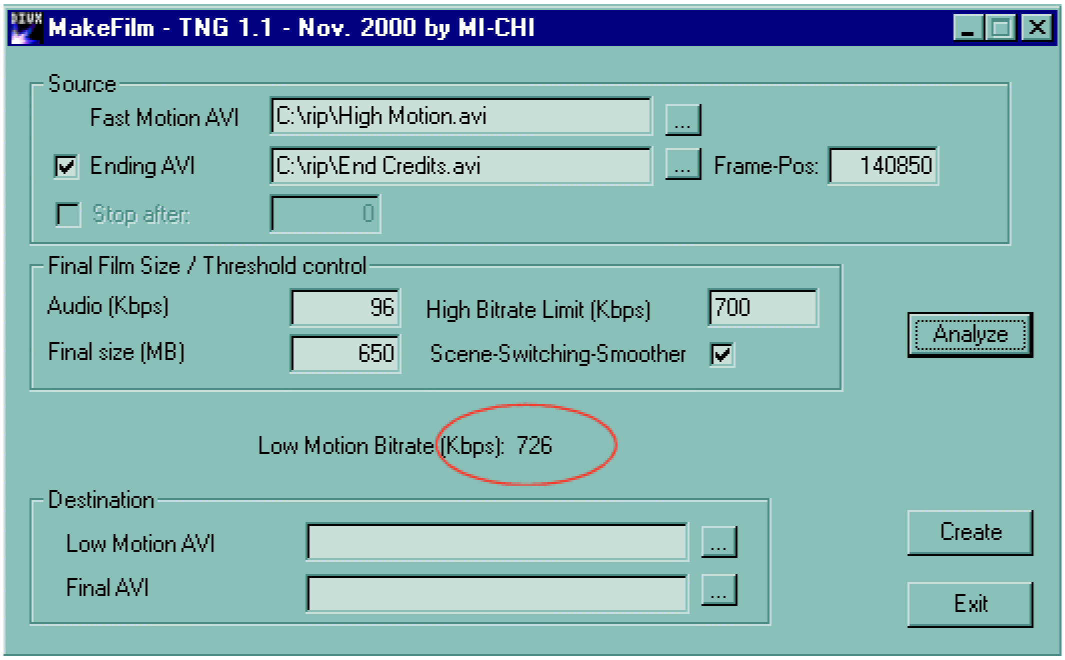
Press OK. Then choose File > Save AVI



Call it something like credits.avi so you know what it is. If you are encoding the End Credits this way you would do well to check after them too. If there are any trailers or unnecessary stuff then delete these off of the end too.

Step 3: The Low Motion Movie

Okay, all the hard work is done. Open MakeFilm and select the High Motion DivX in the first selection box. Then tick the Ending AVI box and select the location of your end credits. Remember the number we wrote down for the start position of the end credits? Put that in the 'Frame-Pos' box.


MakeFilm - TNG 1.1 - Nov. 2000 by MI-CHI

Source

Fast Motion AVI: C:\rip\High Motion.avi

☑ Ending AVI: C:\rip\End Credits.avi — Frame-Pos: 140850

□ Stop after: 0

Final Film Size / Threshold control

Audio (Kbps): 96 — High Bitrate Limit (Kbps): 700

Final size (MB): 650 — Scene-Switching-Smoother ☑

Low Motion Bitrate (Kbps): 726

Destination

Low Motion AVI:

Final AVI:

Bitrate Calculator

This calculator will only work if you encoded your first movie with the Fast Motion codec! In the middle of MakeFilm we have the options 'Audio (kbps)', tell MakeFilm what settings you will or have already encoded your audio. In the above picture I use the bitrate that most Flask user put, namely, 'Mpeg Layer 3 96kbps'.

Just below that selection is the option 'Final size (MB)'. Put in the size of the CD you will be using to store the movie. If you use a 650MB CD type in 650MB or if you use 700MB type that.

Quality Settings

Now, for the joining quality settings. The 'High Bitrate Limit (kbps)' defines at what point the the Low Motion clips will be joined to the Fast Motion movie. For example, at 700 it will replace (with the Low Motion movie clips) all area's of the Fast Motion movie where the bitrate falls below 700kbps. You may wish to experiment with this setting making it a little higher or lower depending on the final result.

Double CD Rips that use two Low Motion movies

As I already explained, this is a little more tricky and is *very* hard to get right. Lets say we have a 150 minuet movie. We could try making the low bitrate movie something like 900 kbps and the high bitrate something like 1200-1500 kbps! Then when we try to join them we will need to put quite a high threshold in the 'High Bitrate Limit (kbps)' section, perhaps something like 950. Then encode it and see what you get. If the movie is too large increase the threshold again to 1000, then 1100, 1200 etc., to see if you can fit it. If you think all this is worth trying, then be my guest. I cannot guarantee the movie will always fit, but in the end this will probably be able to solve any really high bitrate scenes this way.

Scene-Switching-Smoother

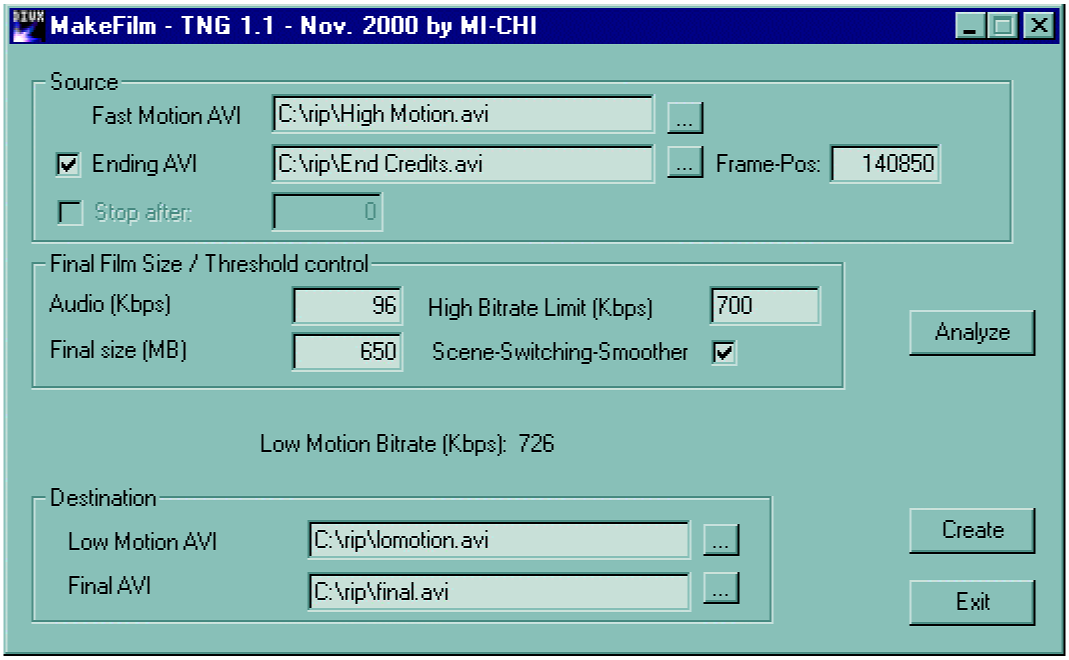
Lastly we have the Scene-Switching-Smoother. This has two options: on or off =). This will depend on your final movie but it smooths scenes where the Fast Motion change is so rapid that you would easily notice the quality change as it turned from Low Motion to Fast Motion. Check your movie and if you see any annoying changes use this option. If you, for whatever reason have decided to use keyframe every 2 or more seconds, then don't use the smoother at all because it will smooth too many scenes and you'll end up with a movie not unlike the one you already started with! For more details and troubleshooting please read through the instructions provided with MakeFilm.

That's it! Press the 'Analyze' button and then MakeFilm will give you the bitrate you should set your second Low Motion movie to (encircled in red above). Again, use the setting keyframe every 1 second and crispiness 75% on your Low Motion movie.

Go! Encode your Low Motion movie with the bitrate given and come back when you have it!

Step 4: The Final Movie

Select your High Motion movie. Select your End Credits clip and put in the Frame-pos number. Select your Low Motion clip. Choose where you want the final movie to be saved. Set your Scene-Switching-Smoother now if you want to use it and set the 'High Bitrate Limit' you prefer.



MakeFilm - TNG 1.1 - Nov. 2000 by MI-CHI

Source

Fast Motion AVI: C:\rip\High Motion.avi

☑ Ending AVI: C:\rip\End Credits.avi Frame-Pos: 140850

□ Stop after: 0

Final Film Size / Threshold control

Audio (Kbps): 96 High Bitrate Limit (Kbps): 700

Final size (MB): 650 Scene-Switching-Smoother ☑

Low Motion Bitrate (Kbps): 726

Destination

Low Motion AVI: C:\rip\lomotion.avi

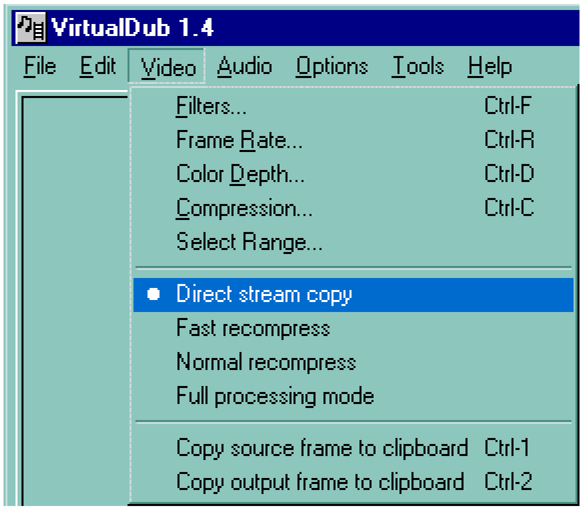
Final AVI: C:\rip\final.avi

There is one last setting I haven't spoken about. The 'Stop after' box (under the Ending AVI selection). If you just want to cut off the end credits instead of replace them with a smaller version, you can put the frame position in this box instead and it will chop the credits off in the final movie. This option is also useful at chopping off trailers and other additional stuff that is sometimes found at the end of DVD movies too.

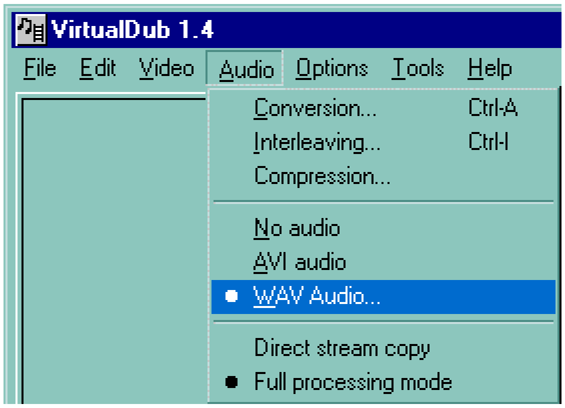
Step 5: Adding the Audio

Time to check your movie. It should be perfect, if not try another threshold or the smoother settings to make a better job of things. Now its time to add the audio back. This is easy enough, open the final movie in VirtualDub.

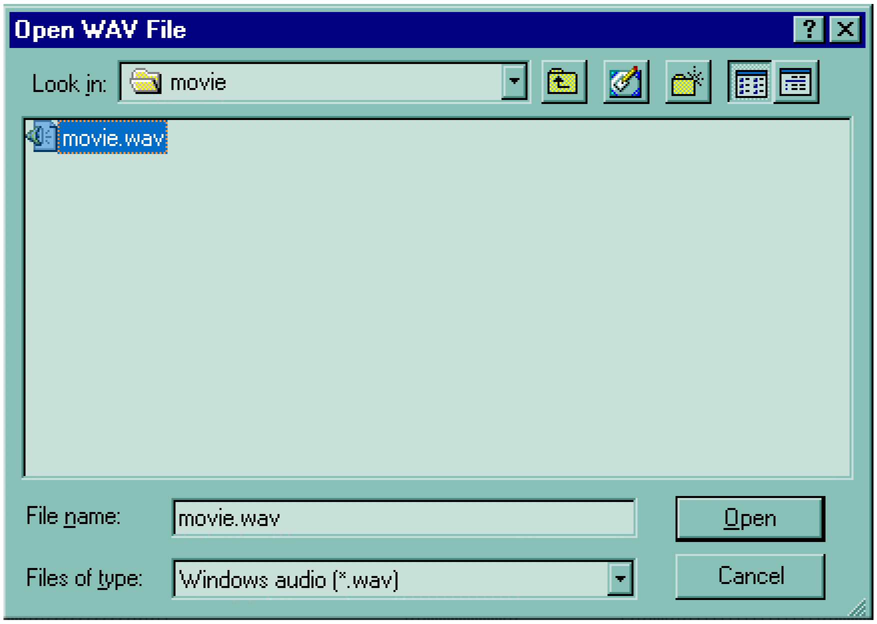
Select: Video > Direct stream copy



Select: Audio > Wav Audio...



Find that wave file.

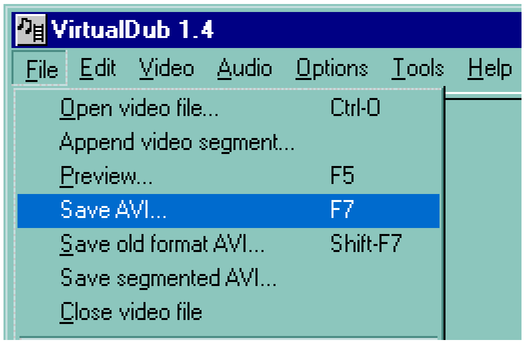


Look in: movie

File name: movie.wav

Files of type: Windows audio (*.wav)

File > Save AVI



Phew! You're done.

TROUBLESHOOTING

Movie freezes half way through but the audio continues, why?!

Check your original movies. If the original movies are the same its just a badly encoded original and nothing to do with MakeFilm. If you are using low bitrates its a good idea to check the MakeFilm's final version. You don't need to actually watch the movie all the way through. Just let it play while you do something else and check it every now and then. You will know if the frozen frame appears because it will stay frozen until it reaches the end. To help prevent framedrops use lower smoothness settings when encoding.

After adding the audio to my final AVI, it is out-of-sync

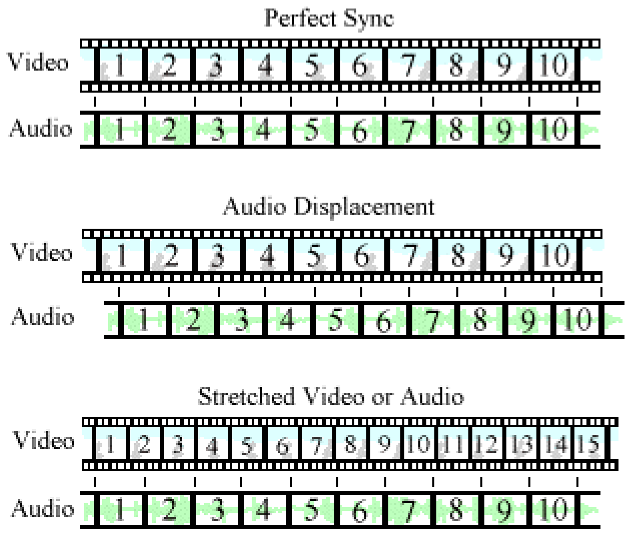
Again try multiplexing it with the original movie and see if its correct. If not, use my audio synching techniques.

More Help

Make sure you go through the manual provided with MakeFilm first. If you have any questions or problems with MakeFilm please do not send them to me because I must then send them to MI-CHI :).

Fixing Audio Synchronization Problems

There are two major problems with audio either it is stretched or displaced. This can be illustrated thusly:



When the synch is perfect both the audio and video will match, as seen in the first image above. Video frame1 matches to Audio frame 1 as does frames 2, 3 and so on. In the second picture it shows the audio as displaced, which means that both audio and video are correct length, but either the video or audio will start at different times causing displacement. In the third sinerio we have an audio and video sequence that are different lengths because one of them has become stretched somehow.

To fix these problems you must check out the film very carefully. If the audio is equally out of synch at the start of the movie as it is the end of the film, then you have an Audio Displacement problem (see Fixing Audio Displacement below). If the audio is very close to correct at the beginning of the movie, but is quite badly wrong at the end of the movie (or vise versa) you probably have a Stretched Video problem (see Fixing Stretched Audio below). It is possible to have both problems of course ;-P. So if you are sure you have both I suggest you fix the Audio Displacement at the very start of the film first and then fix the Stretched Video problem once you know somewhere in the film is absolutely correct!

Before I start here are the things you will need:

VirtualDub 1.4c

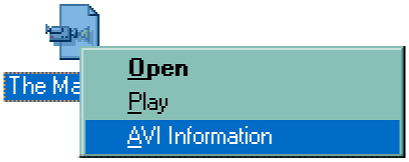
Terabits AVI Info 1.015 also get the latest updates Here

Cool Edit (or simular audio editing tool)

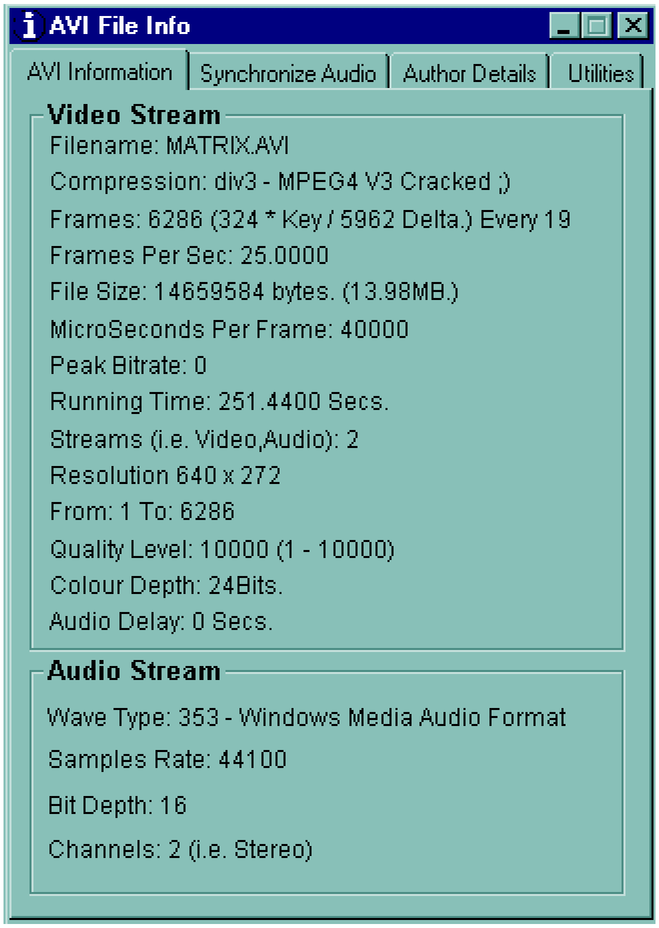
PART 1: Fixing Stretched Audio

Stretched audio is the most common reason a video will be out of synch with a video. Luckly the first release of the Terabit AVI Info tool is now ready :). Designed by yours truly and programmed by my own brother - all credit goes to him, though, for all his hard work. Terabits AVI Info tool is designed to make manual synching much easier and also to make easy some of those little things you'd like to do but do not know how. I will explain these other features at the end of this synching article though, so we can stay on track.

Before we do anything I'd always recommend that you keep a back up of your original AVI file in case something goes wrong. Once installed you do not run AVI Info like most programs, it is more like an extension to Windows. Just find the AVI file you wish to synch and right-click with the mouse. Up will pop an option saying AVI Information - select that.



Then the following box will appear. It is a convenient way of finding out all those little bits of information that neither Windows Media Player nor VirtualDub are tell us about our AVI files.



Video Stream

Filename: MATRIX.AVI

Compression: div3 - MPEG4 V3 Cracked ;)

Frames: 6286 (324 * Key / 5962 Delta.) Every 19

Frames Per Sec: 25.0000

File Size: 14659584 bytes. (13.98MB.)

MicroSeconds Per Frame: 40000

Peak Bitrate: 0

Running Time: 251.4400 Secs.

Streams (i.e. Video,Audio): 2

Resolution 640 x 272

From: 1 To: 6286

Quality Level: 10000 (1 - 10000)

Colour Depth: 24Bits.

Audio Delay: 0 Secs.

Audio Stream

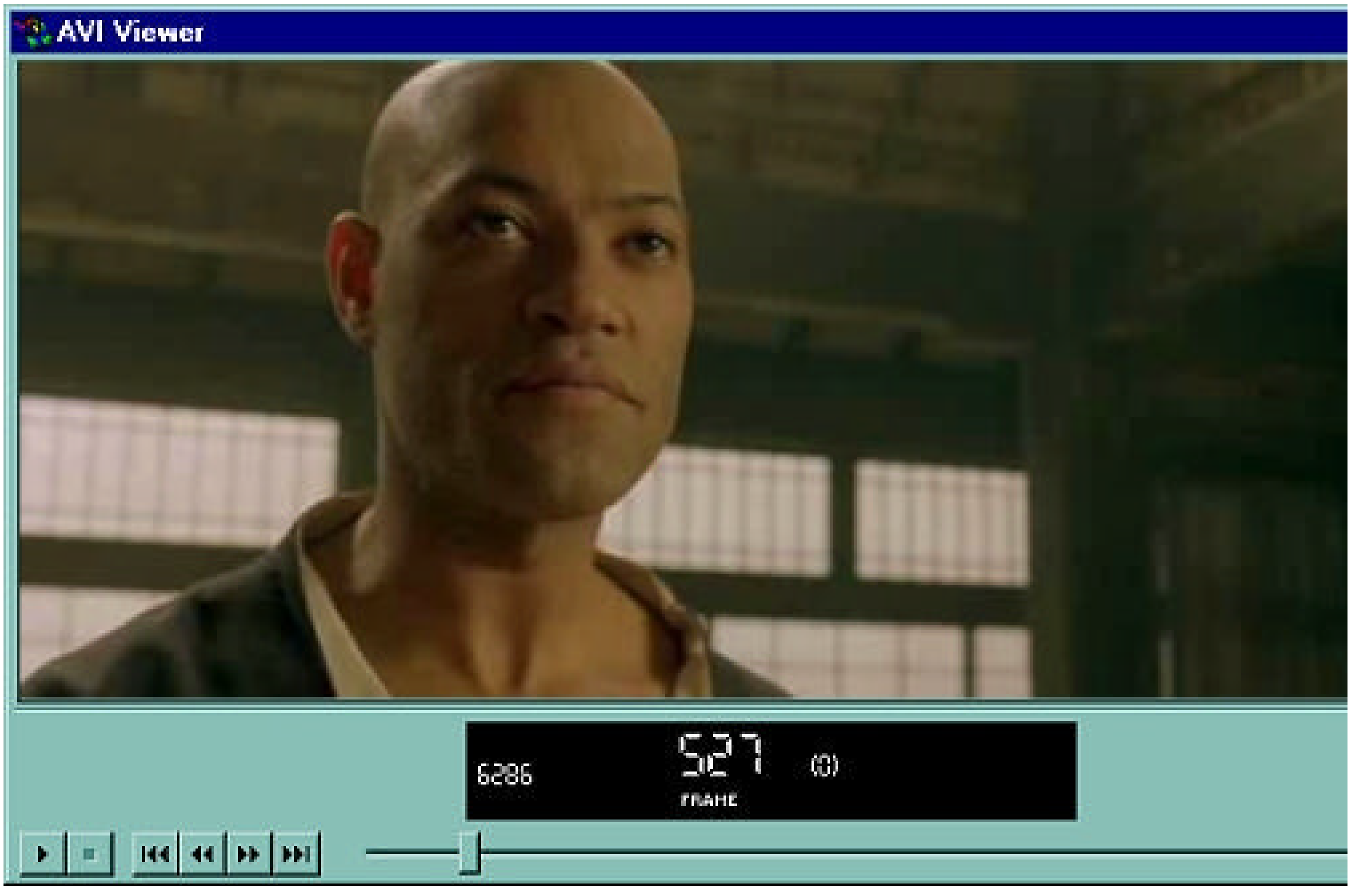
Wave Type: 353 - Windows Media Audio Format

Samples Rate: 44100

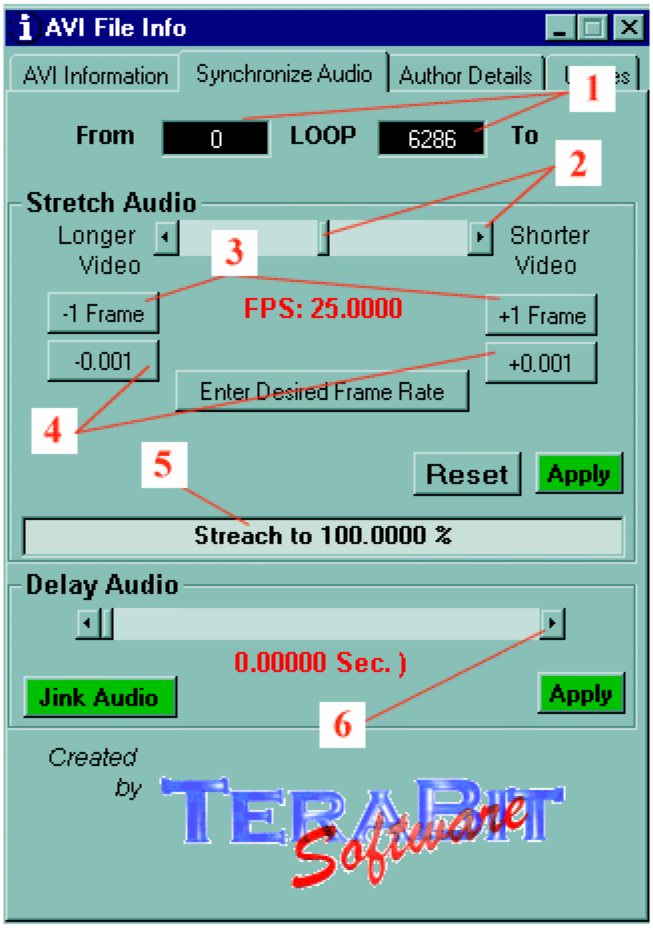
Bit Depth: 16

Channels: 2 (i.e. Stereo)

Click on the Synchronize Audio tab and we get a preview of the video we want to correct, and the important part, a control pannel designed to correct the synching.



Examine your movie carefully using the AVI Player. Preferably very close to the end of the movie to better see how far the audio is out. And also be sure to find a section where their mouths are large enough to see clearly. Locate the frame number of a part just before they are going to speak and put it in the 'From' box below (**1**). Then take note of the frame number from after they stop talking and enter that into the 'To' box (**1**). Great, now the movie will loop between these two ponts. This make life much easier because feedback is instant and we do not need to keep opening the file, finding the right place, guessing how much it is out, changing it, and then starting all over again like we need to do with all other synching methods! Oh, by the way you have to press the apply button or it wont start looping :).



From 0 LOOP 6286 To

FPS: 25.0000

Streach to 100.0000 %

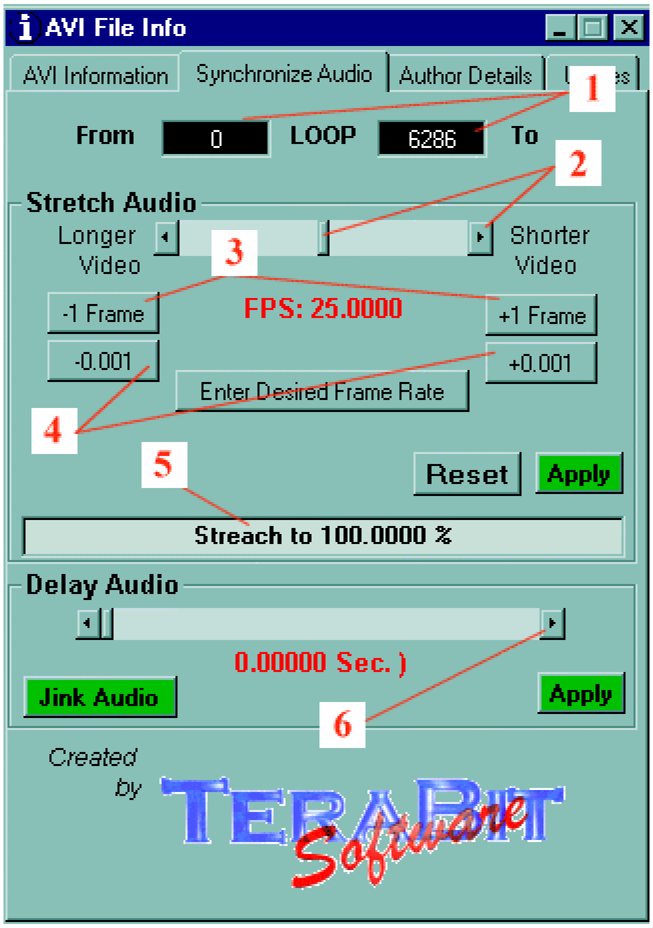
0.00000 Sec.)

Take note of if the audio comes *before* their mouths move or if it comes *after* their mouths move.

1. If the audio comes *after* they speak we must ***decrease*** the frame rate a little!
2. If the audio comes *before* they speak we must ***increase*** the frame rate a little!

It doesn't sound logical at first, so just to clarify: to make the video *longer* we decrease the framerate. To make the video *shorter* we increase the framerate. Imagine it like this: if you had a pack of cards and threw 5 cards onto the floor every second, it would take longer for you to finish the pack of cards than it would if you were to throw 10 cards onto the floor every second. In the same way, a movie will finish earlier if it shows more frames per second. This means a higher framerate makes a shorter movie.

Now the synching is quite straight forward. Play your movie, if the audio comes *after* the mouth moves hit the -0.001 button (**4**) once, and try the movie again. If it is not correct, press the -0.001 button again, and again, and again etc., until it is *almost* perfect. Then use the fine tuning arrows (**2**) to get the final perfect audio. Conversly, if the audio comes *before* the mouth moves we must use the +0.001 button in the same way as already described.



From 0 LOOP 6286 To

FPS: 25.0000

Streach to 100.0000 %

0.00000 Sec.)

If you want to make really *huge* changes you can use the slider bar (**2**) or the increase by a whole frame per second with the +1 and -1 Frame buttons (**3**). There is also the option to enter any framerate you like with the 'Enter Desired Frame Rate' option.

But usually we will only change the framerate by a***very*** small amount! I usually only use 0.001. So if my movie was 25.000 fps and I wanted to make it shorter I would change the framerate to 25.001. If that was too long I'd go to 25.002 then 25.003 and so on. If you want to make the movie longer I would try 24.998 then 24.997 and so on. Once it is almost perfect you can use even smaller amounts such as 0.0001. So imagine 25.007 is the almost perfect, but just needed slightly more. I would try 25.0071 or something like that. It shouldn't take long for you to become an expert.

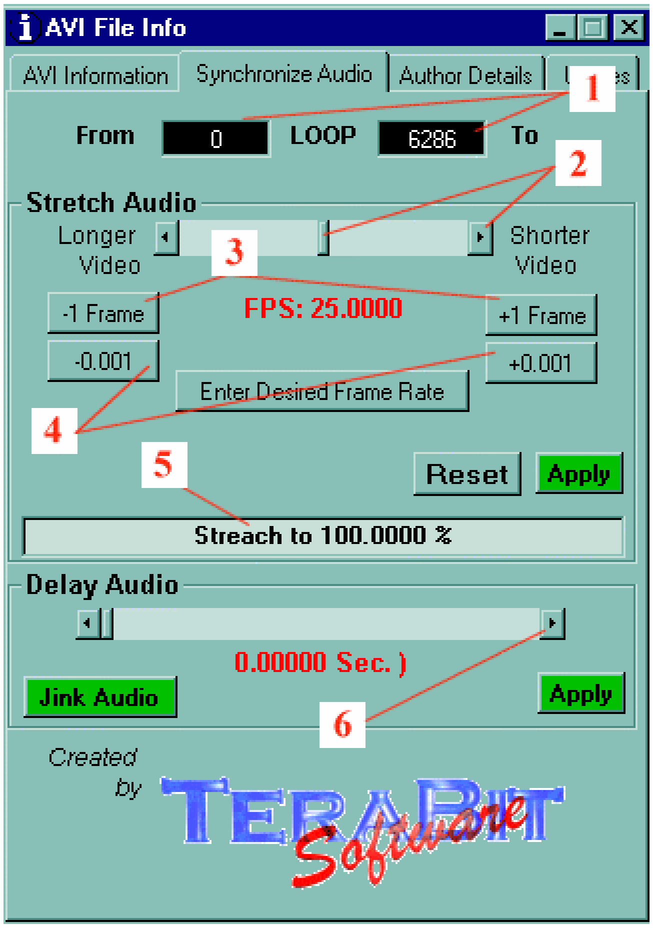
If you think you are totally out you can reset any section back to how it originally was by using the 'reset' button.

To apply it or not to apply it, that is the question?

Whether it be nobler to suffer the slings and arrows of outrageous framerates or not. If we apply it now, the movie will play correctly in Media Player and, in fact, on all PC players. But what if we were synching our AVI file not because it was a DivX movie but because we wanted to convert the AVI to Mpeg or SVCD etc? It is unlikely that the VCD we created would be in synch! This is because all VCD, SVCD and DVD files must be exactly 25 or 29.970 fps to play in a DVD player or a stand alone player! A framerate such as 25.998 will not work!

Other synching tools offer to change the audio samplerate instead. Changing the audio samplerate may be able to put the movie in synch just as effectively as changing the framerate. But again this causes conversion problems. A VCD, DVD or SVCD will only take 44100Hz or 48000Hz respectively. Feeding an encoder 11025Hz will almost always end up with synch problems!

For this problem Terabits AVI Info tool is the only utility so far that can offer a solution. Notice that, as you change the framerate to get the movie in synch, it offers a streach (stretch) amount (5).



From 0 LOOP 6286 To

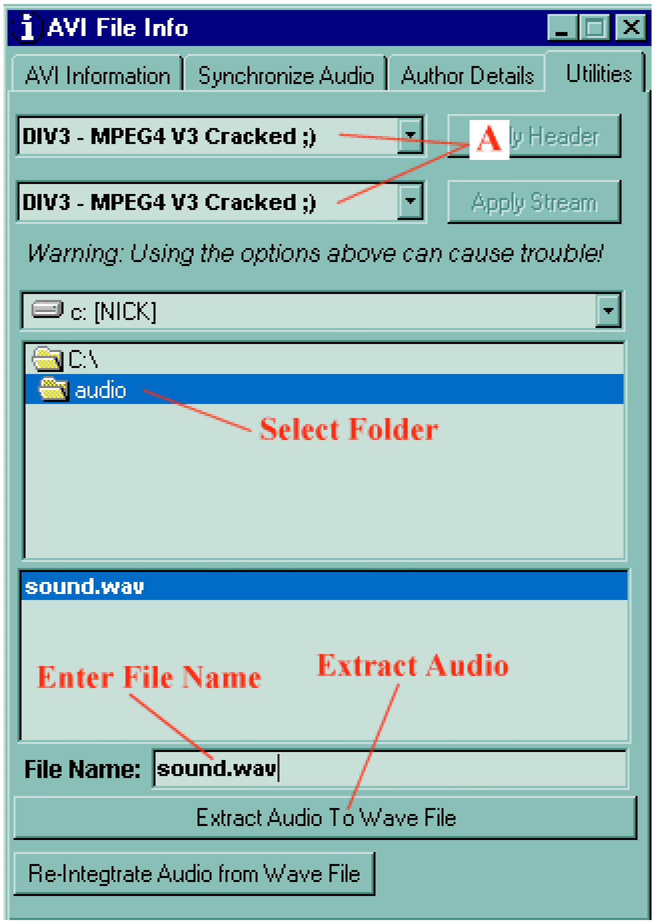
FPS: 25.0000

Streach to 100.0000 %

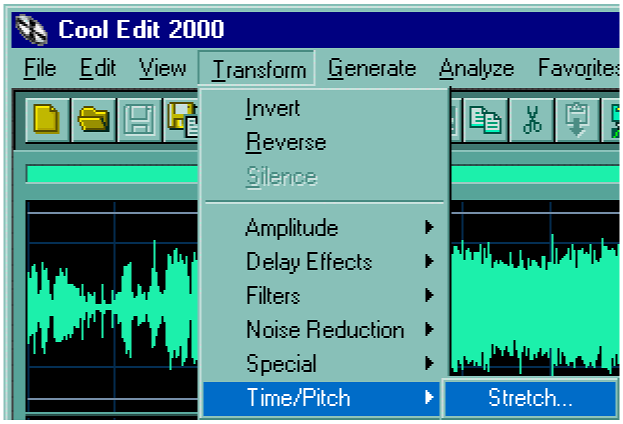
0.00000 Sec.)

So we use the framerate controls as usual to get the movie in synch but this time we***don't*** save the framerate changes! Instead write down (or copy and paste) the 'streach' number exactly. Now we will use this number to stretch the audio in Cool Edit. But before we do that we must extract the audio.

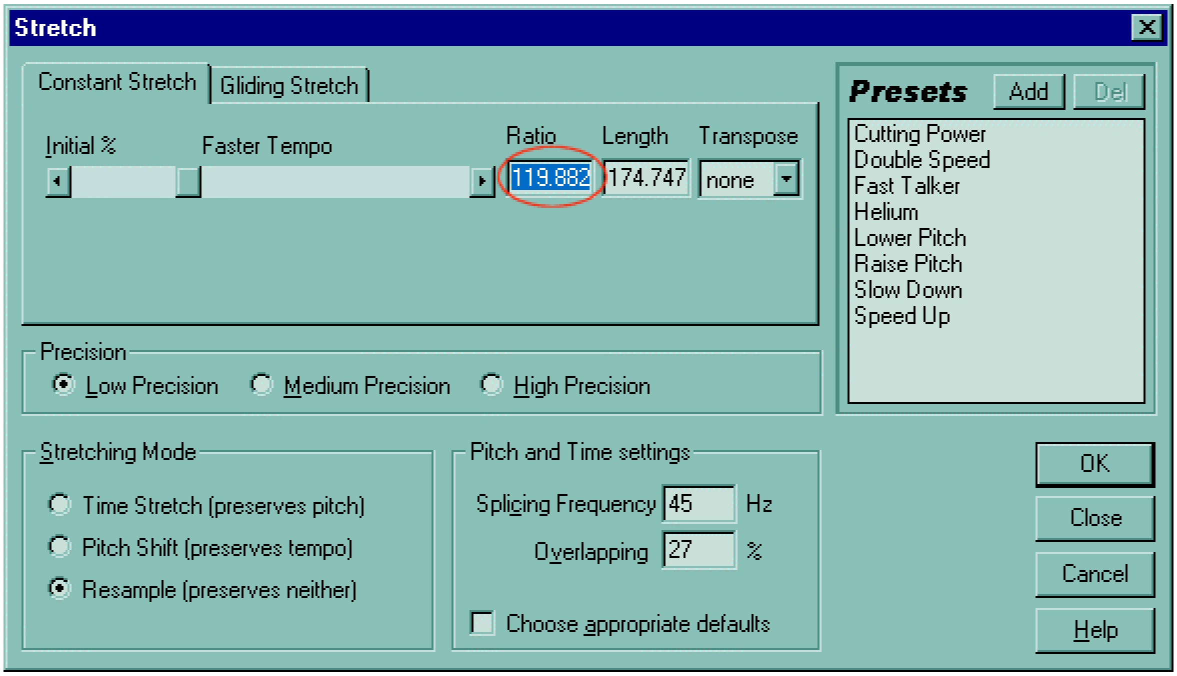
Change to the 'Utilities tab'. Select a folder to save the wave, enter a name for your extracted wave file, and hit the 'Extract Audio To Wave File' button. Alternatively you can use VirtualDub to extract the Wave, but I see no need because this is basically the same thing.



Open the extracted wave in Cool Edit and choose Time/Pitch > Stretch...



Paste the stretch amount into the ratio box (encircled in red). I suggest you use 'Low Precision' option unless you wish to spend many more hours stretching the audio.



Ratio: 119.882
Length: 174.747
Transpose: none

Precision
● Low Precision ○ Medium Precision ○ High Precision

Stretching Mode
○ Time Stretch (preserves pitch)
○ Pitch Shift (preserves tempo)
● Resample (preserves neither)

Pitch and Time settings
Splicing Frequency 45 Hz
Overlapping 27 %
□ Choose appropriate defaults

Note: *The 'High Precision' uses interpolation to keep the audio higher quality, but after we have recompressed it, it will not make any audiable difference so choosing higher quality will probably just waste time.*

Cool Edit 2000 automatically converts the audio to PCM Wave for editing. So be careful because this takes over a gigabyte of space! If your are using another audio editing program then you may have to convert the wave to PCM first. Using Winamp is probably the easiest way so check out my *'Converting Audio*' section for more info on this subject.

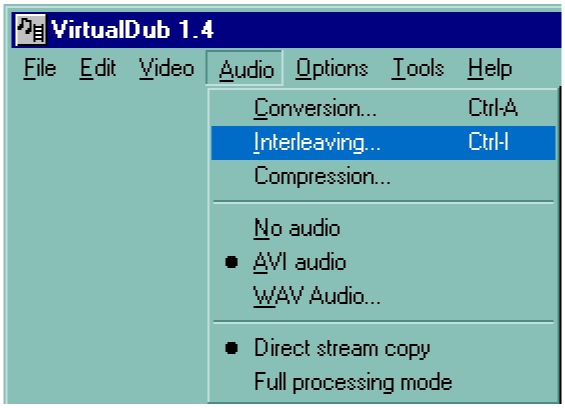
Reintergrating the Audio

Probably the most reliable method now is to save our newly stretched audio as PCM Wave and multiplex it back to our video file. Go to my Multiplexing section for details on this.

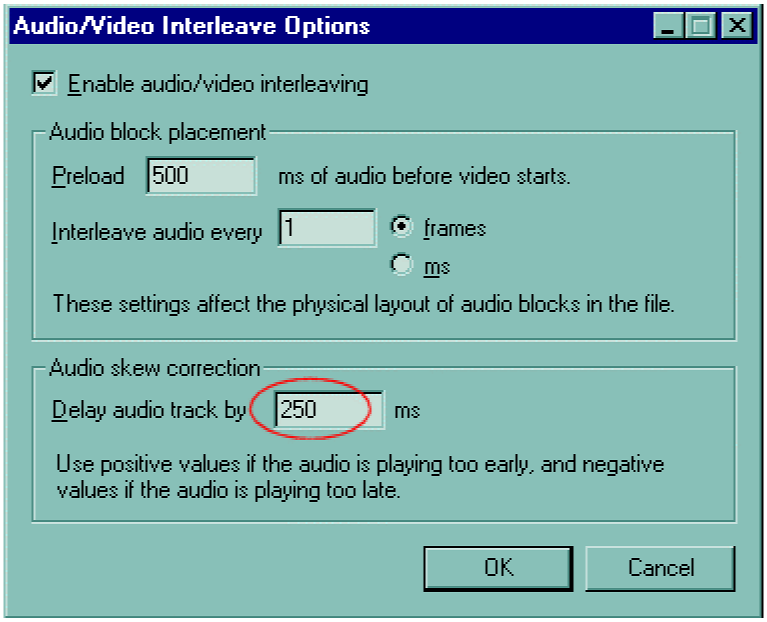
PART 2: Fixing Audio Displacement

Open your video in (you guessed it) VirtualDub. Choose a piece of the movie where people are talking. Their mouths should be clear so you can see easily when the synch is correct or not.

Go to Audio > Interleaving



Up pops the box below. Leave everything as it is except the 'Delay audio track by' part. Look at the movie and see if the sound comes before or after the video. If the sound becomes *before* they speak we will use normal numbers to slide the audio forward. If the audio comes *after* they speak we will use minus numbers (-) to slide the audio backwards.



Audio/Video Interleave Options

☑ Enable audio/video interleaving

Audio block placement

Preload 500 ms of audio before video starts.

Interleave audio every 1 ● frames ○ ms

These settings affect the physical layout of audio blocks in the file.

Audio skew correction

Delay audio track by 250 ms

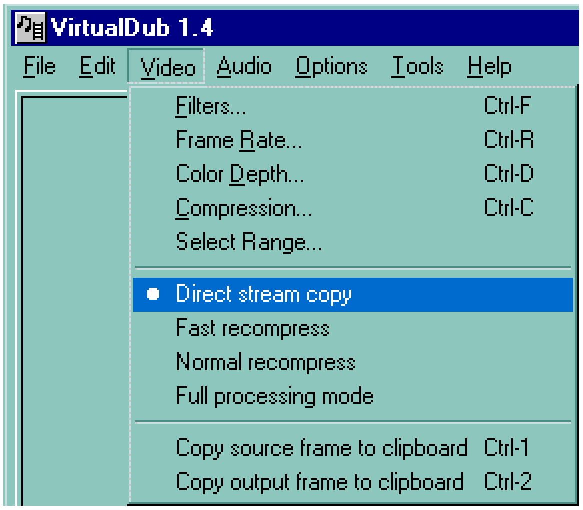
Use positive values if the audio is playing too early, and negative values if the audio is playing too late.

1000 represents one second of time, 500 represents half a second, 250 represents a quarter of a second etc. Lets say the sound appears just after the person speaks. To solve this we must slide the audio to the left a bit. So we can put in minus 250 (ie. -250). Press OK and play the movie in VirtualDub. Is it correct? If not, does the sound come before the person speaks or after? Lets say it is much closer but still comes after the persons mouth moves, we could try increasing the delay by putting in -300. Press OK and check the movie again. Keep doing this until the audio slides into the perfect place!

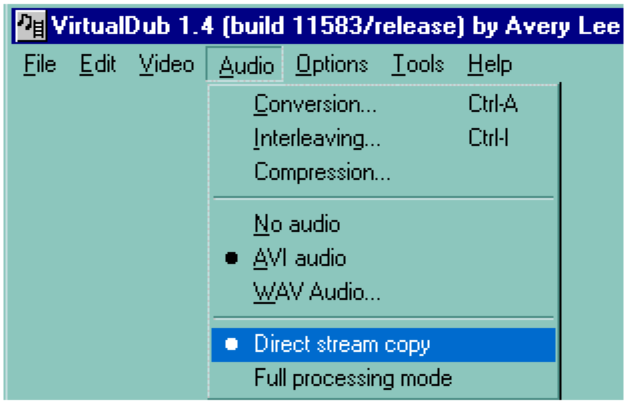
Now lets assume the persons mouth moves and then the audio comes after they speak. We do exactly the same thing except we no longer use the minus sign (-). So we start off small with 250 and press OK to see the results. Then, if its not right, try some more like 300 then 400 or 500 etc. Usually a movie will only be out by about half a second (i.e 500), but I have sometimes found movies to go out by six seconds (i.e. 6000) or more! But usually this is becase I made a mistake in the encoding by setting the wrong samplerate or something, so its easier to grab the audio again.

When the audio is correct save the movie by choosing:

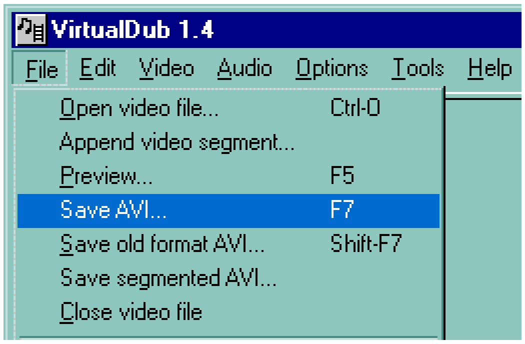
Video > Direct stream copy



Audio > Direct stream copy



File > Save Avi



Fixing Audio Displacement and Stretching

If you are really unlucky you will need to fix both problems! If this happens fix the audio displacement *first* and then the audio stretch. Just repeat steps 1 to 3 and you should be fine:

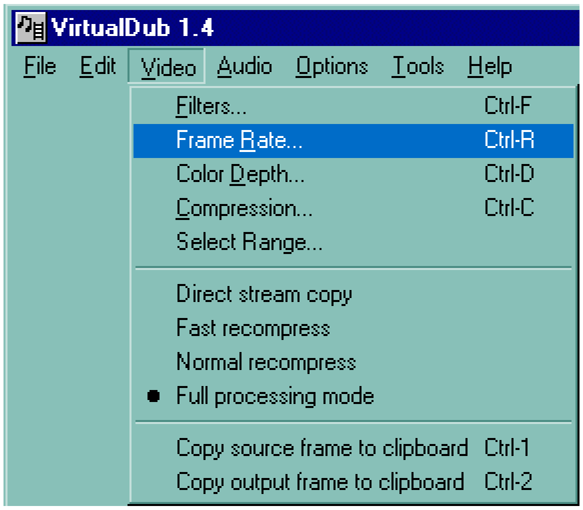
1. Go to the very very start of the movie and find the first part where they speak. Use the displacement methods already described to get the audio starting in the correct place.

2. Then go to the very very end of the movie and use the audio stretch fix method already described to put it back in place.

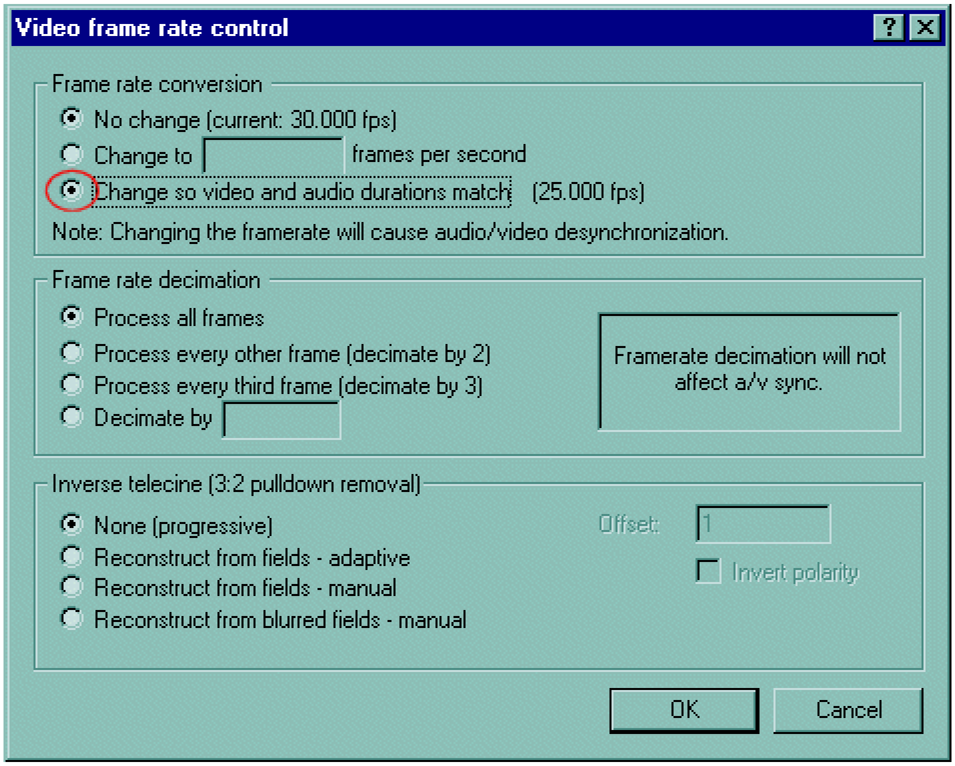
3. Check the movie. It will probably still *not* be correct becase the audio stretch moved the start displacement to the wrong place again! So repeat steps 1 and 2 again :).

PART 3: The Quick Fix Audio Synch Method

You could also try the "quick fix method" for audio synch by opening your movie in VirtualDub and choosing Video > Framerate



Then choose the 'change so video and audio durations match'



Video frame rate control

Frame rate conversion

● No change (current: 30.000 fps)

○ Change to ______ frames per second

● Change so video and audio durations match (25.000 fps)

Note: Changing the framerate will cause audio/video desynchronization.

Frame rate decimation

● Process all frames

○ Process every other frame (decimate by 2)

○ Process every third frame (decimate by 3)

○ Decimate by ______

Framerate decimation will not affect a/v sync.

Inverse telecine (3:2 pulldown removal)

● None (progressive)

○ Reconstruct from fields - adaptive

○ Reconstruct from fields - manual

○ Reconstruct from blurred fields - manual

Offset: 1

□ Invert polarity

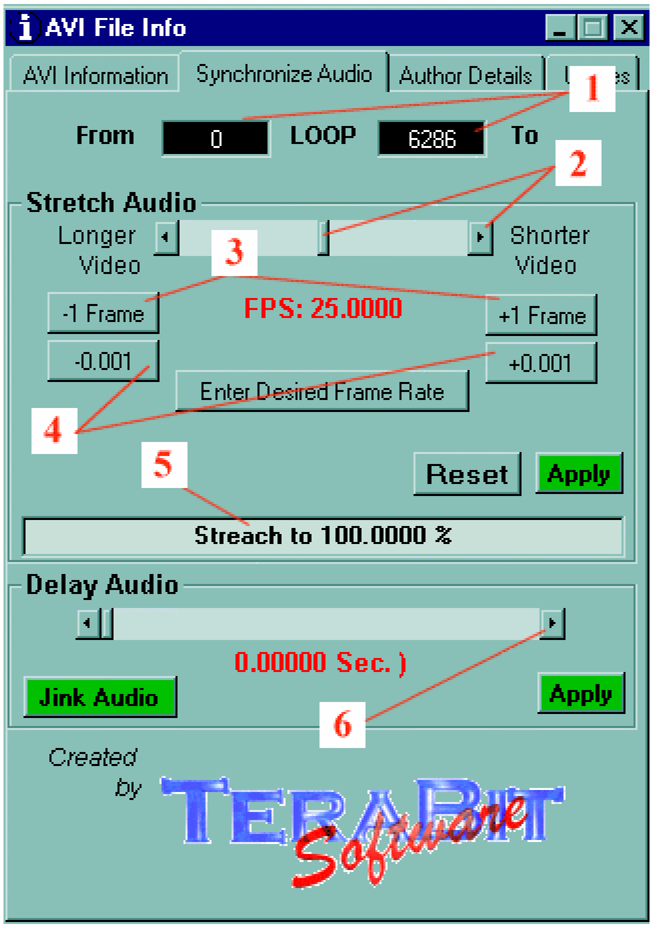
This method sounds amazing but doesn't usually work =(.

More about using the Terabit AVI Info Tool

Finally this looks like a good place to add a few more details on the AVI Info tool we are designing. It isn't completely finished (as you probably guessed) and there are many thing on the cards for this tool. You already know how to extract a wave from an AVI with it and how to use its synching functions.

Delay Audio

We have put in a delay audio function (**6**) but it is only partially working so use it at your own risk. It can only delay the audio's start time and will not slide it to the left or right like VirtualDub. Therefore I suggest you use VirtualDub for audio displacement problems.



AVI File Info

From 0 LOOP 6286 To

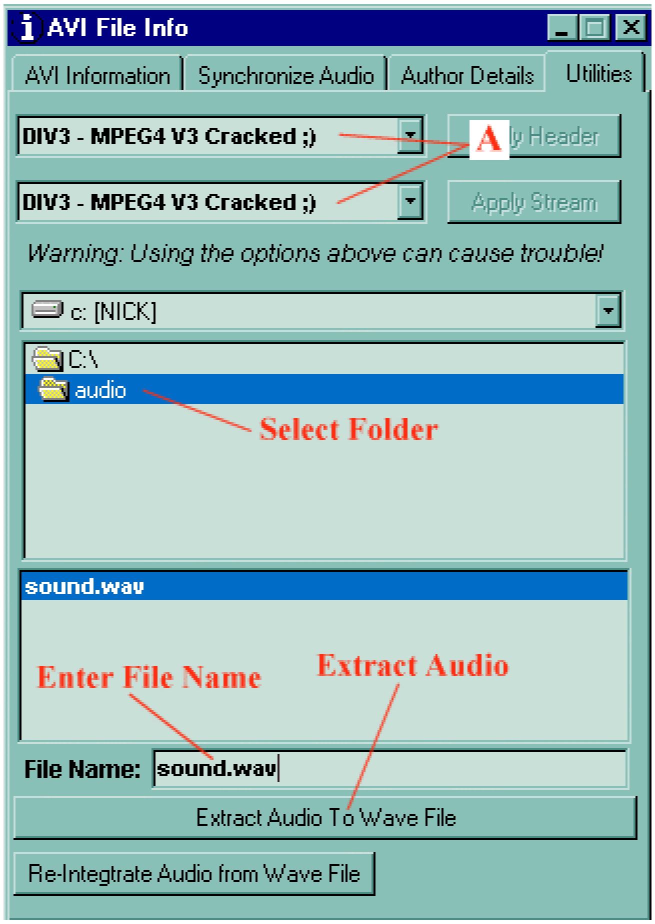
FPS: 25.0000

Streach to 100.0000 %

0.00000 Sec.)

<u>Change FourCC codes</u>

If we go back to the 'Utilities tab' we are given the option to change the Header code of the AVI file (**A**). As you probably realise Windows knows what codec to decode the AVI with by its FourCC codes.



If, for example we wanted to join a High Motion DivX movie clip with a Low Motion DivX movie clip, it would be possible to do this (provided they were the same resolution) by changing one of them to the FourCC code of the other! So a Low Motion video which has:

DIV3

DIV3

Can be changed into a Fast Motion video clip by changing it to:

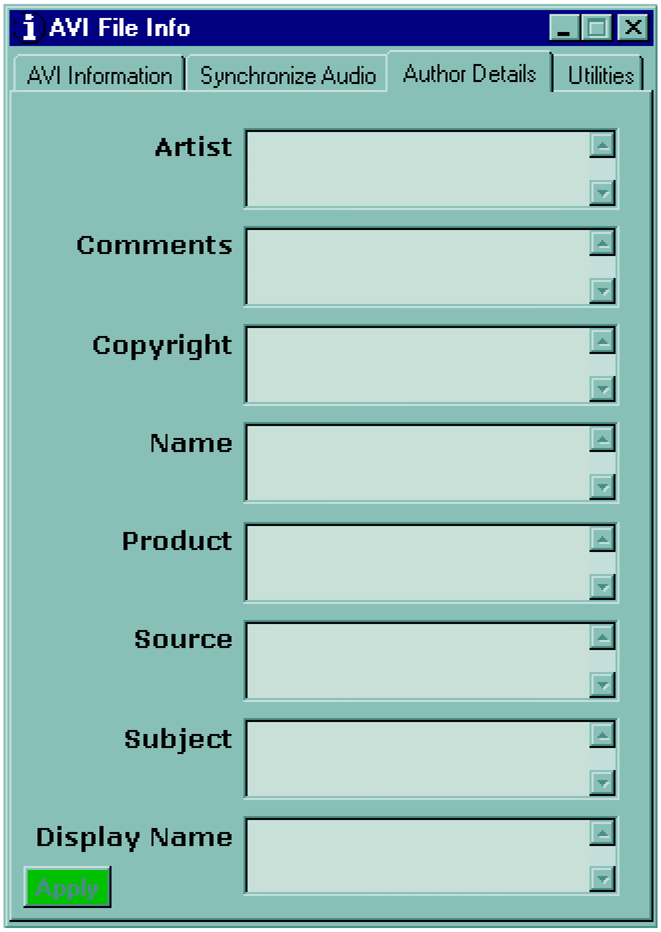
DIV4

DIV3

Once this change has been done you should be able to join them together in VirtualDub. Check out my *'Cutting & Joining'* AVI files section for details on how to do this.

Author Details

Since our DivX method doesn't allow us to add author details to our movies like ASF files we wanted to make a tool for doing this. This tool doesn't work yet BUT you should be able to open a previously authored movie and change it. So you could take someone elses latest ASF movies and change their author details to 'made by joe bloggs' or something crazy like that =o).



Again this will only work with a video file that *already* has someone else's information in it.

Fixing Bad Audio

Uncompressed audio quality is usually dependent on the samplerate and bit depth that it was sampled with. Lets me first explain what these are:

Samplerate

When you record sound into the PC it is broken up into a series of slices. These can be likened to the framerate of a video file, each moment in time is frozen and the information about it is recorded. In the same way, each audio slice represents a number that defines how high or low the audio is at that moment in time. These slices are called samples and the number of samples per second is measured in Hertz (Hz).

Bit Depth

A 'Bit' is a standard storage size used by your computer just like kilobytes and megabytes, it is actually the smallest unit of storage. Each bit can define itself as on or off (1 or 0). Eight bits make a byte, 1024 bytes make a kilobyte (1K) and 1024K make a megabyte (1 MB).

Going back to our video analogy, if the Samplerate is how many "frames" the audio has, then the bit depth is how large the frames are, the resolution or the amount of memory allocated to each frame. The large the bit depth the more possible amplitudes each audio 'slice' can have. Eight bit audio uses 256 amplitudes, 16-bit allows 65536 and 24-bit can have 16 Million!

Flask Mpeg and Samplerates

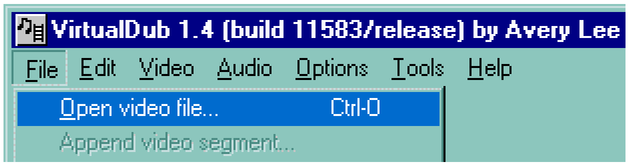
Firstly, it means that the higher the samplerate and the higher the bit depth the higher the quality audio you will get. It also means it will take much more storage space too! 44100Hz 16-Bit represents CD quality audio and 48000Hz is actually even higher than CD quality. This means to get the best quality audio you should also aim for the highest samplerate and bit depth your sound card can handle. But more seriously the analogy to framerate and samplerate is almost perfect because just as a higher framerate will produce a smaller video length a higher samplerate will produce a smaller audio length!!

I know it sounds strange because if you record an hours audio at 44100Hz or at 48000Hz its should still last an hour! And in fact it usually does because it was recorded that way. But when you start converting samplerates that is a completely different matter. Unless the converter takes into account this framerate change there will by audio synch problems.

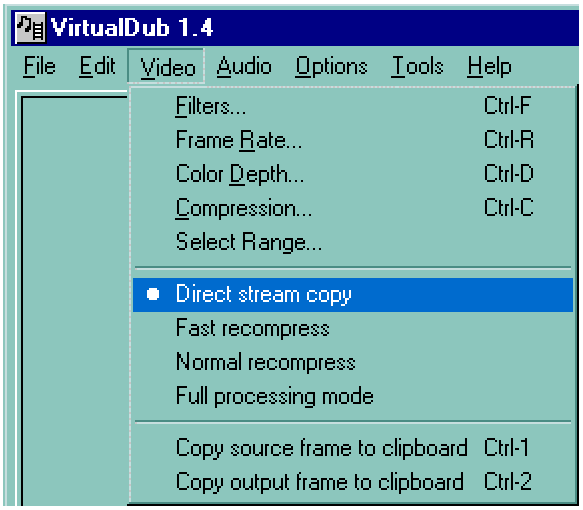
This is a very common reason Flask Mpeg will give audio synch problems. If you convert a DVD 48000Hz audio down into 441000Hz will very often result in audio synch problems. If you find the audio has been recorded at the wrong samplerate with Flask Mpeg it may fix it if you convert it back to the original DVD samplerate i.e. 48000Hz. Theoretically, the same film of 60 mins recorded both separately at 44100Hz and 48000Hz will play for exactly 60 minuets. But when it comes to ‘downsampling’ i.e. converting 48000Hz down to 441000Hz, then problems arise because many programs do not seem to keep the file length the same! Solution? Grab a DVD audio at 48000Hz 16 Bit only! You may then use VirtualDub to convert the audio down to 44100Hz ***after*** because VirtualDub is able to keep the audio length perfect!

Fixing that Dull Sound

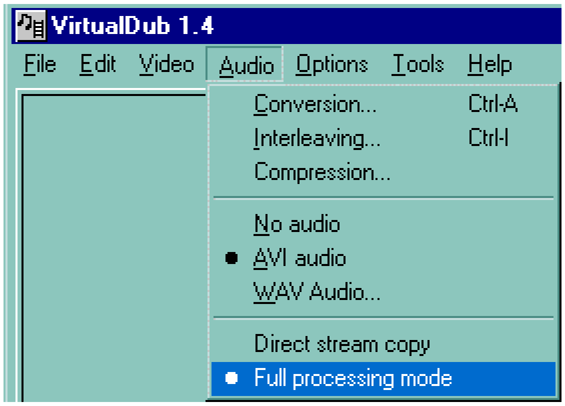
Because many sound cards cannot handle 48000 Hz very well you may find that any method you use to extract the sound will give a fairly dull sound output. This can be prevented by converting the audio to 44100 Hz in VirtualDub. This can be done at the same time that you multiplex the audio and video files together or after when the movie is finished. I'm assuming you are fixing a finished movie here. So open VirtualDub as usual by going to File > Open video file...


VirtualDub 1.4 (build 11583/release) by Avery Lee
File
Edit
Video
Audio
Options
Tools
Help
Open video file... Ctrl-O
Append video segment...

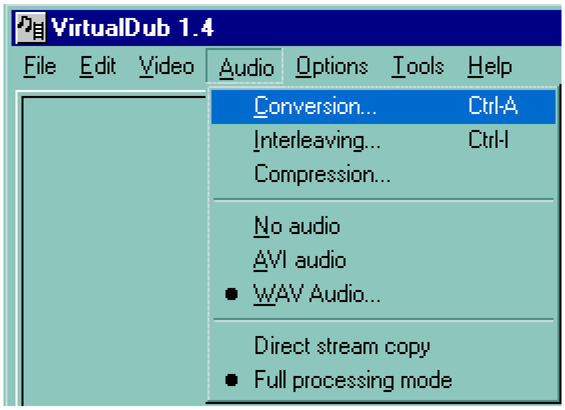
We don't want to effect the video itself so we use Video > Direct stream copy. Whenever we select an option, VirtualDub adds a black dot by it to tell us it is selected.



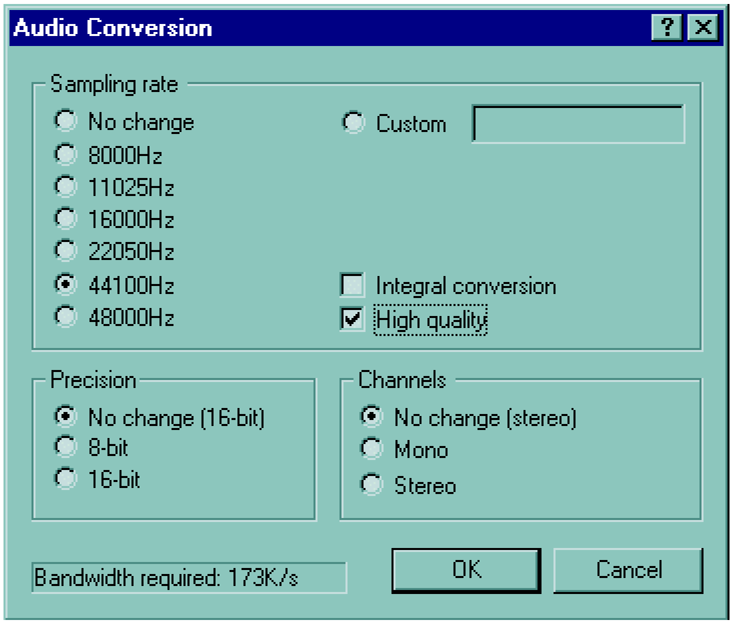
Then select Audio > Full processing mode.



Choose Audio > Conversion...



Choose 44100Hz and make sure the 'High quality' is selected, and press OK.



Audio Conversion

Sampling rate

○ No change
○ 8000Hz
○ 11025Hz
○ 16000Hz
○ 22050Hz
● 44100Hz
○ 48000Hz
○ Custom

☐ Integral conversion
☑ High quality

Precision

● No change (16-bit)
○ 8-bit
○ 16-bit

Channels

● No change (stereo)
○ Mono
○ Stereo

Bandwidth required: 173K/s

Fixing Crackly Sound!

Actually, cracks pop's and hiss are usually a result of badly decoded sound and cannot be fixed other than getting the sound all over again! Ac3Dec is usually the culprit, and since Flask Mpeg uses the same decoding engine as Ac3Dec then it is possible that it will also get some crackly sound.

To be fair I think perhaps Ac3Dec has improved since it was first made, so you may not need to worry. But nevertheless, if you do get bad audio from it I suggest you try another method. The best way I have found is to get the audio with GraphEdit, the quality is much better in my option. Check out my '*Audio with Graphedit*' section for details on this. Then follow the '*multiplexing*' guide to merge it to the video file.

Fixing Very Quiet Sound!

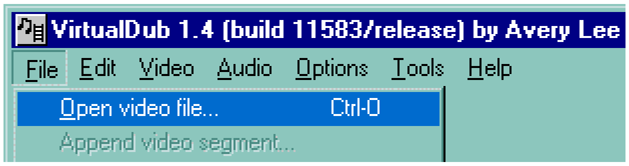
As you would expect, DVD audio has a much higher Dynamic Range than normal TV or Video. This means that it can go from mind-numbingly loud to so quiet that you cannot hear anything! This is fine for our DVD systems but usually a PC will not be able to give enough range. So what you end up with is a soundtrack that is mostly quiet with some very loud bits in it! Software DVD players such as WinDVD have a built in 'Dynamic Normalizer' which usually makes sure the audio is loud enough to hear at all times.

Since you probably used Flask Mpeg or Ac3Dec to create your audio it may be the case that your audio is just too quiet to hear. Again, the best way (in my opinion) to fix this is to get the audio again using GraphEdit and the WinDVD audio filters. Of course, if the audio is badly quiet this *still* may not be enough! The alternative (slower way) is to normalize the sound. Normalizing is a way to make the sound as loud as it will go without clipping. Clipping is when the audio goes off the scale and looses parts of its information. Clipping would cause pops, clicks and other annoying glitches that we obviously don't want. So normalizing is a much better method than amplification.

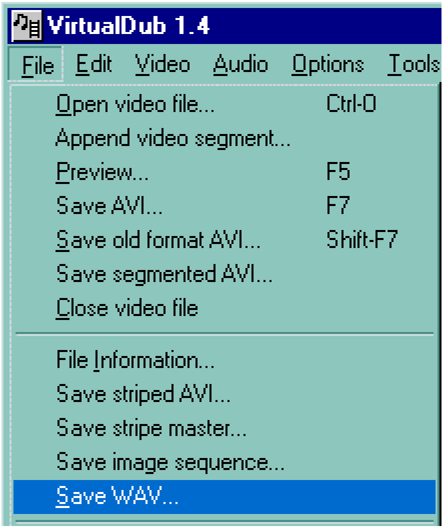
The old method I suggested was to use Cool Edit or SoundForge or a similar audio editing software to normalize because they have this option built in. But there is a more effective way and it doesn't require you buy such expensive software like Cool Edit either =). There is a really good free tool called DeDynamic. It does standard normalization, but it is also able to perform dynamic normalization! What this means is instead of just normalizing the whole file, it normalizes and also adjusts the quiet to loudness ratios. So a loud sound will remain just as loud but a normally quiet sound will be made louder, cool huh! Whatever method you use, the only problem with fixing the audio by normalizing is that it takes anything upto about 4GB of hard drive space to do for a whole movie!

Step 1. Extracting the Audio

I'm assuming you used Flask Mpeg to create your movie so before we go any further we need to extract the audio from your finished movie. Open the movie in VirtualDub:


VirtualDub 1.4 (build 11583/release) by Avery Lee
File Edit Video Audio Options Tools Help
Open video file... Ctrl-O
Append video segment...

Choose File > Save Wav...

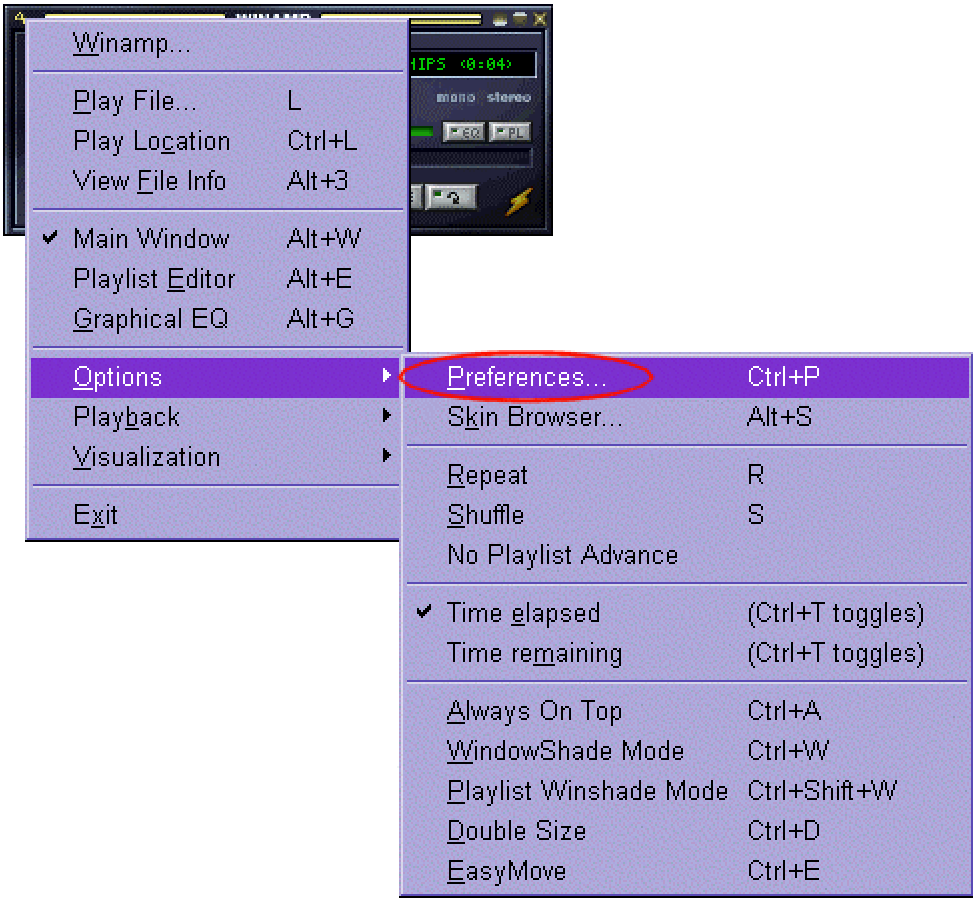


Don't worry about size yet, it shouldn't be larger than about 100MB; in fact, its usually only about 60MB. VirtualDub doesn't change the audio to uncompressed PCM audio, it literally takes out whatever is in there. So if you used Mp3 that is exactly what it will extract...although it will still call it a something.wav and not something.mp3.

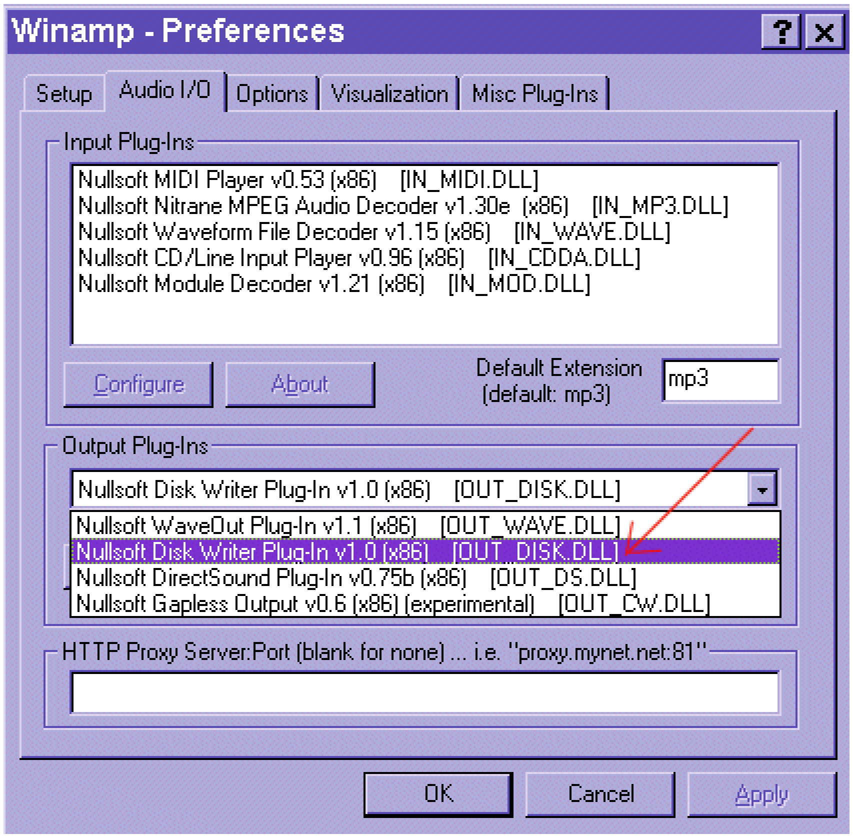
Step 2. Converting it to a PCM Wave

Now we must convert it to uncompressed PCM. For this we can use Winamp because everyone knows how to use Winamp and its free of course ;^). Open your Wav file and press the stop button to stop it playing. Right-click and choose:

Options > Preferences...



Choose the Audio I/O tab. Now instead of the normal OUT_WAVE.DLL option, choose OUT_DISK.DLL. Press 'Apply' and Winamp will ask where you wish to save the file.



Input Plug-Ins

```
Nullsoft MIDI Player v0.53 (x86)   [IN_MIDI.DLL]
Nullsoft Nitrane MPEG Audio Decoder v1.30e  (x86)   [IN_MP3.DLL]
Nullsoft Waveform File Decoder v1.15 (x86)   [IN_WAVE.DLL]
Nullsoft CD/Line Input Player v0.96 (x86)   [IN_CDDA.DLL]
Nullsoft Module Decoder v1.21 (x86)   [IN_MOD.DLL]
```

Default Extension (default: mp3): mp3

Output Plug-Ins

```
Nullsoft Disk Writer Plug-In v1.0 (x86)   [OUT_DISK.DLL]
Nullsoft WaveOut Plug-In v1.1 (x86)   [OUT_WAVE.DLL]
Nullsoft Disk Writer Plug-In v1.0 (x86)   [OUT_DISK.DLL]
Nullsoft DirectSound Plug-In v0.75b (x86)   [OUT_DS.DLL]
Nullsoft Gapless Output v0.6 (x86) (experimental)   [OUT_CW.DLL]
```

HTTP Proxy Server:Port (blank for none) ... i.e. "proxy.mynet.net:81"

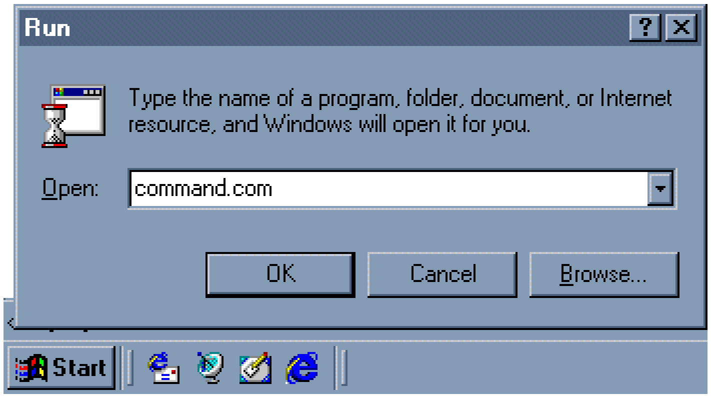
Choose a location and wait until the audio is converted into a ***huge*** one and a half Gigabyte wave file :(.

Step 3. Normalizing

The bad news is DeDynamic is a DOS operated tool. The good news is I'm gonna show ya how to use it :). Make a folder on your main hard drive (usually C:) and call it something simple like dynamic. Cut an paste the gigabyte Wave file you just created with Winamp into it (if you didn't already extract it to there). Also put the DeDynamic.exe file in there as well, so you will have something like this in your dynamic folder:

DeDynamic.exe
audio.wav

Great! Now go to Start Menu > Run and type Command.com to bring the DOS Prompt up i.e.:



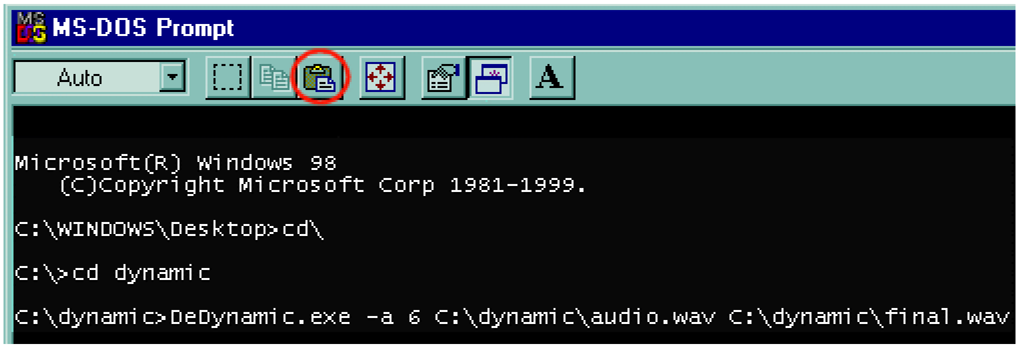
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: command.com

Type: CD\ and press Enter. Type: CD dynamic and press Enter. Now, cut and paste the following line into the DOS prompt:

DeDynamic.exe **-a 6** C:\dynamic\audio.wav C:\dynamic\final.wav

(I have encircled the DOS paste button in red in the picture below).

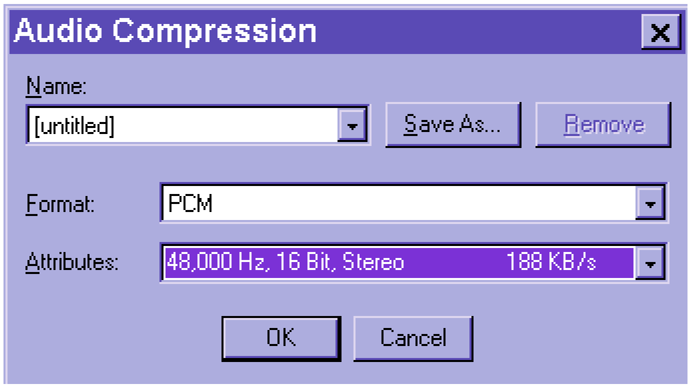


```
Microsoft(R) Windows 98
   (C)Copyright Microsoft Corp 1981-1999.

C:\WINDOWS\Desktop>cd\

C:\>cd dynamic

C:\dynamic>DeDynamic.exe -a 6 C:\dynamic\audio.wav C:\dynamic\final.wav
```

Obviously the text will not be those colours, I'm just using colours so you can see what the line you pasted means. The **-a 6** (highlighted in green text) is the amount of *dynamic* normalization. You can probably quite happily use anything from a setting of 1 to 50 without any problems. If, for example, you wanted to dynamically normalize by 50 you'd put **-a 50** where the **-a 6** was in my above line. If you just want to normalize without the extra dynamics just put **-a 0**.

Using an amount of 50 is severe overkill and it wasn't really designed to do that much. But I've tried it at 50 and above and it sounds quite good, very loud and there is no audio corruption either. But this is a very powerful program, so I would suggest you try a setting of between 4 and 6 and see how that goes. If you still want it louder then you can keep going up until you are happy with it :).

Anyhoo, when you've decided on the setting, paste it, press the 'Enter' key and it will ask what output you'd like it converted into with the following box:



Name: [untitled]

Format: PCM

Attributes: 48,000 Hz, 16 Bit, Stereo 188 KB/s

I would choose the same as the input. So if you had 48000Hz audio originally (all DVDs use this format), then I would choose it again. You can choose another format such as MP3 if you want but this may cause synch troubles and/or difficulty on merging the audio back to the video.

In the end we should end up with another huge gigabyte file, but this time it will be normalized to a really loud kick ass level! That's it! Now go to my '*multiplexing*' guide to see how we use VirtualDub to recombine the new audio to the old video file =o).

How to Convert Mp2 or Most Formats to PCM Wave

Sooner or later you will come to a strange audio format that you cannot convert to wave. But usually if you can play the audio you can convert it with Winamp.

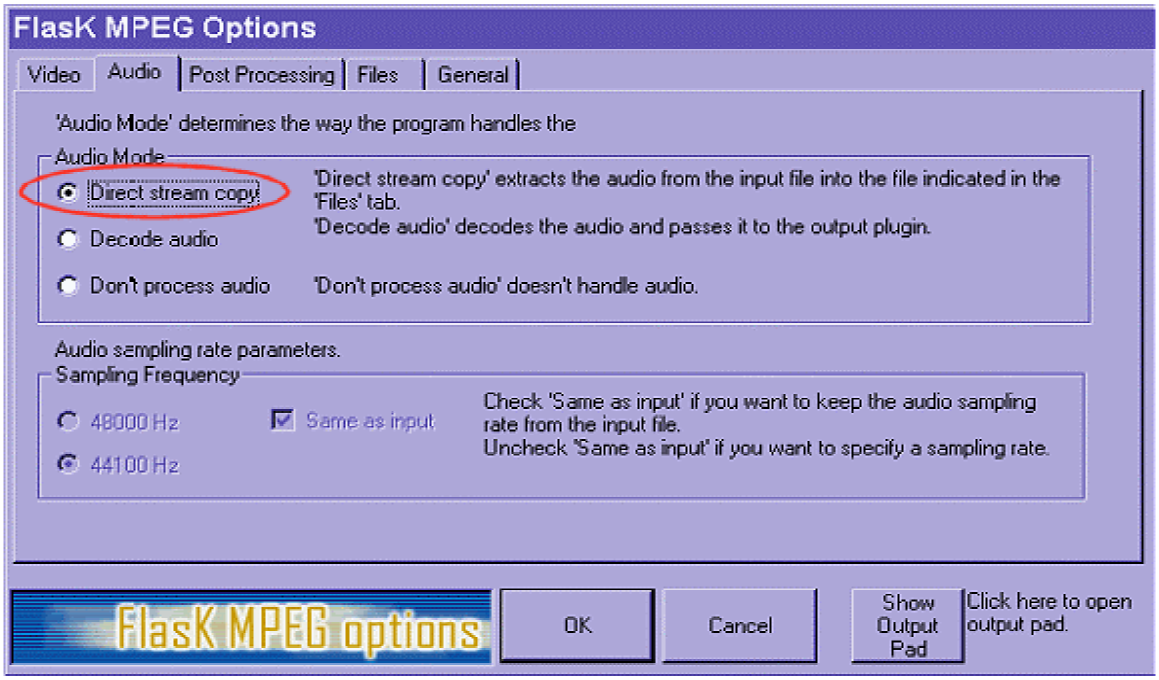
DVD audio specifications state that the audio soundtrack can come in three varieties:

1. Dolby Digital (formerly AC-3)
2. MPEG-2
3. PCM Wave

PCM is lovely to get because it is the computers preferred audio format and this makes it easy to copy, just use direct stream copy and your done. AC3 is the best audio format at the moment although it is soon to be replaced by AAC audio (Advanced Audio Coding) which is like a high quality MP3 version of AC3 audio. AAC used to be called NBC (Non-Backward-Compatible), because it was not compatible with the MPEG-1 audio formats. AAC and AC-3 use similar methods of compression but AAC uses a filterbank with a finer frequency resolution giving a much better compression and playback quality.

Anyway, sometimes you get a movie that uses a MPEG-2 (Mp2) audio track instead of AC3. This presents a problem because neither Flask Mpeg nor Ac3dec are able to decode Mp2 audio. So here are a couple of things you can do:

If you are using Fask Mpeg, use the "Direct stream Copy" option. This will extract the Mp2 audio at the same time your video converts and should keep them in time.



FlasK MPEG Options

'Audio Mode' determines the way the program handles the

Audio Mode

● Direct stream copy

○ Decode audio

○ Don't process audio

'Direct stream copy' extracts the audio from the input file into the file indicated in the 'Files' tab.
'Decode audio' decodes the audio and passes it to the output plugin.

'Don't process audio' doesn't handle audio.

Audio sampling rate parameters.

Sampling Frequency

○ 48000 Hz

● 44100 Hz

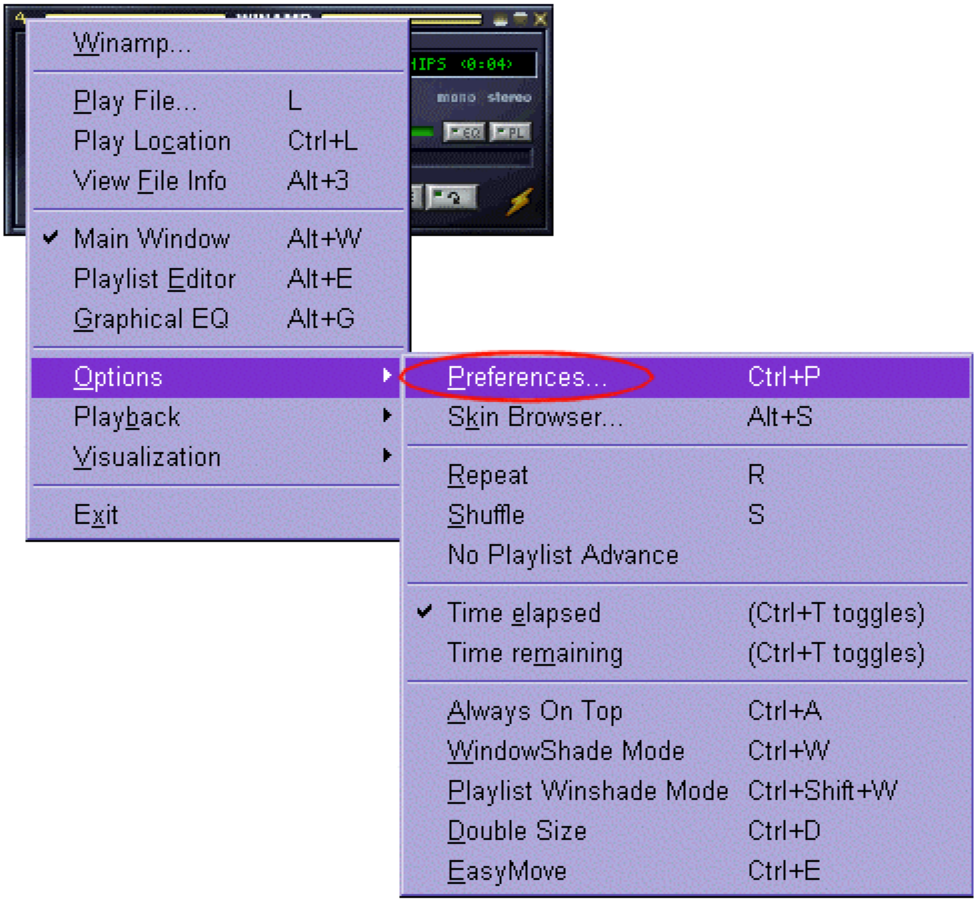
☑ Same as input

Check 'Same as input' if you want to keep the audio sampling rate from the input file.
Uncheck 'Same as input' if you want to specify a sampling rate.

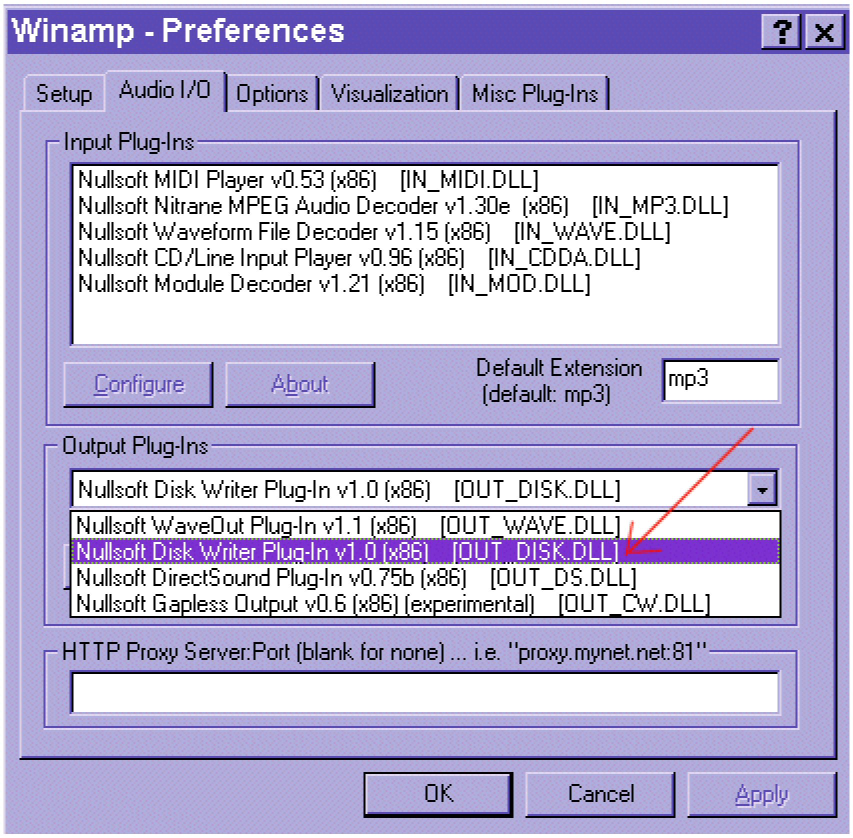
Click here to open output pad.

Open your Mp2 file in Winamp. Everyone knows how to use Winamp and its free of course ;^). Right-Click and choose

Options > Preferences...



Choose the Audio I/O tab. Now instead of the normal OUT_WAVE.DLL option, choose OUT_DISK.DLL. Press 'Apply' and Winamp will ask where you wish to save the file. Choose a location and wait until the audio is converted into a huge one and a half Gigabyte wave file.



Input Plug-Ins

Nullsoft MIDI Player v0.53 (x86) [IN_MIDI.DLL]
Nullsoft Nitrane MPEG Audio Decoder v1.30e (x86) [IN_MP3.DLL]
Nullsoft Waveform File Decoder v1.15 (x86) [IN_WAVE.DLL]
Nullsoft CD/Line Input Player v0.96 (x86) [IN_CDDA.DLL]
Nullsoft Module Decoder v1.21 (x86) [IN_MOD.DLL]

Default Extension (default: mp3): mp3

Output Plug-Ins

Nullsoft Disk Writer Plug-In v1.0 (x86) [OUT_DISK.DLL]

Nullsoft WaveOut Plug-In v1.1 (x86) [OUT_WAVE.DLL]
Nullsoft Disk Writer Plug-In v1.0 (x86) [OUT_DISK.DLL]
Nullsoft DirectSound Plug-In v0.75b (x86) [OUT_DS.DLL]
Nullsoft Gapless Output v0.6 (x86) (experimental) [OUT_CW.DLL]

HTTP Proxy Server:Port (blank for none) ... i.e. "proxy.mynet.net:81"

Now open your video file and wave file in VirtualDub (as explained in the Mpeg2avi tutorial) and convert the wave to MP3 or WMA audio.

If you don't use Flask Mpeg

If you don't extract the audio with Flask Mpeg, you can probably use Vstrip, Vobsnoopy, GraphEdit or the TMPGEnc's Demultiplexer to extract the Mp2 audio. Then you can convert it to Wave with Winamp as already explained. The usage of all these programs are explained in my other guides.

Cutting & Joining DivX, AVI or ASF

Cutting any DivX, AVI or ASF file can be done without problems using VirtualDub. But to cut ASF files you will need VirtualDub version 1.3c because Microsoft put the thumb screws on Avery Lee and so he took it out of all his later versions. Another great program that can chop some mpeg formats is TMPGEnc. Again, all these utilities are free.

As a word of warning, please do *not* use AVIChop, Pecks Power Join or AVI Devil to try and chop an AVI because you are likely to get corrupt files. The following will describe how to cut and join Divx files, but the same method will work with any AVI, MJPEG, ASF or popular format that VirtualDub can open. The exception to this is Mpeg-1 & 2 but I'll explain how to do that in another article =). This method *doesn't* require that we re-encode any file, all we are doing is cutting them.

Note: *ASF's are a very twitchy format and don't seem to be fully supported by anything much except Microsoft Media Encoder. If you have trouble chopping up an ASF file in VirtualDub try Windows Media Indexer instead, it can be downloaded Here. If you have trouble joining two ASF files try opening them in VirtualDub and resaving them again using the Direct Stream Copy option for both Audio and Video. Then you should be able to join them together without problems. If the ASF doesn't open at all there is not much you can do, perhaps its just a corrupted file? The following guide will show you how to do everything spoken of here anyway.*

Before I start here are the things you will need:

VirtualDub 1.4c

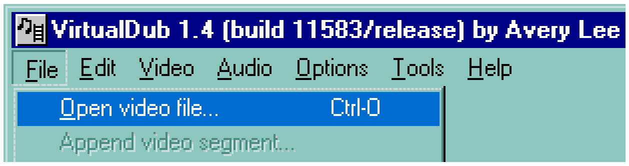
VirtualDub 1.3c (with ASF support)

CUTTING A DIVX INTO TWO PARTS

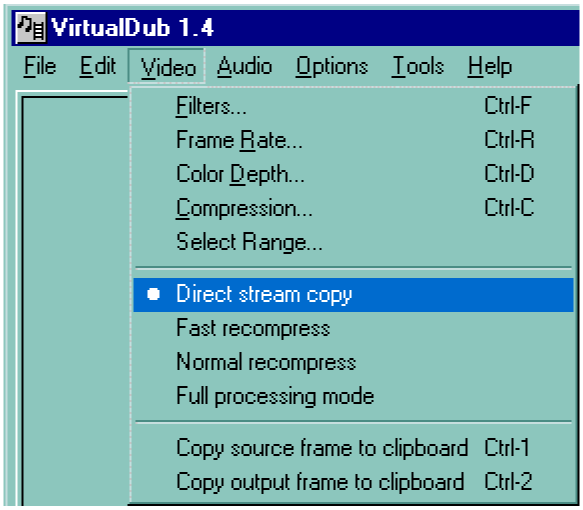
Make sure all previous options you have selected, such as filters etc., in VirtualDub are not selected. The best way to do this is to shutdown VirtualDub and then load it again.

SET IT TO COPY

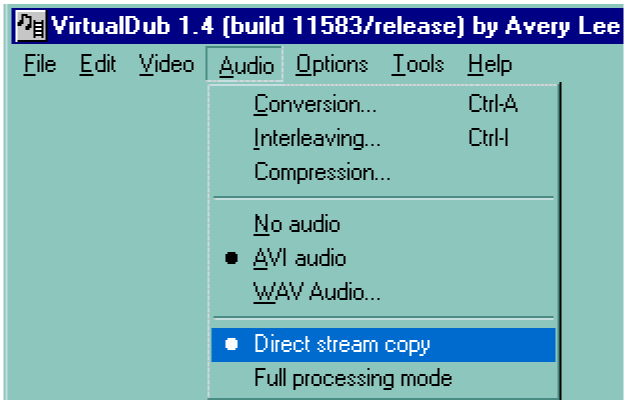
Select File > Open



Go to Video > Direct stream copy. (Note that VirtualDub puts a dot by the options we select)



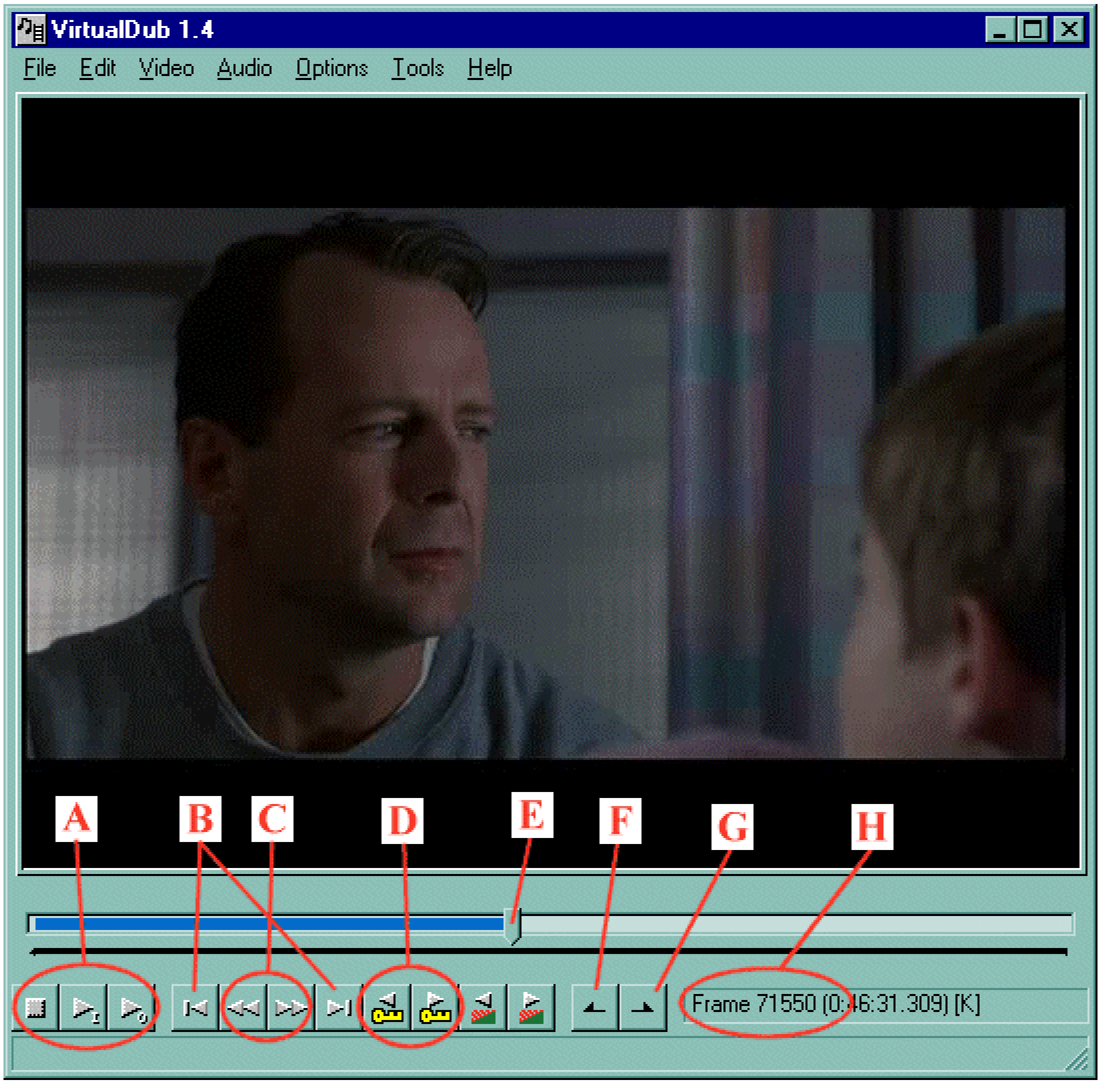
Go to Audio > Direct stream copy.



SELECT WHERE TO CUT

VirtualDub doesn't actually cut a file in half at a certain point, instead it just lets us save any part. So to cut a movie in half we must select the first have, save it, then select the second half and save that. The beauty of this cutting method is we can select exactly where we want to cut it. All the old cutting methods just cut the movie exactly in half at a certain point. The problem with that was it often cut it in the middle of an action scene and totally spoil the movie because we would have to change CD's. With this method you can select to cut it wherever you like. Naturally it is best to cut a movie where one scene ends and another starts. The only thing you have to watch is that each half is not too large for the CD after it is cut =).

If you are unfamiliar with VirtualDubs controls take a look at the picture below. No, its not an alphabet revision. If we look a (**A**) we have our basic video controls: stop and play. There are two play buttons, one plays the input movie and the other plays both the input and output movies together. If you click on the pictures you can stop them playing, and if you right-click on the picture you can select to zoom in or out. This, of course, is just for viewing purposes. Next (**B**) we have the jump to start and jump to end of the movie buttons. And (**C**) allows us to move frame by frame either way.

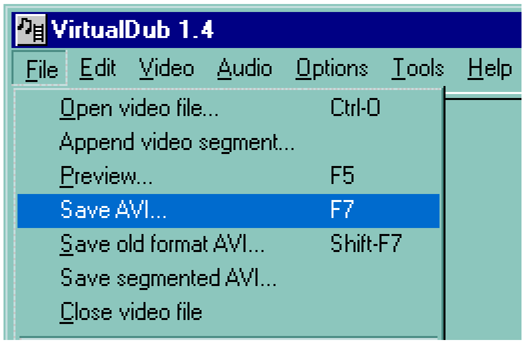

VirtualDub 1.4
File Edit Video Audio Options Tools Help
A
B
C
D
E
F
G
H
Frame 71550 (0:46:31.309) [K]

To cut the movie in half we must move to the middle. First move to the end of the movie by pressing the jump to end button (**B**). Divide in half the number of frames it then shows at (**H**). This gives me something like frame number: 77229. So I use the slider bar (**E**) to move very close to that number, which will be the middle of the movie. Now this is the *important* point we must use the move by keyframe buttons (**D**) to select where to cut our movie. If you don't you could end up with a few seconds of corrupted video where the cut was made! For more information on why check out my article in the appendix called

"Key Frames & Delta Frames Explained". In short, if move using the keyframe buttons you cannot cut it wrong!

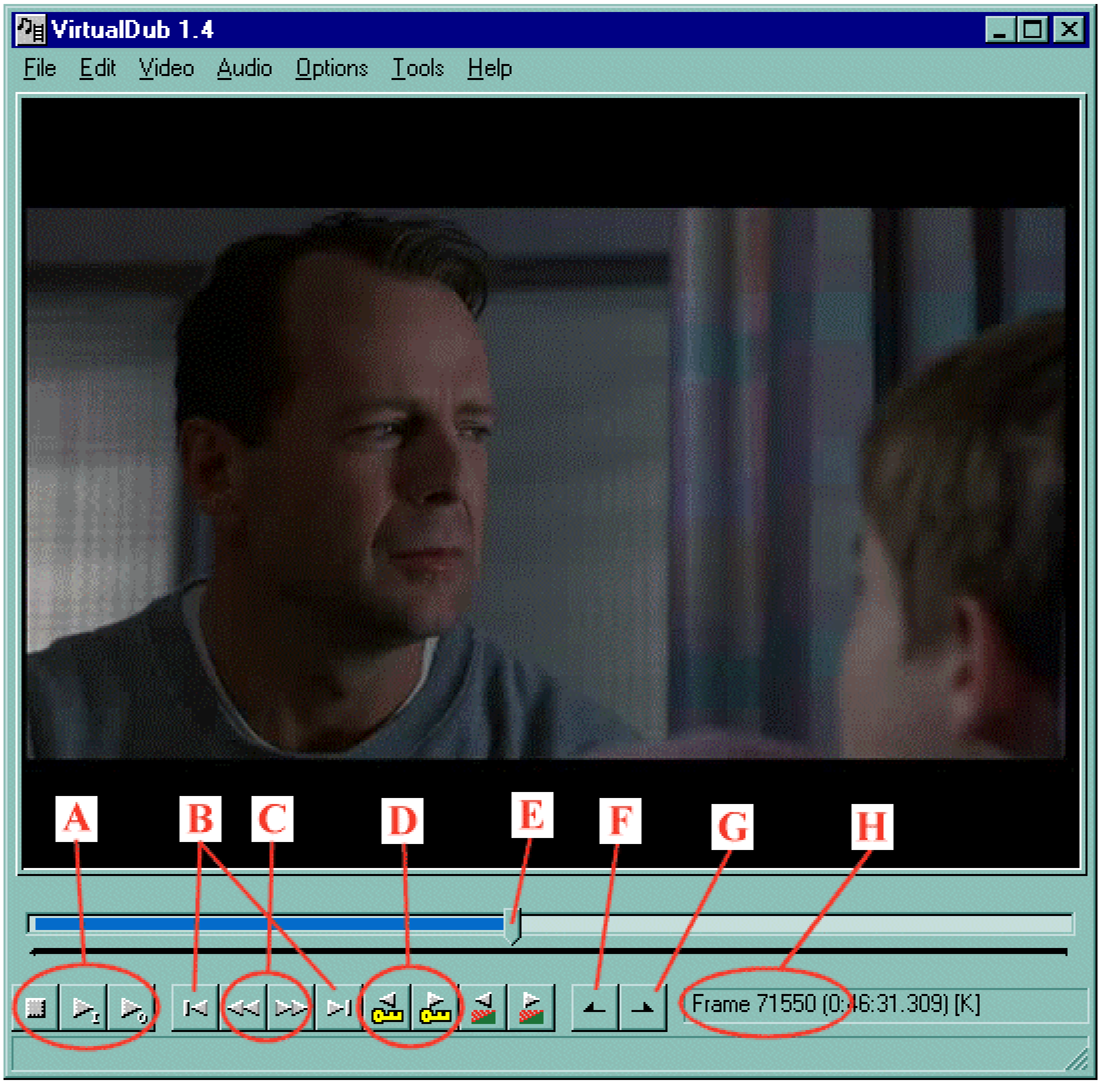
Okay, so we are at the point we wish to cut the movie. Write down the frame number you have selected to cut the movie, I have selected frame 77575 because it fades to black here. Hit the 'mark end point' button (**G**) (also called the "mark out" button). Next, press the jump to the start button (**B**). Then press the 'mark start point' (also called "mark in"). You will notice that the slider bar (**E**) now shows a blue line next to it - this is your selection. You have marked from the middle to the start of the movie.

Then we save it: File > Save AVI...



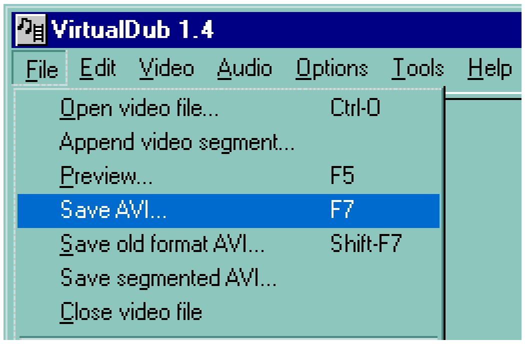
Bingo! Now if you look at the file you just saved it will be the first half of your movie.

Next go to the middle again to the same frame number you cut it before; for me this was: 77575. Hit the 'mark start point' button (**F**). Then press the 'jump to end' button (**B**). And finally the 'mark end point' button (**G**).



Frame 71550 (0:46:31.309) [K]

And again we select: File > Save AVI...



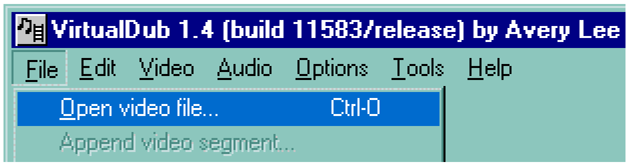
Look at the file you just saved it will be the second half of your movie. That's it! You have cut a DivX movie in half!

Final Note: *You may have noticed that cutting movies using the keyframe buttons can produce a repeated keyframe both at the start of the second file and the end of the first! But, considering there are 25 frames to every second of a movie, this will not even notice, especially when we are splitting a movie onto two CD's anyway =). If you want to delete this, use the 'frame by frame' buttons (**C**) to do so. The only rule to remember here is to never delete a keyframe from the start of a movie clip always delete it off of the end of the clip.*

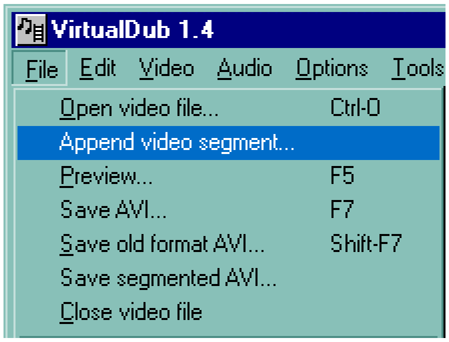
JOINING DIVX FILES

Joining together Divx, AVI or ASF clips is just as easy with VirtualDub once you get the hang of it. Again, if you are cutting up files you must do so on the keyframe to avoid corrupt files. But to join the clips just open them in VirtualDub in the correct order and use the append option. For example:

Open the first file by File > Open video file...


VirtualDub 1.4 (build 11583/release) by Avery Lee
File Edit Video Audio Options Tools Help
Open video file... Ctrl-O
Append video segment...

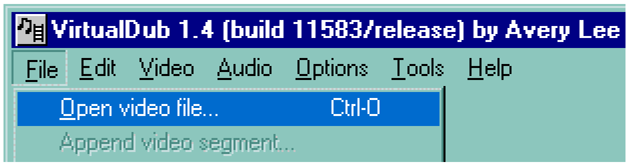
Then to join each file to the previous one. You open them one at a time by going to: File > Append video segment...



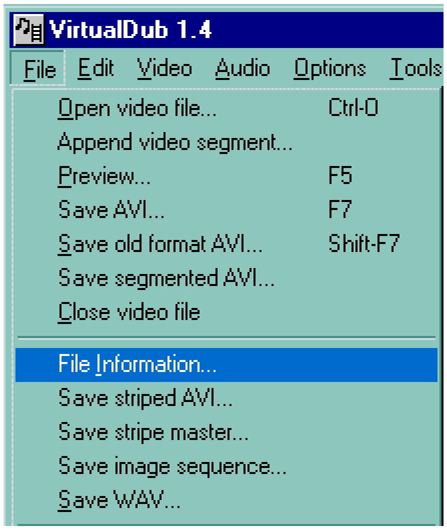
WHAT CAN & CANNOT BE JOINED?

It is very very important that the files you join together are exactly the same format! You cannot join files of two different codecs. For example an DivX will never join with an MJPEG, one of the two must be recompressed first into the others format. Neither will movie clips of two different sizes join together. So you cannot join a 352 x 288 movie clip with a 352 x 240 clip! You cannot join two movie clips together if the audio is a different format. So if one Divx uses Mp3 audio and the other uses WMA audio they will not join and so on. Finally, VirtualDub doesn't support the editing or the joining of Mpeg-1 or Mpeg-2 files yet. To edit them you will need to use TMPGEnc, for that see my guide: *"Cutting & Joining VCD (Mpeg-1 / 2)"*.

To find out exactly what format a file uses just open it in VirtualDub.


VirtualDub 1.4 (build 11583/release) by Avery Lee
File Edit Video Audio Options Tools Help
Open video file... Ctrl-O
Append video segment...

And choose File > File Information...

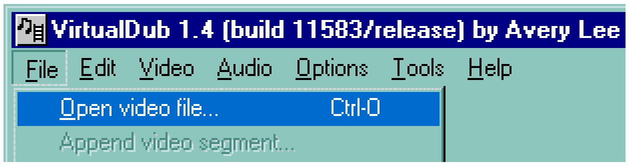


To join any file to any other all you need to do is write down the file information from one and re-encode the other (preferably the shortest clip) with the same setting.

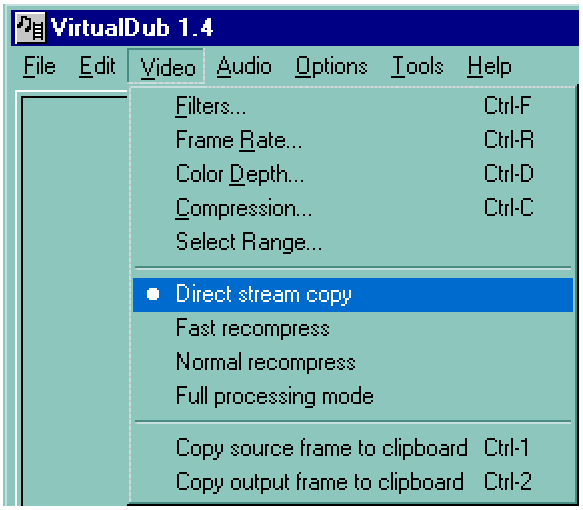
JOINING AND AUDIO SYNCHRONIZATION

Sometimes when you join two files together the second file will go out of synchronization. I think this is because sometimes the audio is clipped too short and doesn't match the video file length. Often you can get around this by creating an offset. To do this, open the first file you wish to join in VirtualDub.

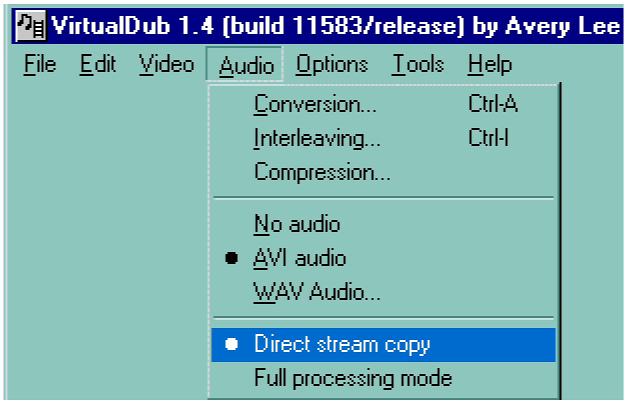
Select File > Open


VirtualDub 1.4 (build 11583/release) by Avery Lee
File Edit Video Audio Options Tools Help
Open video file... Ctrl-O
Append video segment...

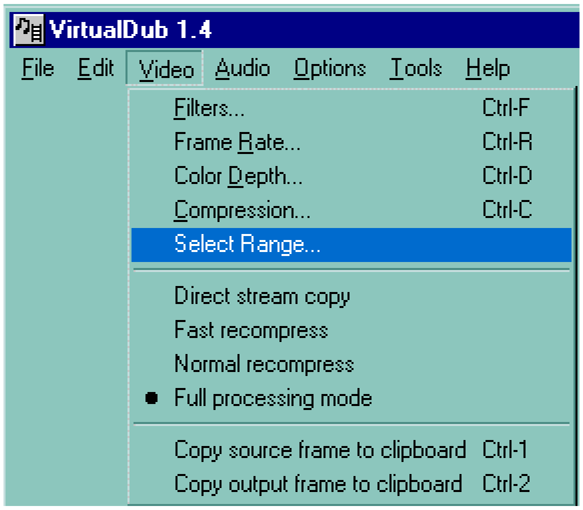
Go to Video > Direct stream copy. (Note that VirtualDub puts a dot by the options we select)



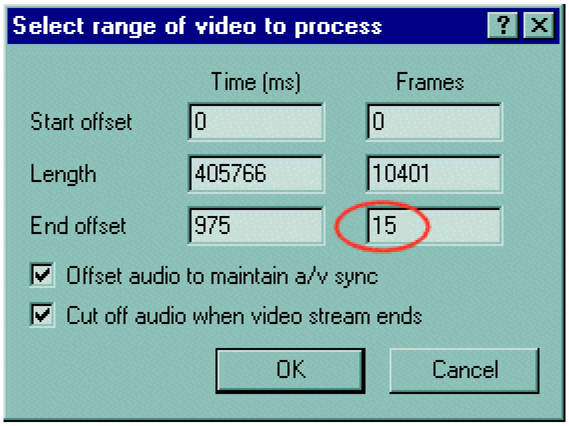
Go to Audio > Direct stream copy.



Go to Select Range...



In the 'End offset' of the picture below (encircled in red) type in about 15 frames.

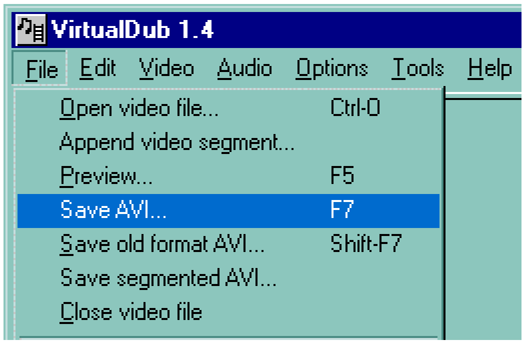

Select range of video to process

	Time (ms)	Frames
Start offset	0	0
Length	405766	10401
End offset	975	15

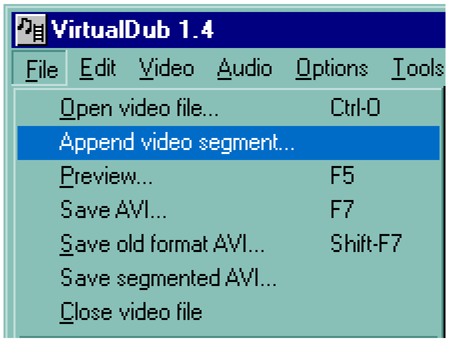
☑ Offset audio to maintain a/v sync

☑ Cut off audio when video stream ends

Then save your newly created first file.



That's it. Now open this new file as your first file to join. Then open the next file you wish to join to it with the Append video segment option as already explained.



15 frames should be enough of an offset to keep it in synch. But you may need to increase or decrease this number a little. If the offset is too much you will get a jump in the movie as it skips a keyframe. If it is too small the picture will freeze and the audio will continue going! So please make sure it is correct before you save your final movie.

COOL THINGS YOU CAN DO WITH JOINING

One of the things I have found fun to do with this joining method I created is to re-add the cut scenes from movies. These cut scenes are usually added to the DVD as specials. But I thought it would be a great idea to put them back in and create my own special edition movies =). For example, I have added all the cut scenes from the Alien movie to my DivX including that Cocoon scene everyone is talking about.

I would also love to add the other Cocoon scene from Aliens, but that was not on the DVD specials as expected. If anyone has it on video from a documentary please compress it and send it my way :). Here is a photo from a magazine:

ADDING THE SPECIALS

Sometimes adding the specials is not as easy as it sounds. So, just for you, I'll let you in on a few tricks I have used that will also help you join specials or compile lists of any files seamlessly.

First thing to do is to get the specials off of the DVD. SmartRipper can extract individual chapters and Vobs so this shouldn't be too difficult. Since most specials are very short its best to either encode them as uncompressed avi, or if hard drive space is limited, as highest quality MJPEG with the Huffyuv or PicVideo codecs etc. This is mainly because we can do perfect frame by frame editing with these formats. It is also an idea to do use PCM audio until *all* editing has been done. This way we will avoid sound degradation and audio synch problems. Finally, watch out for interlacing effects on special which were not on the main movie, and if necessary, deinterlace them.

CROPPING

It's hell trying to match a movie to a special if both have black bars on them! Ideally you should crop both the specials and the original movie to exactly the same size so no boarders exist at all! If the specials are smaller than the movie when cropped, resize them larger and crop them to the same size the original movie was encoded to. This may loose a bit more of the picture than we wanted but won't notice on the final version.

COLOUR MATCHING

Okay, open VirtualDub twice! Once with the Divx you wish to splice the extra scenes into and once with the cut scenes. Find the original scene in the movie, or think of a good place you could cut your new scene into and go to it. Specials and cut scenes are usually good quality video conversions and hence will not always look the same as the main

DVD movie. The colours, saturation and brightness' of scenes may be a little different to match. Colour matching is absolutely vital if they are to look natural. Things you should look out for to make the files match. For detailed explanations on video editing with VirtualDub see the articles I have written in the advanced VirtualDub section.

You should be aware of:

1. Brightness, Contrast & Levels: Try and get the overall tones to match the original movie.

2. Hue & Saturation: Sometimes there will be too much colour flair in the image and reducing the saturation a bit helps. Also, at least in the case of the movie Alien, I needed to cut out some of the red to match the more blue-green trend of the original DVD.

3. Sharpness: Specials are almost always slightly more blurred than the original DVD. The unsharp mask filter gives us the most control here and is probably the one to use.

4. Noise & Clarity: Specials usually have more noise than original movies. Using such filters as the 2D cleaner can do wonders in this area.

You probably will not need to adjust all of the above, but these are the things to look out for. Try each filter and compare the cut scene with the original. When you are happy save the filter settings ready for when you encode from AVI to Divx

SPLICING

Ideally we want frame by frame accuracy. But as you know this will corrupt your Divx so you must edit from keyframe to keyframe. We could encode the whole movie to uncompressed AVI but at something like 1GB per minuet this is not possible. Just to clarify, if you set your keyframes to 1 every second when you encoded your original Divx you should be able to splice in most parts without frame by frame accuracy. This may require you chop out a second or two here or there from the special or original move. But for those tricky scenes try this method.

Open the original movie in VirtualDub and use the keyframe buttons to chop the movie into three parts using *nothing* but the keyframe buttons!

1. From the start to where you want to insert the cut scene.
2. The area where the cut scene will be (i.e. perhaps 5-10 seconds)
3. The part from after the cut scene to the end of the movie.

The second in the above list may be a little confusing at first but you'll see what I mean. Imagine you have found a scene change perhaps, for example, a scene that flicks from the office to one walking home. You think, 'this is the perfect place to add this cut scene'. So we cut out 5-10 seconds of it using the keyframe buttons. So we have a little bit of the office and a little bit of walking home in this clip. Then open the clip again and resave it as an uncompressed AVI or MJPEG.

Bingo! Now it can be edited it frame by frame. Chop it into two parts:

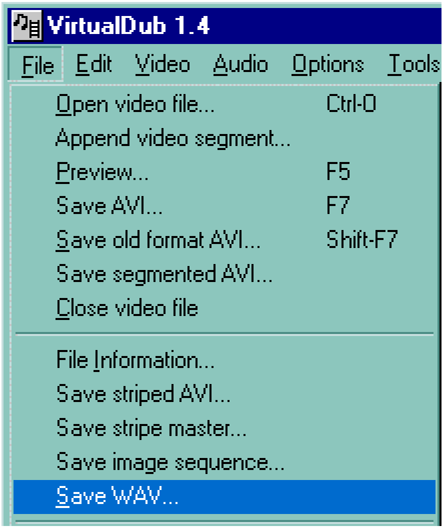
1. Before the change
2. After the change

Finally, open the 'Before the change' in VirtualDub. Append the new cut scene you wanted to put in-between. And append the after the change. Resave and compress it to Divx using the exact same settings you did for your original movie.

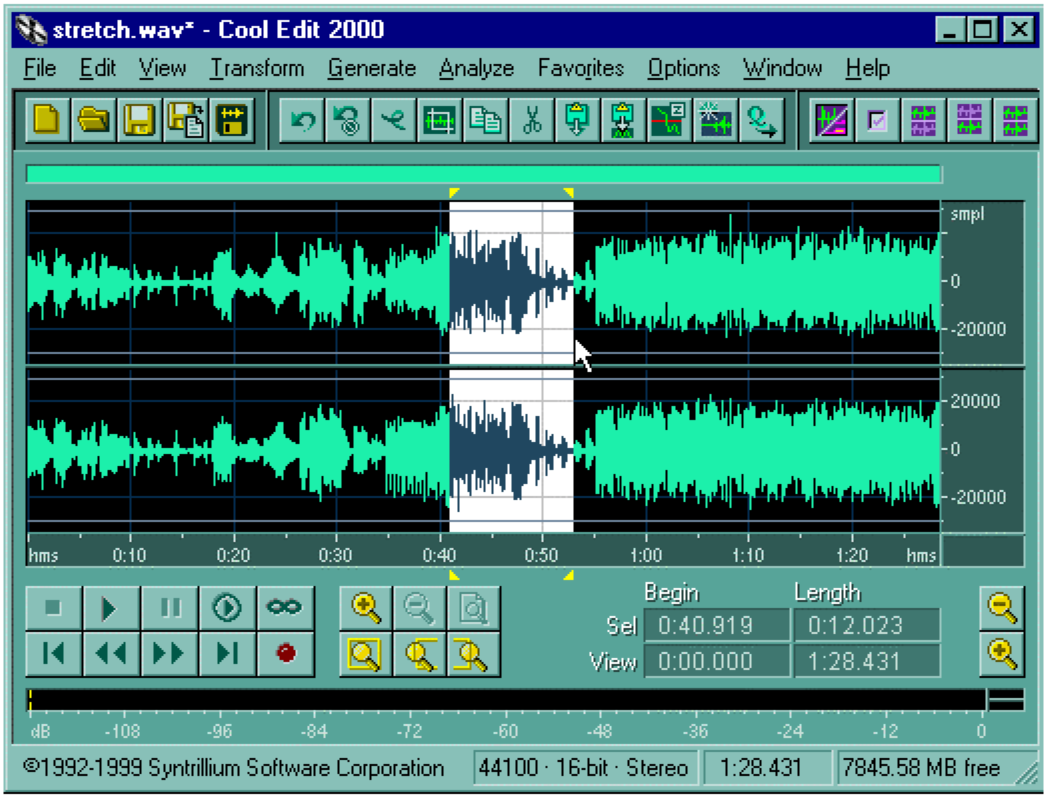
MATCHING THE AUDIO

Nearly done! Just one more thing needed to create the perfect seamless join. Sometimes the audio doesn't quite fit right. For example, the background music may carry on from one scene to another and the cut scene may not have any background music at all! To solve this we must open the audio file in an audio editing program such as Cool Edit or sound Forge. This is another good reason to keep the audio as uncompressed PCM until the cut scenes are done. Save the audio from the newly joined cut scene by opening it in VirtualDub and choosing:

File > Save Wav...



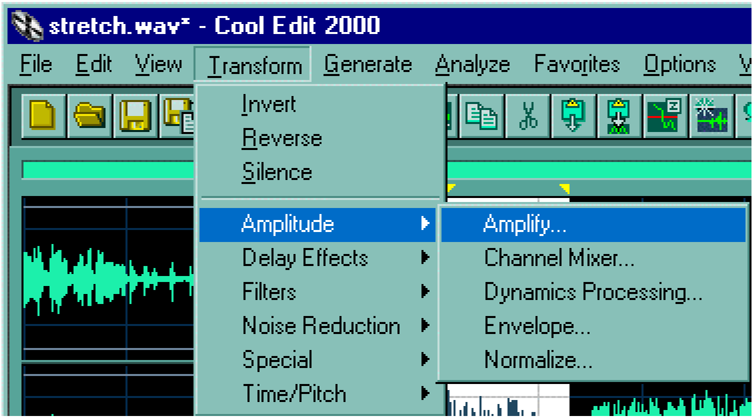
Open it in Cool Edit and find the part where the music, noise, etc., is located. And select just before the change like this:


stretch.wav* - Cool Edit 2000
File Edit View Transform Generate Analyze Favorites Options Window Help
smpl
0
-20000
20000
0
-20000
hms 0:10 0:20 0:30 0:40 0:50 1:00 1:10 1:20 hms
dB -108 -96 -84 -72 -60 -48 -36 -24 -12 0

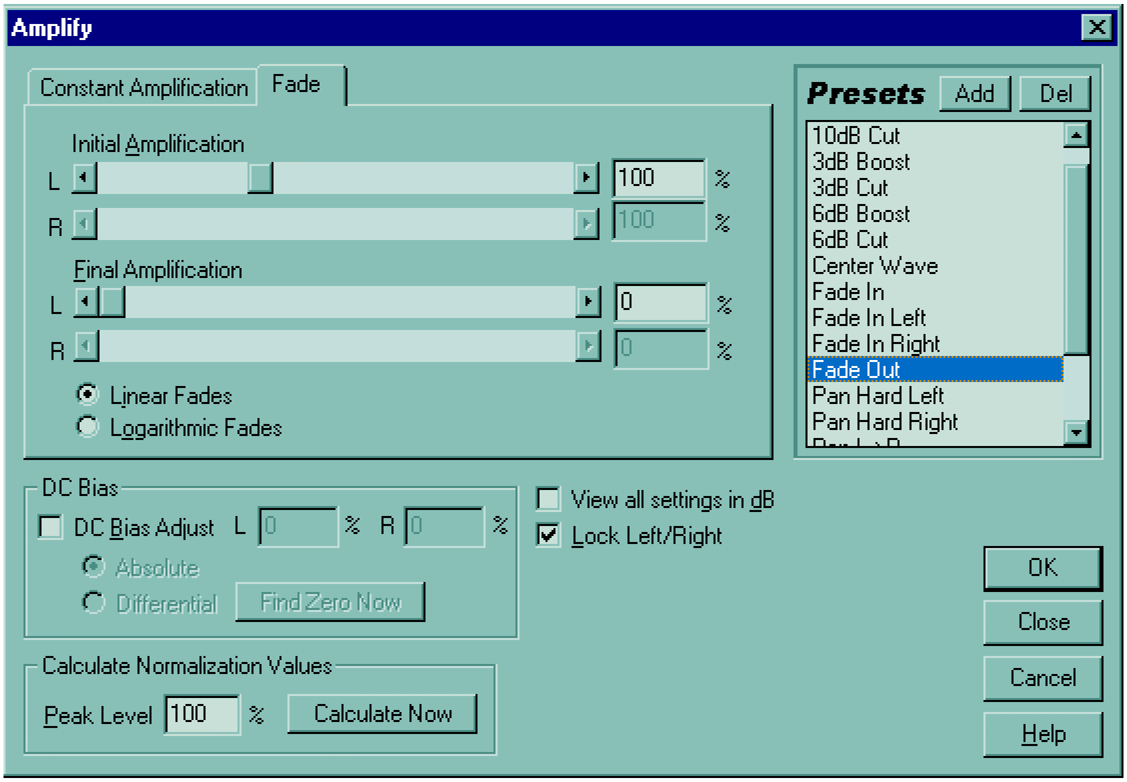
	Begin	Length
Sel	0:40.919	0:12.023
View	0:00.000	1:28.431

©1992-1999 Syntrillium Software Corporation | 44100 · 16-bit · Stereo | 1:28.431 | 7845.58 MB free

Then with that part selected use: Amplitude > Amplify...



Up pops this crazy box. Just hit the fade out button and press ok.



Initial Amplification L 100 % R 100 %

Final Amplification L 0 % R 0 %

● Linear Fades
○ Logarithmic Fades

□ DC Bias Adjust L 0 % R 0 %

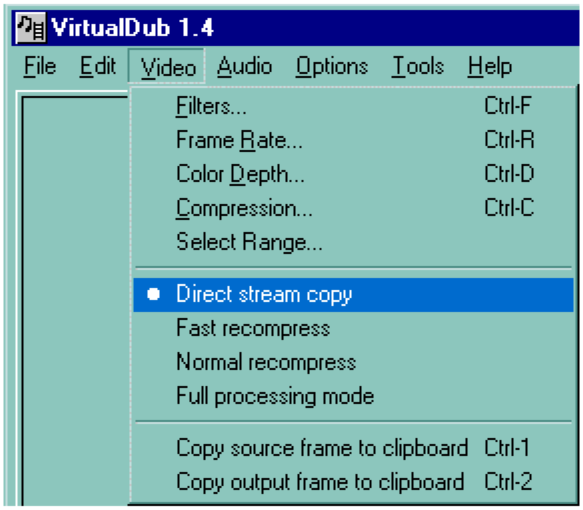
□ View all settings in dB
☑ Lock Left/Right

Peak Level 100 %

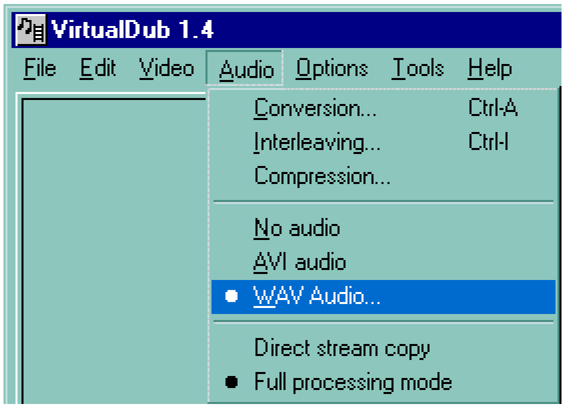
If you selected the area correctly the background music / noise etc., will be faded out. Then, if necessary, select the bit after the scene change but this time choose 'fade in'. This will make the background music fade smoothly in and out between the two scenes leaving no final traces of editing =). Once done just choose File > Save.

To add this audio back to the clip open the clip in VirtualDub. And choose:

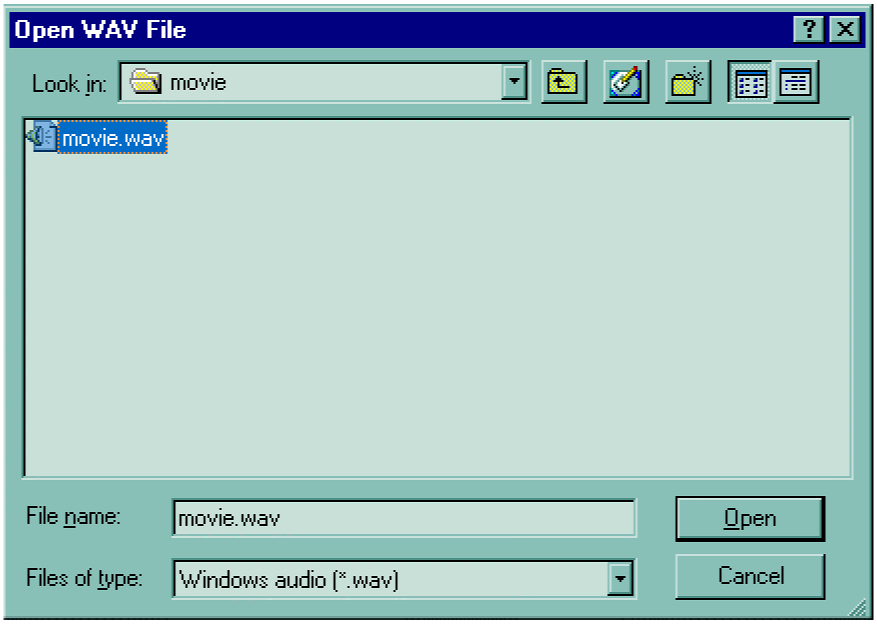
Video > Direct stream copy



Audio > Wav Audio...



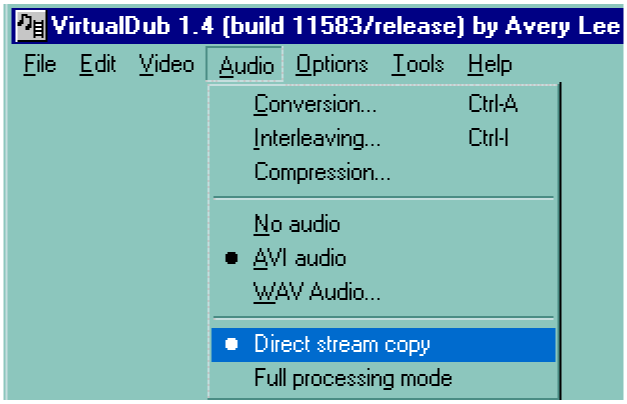
Browse for your altered wave file and press Open:



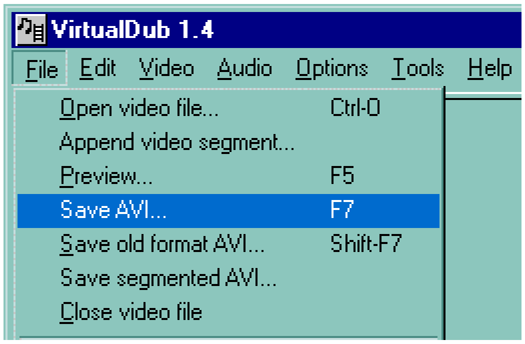
File name: movie.wav

Files of type: Windows audio (*.wav)

Audio > Direct stream copy



Save it baby!



THE FINAL JOIN

Finally, we can join our seamless clip back to the original cut Divx movie by joining:

1. From the start to where we wanted to insert the cut scene.
2. Our NEW cut scene
3. The part from after the cut scene to the end of the movie.

Phew! Well done!

Cutting & Joining Mpeg-1 & 2 with TMPGEnc

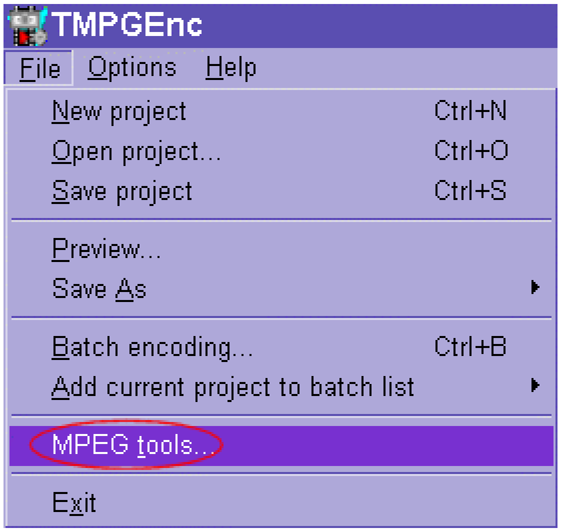
Probably the best utility for chopping Mpeg files is TMPGEnc. You can also use M1-Edit, M2-Edit or IFilmEdit v1.4.5 if TMPGEnc fails, but TMPGEnc is usually better at it and is a free utility too. Some Mpeg files, especially those encoded using the hardware video compression of a capture card, will have problems being cut or joined together. There is no perfect tool for cutting Mpeg files, one may fail when another will work or none of them may work! Mpeg is one of those formats were was never designed to be cut up and edited. But VCD Mpegs made with BBMpeg work perfectly.

Using TMPGEnc is my prefered choice for VCD's and SVCD's because you can cut exactly where you want. Remember, if you have a 650MB CD-R you can fit a 740MB VCD or SVCD on it. A 700MB CD-R will allow us to fit 820MB VCD's or SVCD on it. So when you cut the movie take your time, select a nice place where swapping the CD's will not annoy you (i.e. not in the middle of an action scene). Find a good scene change and cut it there. As long as each part is smaller than the full capacity of each CD you intend to put it on you will be fine.

Before I start you will need:

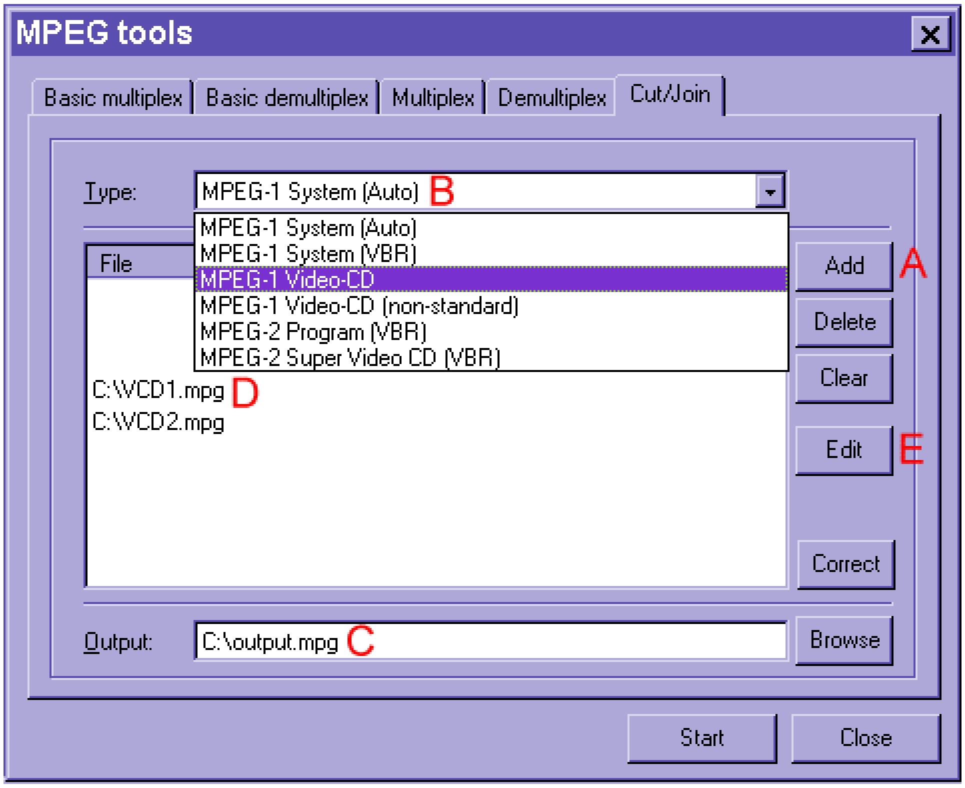
TMPGEnc

Load TMPGEnc and choose: File > Mpeg Tools...



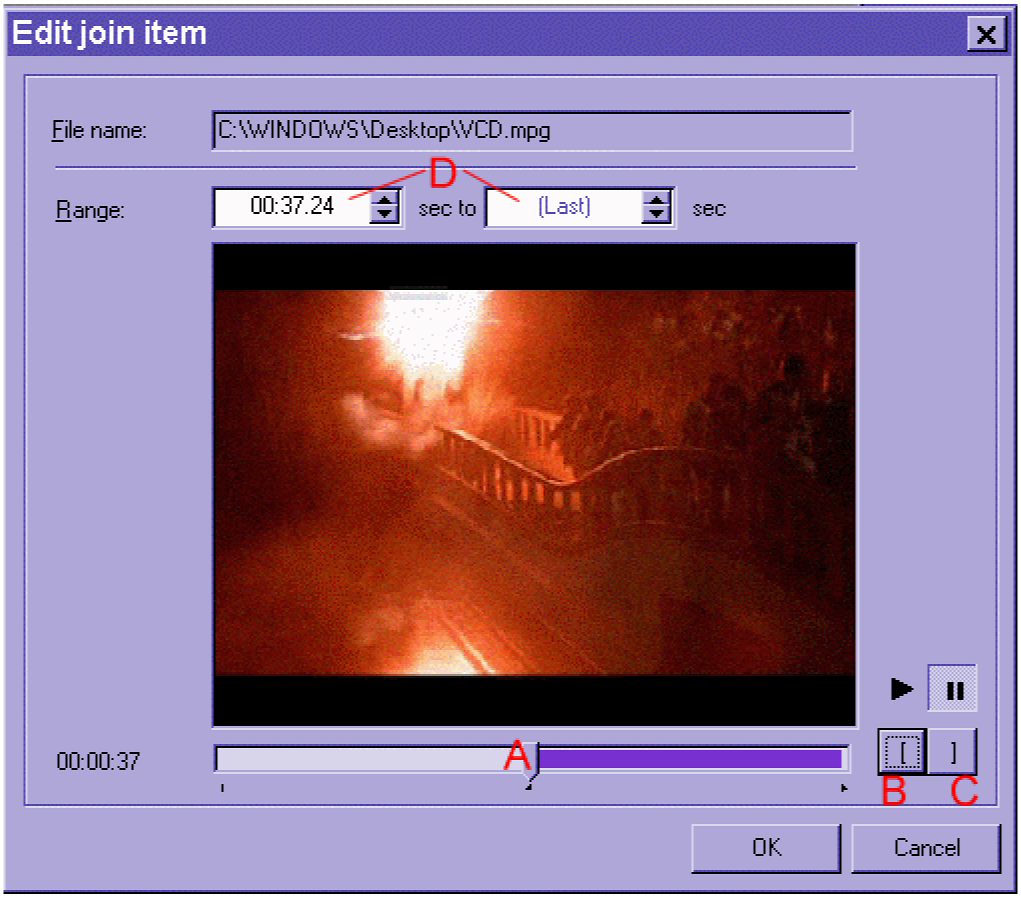
Cutting a Video File

Add the movie clip you wish to cut by pressing the 'Add' button (**A**). Choose the video type you are trying to cut (**B**). So if its a normal Mpeg-1 choose 'Mpeg-1 System (Auto)'. If its a VCD use 'Mpeg-1 VideoCD'. If its SVCD then use the 'Super Video CD (VBR)' and 'Mpeg-2 Program(VBR)' for other Mpeg-2 formats.



Browse for where you wish to save the split mpeg file and give it a name (**C**). Okay highlight the video clip your gonna cut (**D**) and hit the Edit button (**E**).

Up pops the following window. Use the slider bar (A) or the Range settings (**D**) to select where you want this clip to start from. Once you are happy press the start at button (**B**). Next move the slider bar until it reaches where you want the video clip to end and press the mark end button (**C**).



Edit join item

File name: C:\WINDOWS\Desktop\VCD.mpg

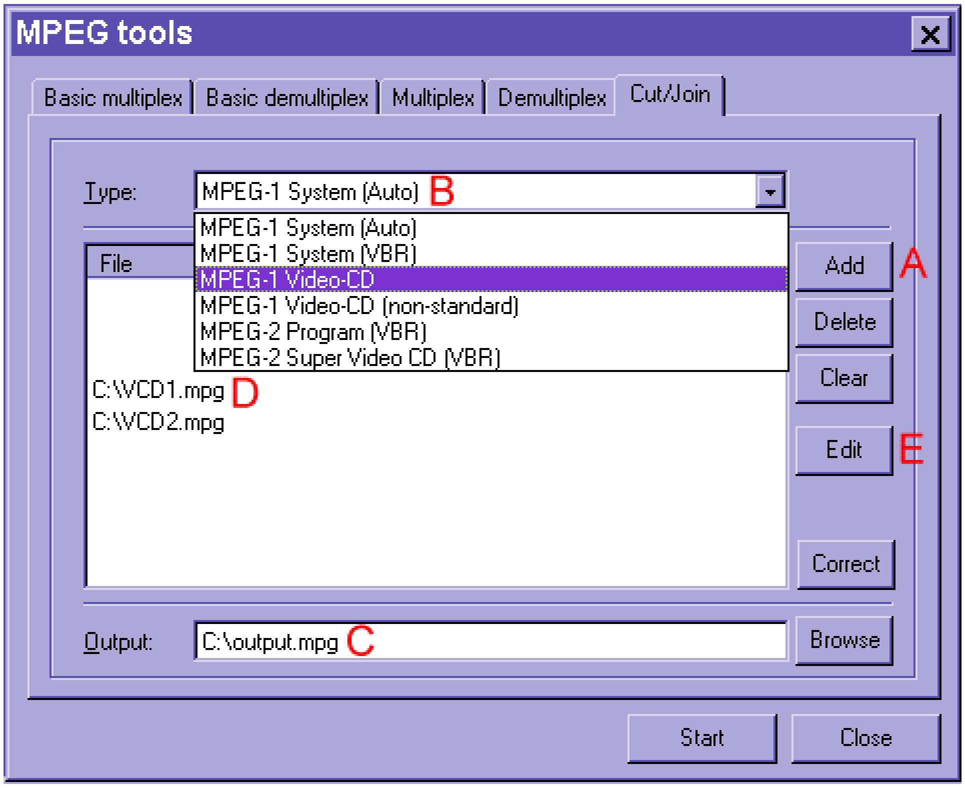
Range: 00:37.24 sec to (Last) sec

00:00:37

When you are done press OK. Then just hit the start button and your new clip will be saved, bingo!

Joining a Video File

Joining files is even easier, you could join 50 Mpeg files in a matter of a few minuets. Again choose the video format that you will be joining clips from with the drop-down menu (**B**). Just keep adding the files in the order you wish to join the together (**A**). If you get the order mixed up you can drag one file above the other.



Type: MPEG-1 System (Auto)

C:\VCD1.mpg
C:\VCD2.mpg

Output: C:\output.mpg

Finally choose where to save the joined mpegs (**C**) and press start.

Obviously if two mpeg files are not the same they will not join. You could not, for example, join a 352 x 288 video clip with a 352 x 240 video clip. Neither could you join a video clip of 25fps with one of 29.97 and so on.

That's all you need to know...get to work =).

Open Any File with AVISynth

AVISynth is a really cool utility, yet it seems very few people are using it! The reason many haven't started is probably not because AVISynth is hard to use, but basically because they downloaded it, double-clicked on each file to try and get it working only to find that it doesn't seem to do anything!? So perhaps it may be a good idea if I explained what AVISynth does and how it can be used!

What can we do with AVISynth?

Lets say you downloaded a weird format video file and wanted to edit it with VirtualDub or convert it to Mpeg with Panasonic Mpeg encoder or with some other Mpeg / AVI etc., encoder. It would normally be impossible to do this because neither VirtualDub nor Panasonic Mpeg Encoder supports this format. Windows Media Player, however, is usually able to play just about any file with no problems. If it cannot play it there will usually be a codec you can install that will allow Media Player to do so. Well, if Media Player can play it usually AVISynth will let you open it in whatever application you need! This even applies to Mpeg-2 files! Yep, that's right! If Media Player can play your Mpeg-2 file then AVISynth will let you open them in almost any video editing or converting utility!

But that is not all! Before it sends the video to be opened in VirtualDub (or whatever application you want) it gives you the option to do all sorts of things to the video! You can resize it, crop it, sharpen, blend, separate interlaced fields, delete frames, duplicate frames, add 3:2 pulldown, reweave interlaced fields, bob deinterlace, convert between YUY2 and RGB and a whole host of other interesting and useful features!

How does it work?

AVISynth acts as a go-between between your video file and any video application you wish to use. You install AVISynth by putting a small .dll program in the Windows System folder. This is almost like a Windows upgrade, it then intercepts all files opened that are called something.**avs** and then runs them through AVISynth. In turn AVISynth decodes them using Media Players codec's and sends them to the application you were opening them with - cool huh!

Basically to do this you need to write a simple text file saying what file you wish to open (just like making a .lst file for Mpeg2avi) and rename the text file with an *.avs extension. Then, when you open this file in VirtualDub or Panasonic Mpeg Encoder (or whatever application you are using) AVISynth will take over and covert it for that application.

Don't worry, it may not sound too clear to you right now, but its *not* hard and after you have followed my step-by-step examples you will be able to set it up and open any file in seconds. So without further delay lets get started.

Lesson 1: DVD to AVI with AVISynth!

Since we are Mpeg-2 obsessed at Digital Digest, what better than to show you how to open a DVD in VirtualDub. As you know VirtualDub doesn't support Mpeg-2 and cannot open or edit Vob files...we will soon change that!

Can you open DVD or Mpeg-2 in Media Player?

As I mentioned, AVISynth can open just about any file that Media Player can handle. But if you cannot open a Vob file in Media Player then you will need to install the Ligos and WinDVD codec's first so Media Player can. This is easy enough to do:

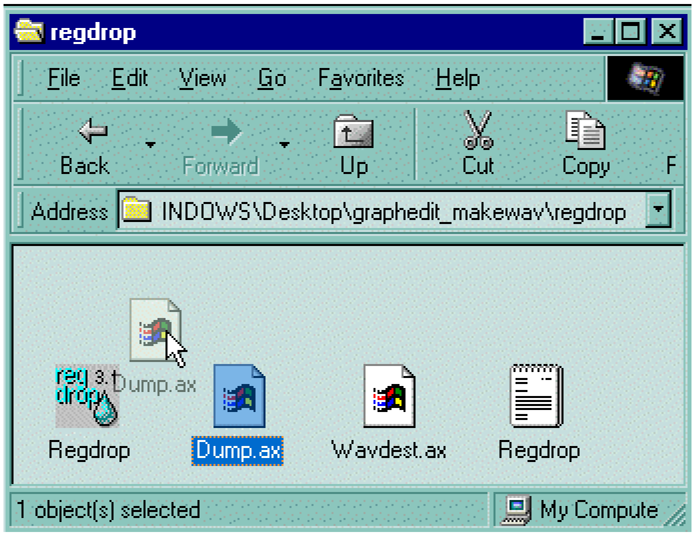
You will need:

RegDrop

Ligos Mpeg-2 Filter

WinDVD Audio Filter

Download the above files and drag the something *.ax file over the RegDrop program and your files are registered. Like this:


regdrop
File Edit View Go Favorites Help
Back Forward Up Cut Copy F
Address INDOWS\Desktop\graphedit_makewav\regdrop
reg drop
3.1
Dump.ax
Regdrop
Dump.ax
Wavdest.ax
Regdrop
1 object(s) selected
My Compute

Note: *the picture above is only an example of what you do, you do not need any Dump.ax or Wavedest.ax filters.*

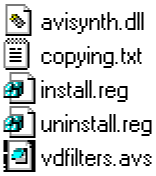
Installing AVISynth

Before I start here are the things you will need:

AVISynth 1.0 Beta 3

VirtualDub 1.4c

After you have downloaded and unzipped AVISynth you should end up with a folder containing the following files:



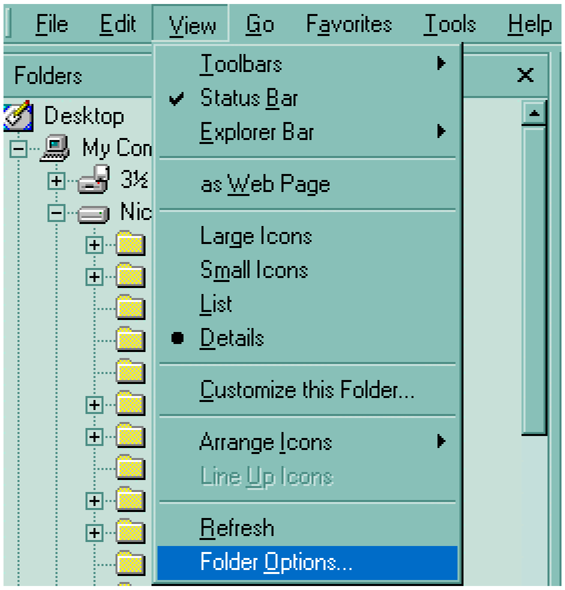
avisynth.dll
copying.txt
install.reg
uninstall.reg
vdfilters.avs

Cut and paste the first file called avisynth.dll into your C:\Windows\System folder. Then double-click on the install.reg file and the program will be installed. To uninstall it double-click on the uninstall.reg. To uninstall you can also delete the avisynth.dll file from your Windows\System folder but that's up to you.

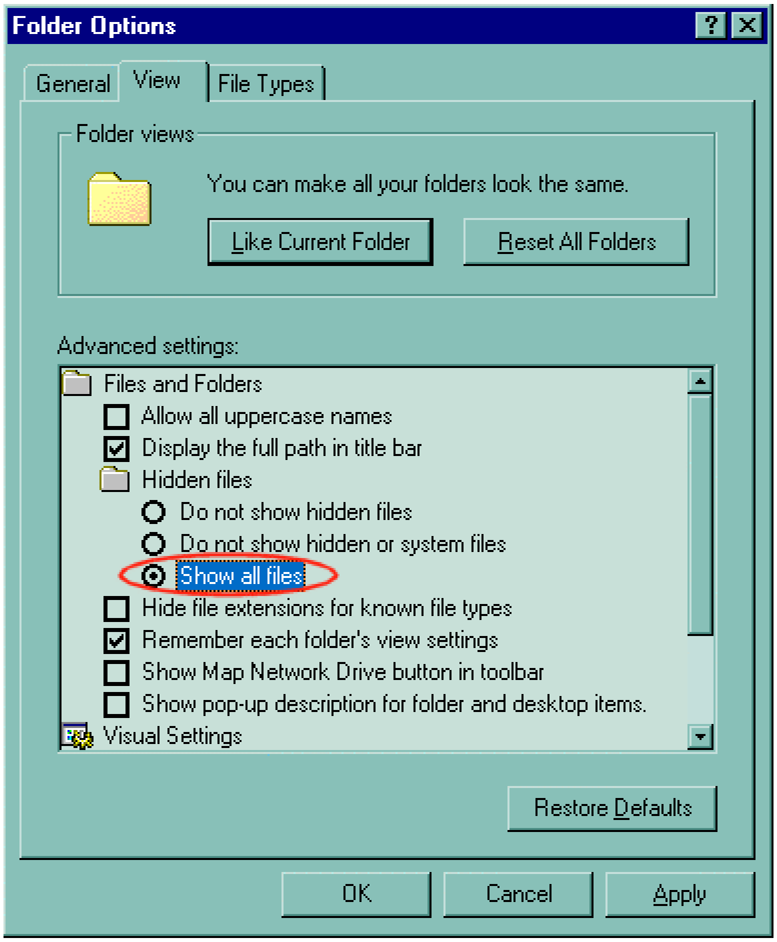
Cannot find the DLL file?

If you cannot find the avisynth.dll file then its very likely that windows is hiding it from you. Windows hides all files that have the DLL extension. in case you accidentally delete any of them and cause windows to become corrupted! To view *all* files you should open windows explorer and go to:

View > Folder Options...



Hit the View tab button and select the option: Show all files



Advanced settings:

Files and Folders
- □ Allow all uppercase names
- ☑ Display the full path in title bar
- Hidden files
  - ○ Do not show hidden files
  - ○ Do not show hidden or system files
  - ● Show all files
- □ Hide file extensions for known file types
- ☑ Remember each folder's view settings
- □ Show Map Network Drive button in toolbar
- □ Show pop-up description for folder and desktop items.

Visual Settings

This will allow you to see the avisynth.dll file and place it in the Windows\system folder :). If you are worried about people accidentally deleting DLL files you can turn the 'hide hidden or system files' option on.

Creating an AVS File

Okay, all the hard work is done! Make a folder on your main drive (usually C:) and put the Vob file you wish to open inside it. I have called my folder vobfiles and the file I'm going to open is called VTS_01_01.Vob.

So open windows notepad (or any text editor) and type the following into it:

```
DirectShowSource("C:\vobfiles\VTS_01_01.Vob")
```

As usual the colours are mine and here to help explain this line means. The DirectShowSource is a command that basically means open anything using Media Players codecs. Then, in brackets and quotation marks, we say what file we wish to open.

In this case I said look for the Vob file in the folder C:\vobfiles and called VTS_01_01.Vob.

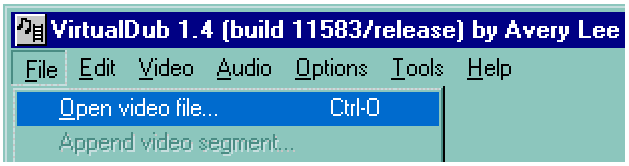
Save it!

Anyway, save the file as whatever you like but put the extension. *.avs on it instead of .txt or .doc etc. I'm calling my text file movie.avs. So when I choose Save in notepad I will type the filename "movie.avs" make sure you use quotation marks (" ") to force notepad to save it as avs instead of txt. Again, it will not work if the text file is called movie.avs.txt or movie.avs.doc or something like that.

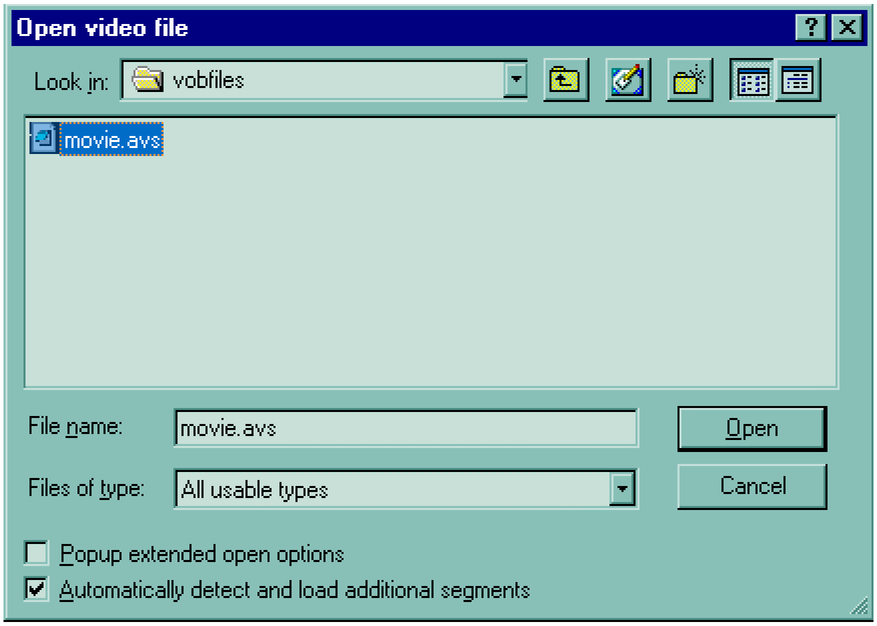
Opening the File in VirtualDub

Okay, time to open our Vob file. Were are going to use VirtualDub, but, as I said before almost any program will open the avs file. Load VirtualDub and go to:

File > Open



Find the avs file you made:



Look in: vobfiles

movie.avs

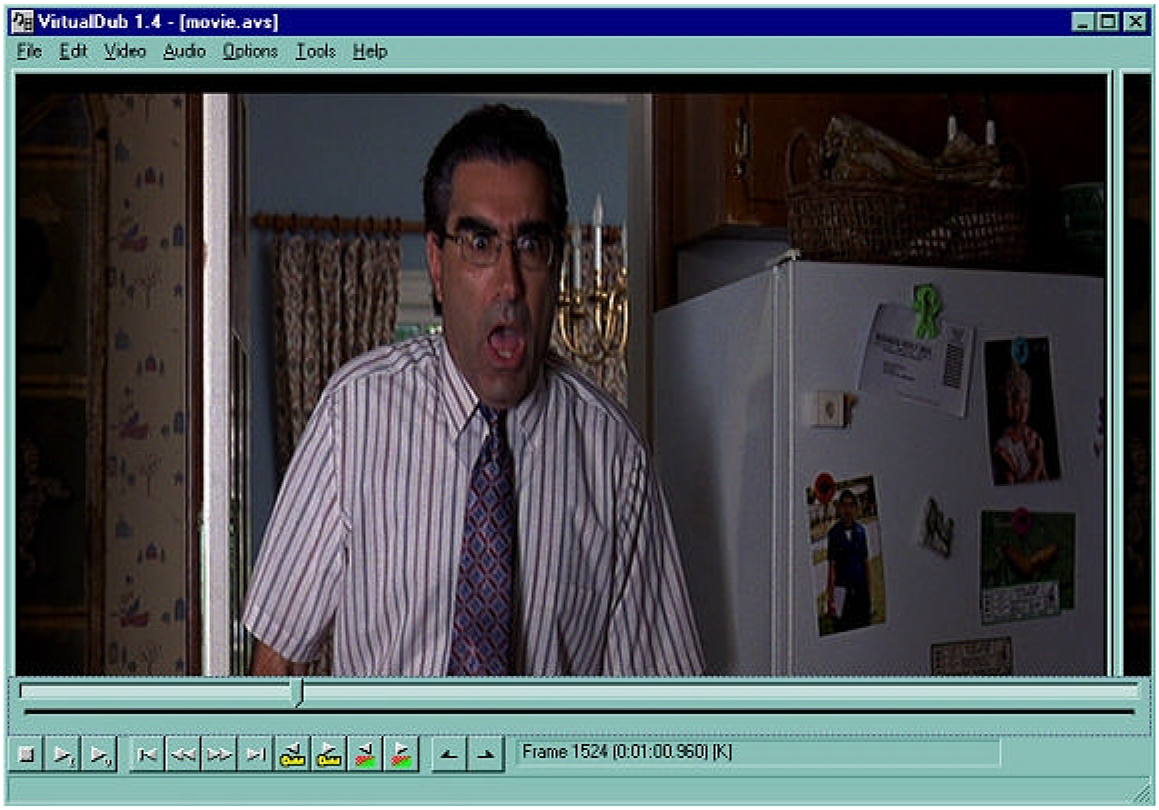
File name: movie.avs

Files of type: All usable types

☐ Popup extended open options

☑ Automatically detect and load additional segments

And this is what you get below - VirtualDub reads it just like it was an AVI file!



Frame 1524 (0:01:00.960) [K]

That's right! Has he discovered how much his son likes apple pie, or has he just learnt about what AVISynth can do? You decide!

AVISynth doesn't offer the ultimate solution for DVD conversion. For one thing it cannot handle subtitles or multiangles directly like Flask Mpeg can. Also, if Media Player cannot open a Vob file or if it cannot play the sound you will not be able to get it in VirtualDub either! All in all VirtualDub may not be as fast or versatile at manipulating Vob files another dedicated program such as Mpeg2avi because it was designed solely for that purpose. Nevertheless, have a play and see what you can come up with :). But don't forget, AVISynth is not just for opening Vob files it can handle almost any file.

Advanced avs Files

That introduction tells us the basics, but there is so much more AVISynth can do with these files *before* they are given to VirtualDub or whatever application you may want to open them in. I will not explain every possibility because that's a mammoth task. Instead, I will explain how to use some of the basic commands and then you will know how to design your own better.

Opening ASF and MOV files

Its not an exact science but you can often open many MOV or ASF files. Both MicroSofts ASF and Apples QuickTime are generic codecs. This means that different codecs are used for both ASF and QuickTime files but they are still called ASF or MOV. What I'm trying to say is you can open some but not others. To open them you usually need to specify the Framerate. You do that like this:

```
DirectShowSource("C:\vobfiles\Video.ASF",fps=29.970)
```

Opening Multiple Files

If you want to open lots of files so they open as one single file like making a mpeg2avi .lst file, all you need to do is add a plus (+) sign and tell it which files to join. For example, instead of this in out text file:

```
DirectShowSource("C:\vobfiles\VTS_01_01.Vob")
```

```
iles\VTS_01_01.Vob+DirectShowSource("VTS_01_02.Vob")+DirectShowSource("VTS_01_03.Vob")+DirectShowSource("
```

That would tell AVISynth to open vob files: 01, 02, 03 & 04 as though it were a single file.

Other File Opening Options

DirectShowSource("*filename&location*"). As we have already seen, this command will open just about any file that Media Player can play. This includes audio files such as WAV, MP3 etc. For example, you could open just about any audio file in an audio editing package such as SoundForge even if SoundForge didn't support that format. Just make an avs text file like we did before and open it in SoundForge the way you did VirtualDub. Unfortunately ASF files do not work correctly and look upside down and ma not have any sound :(.

There is also an experimental Vob opening command called:

VOBVideoSource("*filename&location*")

But this may or may not work for you so you may wish to swap DirectShowSource and VOBVideoSource.

AVISynth also offers other AVI opening commands. The reason for this is because handler may be able to open a certain file type better or process it faster, or may be supported better by the application AVISynth is opening it in.

AVISource("*filename&location***")**

AVIFileSource("*filename&location***")**

OpenDMLSource("*filename&location***")**

So between this lot you should be able to open almost anything.

WAVSource("*filename&location***")**

This command is designed for opening wave files.

Video Editing Commands

Once you've decided how you will open the file you can command AVISynth to do stuff with them beforeit sends them to the application you wish to open it with. This is very useful if you need to crop or resize or something like that but the application you want to open it in doesn't let you. All this crap may sound a little like we are about to learn a programming language, but really its just a matter of listing what things you want done. For example, say we want to open the file but make it half the size it was originally, we could put:

DirectShowSource("*filename&location***")**

ReduceBy2

That's all we need to do! Save the avs text file as usual and open it in VirtualDub to see how it looks. You will find that it opens the video file but at half of its original size!

Okay, lets add to that. Lets say the application you wish to open the file in prefers the RGB AVI format but your original file is in YUY2 format we could put:

DirectShowSource("*filename&location***")**

ReduceBy2

ConvertToRGB

This time it will open the file, half its size and change the format to RGB. Notice also how I have just added this new command to the last one like a shopping list :). Each command can usually be just stuck underneath the last and AVISynth will perform them in that order.

File Command Order

The order in which you ask AVISynth to perform commands can affect both what the final result looks like and how fast the option is completed. For example, say we opened a huge 720 x 576 pixel DVD file. We want to sharpen the image and resize it to 352 x 288 according to a bicubic resize. We could do it like this:

```
AVISource("filename&location")
Sharpen(1.0)
BicubicResize(352,240)
```

But that is a slow way to do it because this way the sharpen filter comes first in the list. Obviously it takes longer to sharpen a 720 x 576 image than it takes to sharpen a 352 x 288 image! So instead its better to do it in this order:

```
AVISource("filename&location")
BicubicResize(352,240)
Sharpen(1.0)
```

This way the image is shrunk first and then sharpened after.

We can also repeat effects. For example, if you wanted to make it twice as sharp you could do this:

```
AVISource("filename&location")
Sharpen(1.0)
Sharpen(1.0)
```

Note: Not all programs will be able to handle every option or some repeated sequences of options so its trial and error. For example, VirtualDub will often crash if you try and feed it a Vob file with multiple AVISynth alterations added it it.

Frames & Fields

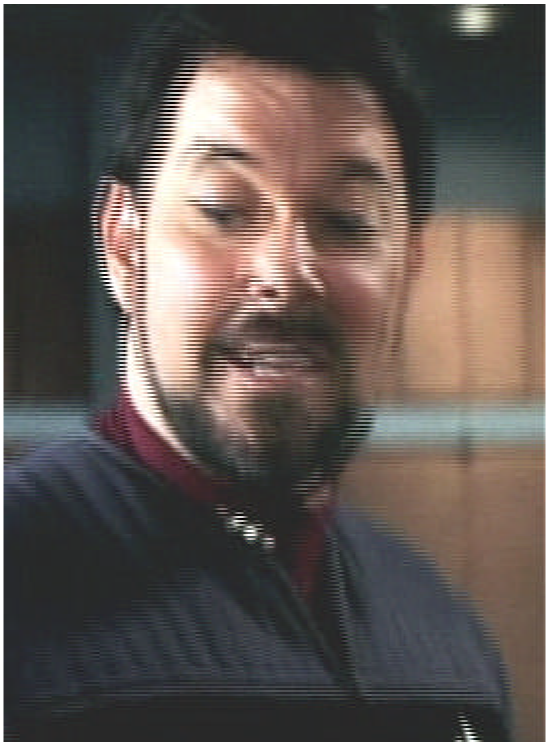
I couldn't finish this article without saying a few words about AVISynth's Field and Framerate commands. So here are a few that should help you get started.

Bob

This is one of the easiest ways to solve those NTSC interlace problems when opening the file in an application that cannot deinterlace it. You could use:

```
DirectShowSource("filename&location")
Bob
```

Simple as that! You may be interested to know that Bob is the method that PowerDVD uses most often to solve those annoying interlace combing problems. This will fix all interlaced files that look corrupted like the following picture:



For more details on interlace problems you could check out my article on '*NTSC / PAL & Interlace*'.

SeparateFields

Another handy little feature that lets you separate the top and bottom frames of an interlaced picture. If for example you will be resizing an video file from 576 pixels high to 288 pixels high or lower an easy way to both speed things up and completely solve all interlace problems without any loss of detail s literally to separate fields. This can be done thusly:

```
DirectShowSource("filename&location")
SeparateFields
```

This will double the amount of frames in the movie because its now giving you frame 1 (top field) frame 2 (bottom field). So a 30fps movie becomes a 60 fps movie. So you may want to use the SelectEven or SelectOdd commands to fix this. Either command will select every other frame in the file. So select odd may just leave you with only bottom or top video fields. To use it just add SelectOdd to the list, like this:

```
DirectShowSource("filename&location")
SeparateFields
SelectOdd
```

This will make the 60 fps back into 30. All these options shouldn't take very much processing power because AVISynth is only taking the information from the file in a certain order, it doesn't need to do heavy calculations on it.

You may be wondering why your movie now looks like a pancake! This is because it only has half the lines it did originally. Like I said you intend on resizing the movie anyway and this solves both interlace problems and cuts down on CPU power because there is of a picture to resize :). To make the final image complete we simply use a BilinearResize option like this:

```
DirectShowSource("filename&location")
SeparateFields
SelectOdd
BilinearResize(384,208)
```

LEARNING MORE COMMANDS

Want to Learn More about AVISynth commands? Here they all are! Just click on each one and it will take you to the info found on the AVISynth's official Webpage:

- Filters that read or produce a video clip
  - AVISource / OpenDMLSource / AVIFileSource / WAVSource
  - SegmentedAVISource
  - DirectShowSource
  - IPCSource
  - Blackness
  - Version
- Filters for changing the frame size
  - HorizontalReduceBy2 / VerticalReduceBy2 / ReduceBy2
  - BilinearResize
  - BicubicResize

  - Crop / CropBottom
  - AddBorders
- Filters for image processing
  - Levels
  - Blur / Sharpen
  - SpatialSoften / TemporalSoften
  - FixLuminance
  - Letterbox
  - Subtitle
  - PeculiarBlend
- Filters for converting between color spaces
  - ConvertToYUY2 / ConvertToRGB
- Filters for dealing with interlaced video
  - AssumeFrameBased / AssumeFieldBased
  - ComplementParity
  - SeparateFields
  - Weave
  - DoubleWeave
  - Bob
  - SwapFields
- Filters for editing
  - Trim
  - AlignedSplice / UnalignedSplice
  - Dissolve
  - Reverse
  - FadeOut / FadeOut2
  - SelectEven / SelectOdd
  - Pulldown
  - Interleave
  - SelectEvery
- Filters for tweaking individual frames
  - FreezeFrame
  - DeleteFrame
  - DuplicateFrame
- Audio-related filters
  - AudioDub
  - DelayAudio
- Debugging
  - Subtract
  - ShowFrameNumber
  - StackHorizontal / StackVertical
  - ShowFiveVersions
- Meta-filters
  - Animate

Editing Video in VirtualDub

As I'm sure you are starting to realize, VirtualDub is one of the easiest to use and yet most useful video editing tools you will ever find. But I still get quite a lot of emails asking how you can crop and resize video files, reduce noise, add logo's or other cool effects with it. This is what this article is about. VirtualDub can do all these things and much, much more! There is no need to explain in detail absolutely everything it can do, because once you've read through this guide you will get the idea and be able to apply that knowledge to everything you want to do.

Video Editing Formats

If you are going to edit and/or compress a video using VirtualDub or any video editing package then the better the original video the better the final result will be. If you are capturing from a TV Tuner or Video capture card use the highest settings you can without going overboard. For example, there is probably no need to capture a video at 640 x 480 pixels if you are going to resize it to 352 x 240 anyway because not only will this take a lot of time to do but it will take twice the hard drive space too. Whenever possible you should capture with CD quality audio (16 bit 44100Hz) and at the TV standard framerate (25fps for PAL TV and 29.97fps NTSC). Usually capturing to MJPEG will produce the best results and will be the easiest format to edit. For more details on video capture check out my capture guide Here.

Compressing a Compressed File

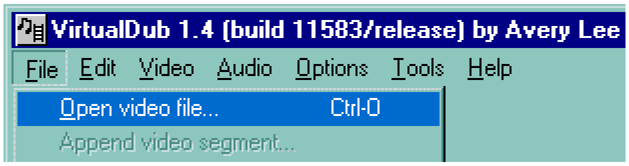
Why not just compress straight to Mpeg or DivX I hear you cry!? Well, you can, but don't forget that if you are going to resize or apply any filters to a video file it *must* be recompressed again! This is not VirtualDub's fault, every existing video editing package is the same! For example, its no good compressing a DVD video into DivX and *then* opening it up to resize it or crop it because to resave that file you will need to recompress it to DivX again. As you can imagine if the video is long then this could take a long time. And since all formats I know (except Huffyuv and uncompressed AVI) compress by discarding fine details, each time you recompress the video some quality will be lost! As you can see, it makes sense to make sure that the initial file as good as we can get it.

Before I start you will need:

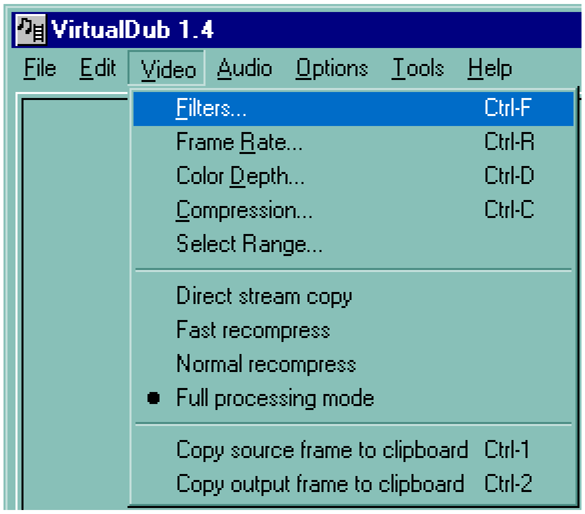
VirtualDub 1.4c (or greater)

CROPPING & RESIZING VIDEO's

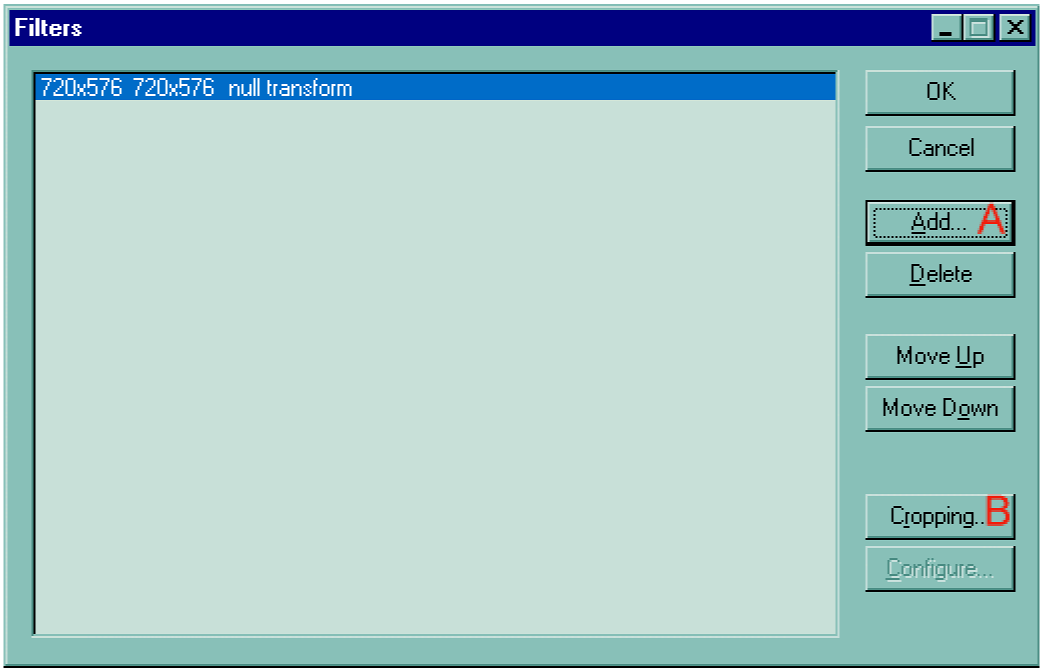
Open your movie file in VirtualDub by going to File > Open video file...


VirtualDub 1.4 (build 11583/release) by Avery Lee
File
Edit
Video
Audio
Options
Tools
Help
Open video file... Ctrl-O
Append video segment...

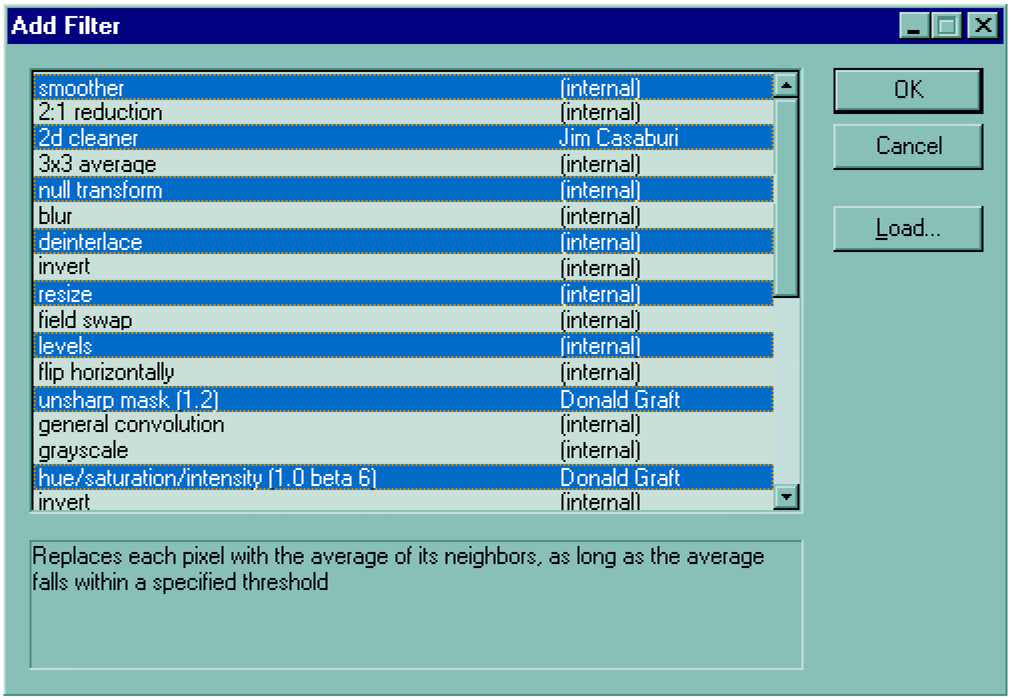
Select File > Filters



Press the Add... button (**A**)



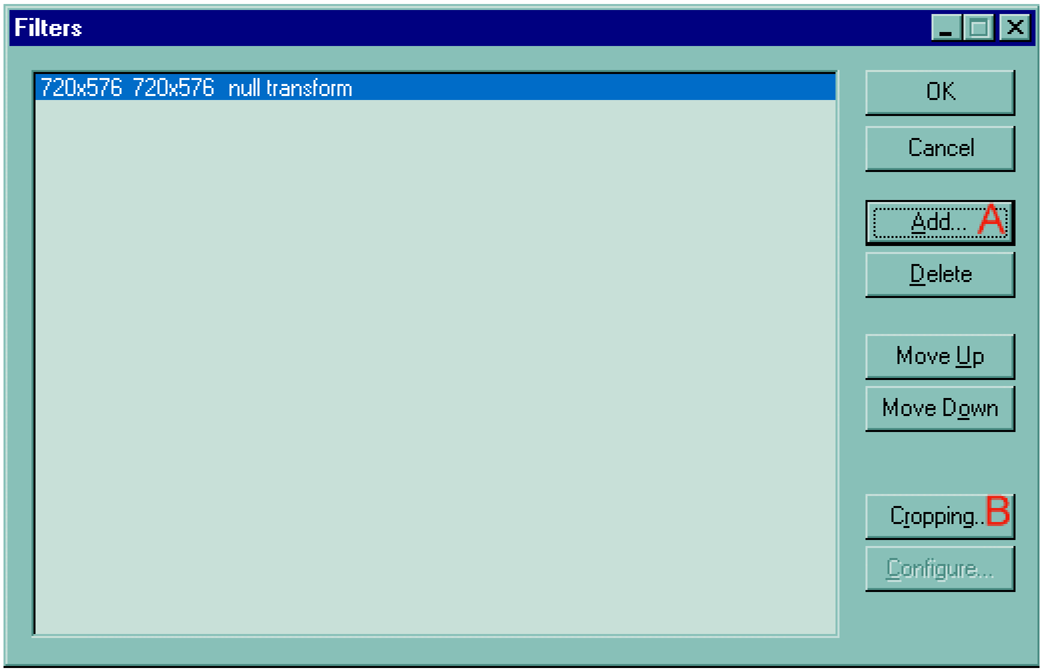
Up pops a list of all the filters that can be applied to a video file. Choose the one called: Null Transform.



smoother	(internal)
2:1 reduction	(internal)
2d cleaner	Jim Casaburi
3x3 average	(internal)
null transform	(internal)
blur	(internal)
deinterlace	(internal)
invert	(internal)
resize	(internal)
field swap	(internal)
levels	(internal)
flip horizontally	(internal)
unsharp mask (1.2)	Donald Graft
general convolution	(internal)
grayscale	(internal)
hue/saturation/intensity (1.0 beta 6)	Donald Graft
invert	(internal)

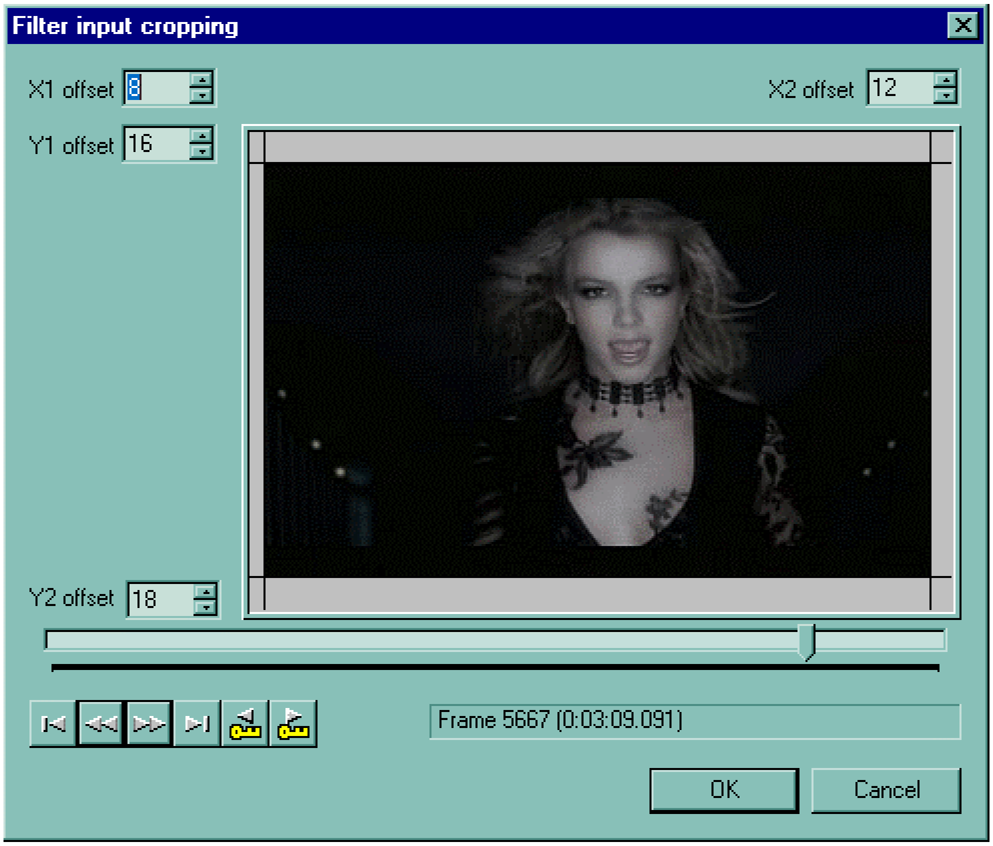
Replaces each pixel with the average of its neighbors, as long as the average falls within a specified threshold

Yes, I know, a weird name that 'null transform' isn't it. But press OK and we're back to this box. Press the Cropping... button (**B**)



720x576 720x576 null transform

Up pops the cropping box (below). Use the slider bar and frame movement buttons to find a good part in the video where you can see clearly where you need to crop. Then change the numbers any of the four number boxes X1, X2, Y1 or Y2. Each box represents one of the four sides of the box. Simply increase or decrease the amount needed and you will see it crop before your eyes :)



X1 offset 8

X2 offset 12

Y1 offset 16

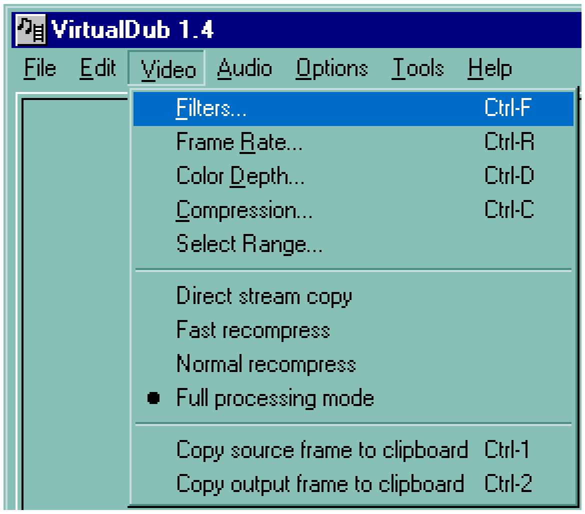
Y2 offset 18

Frame 5667 (0:03:09.091)

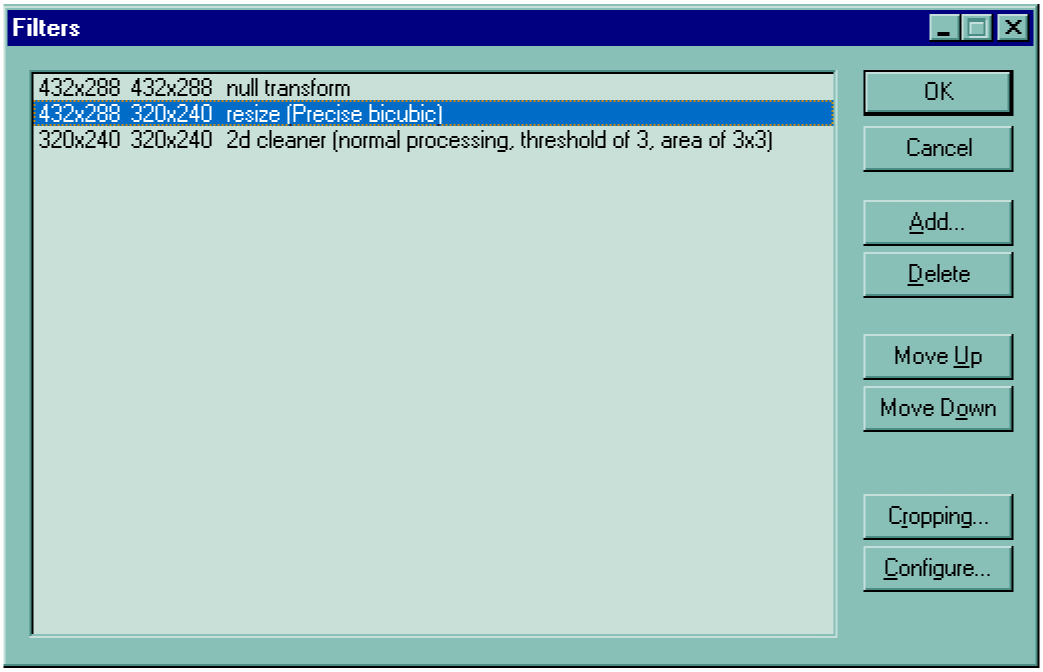
When you are done press OK.

That's it! When you resave the video in VirtualDub the newly saved video will be cropped. But don't forget that you need to recompress again when you save it or it won't work. Cropping is especially useful for getting rid of any hum-bars from the edges of the screen that usually appear with TV Tuner captures.

Resizing is just as easy, go to: File > Filters

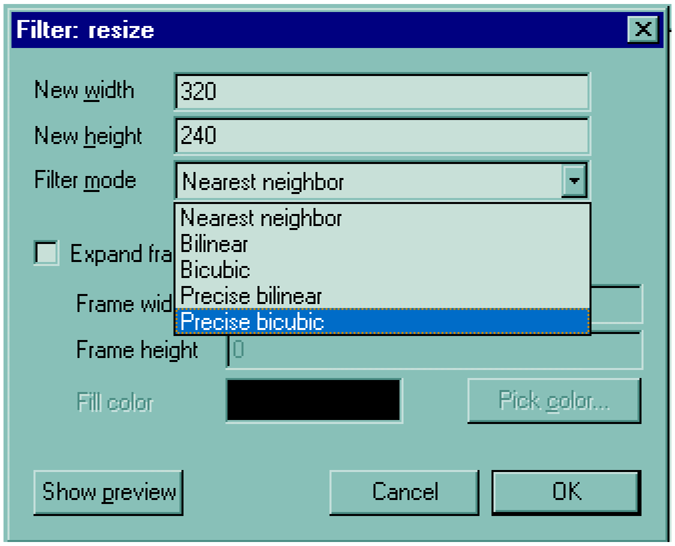


This time press Add... and select the resize filter.



```
432x288  432x288  null transform
432x288  320x240  resize (Precise bicubic)
320x240  320x240  2d cleaner (normal processing, threshold of 3, area of 3x3)
```

Up will pop this box. Just type in the new width and the new height. Then select Precise bicubic from the filter mode. Bilinear is best for shrinking images and bicubic is best for enlarging. Nearest neighbour is the fastest but gives lowest quality for everything but simple shapes.



New width: 320

New height: 240

Filter mode: Nearest neighbor

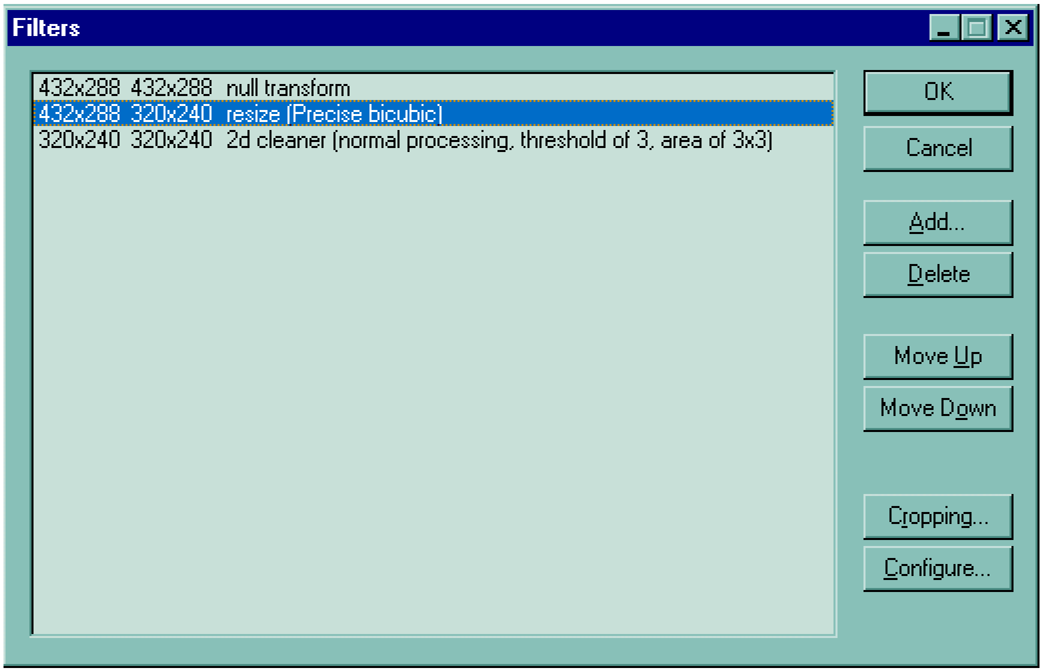
Frame height: 0

You can choose the show preview button too to see how it looks. It is important to remember that, if you intend on compressing the video into Divx (or in fact many other formats), your final movie sizes should be multiples of 16 pixels! Sometimes they only need to be multiples of 8 pixels and sometimes (very rarely) they need to be multiples of

32 pixels. As an example, a 352 x 288 video will compress into Divx without problems, but a 351 x 288 will not!

Filter Order

VirtualDub filters are executed in a specific order. The one at the very top will be executed first and the one at the bottom of the list will be executed last.



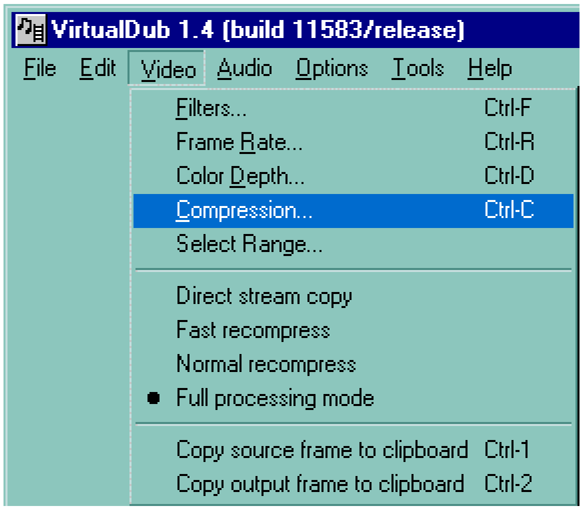
```
432x288  432x288  null transform
432x288  320x240  resize (Precise bicubic)
320x240  320x240  2d cleaner (normal processing, threshold of 3, area of 3x3)
```

It may not seem to matter what order they effect the video, but it *is* an important point. Lets say, for example, you wanted to resize a video from 704 x 576 to 352 x 288, but you also wanted to apply filters to sharpen the image. If you put the sharpen filter first in the list it would take twice as long to convert when you tried to save it! This is obvious because it takes twice as long to sharpen a picture twice the size. It would be far quicker to sharpen the image *after* it had been resized smaller. Because of this, VirtualDub has two buttons called 'Move up' and 'Move down' that allow us to move the order of the filters up and down.

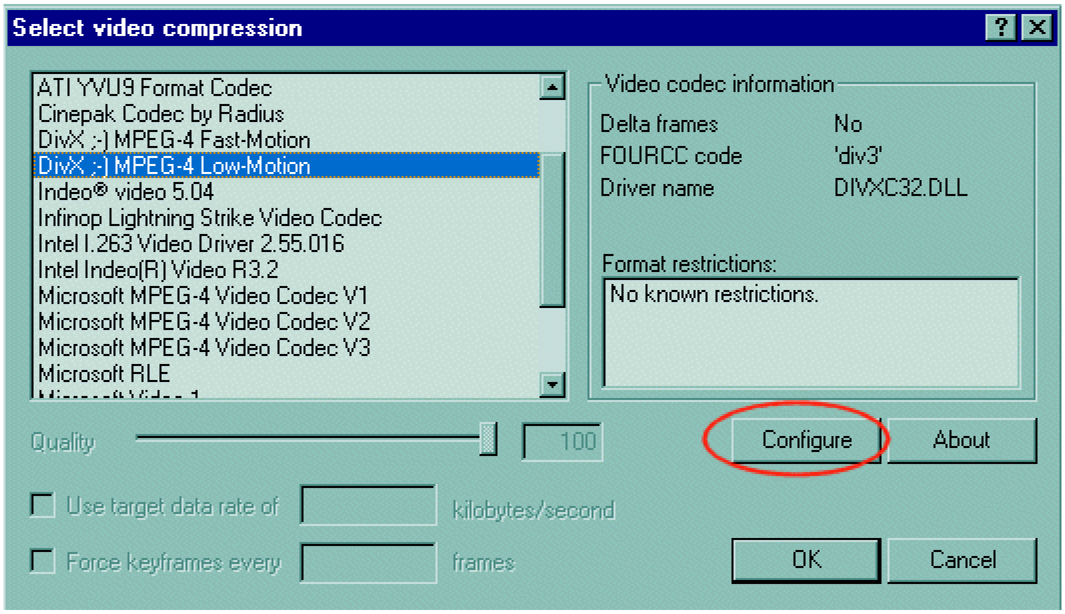
PROCESSING IT

When you are finished all the filters will be applied to the video file you save. Now you must choose the final format you want the video to be. You can, of course, use any video format VirtualDub offers, but just to illustrate, if you wanted to make it to Divx this is how you'd do it.

Go to: 'Video > Compression...'



Up pops the following box below. Choose Mpeg-4 Low-Motion and hit the configure button (highlighted in red).


Select video compression
?
ATI YVU9 Format Codec
Cinepak Codec by Radius
DivX ;-) MPEG-4 Fast-Motion
DivX ;-) MPEG-4 Low-Motion
Indeo® video 5.04
Infinop Lightning Strike Video Codec
Intel I.263 Video Driver 2.55.016
Intel Indeo(R) Video R3.2
Microsoft MPEG-4 Video Codec V1
Microsoft MPEG-4 Video Codec V2
Microsoft MPEG-4 Video Codec V3
Microsoft RLE
Video codec information
Configure
About
OK
Cancel

Delta frames No
FOURCC code 'div3'
Driver name DIVXC32.DLL

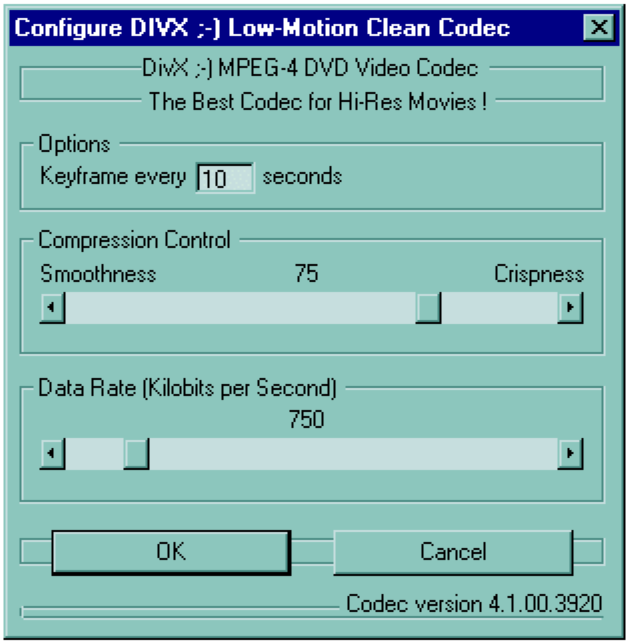
Format restrictions:
No known restrictions.

Quality 100

□ Use target data rate of kilobytes/second

□ Force keyframes every frames

Then up pops the codec options. Use one of the bitrate calculators that can be downloaded from Digital Digest to make sure the movie will fit into the size you need. 650kbps will usually fit *any* movie on a single CD, but you can usually get away with between 900 and 1000kbps for a 352 x 240 VCD movie for better quality see my quality guide if you are unsure about correct bitrate settings.



Configure DIVX ;-) Low-Motion Clean Codec

DivX ;-) MPEG-4 DVD Video Codec
The Best Codec for Hi-Res Movies !

Options
Keyframe every 10 seconds

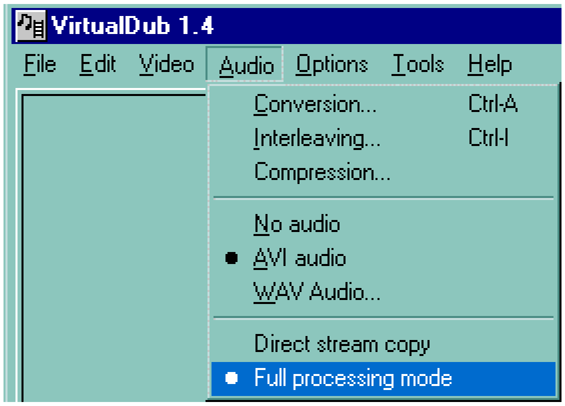
Compression Control
Smoothness 75 Crispness

Data Rate (Kilobits per Second)
750

Codec version 4.1.00.3920

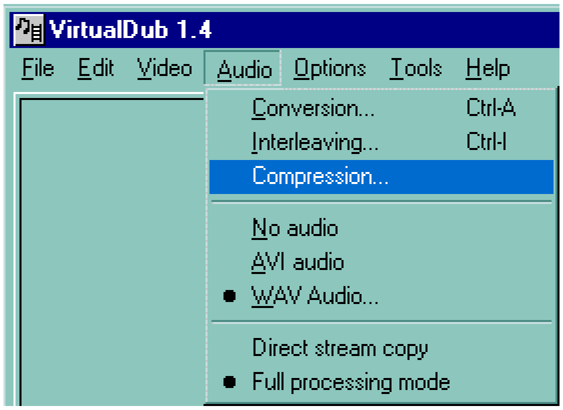
Audio Compression

Now select the audio settings by going to: Audio > Full processing mode (so the black dot appears by it).

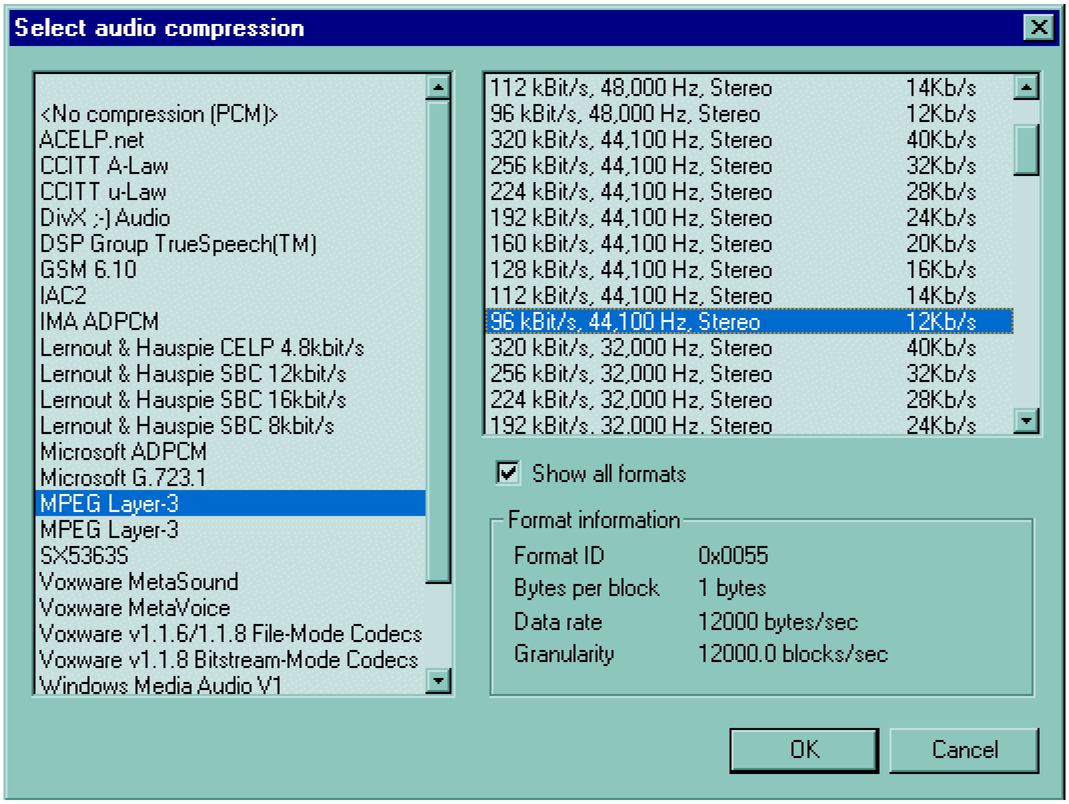


VirtualDub 1.4

Then Audio > Compression.



Up pops this box below. Choose Mpeg Layer 3 or Divx ;-) Audio 64kbps, 441 kHz, stereo. And press OK.



<No compression (PCM)>
ACELP.net
CCITT A-Law
CCITT u-Law
DivX ;-) Audio
DSP Group TrueSpeech(TM)
GSM 6.10
IAC2
IMA ADPCM
Lernout & Hauspie CELP 4.8kbit/s
Lernout & Hauspie SBC 12kbit/s
Lernout & Hauspie SBC 16kbit/s
Lernout & Hauspie SBC 8kbit/s
Microsoft ADPCM
Microsoft G.723.1
MPEG Layer-3
MPEG Layer-3
SX53633S
Voxware MetaSound
Voxware MetaVoice
Voxware v1.1.6/1.1.8 File-Mode Codecs
Voxware v1.1.8 Bitstream-Mode Codecs
Windows Media Audio V1

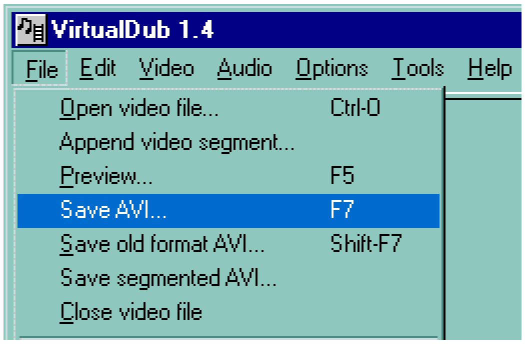
112 kBit/s, 48,000 Hz, Stereo	14Kb/s
96 kBit/s, 48,000 Hz, Stereo	12Kb/s
320 kBit/s, 44,100 Hz, Stereo	40Kb/s
256 kBit/s, 44,100 Hz, Stereo	32Kb/s
224 kBit/s, 44,100 Hz, Stereo	28Kb/s
192 kBit/s, 44,100 Hz, Stereo	24Kb/s
160 kBit/s, 44,100 Hz, Stereo	20Kb/s
128 kBit/s, 44,100 Hz, Stereo	16Kb/s
112 kBit/s, 44,100 Hz, Stereo	14Kb/s
96 kBit/s, 44,100 Hz, Stereo	12Kb/s
320 kBit/s, 32,000 Hz, Stereo	40Kb/s
256 kBit/s, 32,000 Hz, Stereo	32Kb/s
224 kBit/s, 32,000 Hz, Stereo	28Kb/s
192 kBit/s, 32,000 Hz, Stereo	24Kb/s

☑ Show all formats

Format information

Format ID	0x0055
Bytes per block	1 bytes
Data rate	12000 bytes/sec
Granularity	12000.0 blocks/sec

That's it! Save AVI...



Okay that's it for this one. Check out part two and I'll try and explain a few more tricks =).

More Filter Tricks:

Cleaning Up Video With VirtualDub Filters

The amount of filters available for VirtualDub is growing fast. If you want to see the latest selection get 'em Here. To install new filters is really easy, just download, unzip, and put them inside the the VirtualDub folder called 'Plugins'. Then the next time you run VirtualDub they will appear. And just to recap, to use the filters we go to: File > Filters. If you haven't read my last article on Filters go Here.

Lets face it most video capture has at least some noise or sharpness problems unless we are using very expensive equipment or have a very good quality source to record from. This is why VirtualDub has a selection of useful filters designed to correct bad colour, brightness, saturation, noise, smoothness and a whole bunch of other things. So here I will give you a few tips on some of the filters that I use and how they can be applied effectively.

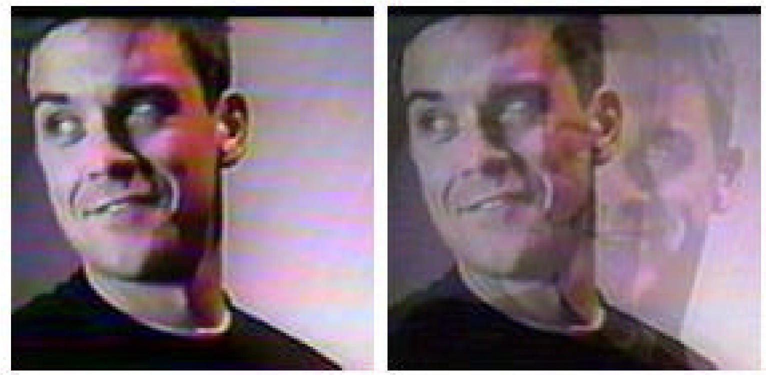
SUBTLE NOISE REDUCTION

Ideally we want to do as little to the original video as is needed to make it look good. Its very easy to go overboard and stick five or more powerful filters in the hope that the final result will look great. In actual fact the least changes necessary will usually results in the best results. Using filters seriously slows down compressing time so I suppose this will help to limit your use of them a little too.

Don't expect any of these subtle noise reduction filters to work miracles because they wont. Noise reduction without loss of detail is almost impossible because the original information is not there. There are another selection of what I like to call 'Hard Core' filters that will do some amazing things but these work by making large changes to the original image. But it is possible to use these in conjunction with the more subtle ones to achieve a good balance.

Using the Dynamic Noise Reduction Filter

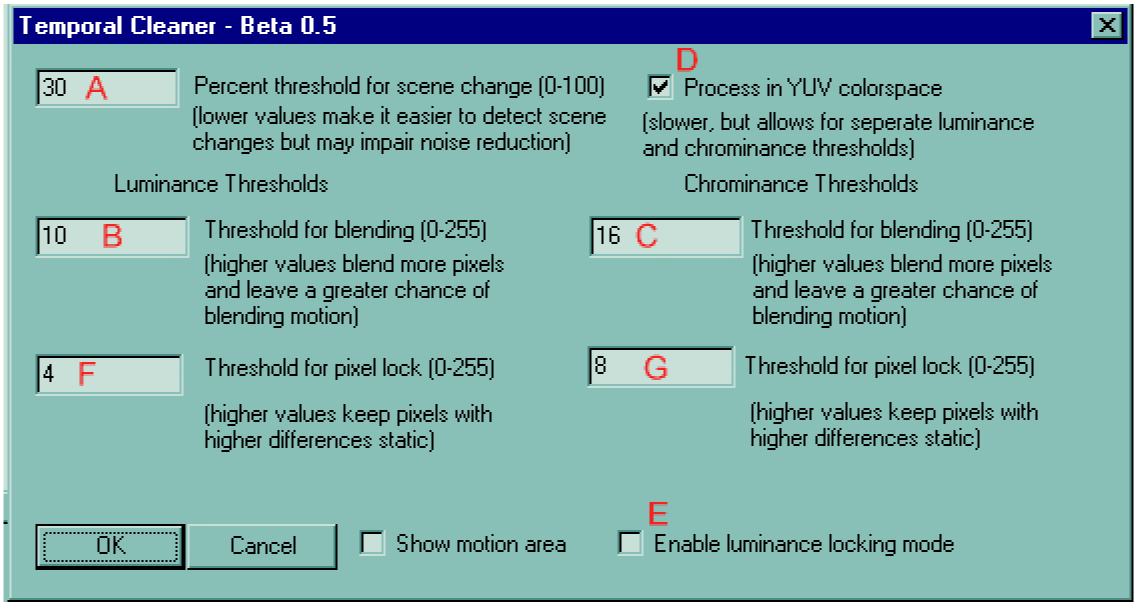
Most noise reduction filters work on the idea that the same noise disruption will not appear in two or more frames in exactly the same place. Their solution is to compare the first frame to the second frame in the sequence and get rid of minor changes. This is basically what the Dynamic Noise Reduction filter does. It doesn't need much explanation, really, because it only has one slider bar. The higher you set it the more it will reduce subtle noise errors found in your video. The only problem with this is the higher you set it the more it will tend to merge two frames together. This produces an annoying ghosting effect like this:



The one on the left is the original. The one on the right has ghosting because of the high level of noise reduction!

Using the Temporal Cleaner Filter

The Temporal Cleaner is a bit better than the Dynamic in some ways although its based on the same idea. The reason it is slightly better is mainly because it is designed to only copy across parts of the image that don't move at all. This helps greatly to reduce the ghosting effect. The down side is controls are more confusing and its easy to get weird effects from setting them.



Temporal Cleaner - Beta 0.5

30 — Percent threshold for scene change (0-100) (lower values make it easier to detect scene changes but may impair noise reduction)

☑ Process in YUV colorspace (slower, but allows for seperate luminance and chrominance thresholds)

Luminance Thresholds

10 — Threshold for blending (0-255) (higher values blend more pixels and leave a greater chance of blending motion)

4 — Threshold for pixel lock (0-255) (higher values keep pixels with higher differences static)

Chrominance Thresholds

16 — Threshold for blending (0-255) (higher values blend more pixels and leave a greater chance of blending motion)

8 — Threshold for pixel lock (0-255) (higher values keep pixels with higher differences static)

□ Show motion area

□ Enable luminance locking mode

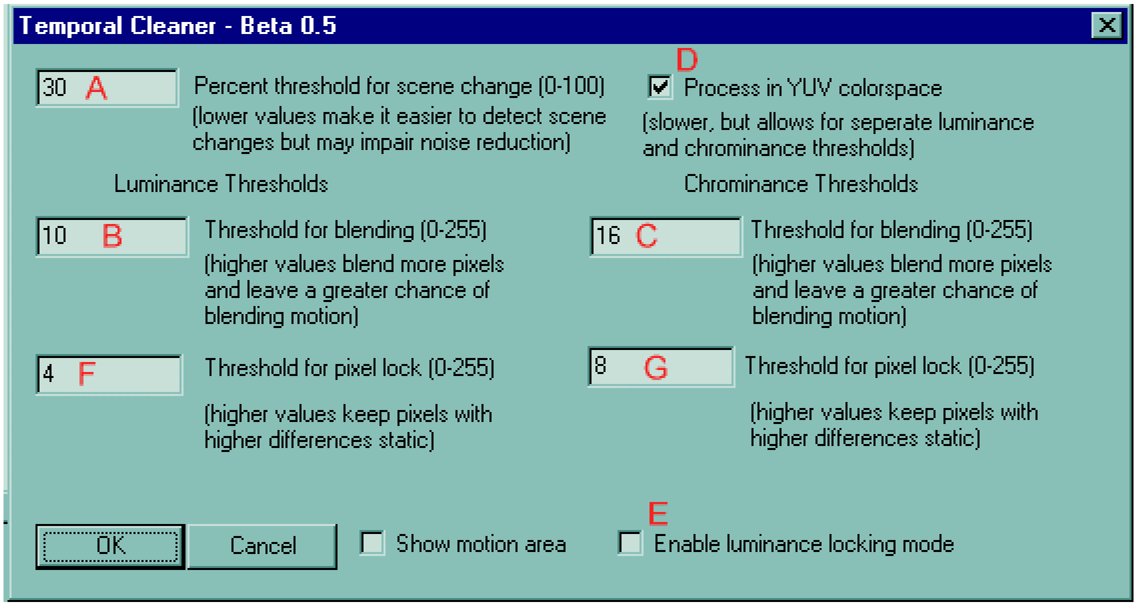
A threshold is how different one part of the picture is to the one next to it. If you are used to using the 'magic wand' option in Adobe Photoshop or Paint Shop Pro then you should have a good idea of what this means for a picture. Imagine a light colour and a dark colour ten times darker than it. If we set a threshold of 10 then both colours would be considered as exactly the same. If the threshold was set to 9 or below then the colours would be considered as not the same. The average threshold of any high colour image is about 35 pixels which is Adobe's default setting for Photoshop.

(**A**) This defines how different whole one frame is to the one next to it. It describes the whole frame and not just parts of it. If it decides one frame is totally different from the next it restarts the comparison between it and the next frame. This is designed as a safety function to prevent parts of one scene appearing in the next. In other words if its too high you will get some ghosting effects, if its too low it will lessen the amount of noise it can delete.

(**D**) Process in YUV colorspace allows you to set both the brightness thresholds (luminance) and colour threshold (chrominance) more exactly. Basically you should check this box for best quality and uncheck it if speed is more important.

The settings are now split into two parts. Brightness thresholds are listed on the left hand side of the GUI and Colour thresholds are described on the right hand side.

(**B**) Luminance Threshold. This can be any number from 0-255. Luminance basically means brightness. Just imagine the picture was only black and white. In that case this threshold ignores all colour decides just how different the shadows are between one frame and the next.



Temporal Cleaner - Beta 0.5

30 — Percent threshold for scene change (0-100) (lower values make it easier to detect scene changes but may impair noise reduction)

☑ Process in YUV colorspace (slower, but allows for seperate luminance and chrominance thresholds)

Luminance Thresholds

10 — Threshold for blending (0-255) (higher values blend more pixels and leave a greater chance of blending motion)

4 — Threshold for pixel lock (0-255) (higher values keep pixels with higher differences static)

Chrominance Thresholds

16 — Threshold for blending (0-255) (higher values blend more pixels and leave a greater chance of blending motion)

8 — Threshold for pixel lock (0-255) (higher values keep pixels with higher differences static)

☐ Show motion area

☐ Enable luminance locking mode

(**F**) Brightness pixel lock feature. This keeps pixels that are almost the same brightness between two or more frames locked exactly the same position for as long as it can. This setting should usually be very low because you will start to see a spray of dots where things move from one frame to the next.

(**C**) Colour threshold setting. This refers to how deferent colours are from each other between two or more frames. The human eye is less sensitive to colour settings than it is to brightness so these can be made higher thresholds than the same brightness settings on the left. This seems to be the main reason the default settings are higher.

(**G**) Colour pixel lock feature. This keeps pixels that are almost the same colour between two or more frames locked exactly the same position for as long as it can. This setting should usually be very low because you will start to see a spray of dots where things move from one frame to the next.

(**E**) Allow to lock luminance. This is a kind of add on feature that may not help much with noise reduction. It compares the chrominance threshold to the luminance threshold on any locked pixels and tries to make a guess on what should be kept or not.

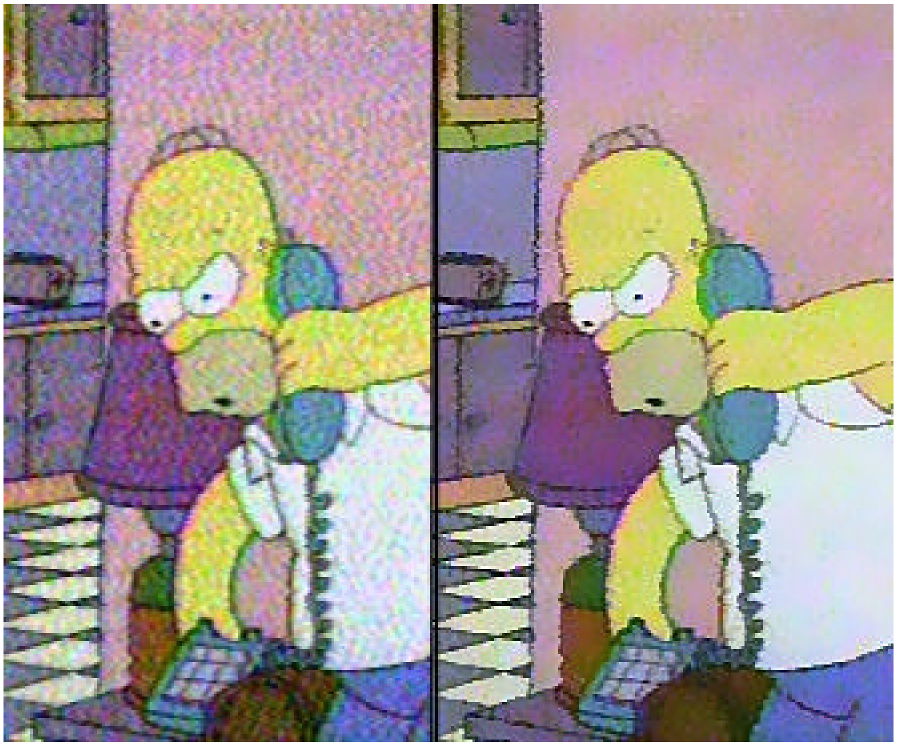
Well that's about it! The show motion area is not a noise reduction option. It is only there to show you what area of the picture is being changed by the noise reduction. Unchanged parts will appear as blue. Moving parts will be black. Gray shows brightness locked pixels. Usually it produces one big mess but it may help you to see what is going on with the setting you are using.

Phew! After all that explaining you'd expect some big results from this thing! But, again, I'm sorry to say that no noise reduction can truly reduce noise because noise has become part of the picture. You can no more tell a computer to remove the noise dots then you can tell it to remove an eye from a person.

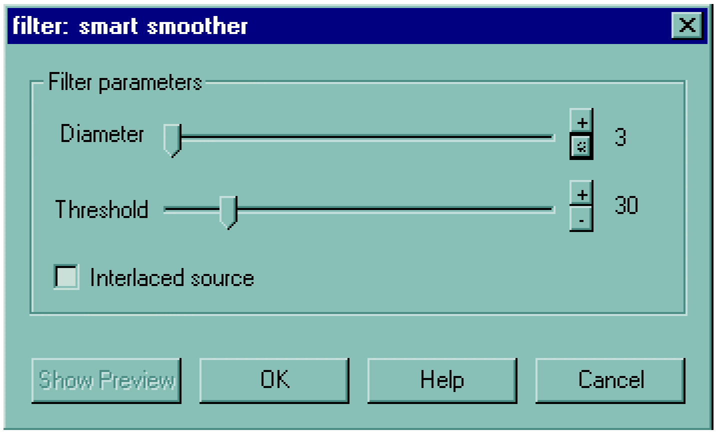
HARD CORE NOISE REDUCTION

Using the Smart Smoother Filter

Take a look at the picture on the left. I captured it when the aerial was not tuned in correctly to simulate a lot of noise. Then I used the Smart Smoother filter (Diameter 11; Threshold 200) to clean it up. Its not perfect, but I'm sure you will agree that the noise is a lot less.



The filter work's with two settings the diameter and threshold. The Diameter is how many pixels it effects at a time. The more you use the more blurry the image will appear. And the Threshold is a check to see how different one colour is from the next. A lower threshold is quite good for cartoons because it tends to make solid blocks of colour. As always experimentation is he best method.



filter: smart smoother

Filter parameters

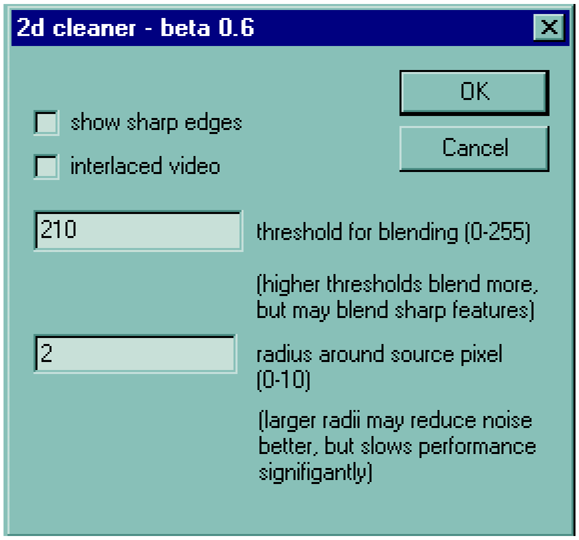
Diameter 3

Threshold 30

□ Interlaced source

Using the 2D Cleaner Filter

This is one of my favorite filters because it can be very subtle and yet powerful at the same time. Its basically very similar to the previous Smart Smoother in operation but after a while of using I think you will prefer it for most stuff.



□ show sharp edges

□ interlaced video

210 threshold for blending (0-255)

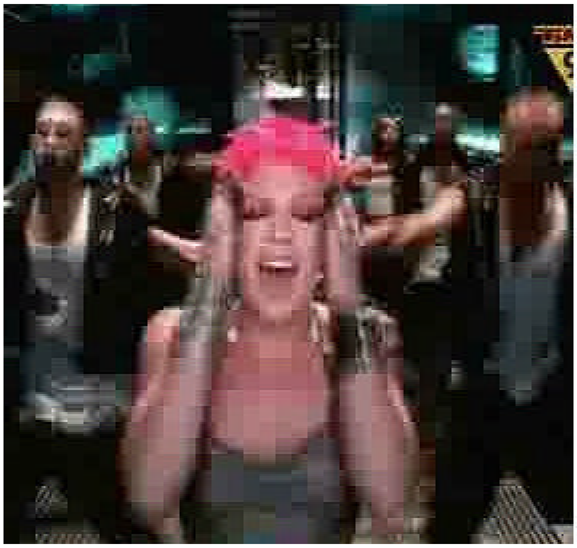
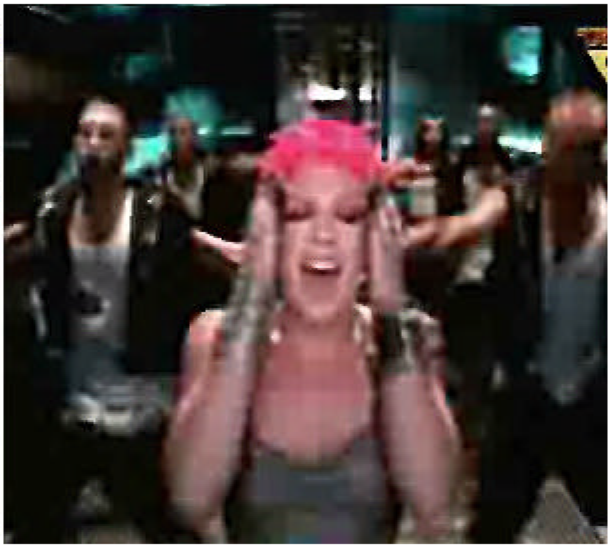
(higher thresholds blend more, but may blend sharp features)

2 radius around source pixel (0-10)

(larger radii may reduce noise better, but slows performance signifigantly)

All these filters are basically selective blurring filters. They allow you to control how much of the picture is blurred and how much of it is kept sharp. For most images a radius of 2 enough, in fact a radius of 1 may be better if the noise is not too bad. The higher the radius the longer it will take to reduce noise and the more blurry the overall picture will be. The threshold doesn't exactly say how blurry the picture will be but defines how sharp fine details will be. In laymen's terms this means that the higher the threshold the smoother the look at the expense of fine detail, but the lower the threshold the more 'plastic' the picture will appear. The first thought is to use a threshold setting that is lowest so as to keep the sharpest edges, but this results in a plastic cartoon-like effect. I suggest you start at in the hundreds. I usually start at 150 and work my way down a little until it looks good. But again, this is all dependent on the picture you may need to go even higher or much lower.

The picture below is a good example of how it works. Getting rid of macroblocks is almost impossible. So I have encoded a video clip with very low bitrate to get some macroblocks and then I used the 2D Cleaner (Radius 2; Threshold 90) to try and delete them. As you can see, the 2D cleaner is able to blur out the small block-like areas but still keep a fair amount of detail.

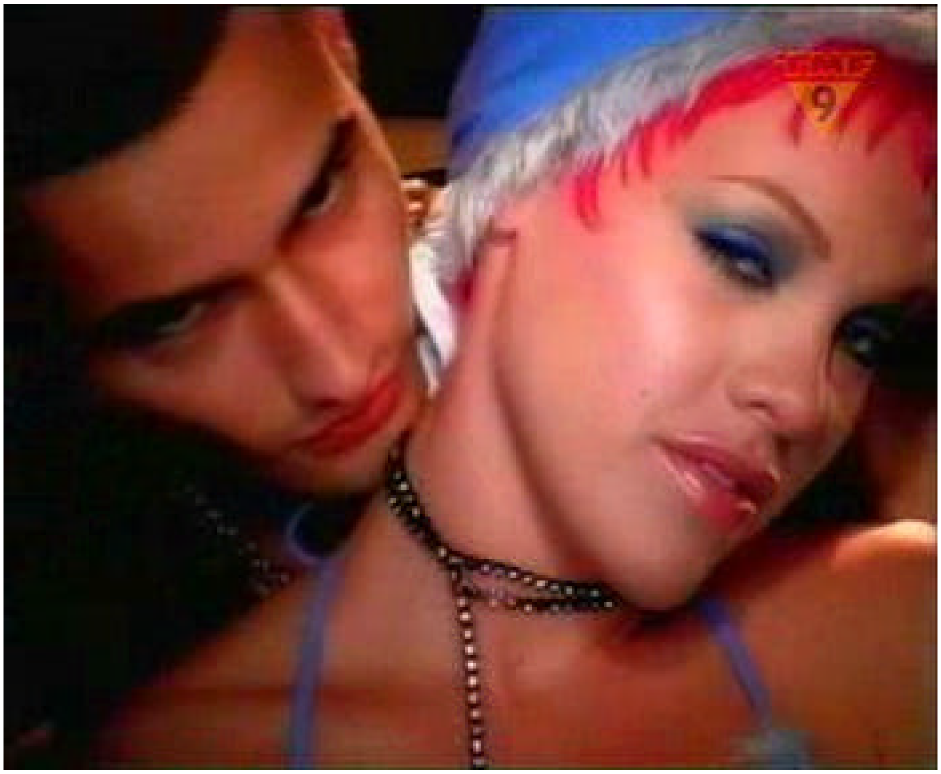


HUE, SATURATION, LEVELS & FOCUS

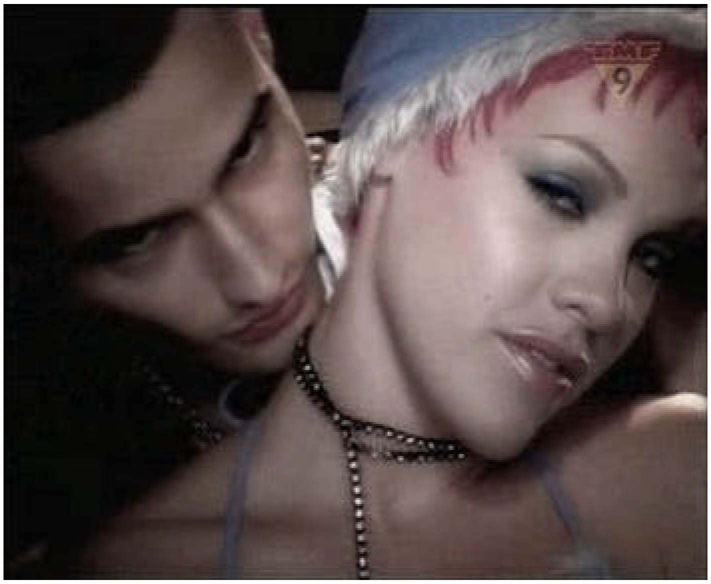
Saturation

Saturation is the strength of a the colours in a video. If you have too much colour (as is very often the case with video capture cards) then you may need to turn the saturation down. A completely non-saturated image is gray and a totally saturated image is almost solid blocks of colour.

This first image is deceptive because it still looks quite nice (I'm talking about the colour not the girl...well, her too :). Anyhoo, people are naturally drawn to highly coloured images and it gives over a dreamy effect. Nevertheless, there is far too much colour in this image and it is not at all true to the original video it was taken from. To fix it you must use the Hue / Saturation / Intensity filter. Just turn the saturation down a little and it will look okay.



This next image has far too little colour in it and is starting to look almost black and white. This is a more obvious problem and the saturation needs to be turned up a little this time - be careful not to overdo it though.



This final image has a good balanced saturation and matches the original video well. Provided you have a good video and TV, its always a good idea to view the captured image alongside the original so you can be more objective about the true colours.

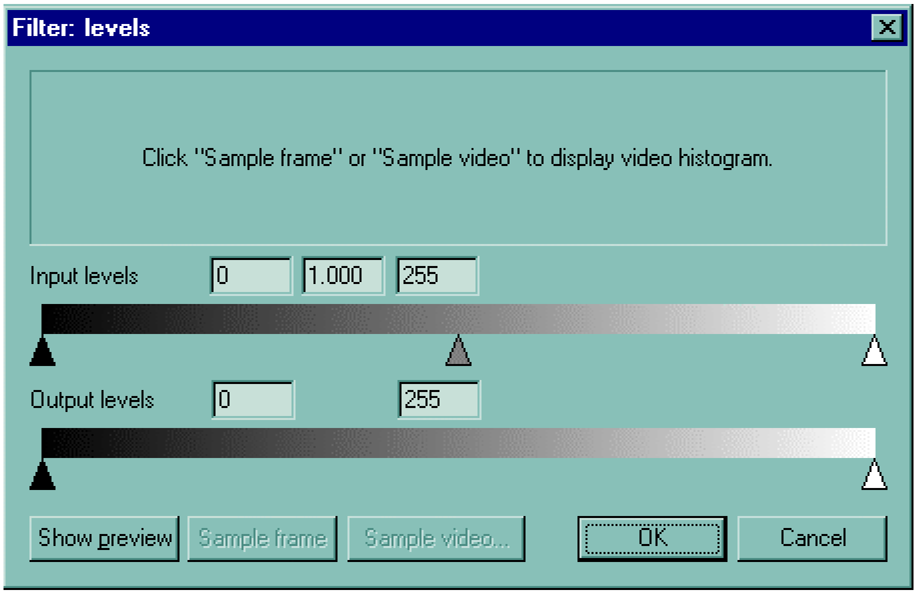
Hue

Hue is basically the overall colour tendency. You rarely need to adjust the hue of a video but just to keep it complete I'll explain a bit about it. Sometimes a image will be a little more red than it should be or a little more green etc. If you notice that this is the case then I suggest you tweak the hue a little using the Hue / Saturation / Intensity filter again. Usually you will only need to adjust it by a very very small amount. Its really just a matter of looking and seeing what is right. Forget about the intensity option with the Hue / Saturation / Intensity filter because the levels filter is more exact in this area.

Levels

Take a look at the picture below, no, don't adjust your computer screens its meant to look like that :). A photo can be described as being made up of three parts shadows, midtones and highlights. I have simplified the picture below to show them better.

Forget colour for a second, the level control only effect the brightness and grey tones of an image. Lets say a black and white photo is only 255 tones. Tone 0 is pure black and 255 is pure white. Now look at the two bars below:



Filter: levels

Click "Sample frame" or "Sample video" to display video histogram.

Input levels 0 1.000 255

Output levels 0 255

These bars represent the brightness of the picture from left being darkest to the right as the lightest. The bottom 'output levels' bar represents the *overall* colour value, that is, how dark 0 starts at and how light 255 will finish. Remember that 255 represents the brightest colour and 0 the darkest. So if 0 actually starts as a gray then the overall picture will look much lighter!

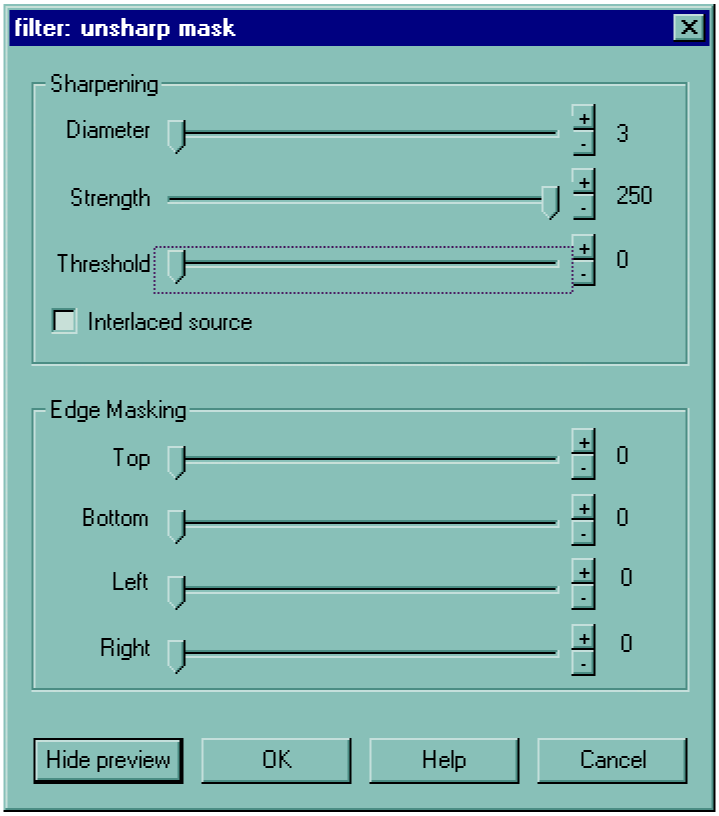
If we move the black arrow of the bottom left bar to the right, it will make all the dark parts brighter. This gives almost exactly the same effect you would get from turning the brightness up (but without increasing the intensity). Conversely, the white arrow on the bottom right will move the bright values down. The main difference between this and adjusting the brightness is that the contrast (the ratio between how much dark and how much light exists) is not effected - it just 'washes' the image darker or lighter.

Next the 'input levels' bar above still represents all the tone values in the image from 0-255 as the one below, but this time it lets us change the ratios between them. For example, move the top left arrow to the right and you will increase the shadows in the picture. If you move the far-right arrow left you will increase the highlights. The center arrow represents the ratio between the two. If we move it to the left it will give us much less shadows, making the picture appear more flat. If we move it to the right it will give much more shadows, making the picture appear hard and defined. The default value is 1.0.

As I'm sure you already realise, this filter is more powerful than just changing the brightness and contrast of a picture and gives you a lot of control to fine tune your image and get the best effect.

Using the Unsharp Mask Filter

To sharpen or blur an video using a sharpen or blur filter is easy enough to do, just set the amounts. But if you really want control over how your image is sharpened then you cannot beat the unsharp mask filter. It has a strange name but it lets you control what areas are sharpened better.



Diameter 3

Strength 250

Threshold 0

☐ Interlaced source

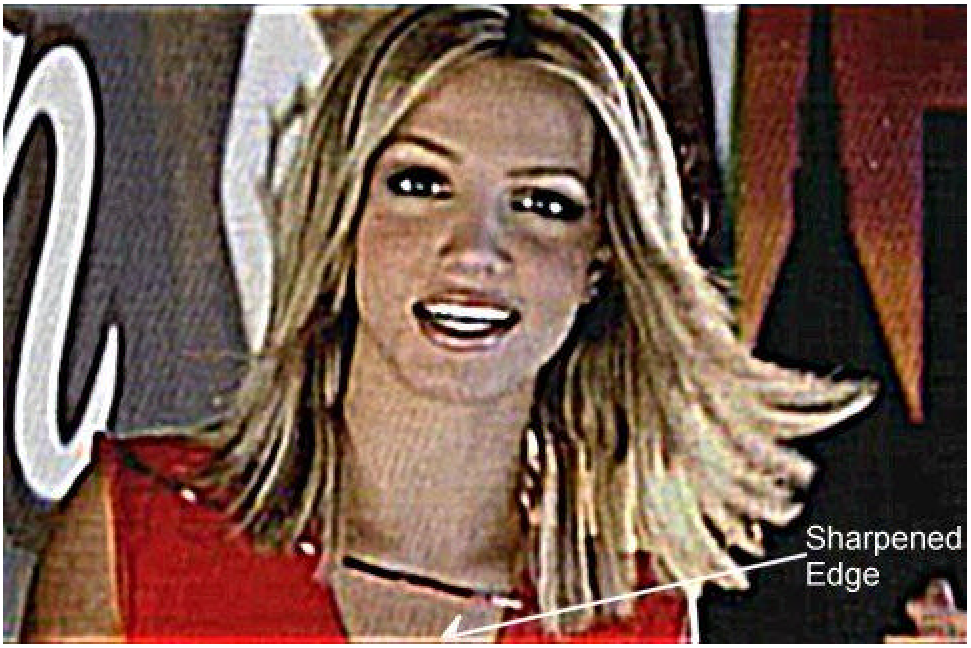
Top 0

Bottom 0

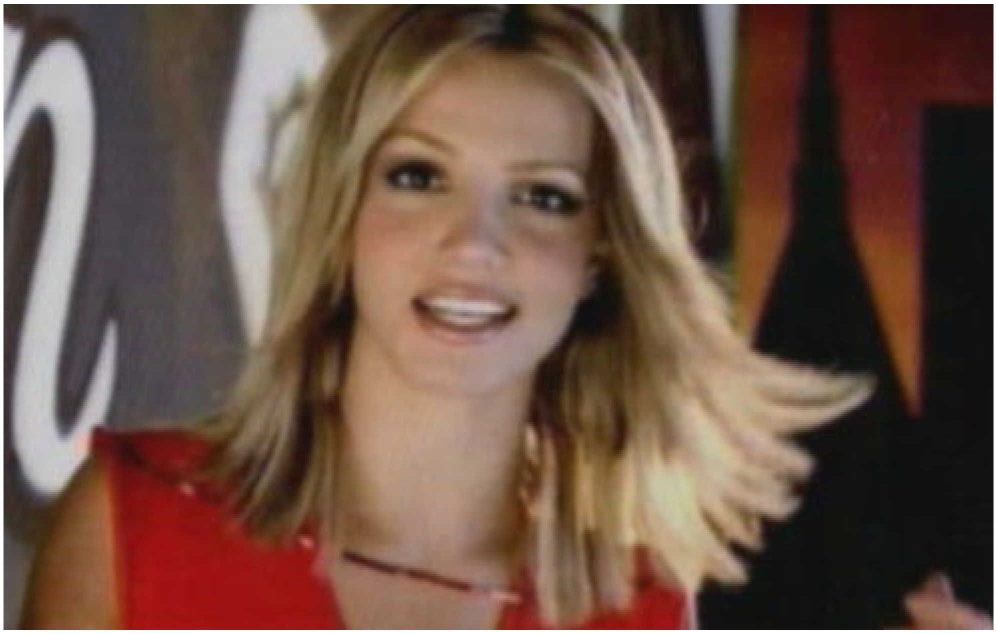
Left 0

Right 0

First lets explain what Edge Masking means. When you sharpen a video it sees the edges of the picture as black. Because sharpening increases the difference's between adjacent colours it will automatically make a white mark around the edges that could get annoying (see the picture below). Edge masking simply lets you take out the sharpening effects on each edge. By moving the Top, Bottom, Left and Right slider bars you can delete all sharpening effects to any of these areas.

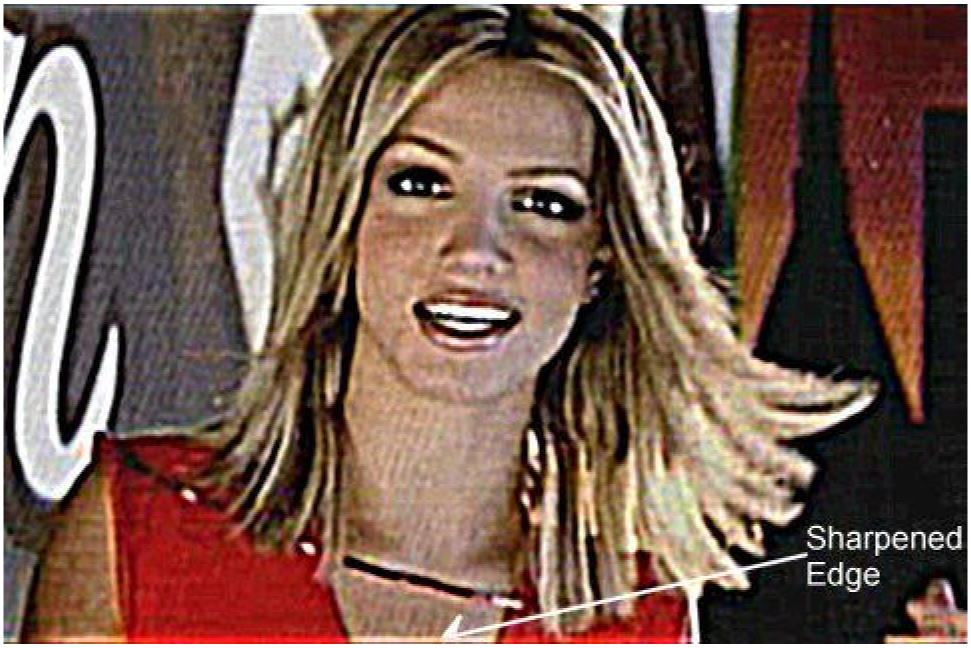


Now edge masking is explained all we need to worry about is the top three slider bars, the Diameter, Strength and Threshold. But first, take a look at the original captured image. It looks a little too blurry and we want it to look sharper.

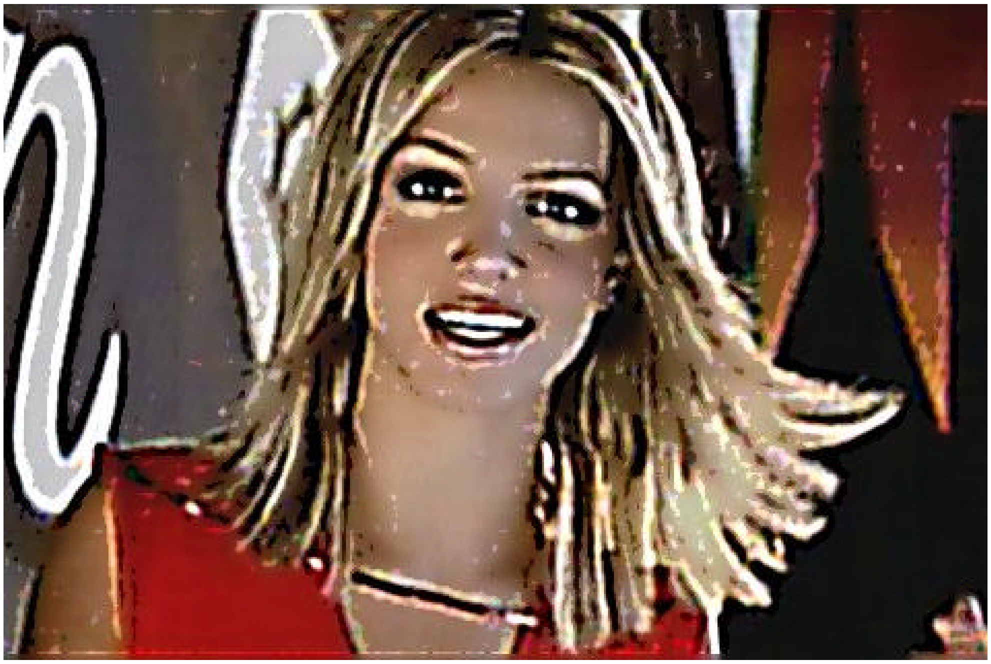


To see just what is going on I think its best to start with quite a high sharpening strength. So keep everything on its lowest setting and turn it up until you see large changes. I've used the full strength of 250 below (Diameter 3; Strength 250; Threshold 0).

Now we can see what the Diameter does. It basically shows what parts of the picture will become sharpest. A setting of 3 is the smallest amount and 11 the largest. Usually you will not need to go much higher than 3 unless you want to get a certain effect like to enhance highlights etc. But the picture below is Diameter 7; Strength 250; Threshold 0.

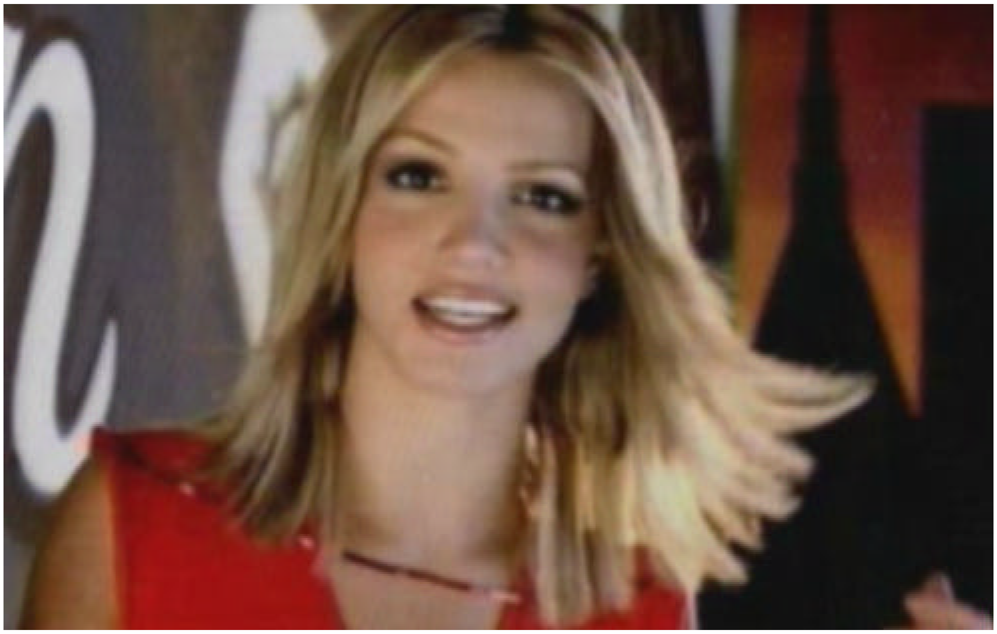


Finally we can see what the Threshold does. The Threshold controls what percentage of the picture is effected by the sharpening. If you have a large threshold only the finest details of picture will be affected and the rest of the picture unaltered. If you have a very small threshold then most of the image will be sharpened. The picture below has Diameter 7; Strength 250; Threshold 4.



Okay, 250 Strength is far too strong, so lets lower it a bit until the picture looks normal again. Then all we need to do is increase the strength a little bit at a time until the picture looks nice.

Here is the original picture below...



And here is the final sharpened version (Diameter 5; Strength 31; Threshold 2).

This picture could actually look a lot sharper but I felt that the diagonal lines (due to the aerial signal) were too noticeable so I used a slightly more blurred effect.

Final Notes

Well, I have explained some of the more complicated but most useful filters for video enhancing. There are a whole bunch of others too but doing a guide on them all would be crazy and take ages. I recommend you take some time to try out each filter. Read the instructions about them on the creators websites or instructions. The links to each website will be found on the VirtualDub homepage. For those with interlace problems don't forget that there are a bunch of smart deinterlacer filters. And for those who struggle with flickery video there is at least one anti-flicker filter.

I think it is important now to give one final warning! A small picture may look great with just about any filter you use on it, but *always* check the final movie (or a small clip of it before you apply the filter to the whole movie). Do this by playing it full screen with your monitor brightness both normal and turned up much higher than you normally would have it.

Have Fun!

Video Capture Guide

Part 1: The Basics

Hardware

Obviously my experience of video capture hardware is quite limited because I don't have unlimited money and I have only dealt with a couple of capture cards. Also keep in mind that hardware is improving all the time so when you search there may be more options than when this article was written. You will also be limited by how much you are willing to spend on your hardware and what you need it to do with it. In light of these facts I cannot recommend any particular card or TV tuner to use. I suggest you search around and read some reviews. Toms Hardware Guide is where I usually go to check out the latest info on hardware so check him out at: http://www.tomshardware.com/.

The expensive dedicated video capture cards designed for professional video editing are obviously the best quality but are usually out of most peoples price ranges. If you are very serious about it, there are some good quality and yet nicely priced cards sold for both home and proffesional use by Pinnacle Systems such as the DC10plus or perhaps even the DC50.

The less expensive Graphics Cards like the Nvidia GeForce, Matrox and ATI All In Wonder, and more recently the Radeon are good cheap alternatives. Just remember, despite claims of professional video quality, these all-singing-all-dancing graphics cards almost always produce worse results than a dedicated capture card. Of these graphics cards I think the current opinion is that the Matrox is the best quality, then the ATI (almost as good), then the GeForce and Voodoo ranges respectively. Many of these cards give the option of either Mpeg-2 or MJPEG hardware capture. Mpeg-2 gives best compression and MJPEG best quality especially if it is hardware based like the Matrox cards. If it doesn't offer *any* hardware capture option then its probably no better than a normal graphic card with a cheap TV Tuner stuck to it!

Basic TV Tuners cards are the cheapest option if you don't want a big upgrade and come with many low priced graphics cards. Because they are so cheap people assume they are not much good. Truth is, if your computer is 'man enough' then you will often be able to get equal or superior capture to those all-in graphics cards! In fact, I'd go as far as to say that if you have good enough hardware then you may be able to improve the quality of many graphic card captures by turning off hardware compression and using it like a TV Tuner instead!

What do I mean by 'man enough'?

The most perfect graphics format for video capture would be 'uncompressed AVI' with 'uncompressed PCM audio'. More specifically that is one uncompressed bitmap image for every frame along with a Wave file played in time to it. Many TV Tuners and Capture Cards will offer such an option or an equivalent lossless compression. But, especially at high resolutions most computer hardware will not be able to handle such an influx of data and you will get bad quality!

In light of this, most graphics cards use video compression as a *standard* part of capture to avoid problems. The ATI graphics cards may say something like 'AVI 1.0' but don't let the name AVI confuse you, it is actually a *highly* compressed format! Your hard disk can only save so much information per second. If you are capturing more information than your hard disk can take, you will loose frames and get bad, jerky video. To illustrate this effect, just try copying a 500MB file from one folder on your hard drive to another (don't cut and paste, just *copy* and paste). How long does it say it will take to copy? 3 minutes? 5 minutes? longer? So what happens if you are capturing 30 seconds of video at 640 x 480 in uncompressed AVI? Only 30 seconds of this format will take over 500MB of hard disk space! Your hard disk will just not be able to keep up! And this also highlights the other huge downside of uncompressed AVI - it takes a huge amount of space to store! For all these reason almost all video capture cards will offer to compress the video file directly to MJPEG, Mpeg-1 or Mpeg-2. But, if set correctly, any of these formats can give us VHS quality video.

A TV Tuner must use your computers Hard Disk and CPU power to compress or store the AVI in real time, so you will need to know that your machine is able to handle it. You certainly need a fast hard drive if you are capturing at 640 x 480. About 7200 RPM should be enough or 5400 RPM for 352 x 288 capture. I think Seagate do the best Hard Disks. If money is not a problem get SCSI HDD's because you can buy ones that reach speeds of 10000 RPM which will give you unlimited power in the storage department! On the other hand some of the newer IDE's are able to handle 25MB/s or more without trouble and are much cheaper. You will always need lots HD space, you can never have enough with video capture and 20 GB or more is probably a conservative estamate. I also suggest at least a 300Mhz CPU and an absolute minimum of 64MB RAM - the more the better of course.

Please bare in mind that all this is a rough estimate. Although most PC's exceed these requirements today, there are so many factors including slow graphics cards or motherboards that it is difficult to give exact requirements. If you are unsure I suggest you just invest in an ATI All in Wonder card or something like that rather then just waste money in the hope you can upgrade your system and capture better quality without one - you have been warned!

Video Capture Formats

MJPEG: will usually give the highest quality video especially if its hardware based capture. Instead of using bitmaps for each frame it uses compressed JPEG's images. As you know the quality of JPEG is almost as good as Bitmaps but very highly compressed. If you are capturing with a TV Tuner this is just about the only choice. The Picvideo MJPEG codec is the best quality I know of for real time capture.

Mpeg-1: Mpeg-1 capture is not fast enough to do in real time unless it is hardware based capture. I know there are utilities that will capture in Mpeg-1 in "real time" but the quality is crud even on a machine with a very high CPU. Lets face it, why does it take many hours to compress to Mpeg-1 or Mpeg-2 from an AVI with Panasonic Encoder or TMPGEnc etc., if you could do it in real time with the same quality?! Most of these real time captures use the ligos codec to compress, but the Ligos Mpeg Encoder cannot capture real time, so I think that says it all!

Mpeg-2: is the next highest quality to MJPEG But there are many problems with this format. Again it must be done by hardware compression because software compression is far too slow to produce any good quality. It is also a format that is very hard to edit. VirtualDub will not open an Mpeg-2 file unless we use Flask Mpeg or AVISynth to frameserve it. This makes editing very slow indeed.

Note: As you probably know by now, Mpeg-1, 2 & 4 compress like JPEG, but to increase effectiveness they also compare two or more frames in a sequence and only save the differences. This means they contain frames with only parts of the full picture in them. These half frames are called B frames and P frames. The full JPEG-like frames are called I-Frames. Many Mpeg capture cards such as the ATI All In Wonder will offer to capture as I-Frame only - in other words, as 'full frames' only. Capturing this way will increase the quality a lot but it will take up almost as much space as an MJPEG. This is because, with all the technical difference aside, I-Frame is basically the same compression as MJPEG. Other advantages are it takes less CPU power to compress and I-Frame only Mpeg-1 is fully editable in VirtualDub! You will also have the advantage of Mpeg-2 audio compression so you can capture for a little longer than MJPEG. If you are going to capture this way I suggest you turn off all motion compensation settings too.

Mpeg-4: is the best compressed format and you can use VirtualDub to capture directly to Divx or MS Mpeg-4. But again this will not produce good quality results because, like the other Mpeg formats, it takes too long to compress. It would only be useful if you intended on very low bitrates where quality was not a high factor.

Framerates & Resolution

If you are going to edit and/or compress a video using VirtualDub or any video editing package then the better the original video the better the final result will be. Try to use the highest settings you can without going overboard. For example, there is no need to struggle to capture a video at 640 x 480 pixels if you are going to resize it to 352 x 240 because not only will this take a lot of time to resize, but it will take twice the hard drive space! The difference in quality between the two will be very small so why not just keep to 352 x 240 instead.

VHS Resolution

Concerning resolution, you should always keep in mind that VHS video is roughly 320 x 240 pixels. Even so it may be best to keep to 352 x 288 PAL or 352 x 240 NTSC if you wish to follow VCD standards. This fact sounds really weird because a TV's active picture has 480 lines and a DVD resolution is 720 x 480 pixels. Some, in a desperate attempt to capture full quality video, will make DVD resolutions of 720 x 480. Let me tell you now that these captures are *not* even close to DVD quality no matter how good the capture! Neither have I just picked the size 320 x 240 out of thin air either. The reason for this is based on a value of clarity that mathematicians call the Kell factor. The Kell refers to the amount of detail displayed, in this case, on an analogue TV. To simplify a rather complex subject down to the bare bones we could say that a TV picture is very fuzzy and a TV "pixel" works out to less than half the clarity of a computer pixel! This means a computer resolution of less than half that of a TV is needed to store the same information. What is worse is the fact that, if the quality of your VCR or recording is not very good, or if you are using a camcorder, you are probably looking at little more than 280 x 210 pixels or something like that!

Note: I haven't done the exact mathematics on this yet, but a DVD resolution must work out to over double the quality of VHS video! Which brings up the subject 'why do we need high definition DVD's if they are already so close to HDTV quality, or have I just confused myself on this one?'

What does this mean? That capturing at 352 x 240 will give full VHS resolution? Possibly, but its not quite that simple! It may depend on how your capture card works. If it disregards lines then actual information is missing when it resizes. What would be needed is to construct a full image from *all* 480 TV lines and resize them down to the equivalent PC pixels. For example, if you want full detail you can capture at 352 x 480 and then resize precise Bicubic with VirtualDub to 352 x 240 (or 288). This should preserve most of the fine details and I personally think this is a good alternative. As you can see, assuming we need 480 pixels high for total quality, the pixels across still remain the same, about 352 pixels. In light of this problem Mpeg-2 was developed with yet another compromise which is supported by the ATI capture card. It can capture at 352 x 480 and then when played back stretch the video back to 720 x 480 to keep the correct aspect ratio. This takes much less space since it is resized *only* when it is played back and stops all the arguments.

Interlace

I have explained the problem of interlace in great detail Here. So I don't need to spend too much time on it. Basically there is little you can do to solve it other than using VirtualDubs deinterlace filters. How bad the problem is, or if you get it at all, will depend on how your graphics card captures. If you only capture one field (i.e. every other line making 240 or 288 pixels) then usually you won't have the problem. Sometimes you will be given the option to capture both fields (i.e. 480 or 576 pixels) and blend both fields together, check the settings of your capture device. There is a rumour going around that capturing to Mjpeg or Mpeg-2 will solve interlace problems. This is not true, at least with every set up I've tried so far.

Audio

Since PCM Wav audio is not too large and yet still the best quality I suggest that you use that. It puts a large strain on your computer to compress to Mp3 or WMA audio and there may be more problems with audio synch if you don't. I suggest you also use 44100Hz 16 bit stereo which is CD quality audio and best for music. This is also the most compatible format. You can, of course, use any other setting such as 22050Hz if you prefer and this will to save some space at the expense of a little quality. But then again if you are going to compress the audio anyway I wouldn't worry too much.

VHS Framerates

There is never any need to capture higher than 30 frames per second. I'd also recommend that you stick to 25fps for PAL TV capture and 29.970fps for NTSC TV capture. This is because most capture cards are designed for this kind of capture and you may get bad results or synch problems if you don't.

Dropped Frames

Both VirtualDub and most capture cards will tell you if your system cannot capture all the frames you need per second for smooth playback. I think that if your computer cannot take the strain and starts dropping frames it is better to capture at a lower resolution! I would *always* sacrifice picture size (resolution) in favor of a high framerate unless the picture was very small. If you have the framerate dropping problem with an NTSC TV which is 29.97fps then you could try capturing 25fps instead, but, as I said, this may cause problems. Also if you intend on making a VCD at 25fps then you should also capture at 352 x 288 instead of 352 x 240.

If your computer starts dropping frames you will almost always get audio synch problems and annoying jerky playback problems. Its easy for the novice to think that if they get dropped frames at 25fps then they should set it at 30fps to compensate. Then, even if some frames were dropped, we would still get smooth playback. This is a big mistake! If your machine is dropping frames at 25fps then asking it to capture more per second puts even more strain on it and you will get much worse quality. The only answer is to lower the resolution or lower the framerate or upgrade your system =oP.

Starting Capture

Now you know the basics and have the equipment, I will explain in the next article how to set up and capture using VirtualDub and your capture device. Additionally, it is always a good idea to make some test clips before you capture and see how they play before you do a long capture. You will also need to watch the 4GB limit, Windows 98 cannot save single files larger than 4GB each so that restricts what we can do even more. If you have Windows 2000 this shouldn't be a big problem. Nevertheless all these issues will be dealt with in following articles.

Converting VCD to DivX

Converting a Video CD (VCD) to DivX is usually easier to do than converting a DVD CD. But why convert? Simple, VCD's take up two 650MB CD-R's whereas DivX can fit on one CD-R with similar quality.

Before I start here are the things you will need:

DivX ;-) Mpeg-4 Codec 3.1alpha (or greater)

Fhg Radium MP3 codec

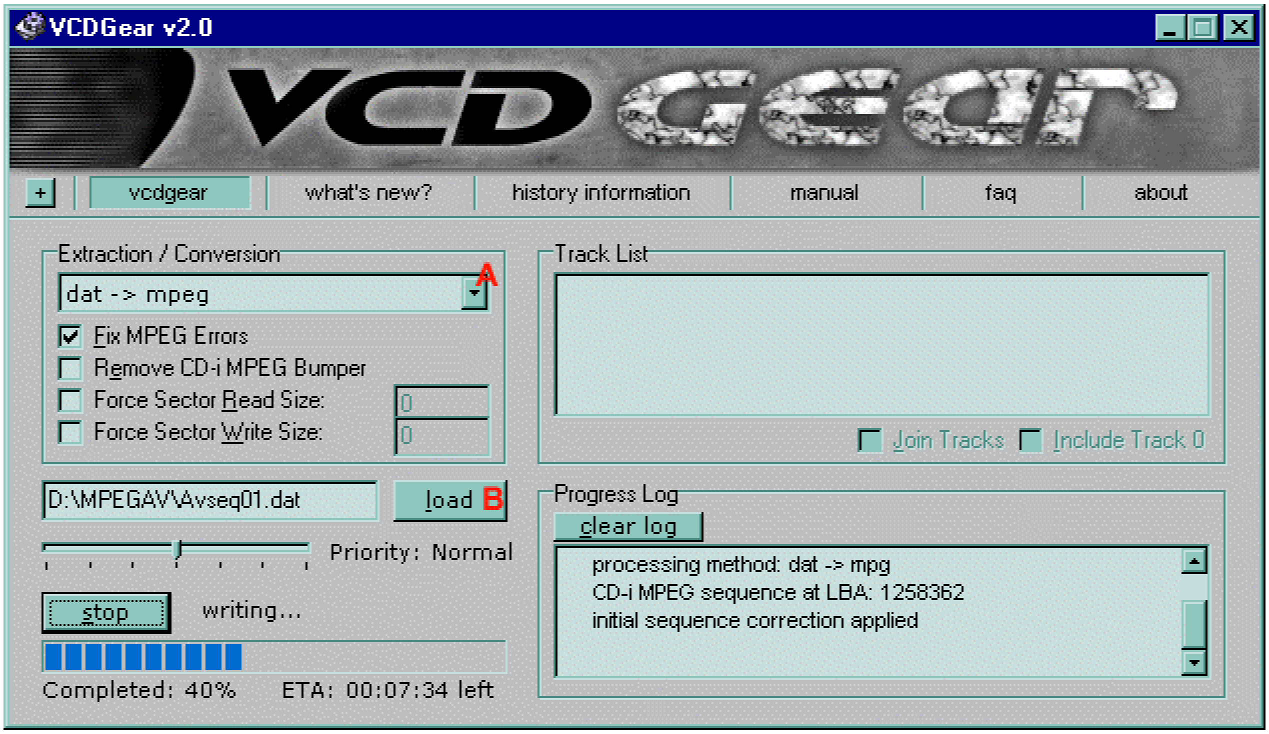
VCDGear 2.0

VirtualDub 1.4c (or greater)

Advanced Bitrate Calculator 1.8

Extracting the Dat File

Open VCDGear 2.0 by double clicking on it. Up pops this:



Extraction / Conversion: dat -> mpeg

☑ Fix MPEG Errors

□ Remove CD-i MPEG Bumper

□ Force Sector Read Size: 0

□ Force Sector Write Size: 0

D:\MPEGAV\Avseq01.dat

Priority: Normal

writing...

Completed: 40% ETA: 00:07:34 left

Progress Log:

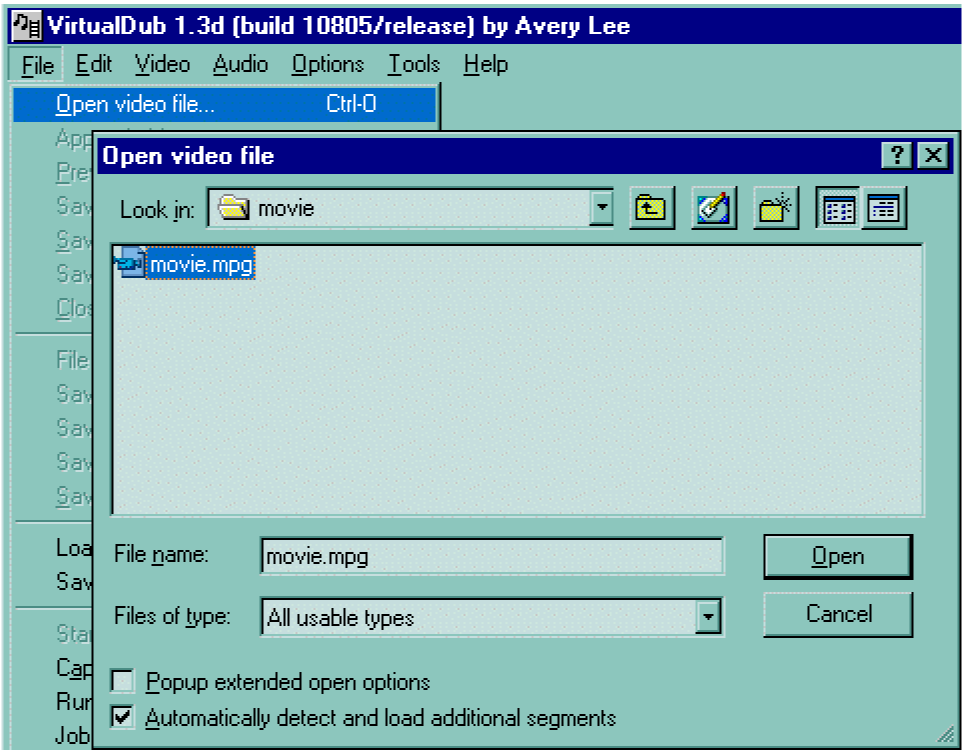
```
processing method: dat -> mpg
CD-i MPEG sequence at LBA: 1258362
initial sequence correction applied
```

Put your first VCD into your CD-Rom drive and choose the 'dat -> mpeg' option from the drop down list (**A**); then press load (**B**). Make sure 'Fix MPEG Errors' is ticked but uncheck the 'Remove CD-i MPEG Bumper'. Leave the rest default. Bingo! You have and mpeg file!

Note: *Many people are of the opinion that because they can play VCD *.dat files in most Mpeg players the *.dat file is the same as an Mpeg. This is not the case! A VCD mpeg is basically an Mpeg-1 file but just like the DVD VOB files the *.dat file also contains other pieces of information needed to play the VCD in a commercial VCD player. These files need to be deleted otherwise you will not be able to do anything with the Mpeg movie. This is why we need VCD Gear.*

Converting the Video and Audio to DivX with VirtualDub

It seems that the latest version of VirtualDub does a much faster conversion of Mpeg-1 to DivX so I suggest you try this first! Also there is less chance of the audio going wrong this way because VirtualDub usually sees to that.


VirtualDub 1.3d (build 10805/release) by Avery Lee
File Edit Video Audio Options Tools Help
Open video file... Ctrl-O
App
Pre
Sav
Sav
Sav
Clos
File
Sav
Sav
Sav
Sav
Loa
Sav
Sta
Cap
Rur
Job
Open video file
?
X
movie.mpg
Open
Cancel

Look in: movie

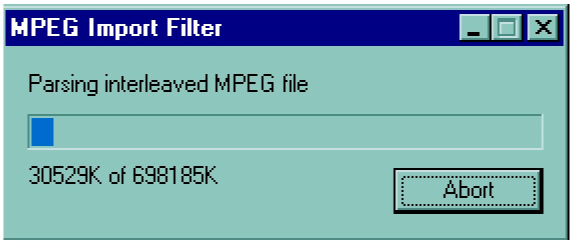
File name: movie.mpg

Files of type: All usable types

☐ Popup extended open options

☑ Automatically detect and load additional segments

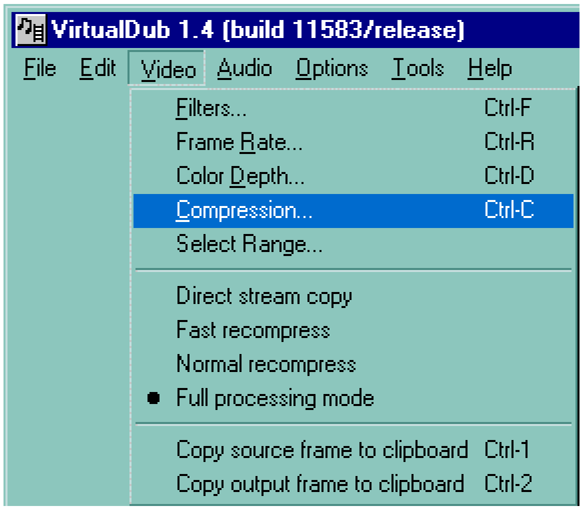
Wait while it checks the movie to make sure its in the correct format to convert.



Parsing interleaved MPEG file

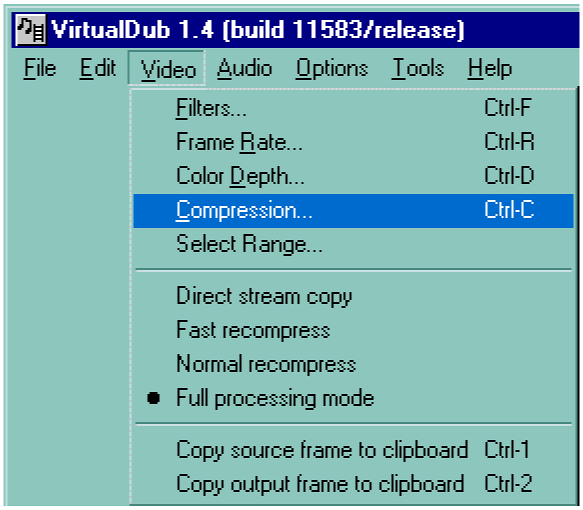
30529K of 698185K

Choose: Video > Full Processing Mode (checked below with a black dot).

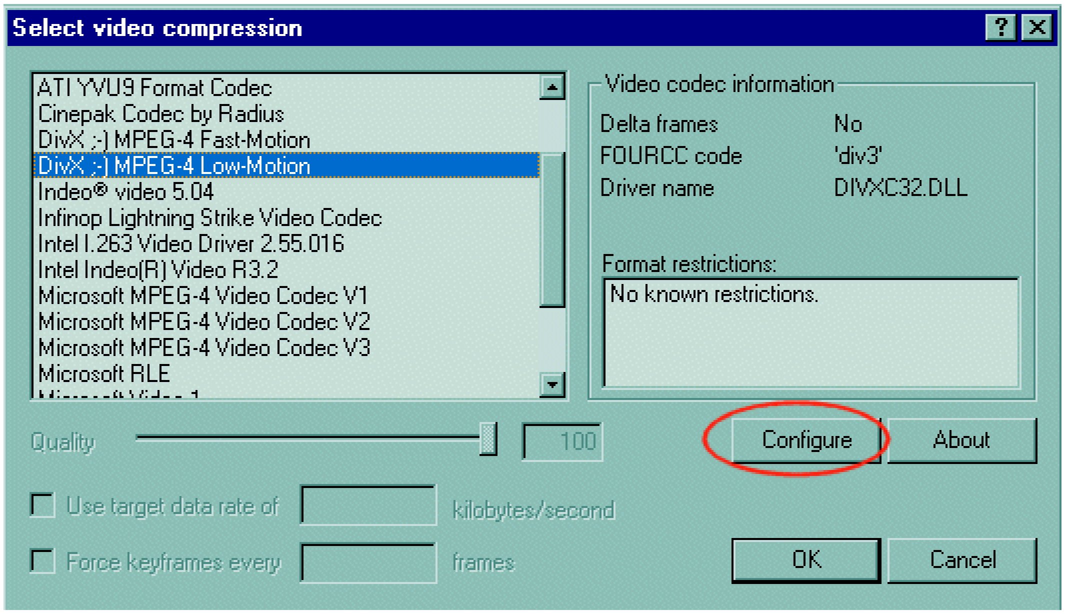


Video Compression

And then: 'Video > Compression...'



Up pops the following box below. Choose Mpeg-4 Low-Motion and hit the configure button (highlighted in red).



Select video compression

ATI YVU9 Format Codec
Cinepak Codec by Radius
DivX ;-) MPEG-4 Fast-Motion
DivX ;-) MPEG-4 Low-Motion
Indeo® video 5.04
Infinop Lightning Strike Video Codec
Intel I.263 Video Driver 2.55.016
Intel Indeo(R) Video R3.2
Microsoft MPEG-4 Video Codec V1
Microsoft MPEG-4 Video Codec V2
Microsoft MPEG-4 Video Codec V3
Microsoft RLE
Microsoft Video 1

Video codec information

Delta frames	No
FOURCC code	'div3'
Driver name	DIVXC32.DLL

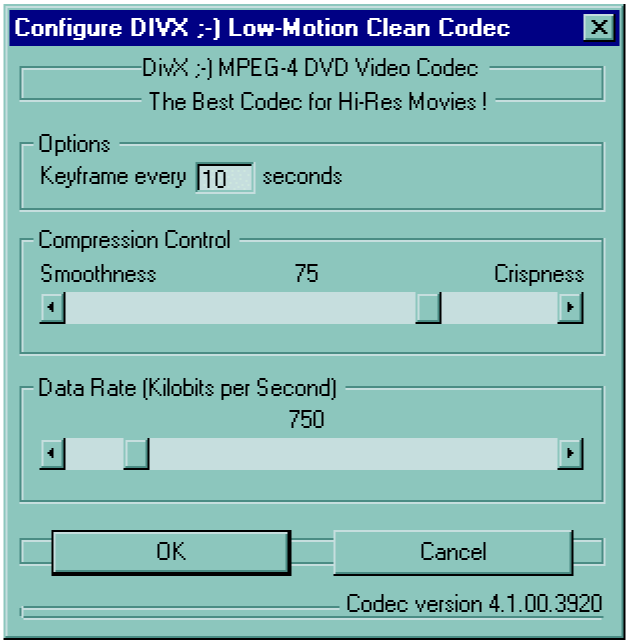
Format restrictions:
No known restrictions.

Quality 100

□ Use target data rate of kilobytes/second

□ Force keyframes every frames

The Up pops the codec options. Use one of the bitrate calculators that can be downloaded from Digital Digest to make sure the movie will fit into the size you need. 650kbps will usually fit ANY movie on a single CD, but you can usually get away with between 900 and 1000kbps for a 352 x 240 VCD movie for better quality see my quality guide if you are unsure about correct bitrate settings.



Configure DIVX ;-) Low-Motion Clean Codec

DivX ;-) MPEG-4 DVD Video Codec
The Best Codec for Hi-Res Movies !

Options
Keyframe every 10 seconds

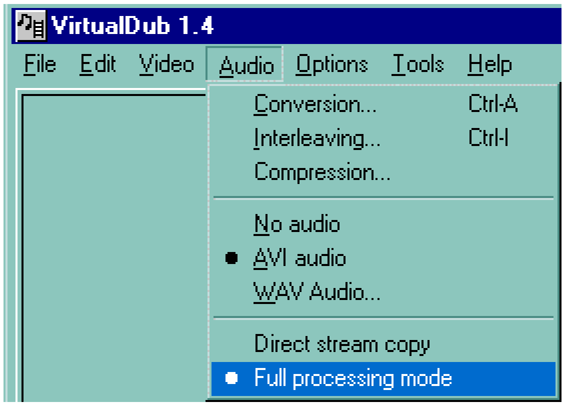
Compression Control
Smoothness 75 Crispness

Data Rate (Kilobits per Second)
750

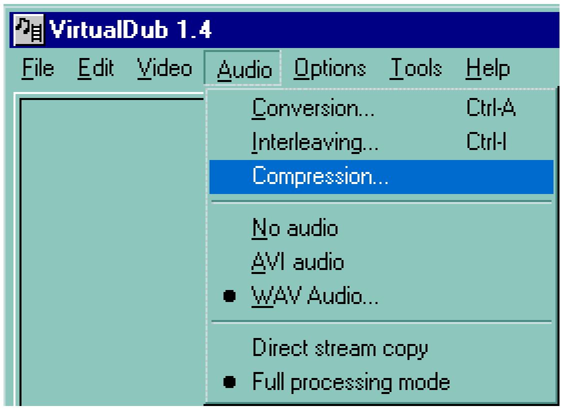
Codec version 4.1.00.3920

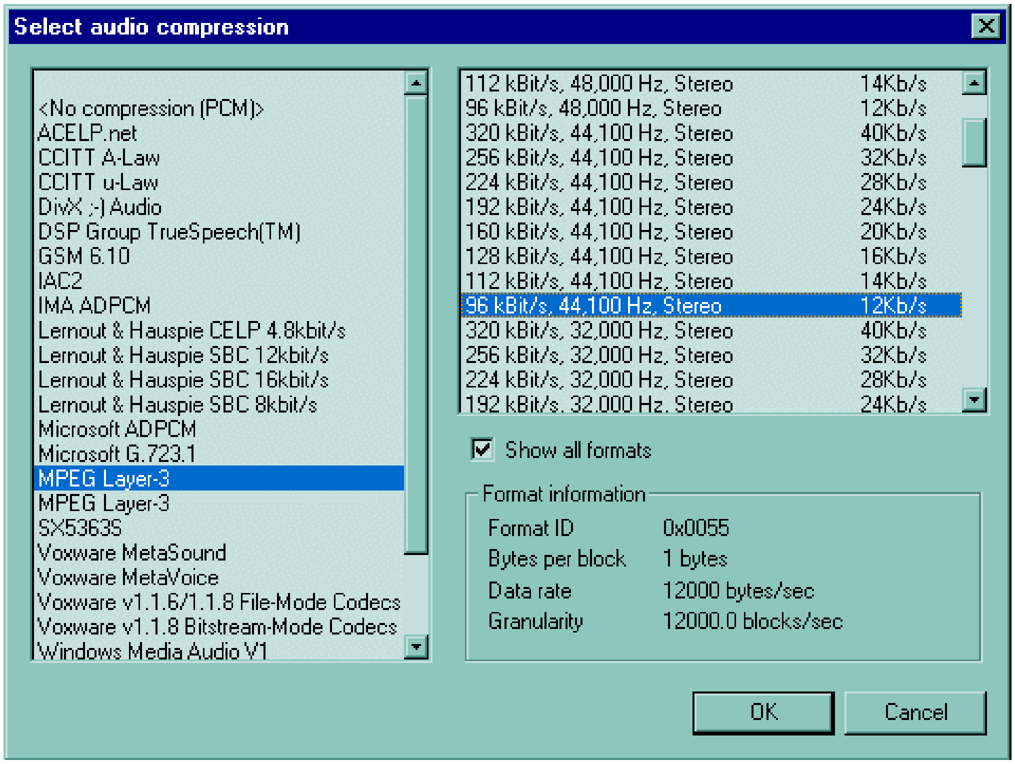
Audio Compression

Now select the audio settings by going to: Audio > Full processing mode (so the black dot appears by it).



Then Audio > Compression.




Select audio compression

<No compression (PCM)>
ACELP.net
CCITT A-Law
CCITT u-Law
DivX ;-) Audio
DSP Group TrueSpeech(TM)
GSM 6.10
IAC2
IMA ADPCM
Lernout & Hauspie CELP 4.8kbit/s
Lernout & Hauspie SBC 12kbit/s
Lernout & Hauspie SBC 16kbit/s
Lernout & Hauspie SBC 8kbit/s
Microsoft ADPCM
Microsoft G.723.1
MPEG Layer-3
MPEG Layer-3
SX5363S
Voxware MetaSound
Voxware MetaVoice
Voxware v1.1.6/1.1.8 File-Mode Codecs
Voxware v1.1.8 Bitstream-Mode Codecs
Windows Media Audio V1

112 kBit/s, 48,000 Hz, Stereo	14Kb/s
96 kBit/s, 48,000 Hz, Stereo	12Kb/s
320 kBit/s, 44,100 Hz, Stereo	40Kb/s
256 kBit/s, 44,100 Hz, Stereo	32Kb/s
224 kBit/s, 44,100 Hz, Stereo	28Kb/s
192 kBit/s, 44,100 Hz, Stereo	24Kb/s
160 kBit/s, 44,100 Hz, Stereo	20Kb/s
128 kBit/s, 44,100 Hz, Stereo	16Kb/s
112 kBit/s, 44,100 Hz, Stereo	14Kb/s
96 kBit/s, 44,100 Hz, Stereo	12Kb/s
320 kBit/s, 32,000 Hz, Stereo	40Kb/s
256 kBit/s, 32,000 Hz, Stereo	32Kb/s
224 kBit/s, 32,000 Hz, Stereo	28Kb/s
192 kBit/s, 32,000 Hz, Stereo	24Kb/s

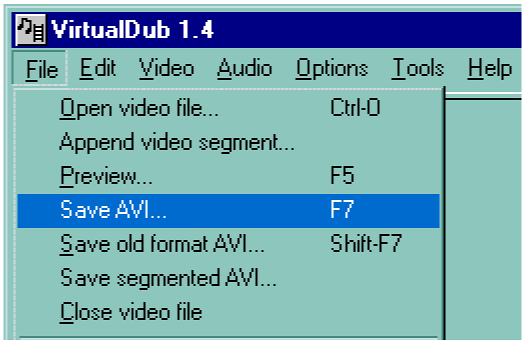
☑ Show all formats

Format information

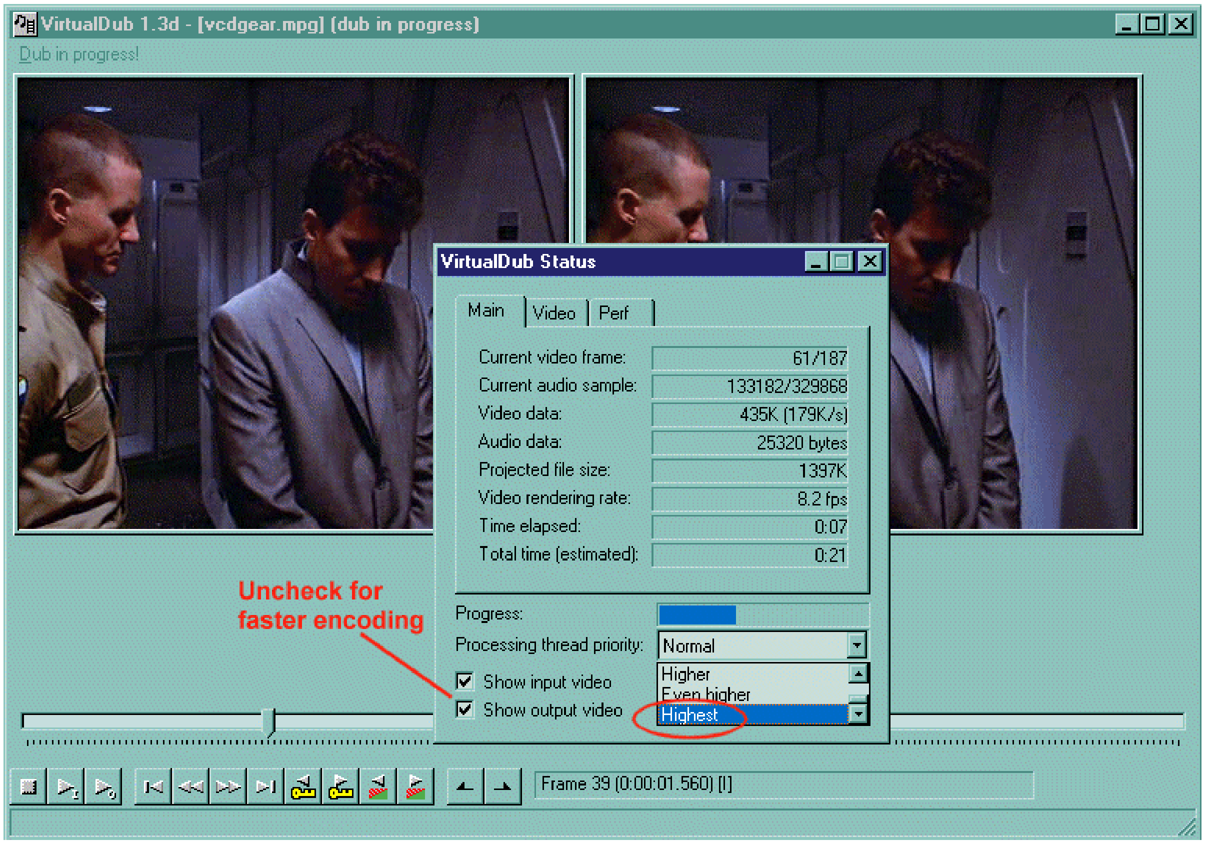
Format ID	0x0055
Bytes per block	1 bytes
Data rate	12000 bytes/sec
Granularity	12000.0 blocks/sec

Up pops this box below. Choose Mpeg Layer 3 or DivX ;-) Audio 64kbps, 441 kHz, stereo. And press OK.

That's it! Save AVI...



Here we go! The picture on the left represents the original VCD movie and the one on the right represents the DivX version. This is great because unlike Flask Mpeg you can see what the quality will be like and change it before the file is finished =^). For even faster encoding you can set the priority settings to 'Highest' and turn off the 'Show input' and 'Show output' video by unchecking the boxes. These can be turned on at any time to see your progress.



Current video frame: 61/187
Current audio sample: 133182/329868
Video data: 435K (179K/s)
Audio data: 25320 bytes
Projected file size: 1397K
Video rendering rate: 8.2 fps
Time elapsed: 0:07
Total time (estimated): 0:21
Progress:
Processing thread priority: Normal
Higher
Even higher
Highest
☑ Show input video
☑ Show output video
Frame 39 (0:00:01.560) [I]

Uncheck for faster encoding

Troubleshooting

Some VCD's are just badly encoded! I have even bought "proffesional" VCD's that would not play on my VCD player! The weird thing is they play back perfectly in Media Player but when we encode them they sometimes have sections with pink and green blocks in them.

On such substandard, or perhaps just corrupted mpeg files, I have tried all the main utilities: Flask Mpeg, Mpeg2avi and VirtualDub with no luck. I have tried fixing them with IFilmEdit, VCDGear and a whole bunch of others. The only thing that did a good job was Graphedit so check out my guide for converting "Stubborn Mpeg's to DivX (AVI) with Graphedit".

How to Convert QuickTime MOV Files

This seems to be the big issue that no one knows how to do easily. Everyone who has ever asked this on the Newsgroups will either get the age old sarcastic remark: "just buy Quicktime" or "Premiere does a great job!" And this is usually from people who *pretend* they actually bought the products instead of going to astalavista.box.sk and cracking them! Well dargsarnit! Who wants to buy Quicktime just to convert from its crappy uneditable format? Or spend hundreds of bucks buying Premiere for the same reason?! In that case try my method that doesn't require illegal software or the need to shell out megabucks!

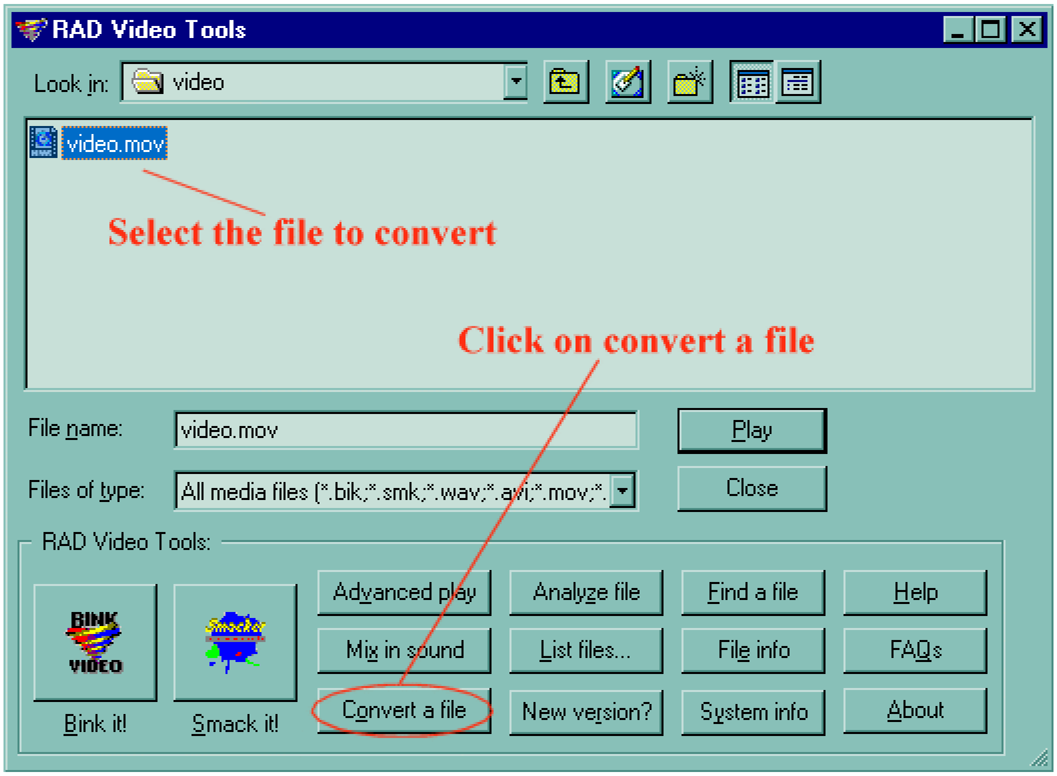
I won't lie to you, though, there is no easy way to convert every Quicktime movie. So this is just the easiest method (other than Quicktime Pro) that I have found.

Before I start here are the things you will need:

DivX ;-) Mpeg-4 Codec 3.1alpha (or greater)

RAD Video Tools Bink

Install Bink and start it up, you will then get the following window. Select your Quicktime movie and hit the 'convert a file' button.



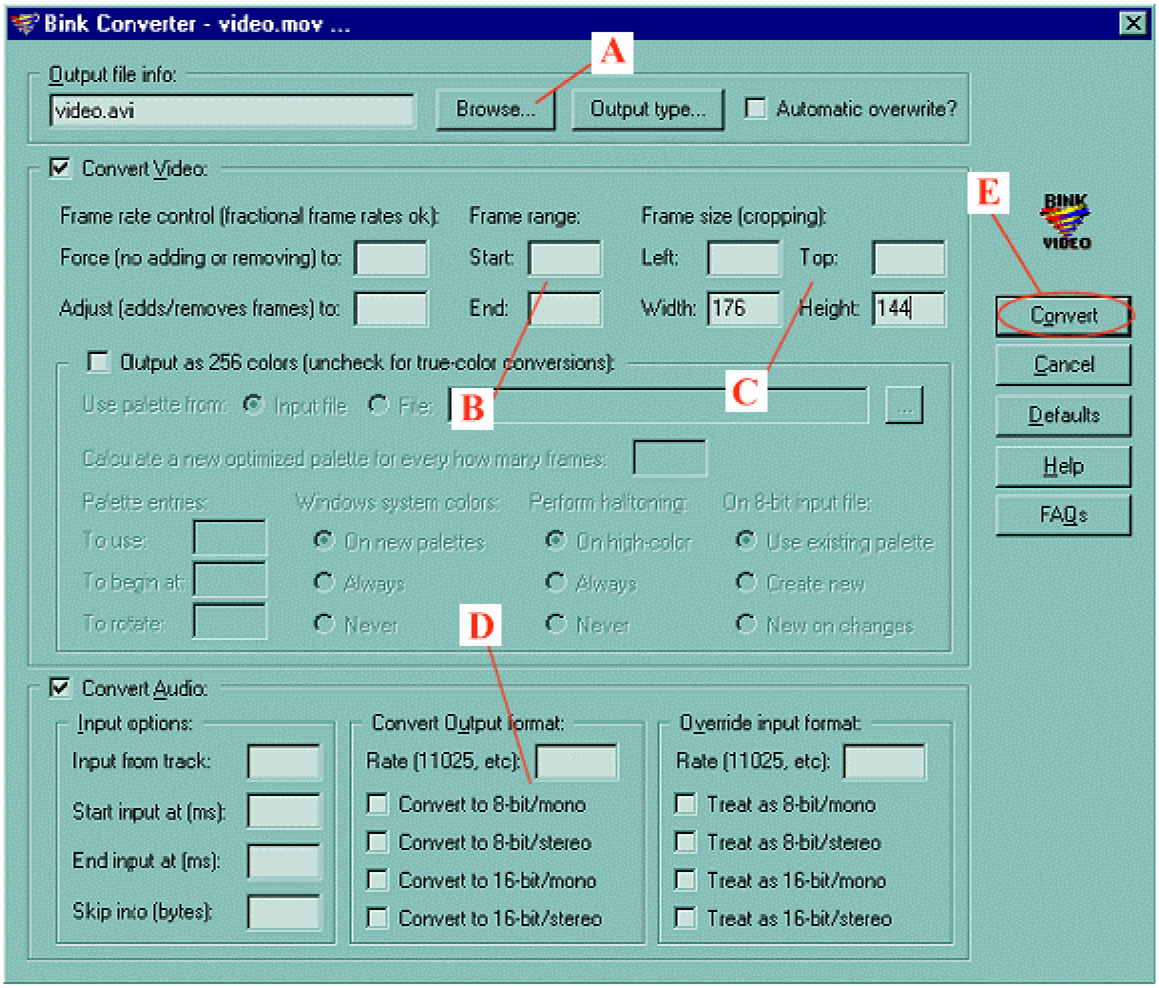
File name: video.mov

Files of type: All media files (*.bik;*.smk;*.wav;*.avi;*.mov;*.

Select the file to convert

Click on convert a file

Up will appear the conversion options. First tell it where to save your final AVI file (**A**). Then all you need to do to start the conversion is to press 'convert' (**E**).



Bink Converter - video.mov ...

Output file info: video.avi

□ Automatic overwrite?

☑ Convert Video:

Width: 176

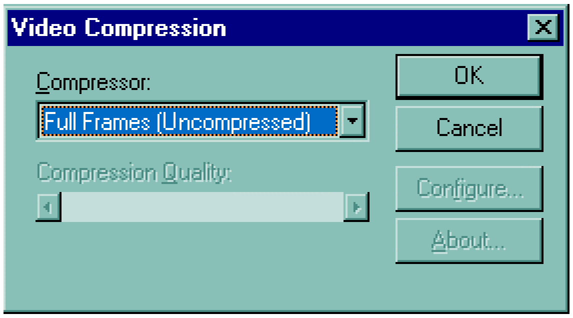
Height: 144

□ Output as 256 colors (uncheck for true-color conversions):

☑ Convert Audio:

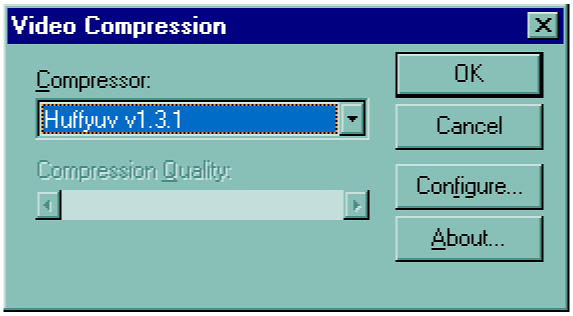
But there are a few other options you may wish to use before you convert. You can crop the image to a certain size (**C**) - it doesn't support resizing unfortunately. If you need to resize or edit the video in any way I suggest you convert it and then do so in VirtualDub. You can also select how much of the video is to be converted (**B**) by selecting what frame to start the conversion from and what frame to end it. Finally we have the Audio Samplerate conversion (**D**). This is not really needed most of the time because it may cause synch problems if we change it from the original. The only time you may want to use it is if you intend on converting straight to VCD which requires you use a 44100Hz samplerate. But I'd prefer to prepare my audio with VirtualDub instead. Anyway, you can experiment with the rest of the settings if you find the final video doesn't look as good as you hoped.

Once you click on the Convert button you will be asked what format you wish to convert the video into. If it is a short clip and you intend on editing it in VirtualDub, you would probably be best off using Uncompressed AVI i.e.:


Video Compression

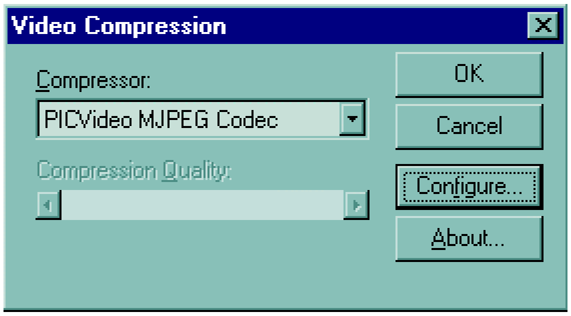
Compressor: Full Frames (Uncompressed)

If it is quite big but you still want to do lots of editing I suggest you choose MJPEG because this offers the best editing. The best quality MJPEG is the uncompressed Huffyuv format. But this sometimes has framerate problems so check your final file.


Video Compression

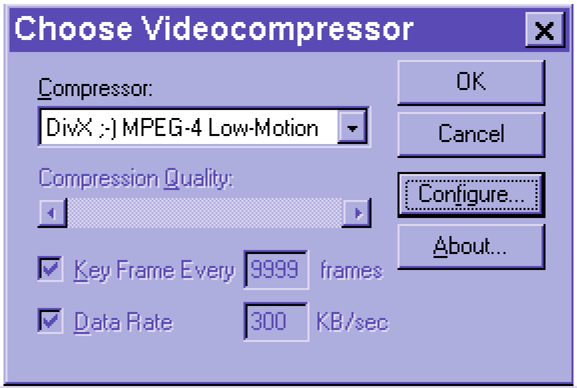
Compressor: Huffyuv v1.3.1

Another alternative is the PICVideo MJPEG codec which is a good trade between compression and quality.


Video Compression

Compressor: PICVideo MJPEG Codec

If you don't want to do much editing or you just want to convert straight to Divx just choose the Divx codec you like best:



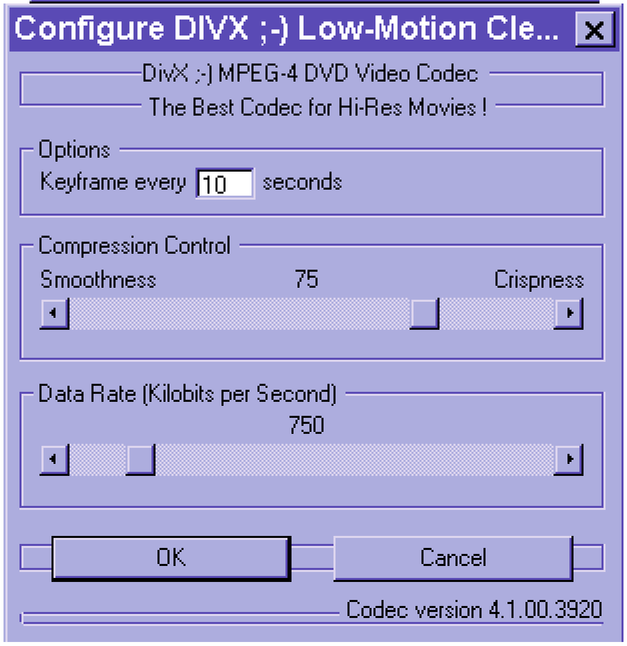
Choose Videocompressor

Compressor: DivX ;-) MPEG-4 Low-Motion

Compression Quality:

☑ Key Frame Every 9999 frames

☑ Data Rate 300 KB/sec



Configure DIVX ;-) Low-Motion Cle...

DivX ;-) MPEG-4 DVD Video Codec

The Best Codec for Hi-Res Movies !

Options

Keyframe every 10 seconds

Compression Control

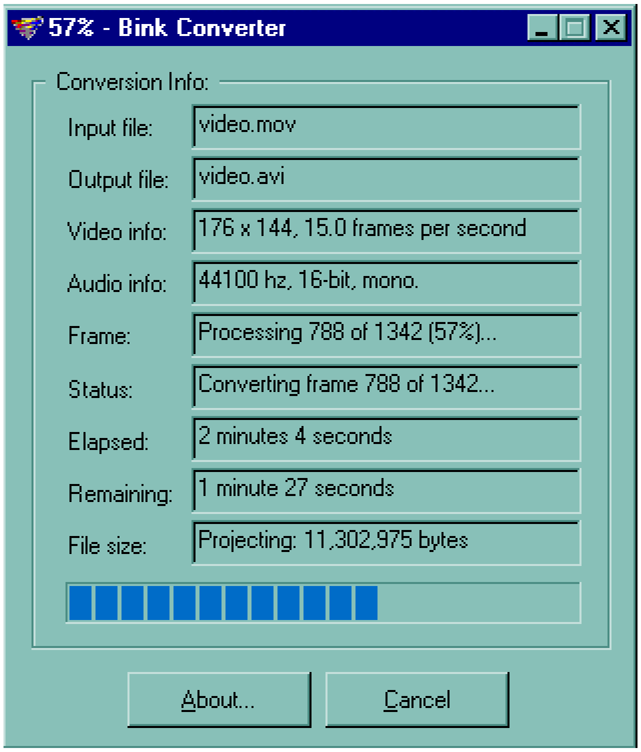
Smoothness 75 Crispness

Data Rate (Kilobits per Second)

750

Codec version 4.1.00.3920

After you have decided press OK and you will see the progress box below:



57% - Bink Converter

Conversion Info:

Input file: video.mov

Output file: video.avi

Video info: 176 x 144, 15.0 frames per second

Audio info: 44100 hz, 16-bit, mono.

Frame: Processing 788 of 1342 (57%)...

Status: Converting frame 788 of 1342...

Elapsed: 2 minutes 4 seconds

Remaining: 1 minute 27 seconds

File size: Projecting: 11,302,975 bytes

That's it! The audio is always uncompressed PCM as far as I can tell, so you will need to recompress it using VirtualDub as usual. See my multiplexing guide for details of that.

Troubleshooting

The audio is out of synch, what can I do?

Make sure you are using the correct samplerate used in the original movie. Then use my AVI synching guide to put things right again.

Bink will not convert the MOV file at all?!

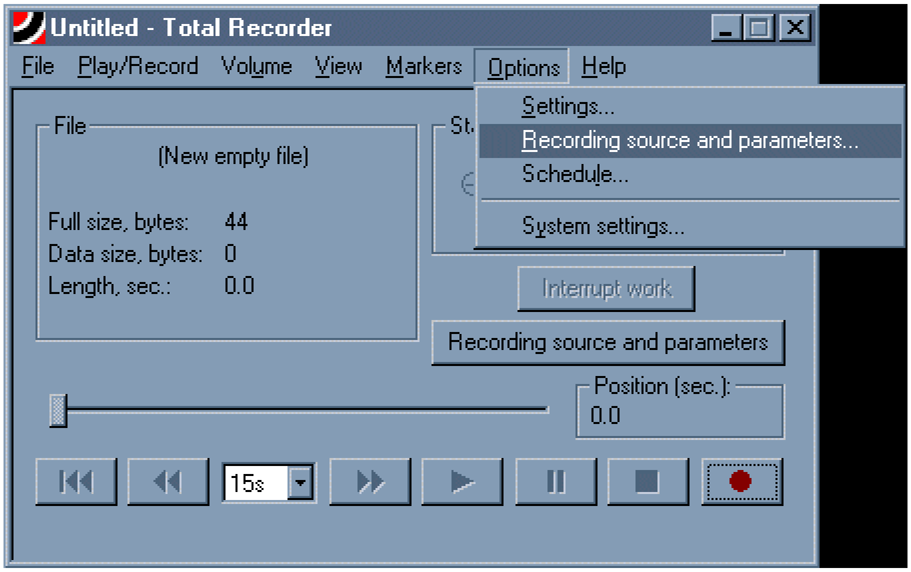
It is not a guarantee that Bink can convert every kind of Quicktime format (there are a few kinds). But I have tried it on a few including the best format i.e. Sorenson and it worked fine.

There is no audio, why?

Quicktime movies sometimes have copy protection installed which will disable the audio. If you have Total Recorder you can get it with that after and then multiplex it to the AVI using VirtualDub (see my multiplexing guide for details of this).

<u>Getting the audio with Total Recorder is easy enough:</u>

Open your video in Quicktime or Media Player depending on what plays it. Then press the red recording button at the bottom right of the picture below. It shouldn't start recording until you play your video file. Start your video file playing.



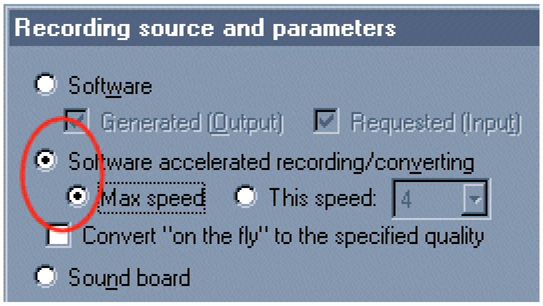
File

(New empty file)

Full size, bytes: 44
Data size, bytes: 0
Length, sec.: 0.0

Position (sec.): 0.0

If the format supports it, you can speed up the process using the following method: Go into Options > 'Recording source parameters...



Recording source and parameters

○ Software

☑ Generated (Output) ☑ Requested (Input)

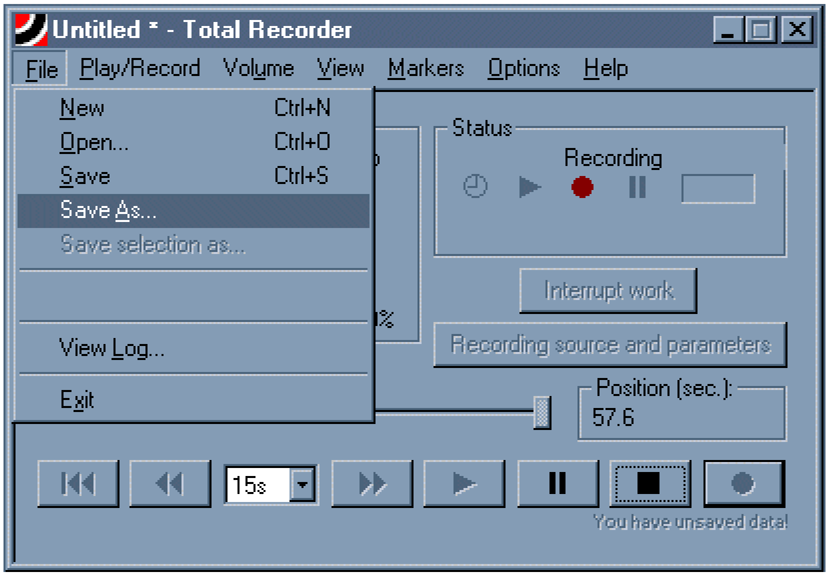
● Software accelerated recording/converting

● Max speed ○ This speed: 4

□ Convert "on the fly" to the specified quality

○ Sound board

Choose the Software accelerated recording/converting option and also check the Max Speed option. This will speed up the conversion to many times the speed it would normally take.



When you are finished save the wave file and you are done :-)

Note*: if it starts to work fine and then you get nothing but garbage for sound this is because you have a messed up cracked version of Total Recorder. You need to fully register the version of Total Recorder which requires a serial number for it to work correct. Most cracks will cause this security measure.*

AVI to DivX

This is the easiest of all ways to convert to DivX because this AVI is the prefered windows format. So lets keep this simple huh =)

Before I start here are the things you will need:

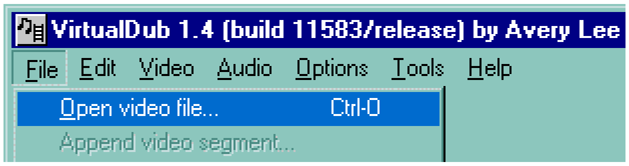
DivX ;-) Mpeg-4 Codec 3.1alpha (or greater)

Fhg Radium MP3 codec

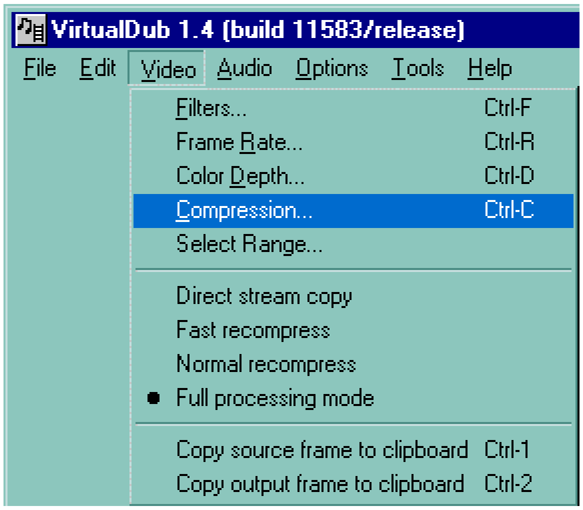
VirtualDub 1.4c (or greater)

Advanced Bitrate Calculator 1.8

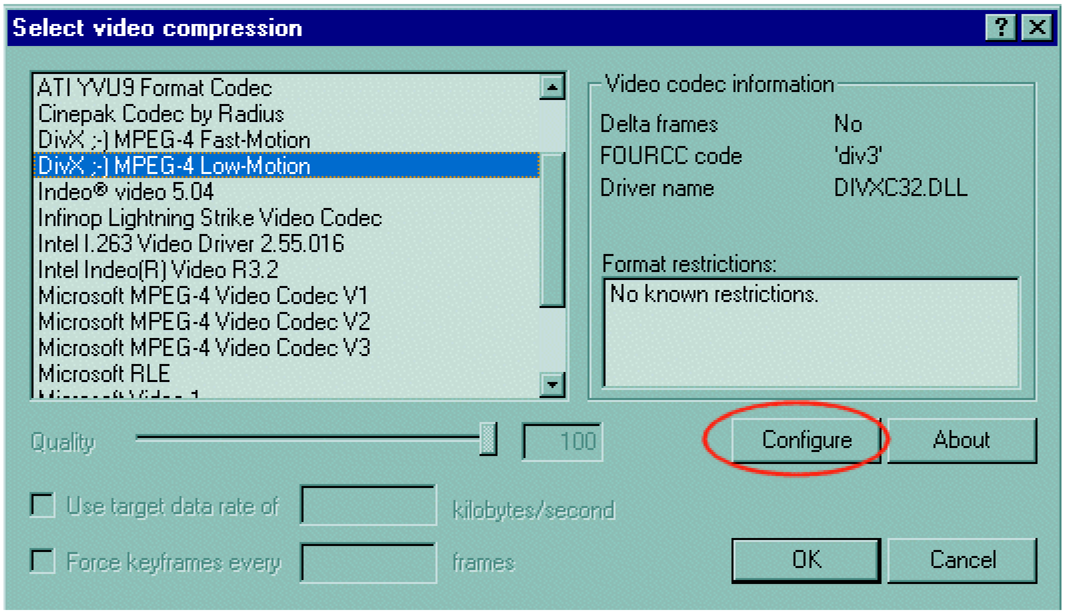
Open the AVI file you wish to convert in VirtualDub.


VirtualDub 1.4 (build 11583/release) by Avery Lee
File
Edit
Video
Audio
Options
Tools
Help
Open video file... Ctrl-O
Append video segment...

And then go to: 'Video > Compression...'



Up pops the following box below. Choose Mpeg-4 Low-Motion and hit the configure button (highlighted in red).



Select video compression

ATI YVU9 Format Codec
Cinepak Codec by Radius
DivX ;-) MPEG-4 Fast-Motion
DivX ;-) MPEG-4 Low-Motion
Indeo® video 5.04
Infinop Lightning Strike Video Codec
Intel I.263 Video Driver 2.55.016
Intel Indeo(R) Video R3.2
Microsoft MPEG-4 Video Codec V1
Microsoft MPEG-4 Video Codec V2
Microsoft MPEG-4 Video Codec V3
Microsoft RLE

Video codec information

Delta frames	No
FOURCC code	'div3'
Driver name	DIVXC32.DLL

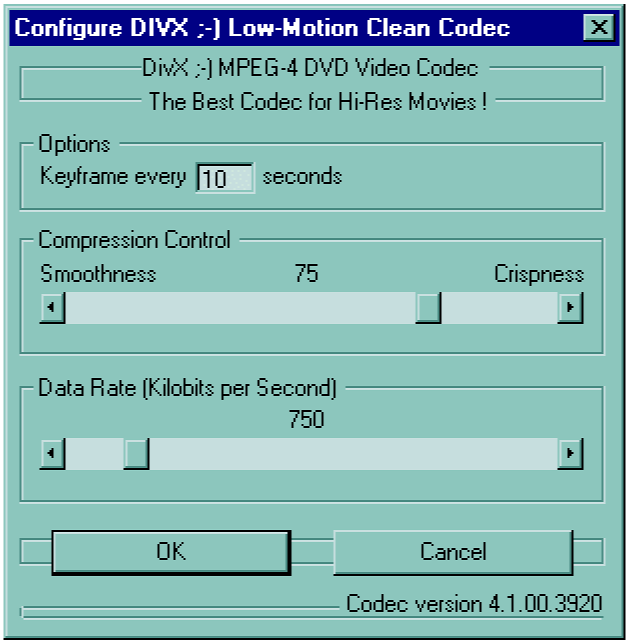
Format restrictions: No known restrictions.

Quality 100

□ Use target data rate of kilobytes/second

□ Force keyframes every frames

Then up pops the codec options. Use one of the bitrate calculators that can be downloaded from Digital Digest to make sure the movie will fit into the size you need. 650kbps will usually fit ANY movie on a single CD, but you can usually get away with between 900 and 1000kbps for a 352 x 240 VCD movie for better quality see my quality guide if you are unsure about correct bitrate settings.



Configure DIVX ;-) Low-Motion Clean Codec

DivX ;-) MPEG-4 DVD Video Codec
The Best Codec for Hi-Res Movies !

Options
Keyframe every 10 seconds

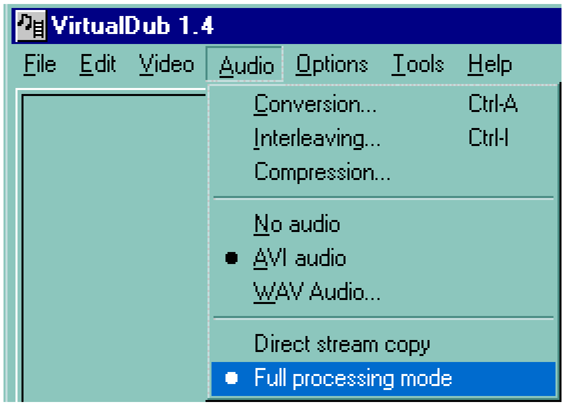
Compression Control
Smoothness 75 Crispness

Data Rate (Kilobits per Second)
750

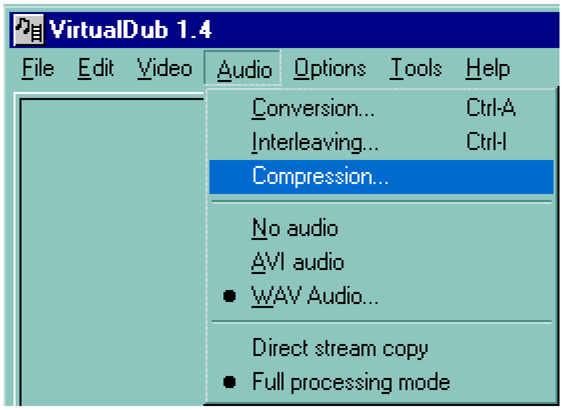
Codec version 4.1.00.3920

Audio Compression

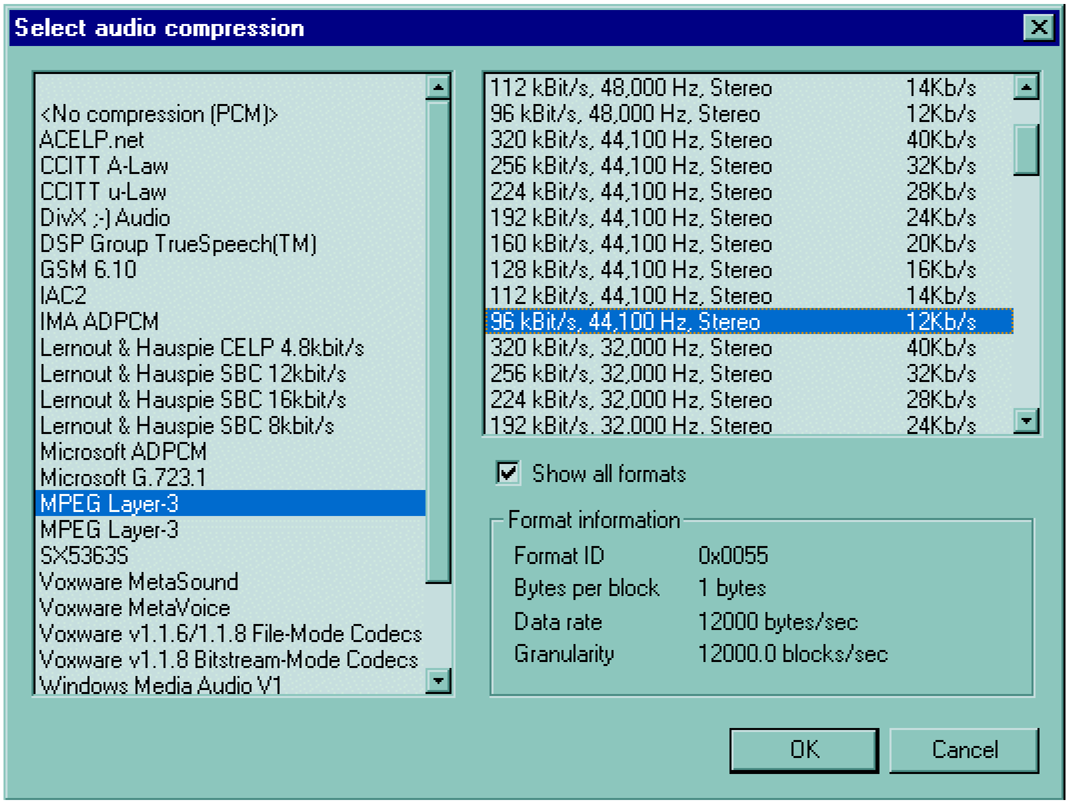
Now select the audio settings by going to: Audio > Full processing mode (so the black dot appears by it).



Then Audio > Compression.



Up pops this box below. Choose Mpeg Layer 3 or DivX ;-) Audio 64kbps, 441 kHz, stereo. And press OK.



Select audio compression

- <No compression (PCM)>
- ACELP.net
- CCITT A-Law
- CCITT u-Law
- DivX ;-) Audio
- DSP Group TrueSpeech(TM)
- GSM 6.10
- IAC2
- IMA ADPCM
- Lernout & Hauspie CELP 4.8kbit/s
- Lernout & Hauspie SBC 12kbit/s
- Lernout & Hauspie SBC 16kbit/s
- Lernout & Hauspie SBC 8kbit/s
- Microsoft ADPCM
- Microsoft G.723.1
- MPEG Layer-3
- MPEG Layer-3
- SX53635
- Voxware MetaSound
- Voxware MetaVoice
- Voxware v1.1.6/1.1.8 File-Mode Codecs
- Voxware v1.1.8 Bitstream-Mode Codecs
- Windows Media Audio V1

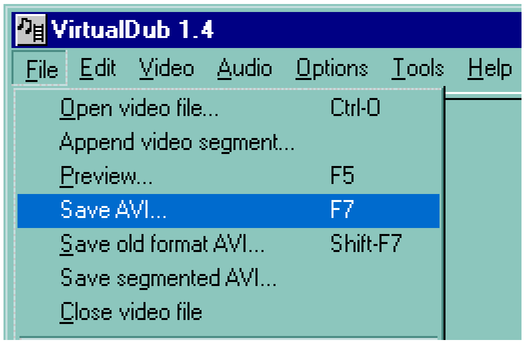
112 kBit/s, 48,000 Hz, Stereo	14Kb/s
96 kBit/s, 48,000 Hz, Stereo	12Kb/s
320 kBit/s, 44,100 Hz, Stereo	40Kb/s
256 kBit/s, 44,100 Hz, Stereo	32Kb/s
224 kBit/s, 44,100 Hz, Stereo	28Kb/s
192 kBit/s, 44,100 Hz, Stereo	24Kb/s
160 kBit/s, 44,100 Hz, Stereo	20Kb/s
128 kBit/s, 44,100 Hz, Stereo	16Kb/s
112 kBit/s, 44,100 Hz, Stereo	14Kb/s
96 kBit/s, 44,100 Hz, Stereo	12Kb/s
320 kBit/s, 32,000 Hz, Stereo	40Kb/s
256 kBit/s, 32,000 Hz, Stereo	32Kb/s
224 kBit/s, 32,000 Hz, Stereo	28Kb/s
192 kBit/s, 32,000 Hz, Stereo	24Kb/s

☑ Show all formats

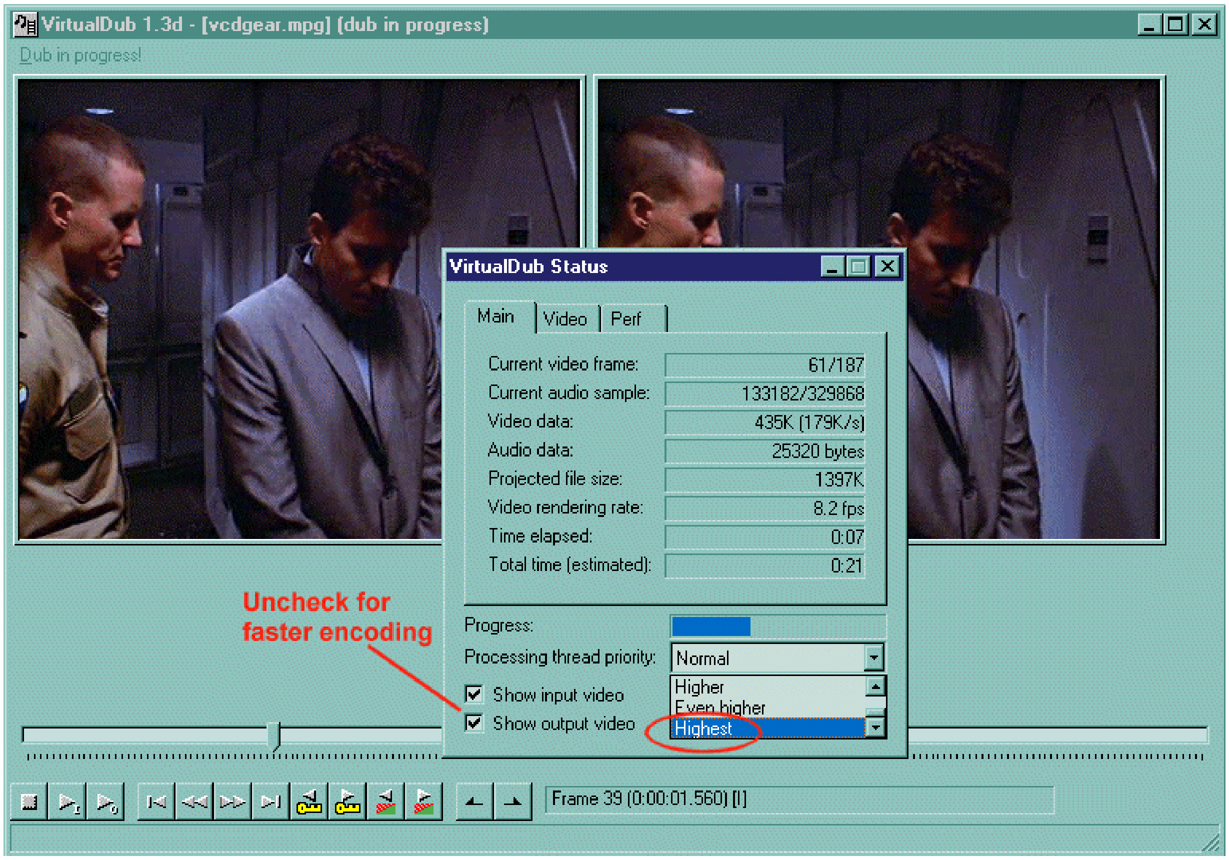
Format information

Format ID	0x0055
Bytes per block	1 bytes
Data rate	12000 bytes/sec
Granularity	12000.0 blocks/sec

That's it! Save AVI...



Here we go! The picture on the left represents the original AVI movie and the one on the right represents the DivX version. For even faster encoding you can set the priority settings to 'Highest' and turn off the 'Show input' and 'Show output' video by unchecking the boxes. These can be turned on at any time to see your progress.



VirtualDub 1.3d - [vcdgear.mpg] (dub in progress)

Dub in progress!

VirtualDub Status

Current video frame: 61/187

Current audio sample: 133182/329868

Video data: 435K (179K/s)

Audio data: 25320 bytes

Projected file size: 1397K

Video rendering rate: 8.2 fps

Time elapsed: 0:07

Total time (estimated): 0:21

Progress:

Processing thread priority: Normal

☑ Show input video

☑ Show output video

Uncheck for faster encoding

Frame 39 (0:00:01.560) [I]

DVD to VCD with Flask Mpeg

Just in case you didn't know, VCD is about the oldest highly compressed video format we have and uses the Mpeg-1 codec. It generally gives the lowest overall quality picture when compared to DivX & SVCD's but can still produce near VHS quality video when split over 2 CD's. Another benefit is it can be played on just about every standalone VCD, SVCD and DVD player. Not only that but a VCD can be played on a PC with only 100Mhz CPU as long as it has the graphics card isn't really weak.

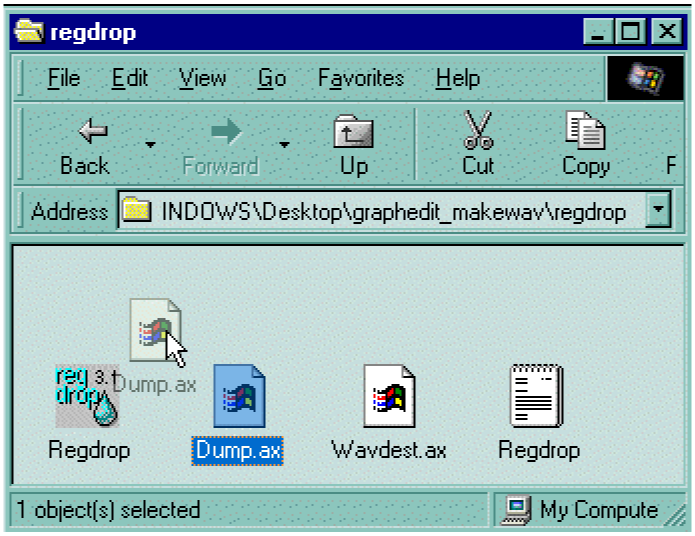
The old way to make VCD's was to use Xing or the Panasonic Mpeg encoder and of course both encoders cost money to buy and were awkward to use with DVD's. But now since Flask Mpeg comes with the BBMpeg plugin there has been no reason to choose that path anymore. The free BBMpeg plugin produces slightly better quality then Xing and about the same quality or better as the Panasonic encoder. This quality is usually as good as most professionally made VCD's too so there is probably no need to change it. The only downside is its not an amazingly fast method.

Various Methods of Making a VCD

Before I go any further in explaining how to use BBMpeg you should know that there are a few other methods for making VCD's but *all* of them require that you Frameserve to the application you want to use! To frameserve with Flask you must check out my frameserving guide Here. Many prefer using TMPGEnc for making both SVCD's and VCD's and I see no reason to frameserve to any other application since its the best there is both for quality and features! I think it is a waste of my time to explain all of the methods you can use - you want the best ones, right :o). For more details on setting up TMPGEnc after you have frameserved to it just follow my guide: 'AVI to VCD'.

In the good old days it was always very hard to convert a DVD to VCD without problems. These days its becoming much easier to do. I remember doing the Matrix was a big pain. I had to split the Vobs into parts and delete repeated sequences, slice it into parts - it took ages to work it all out! Nowadays you just open it in Flask and press go! But in my search I found one of the easiest method was to use the Panasonic Mpeg Encoder. So I think this method is worth a mention only because the same method works with many applications including TMPGEnc and you may find it useful to know sometime. By using this method it *may* be possible to cut out the frameserving completely and convert straight to SVCD or VCD! In reality, because of the way DVD's are made, this option is probably not the way to go :o(.

All it required is you install the WinDVD Audio Filter and the Ligos Mpeg-2 Decoder Ligos Mpeg-2 Video Filter. Easiest way to install them is to use RegDrop. Just drag them over the utility like this:



Or you can just use the DOS line command Regsvr32.exe (i.e.. regsvr32 "C:\folder\filename.ax").

After they are installed you should be able to open any DVD Vob file in TMPGEnc or Panasonic Mpeg Encoder and encode them! The first problem you will see is that Windows 9x only handles files of 4GB at a time! So if your DVD is larger than 4GB it wont do a whole movie. AVISynth is one solution to this because now you have the WinDVD and Ligos filters installed it will let you open all the Vob files in a long list. Check out how to use AVISynth Here.

Flask Plug-in's

Flask also lets you use a few other plug-IN's other than BBMpeg to convert to VCD. The most common one is the Panasonic MPEG1 Encoder Plug-In for Adobe Premiere5.x. Install the plugin like you would with Premiere. *You don't actually need Adobe Premiere, just install the plug-in!* When installing the plug-in choose a folder to store it; it doesn't really matter where. Once its installed find the plugin, it's usually called: cm-mpeg-pwi2.0e.prm. Copy and paste it into the Flask Mpeg folder and rename it to panplug.cm.flask. Actually it doesn't matter what you call it, Flask tries to use any plug-in that are named to something.**cm.flask.**

All you need to do then is select the Panasonic option in Flask instead of BBMpeg. Up will pop the selection box, choose the Panasonic's VCD option - Bingo! You're done! The problem with the Premiere plugin is the quality is not quite as good as BBMpeg, although it's still nearly as good. It can sometimes also mess up some of your system codecs so you may need to reinstall some of them after if you get problems.

Okay, you've seen the rest not see one of the best!

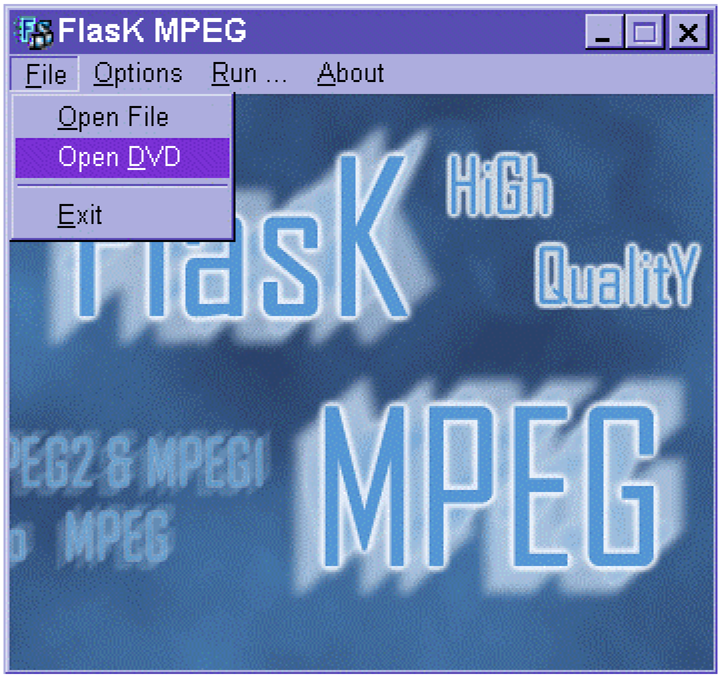
The BBMpeg Method Explained

Before I start you will need:

Flask Mpeg 0.594 (or greater)

As always, you must have the DVD files ripped and decoded to your Hard Disk first if you want to convert a DVD. You can use CladDVD or VobDec to do this but I prefer SmartRipper. They are all legal (because they do not use stolen DVD codes) so there is nothing to worry about. Flask has two basic conversion modes:

> **Open File:** This will open any mpeg file and try to convert it to whatever you want.
>
> **Open DVD:** This is a special mode that reads a DVD in the same way a DVD player would. This means you will be able to select to convert with subtitles; it will also make sure it only rips one angle in a multi-angle DVD. This prevents repeated scenes spoiling your movie!

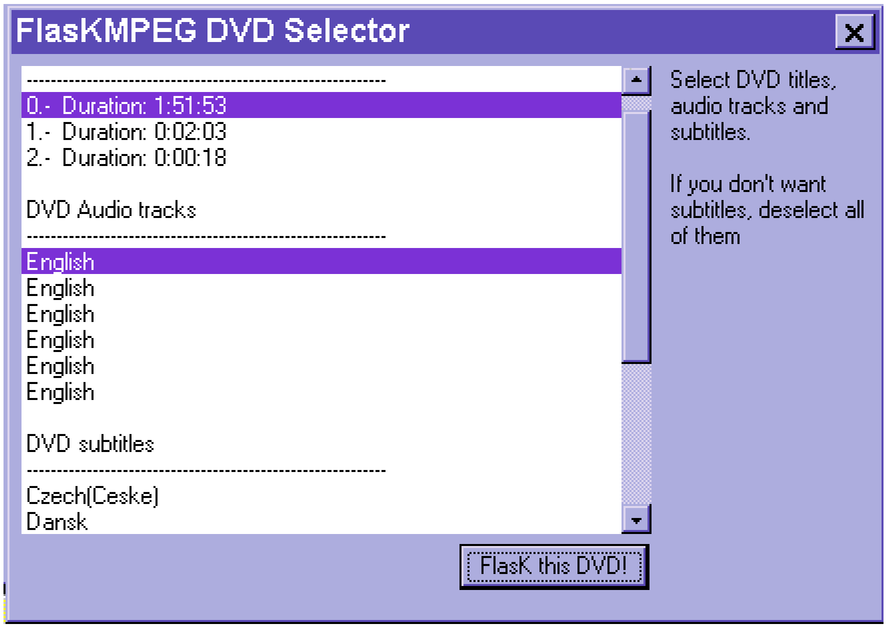


If you want to use Flask's 'Open DVD' mode you must copy the files exactly as they are on the DVD. This means you must rip them with SmartRippers file mode, do *not* use its movie mode. With VobDec or CladDVD you must turn off the multi-angle functions first - DeCSS is old and cannot do all DVD anymore, so don't use it!

Also, if you are using Flask's 'Open DVD' mode you must copy the IFO file that belongs to the movie you wish to convert. This will be called by the same name as the first movie file. For example, if the first movie file is called **VTS_01_0.VOB** the IFO file to copy with it will be **VTS_01_0.IFO**. Remember, the main movie will not always have the same name. So if first file of the main movie were called **VTS_02_0.VOB** then the IFO file to copy with it will obviously also be **VTS_02_0.IFO** For more details on this subject read the article called "*Introduction to the Structure of a DVD*" in the appendix section.

OPEN THE FILE WITH 'Open DVD'

Select 'Open DVD' and find the .IFO file for your movie. Up will pop something like the picture below:



FlasKMPEG DVD Selector

```
------------------------------------------------------------
0.- Duration: 1:51:53
1.- Duration: 0:02:03
2.- Duration: 0:00:18

DVD Audio tracks
------------------------------------------------------------
English
English
English
English
English
English

DVD subtitles
------------------------------------------------------------
Czech(Ceske)
Dansk
```

Select DVD titles, audio tracks and subtitles.

If you don't want subtitles, deselect all of them

Select the movie Duration, in this case 1.51.53. This will usually be the first one in the list, but you can usually see from the length which one to choose. Its a good idea to take a note of this time for later on. If you are not converting using the IFO file then get the length of the video from your PC DVD player (i.e. PowerDVD etc).

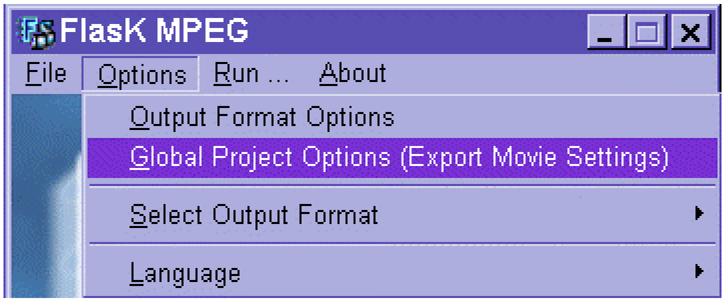
Next choose the language. Obviously they cannot ALL be English so choose the first and encode one min of the film and listen to it. If its not English choose the next in the list, and the next and the next etc., until you find the correct one. Or open the DVD in a DVD player and see the order they are in, usually it will be the same.

Lastly we have subtitles. I don't usually select any because I don't want them on my movie. If you choose subtitles then you will not be able to turn them off, they will ALWAYS be on your movie! Flask is not always able to do subtitles correctly, so again, you will have to try it and see. There are other ways to get subtitles, though; why not check out my subtitle ripping section for this.

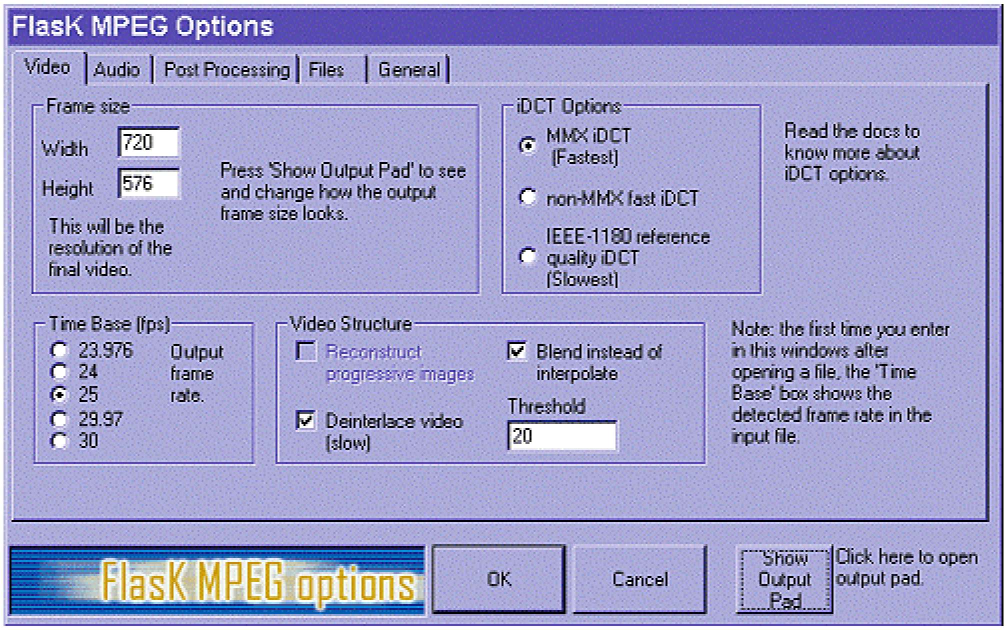
Now Press FlasK this DVD!

GLOBAL PROJECT OPTIONS

Select Global Project options.



VIDEO TAB


FlasK MPEG Options

Video | Audio | Post Processing | Files | General

Frame size

Width 720

Height 576

This will be the resolution of the final video.

Press 'Show Output Pad' to see and change how the output frame size looks.

iDCT Options

● MMX iDCT (Fastest)

○ non-MMX fast iDCT

○ IEEE-1180 reference quality iDCT (Slowest)

Read the docs to know more about iDCT options.

Time Base (fps)

○ 23.976
○ 24
● 25
○ 29.97
○ 30

Output frame rate.

Video Structure

□ Reconstruct progressive images

☑ Blend instead of interpolate

☑ Deinterlace video (slow)

Threshold 20

Note: the first time you enter in this windows after opening a file, the 'Time Base' box shows the detected frame rate in the input file.

Click here to open output pad.

Frame Size: In the Width and Height sections you can put the size you wish your final video to be. If you wish to crop your movie too you can select the 'Show Output Pad' at the bottom right. BUT your final movie size *must* be either 352 x 288 if your VCD is PAL or 352 x 240 if it is NTSC. So crop and resize how you like but make sure the final image is the right size.

Time Base (fps): All PAL movies (European) are 25 frames per second (fps) no other setting will work. All North American movies are NTSC which means they are 29.97 fps, no other setting will work!

iDCT Options: Just leave these alone its overkill to use IEEE and the quality will not look better.

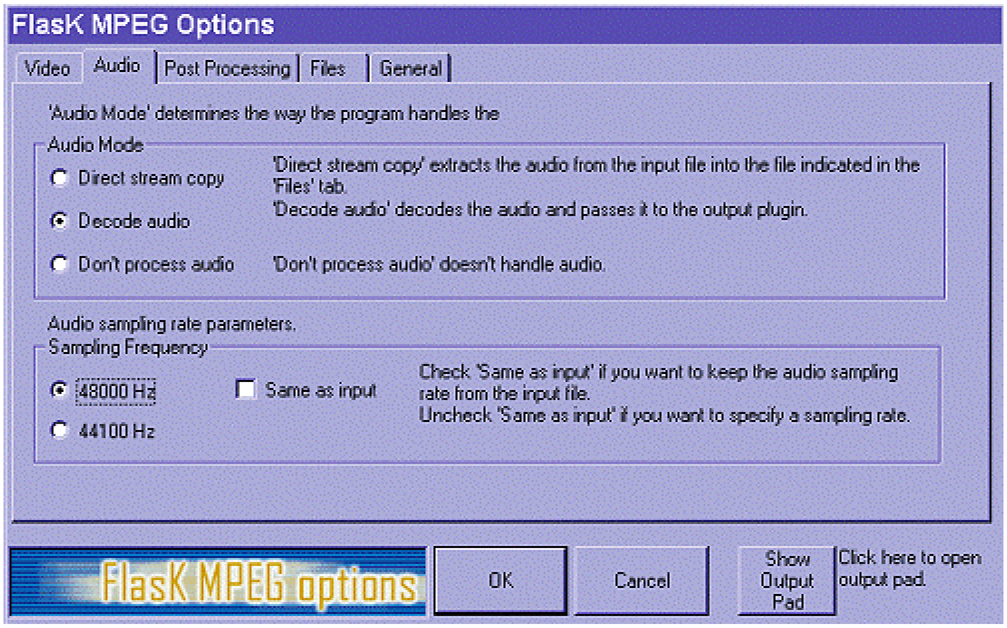
Video Structure: Do *not* check the deinterlace button unless you really need it because the video will decode very slow. For a full explanation of the interlace problem, check out my article: "Video Formats: NTSC & PAL / Telecine" in the appendix section.

Deinterlace video

For this kind of Deinterlacer blend gives best quality. The Threshold setting is basically how much it blurs problem lines. The lower the number the more it blurs it. If the interlace problem is really bad you should use a setting of 1-5 instead of 20. If its not very bad you could use 15-20.

AUDIO TAB

On to the Audio tab select 'Decode audio' if you want sound. You need to choose 44100Hz to be compatible with VCD standards.



'Audio Mode' determines the way the program handles the

Audio Mode

○ Direct stream copy
● Decode audio
○ Don't process audio

'Direct stream copy' extracts the audio from the input file into the file indicated in the 'Files' tab.
'Decode audio' decodes the audio and passes it to the output plugin.
'Don't process audio' doesn't handle audio.

Audio sampling rate parameters.

Sampling Frequency

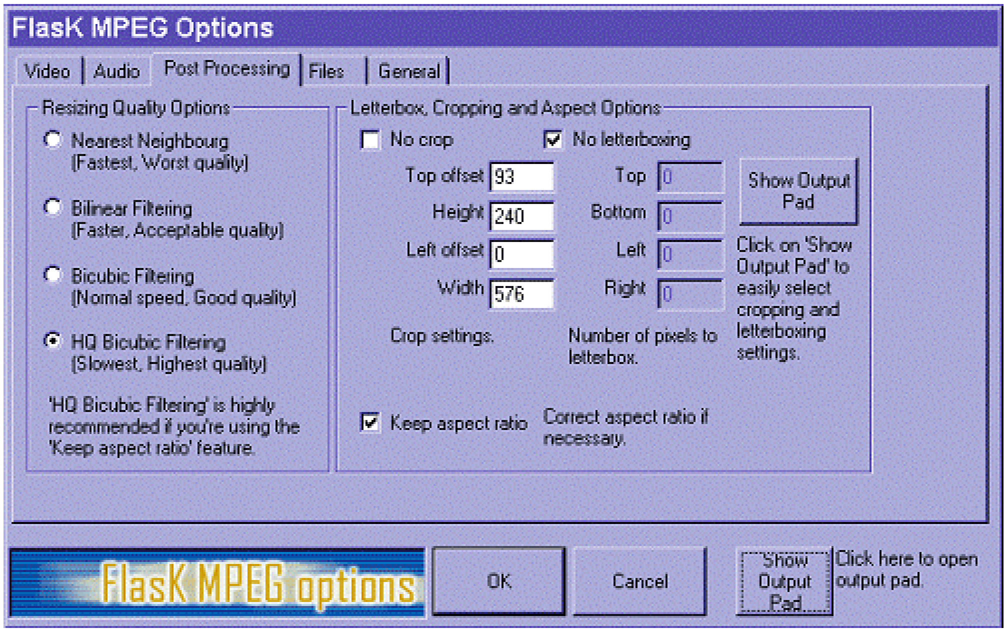
● 48000 Hz
○ 44100 Hz
□ Same as input

Check 'Same as input' if you want to keep the audio sampling rate from the input file.
Uncheck 'Same as input' if you want to specify a sampling rate.

If you just want to copy the DVD's AC3 audio across instead of converting it you can use the 'Direct Stream copy' option. This could be the way to go if you have synch problems. But that way the audio must be decoded to wave using another method such as explained in my Mpeg2avi guide. Then you can use something like TMPGEnc to encode the audio to Mp2 and multiplex it to the video file. It is also possible to fix synching by opening the video in Cool Edit or another audio editing tool to stretch the audio a bit. Then you can again encode and multiplex. All this is a real pain in the neck so I hope you never need to try any of them.

POST PROCESSING TAB

This section deals almost exclusively with resizing. Never use 'Nearest Neighbouring' unless you are not resizing the picture because the quality is crud. Contrary to popular belief Bilinear looks just as nice as Bicubic, except it is twice as fast. JASC (makers of Paint Shop Pro) recommend Bilinear for shrinking images and Bicubic for enlarging them. But use what you think looks best.



Resizing Quality Options

○ Nearest Neighbourg (Fastest, Worst quality)

○ Bilinear Filtering (Faster, Acceptable quality)

○ Bicubic Filtering (Normal speed, Good quality)

● HQ Bicubic Filtering (Slowest, Highest quality)

'HQ Bicubic Filtering' is highly recommended if you're using the 'Keep aspect ratio' feature.

Letterbox, Cropping and Aspect Options

□ No crop ☑ No letterboxing

Top offset 93
Height 240
Left offset 0
Width 576
Crop settings.

Top 0
Bottom 0
Left 0
Right 0
Number of pixels to letterbox.

Show Output Pad

Click on 'Show Output Pad' to easily select cropping and letterboxing settings.

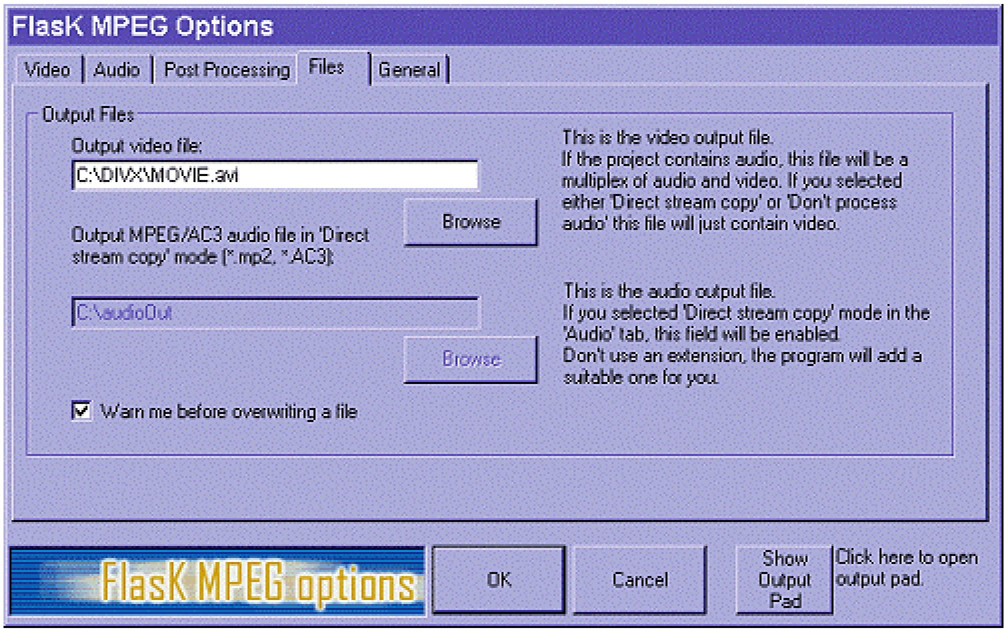
☑ Keep aspect ratio — Correct aspect ratio if necessary.

Show Output Pad — Click here to open output pad.

Keep aspect ratio: PAL users should always tick this box unless you know you don't need it. This is even more important with Widescreen DVD's. If you use NTSC DVD's the image may become stretched slightly wrongly. If you notice this uncheck the 'keep aspect ratio' and work out the ratio yourself. See the article *"Resizing DVD's with Correct Aspect Ratios"* in my appendix. Don't forget that the final movie must be either 352 x 288 or 352 x 240 respectively.

Crop & Letterboxing: All the settings for cropping and letterboxing the DVD can be entered here or the output pad can be used. For detailed information on how to resize a movie in Flask read section 2nd of this guide: *"resizing the video"*. I suggest you don't mess about with all that though, just check the no crop and the no letterboxing options.

FILES TAB



Output Files

Output video file: C:\DIVX\MOVIE.avi

This is the video output file.
If the project contains audio, this file will be a multiplex of audio and video. If you selected either 'Direct stream copy' or 'Don't process audio' this file will just contain video.

Output MPEG/AC3 audio file in 'Direct stream copy' mode (*.mp2, *.AC3): C:\audioOut

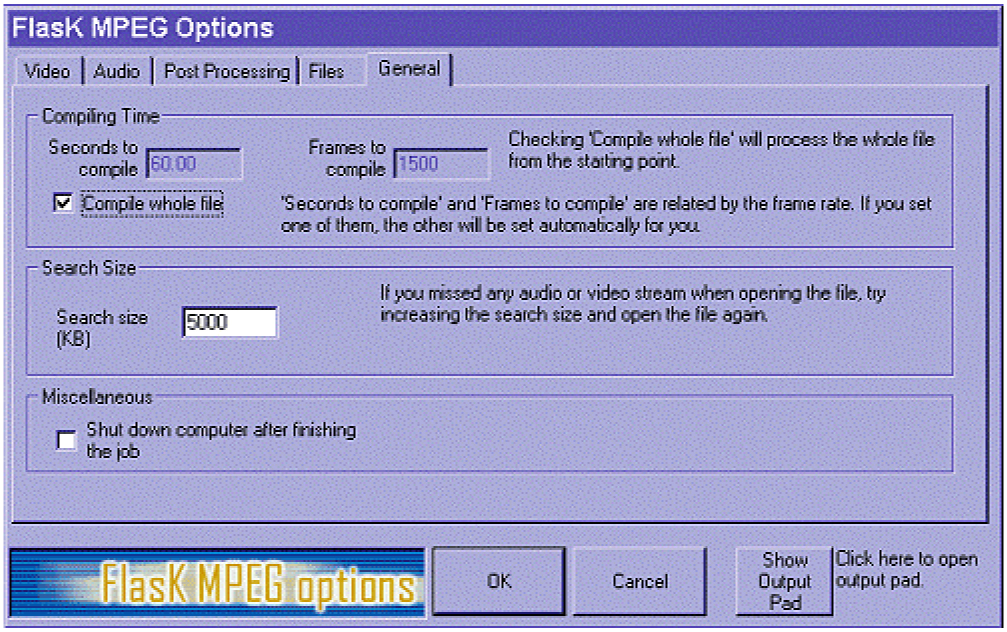
This is the audio output file.
If you selected 'Direct stream copy' mode in the 'Audio' tab, this field will be enabled.
Don't use an extension, the program will add a suitable one for you.

☑ Warn me before overwriting a file

Click here to open output pad.

Choose where you want to save your final movie. The audio save option is grayed out because you are encoding the video with audio in it. If you selected 'Direct Stream Copy' on the previous Audio Tab then you could say where you wanted it saved.

GENERAL TAB



Compiling Time

Seconds to compile 60.00

Frames to compile 1500

Checking 'Compile whole file' will process the whole file from the starting point.

☑ Compile whole file

'Seconds to compile' and 'Frames to compile' are related by the frame rate. If you set one of them, the other will be set automatically for you.

Search Size

Search size (KB) 5000

If you missed any audio or video stream when opening the file, try increasing the search size and open the file again.

Miscellaneous

☐ Shut down computer after finishing the job

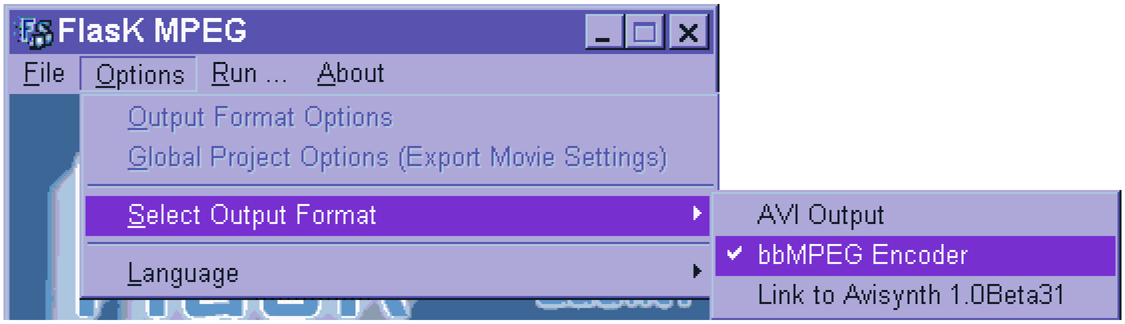
Click here to open output pad.

Compiling Time: This is important. *You must tell Flask how long the movie is.* This is why I asked you to take note of the movie length. You must work out exactly how long the movie is. If you don't tell Flask the movie length could end up 300,000 frames even if your movie is only 30 seconds long! As you can imagine this is a big waste of time.

There are 25 frames to every second of a PAL video and 30 for every NTSC. So lets say your PAL movie was one hour long. First you would work out how many seconds on an hour of video (i.e: 60 x 60 =3600 seconds). And then you would multiply that by 25 (i.e: 25 x 3600 = 90,000). You can also encode a small test clip to see the quality before you start. But I still recommend you shut down and restart Flask just before you start to encode a long movie to reduce chances of it crashing.

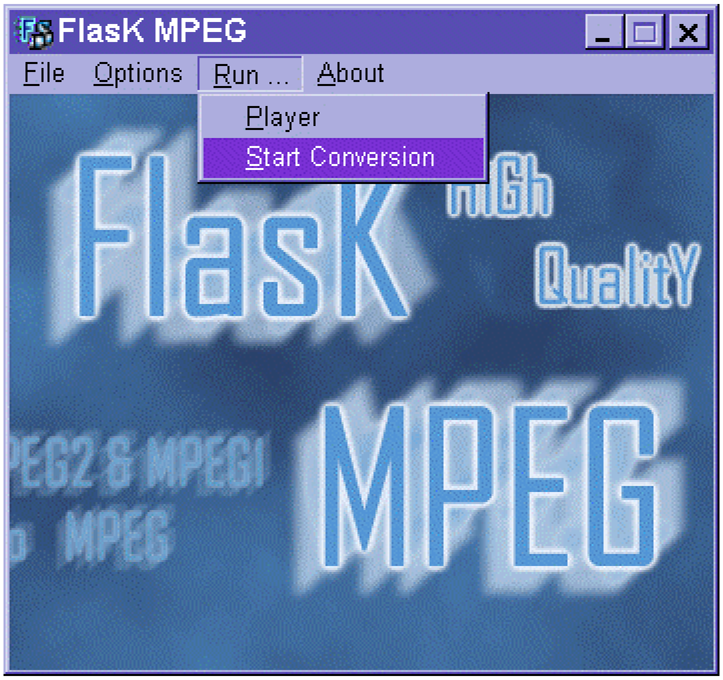
Search Size: This searches the DVD for the audio. As it says, if you have problems make the number bigger. If all else fails and you cannot find audio you can try one of the other methods I explain in the section:*"DVD Audio Extraction"*. This means you will need to encode the audio to Mp2 and multiplex it with the original with TMPGEnc or something like that. There is no fail-safe way to convert a DVD yet.

SELECT THE BBMPEG PLUG-IN

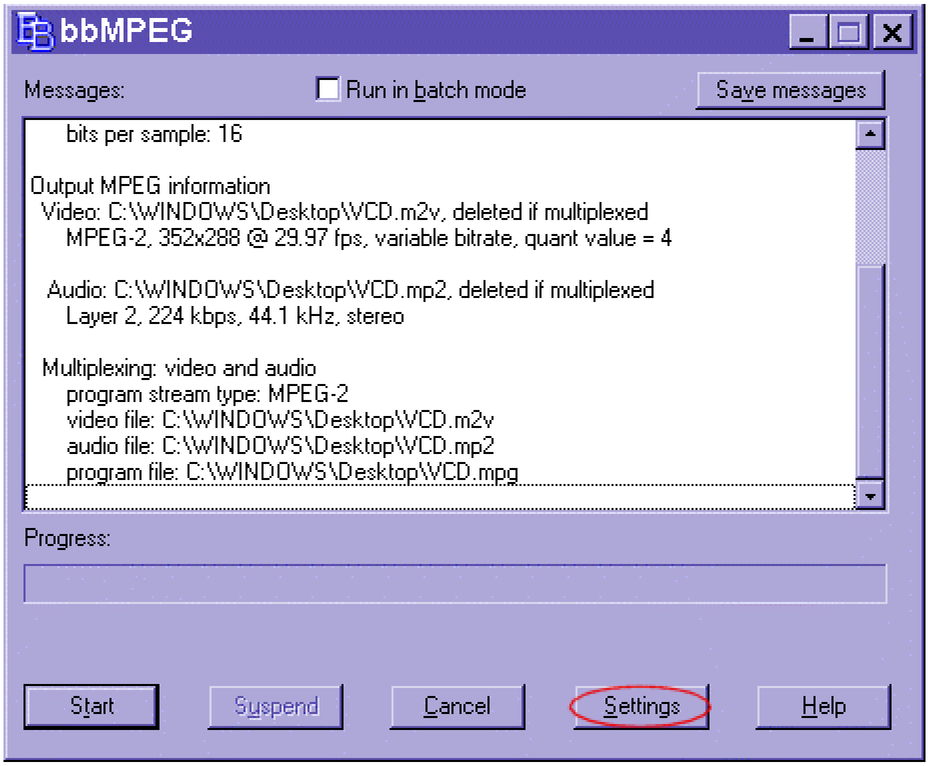


PRESS GO!

Once you have finished setting up your movie just go to Run > Start Conversion.

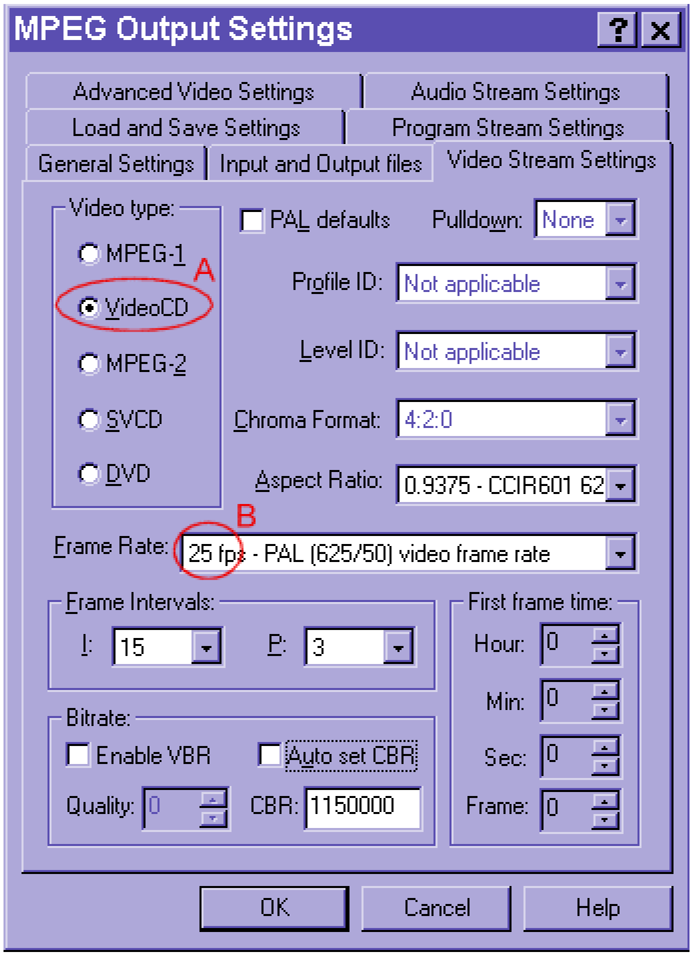


Up will pop BBMpegs info window showing the settings you put into Flask. Next choose the 'settings' option.



```
     bits per sample: 16

Output MPEG information
  Video: C:\WINDOWS\Desktop\VCD.m2v, deleted if multiplexed
     MPEG-2, 352x288 @ 29.97 fps, variable bitrate, quant value = 4

  Audio: C:\WINDOWS\Desktop\VCD.mp2, deleted if multiplexed
     Layer 2, 224 kbps, 44.1 kHz, stereo

  Multiplexing: video and audio
     program stream type: MPEG-2
     video file: C:\WINDOWS\Desktop\VCD.m2v
     audio file: C:\WINDOWS\Desktop\VCD.mp2
     program file: C:\WINDOWS\Desktop\VCD.mpg
```

Go to the Video Stream Settings option and choose the VideoCD (**A**) option below. If your movie is NTSC change the 25fps option (**B**) to 29.976. That's it!



Video type:
○ MPEG-1
● VideoCD
○ MPEG-2
○ SVCD
○ DVD

□ PAL defaults
Pulldown: None
Profile ID: Not applicable
Level ID: Not applicable
Chroma Format: 4:2:0
Aspect Ratio: 0.9375 - CCIR601 62
Frame Rate: 25 fps - PAL (625/50) video frame rate

Frame Intervals:
I: 15
P: 3

Bitrate:
□ Enable VBR
□ Auto set CBR
Quality: 0
CBR: 1150000

First frame time:
Hour: 0
Min: 0
Sec: 0
Frame: 0

Now here is where you must use some restraint; the VCD format is very exacting. When we used the 'VideoCD' option BBMpeg sets the Whitebook standard. VCD doesn't allow you to set your own bitrate it must be a constant bitrate (CBR) of 1150kbps (1150000). The I, P and B frames are set to 15/3. The packing is 2324 and the audio is 16bit 44100Hz stereo 224kbps Mpeg-1 Layer 2 (Mp2).

BBMpeg looks very difficult to use at first because it has lots of options but, at least for VCD, you cannot use them. The only ones we can safely change are the framerate to either 25fps or 29.97fps and what processor optimizations to use in the General Settings tab.

Bingo! Thats it! Wait for the movie to finish, it should be anything upto 1480 MB in size to fit on 2 CD's! Don't worry, VCD's and SVCD's are not written to a CD-R in the same way we would write other data! CD-R's actually waste about 90MB in order to produce a filing system that we can use in Windows. This means that you can fit an Mpeg video of about 740MB on a single 650MB CD-R or 820MB for playing in a standard VCD player. The same idea has aways applied to any audio CD's you make. As long as it is 74 mins it doesn't matter how many megabytes the audio takes on your hard disk. Either way your CD-R burning software will usually tell you if the Mpeg is too large.

Nero and Easy CD Creator are the best CD-R 'Burning' software for making both VCD's and SVCD's. In fact Easy CD is the way to go if you are buying something for doing VCD's becasue it will even allow you to create VCD menus too! I will write some short guides on using these applications too soon, I hope.

But wait! Before we can burn anything we need to chop our movie into two parts. For details on how to do this check out my 'Cutting and joining Mpeg's guide' Here. Once that's done you can burn them to CD and you're done!

AVI to VCD with TMPGEnc

No doubt about it! If you want the best quality SVCD or VCD then TMPGEnc is the way to go and what's more, its free baby! The bad news is only the older versions gave us full support of SVCD and VCD. This is a very important point, *only* use TMPGEnc upto version 12a don't download version 12b or any later versions unless you want to buy the product of course! And you will need the English patched version too or it will be unreadable.

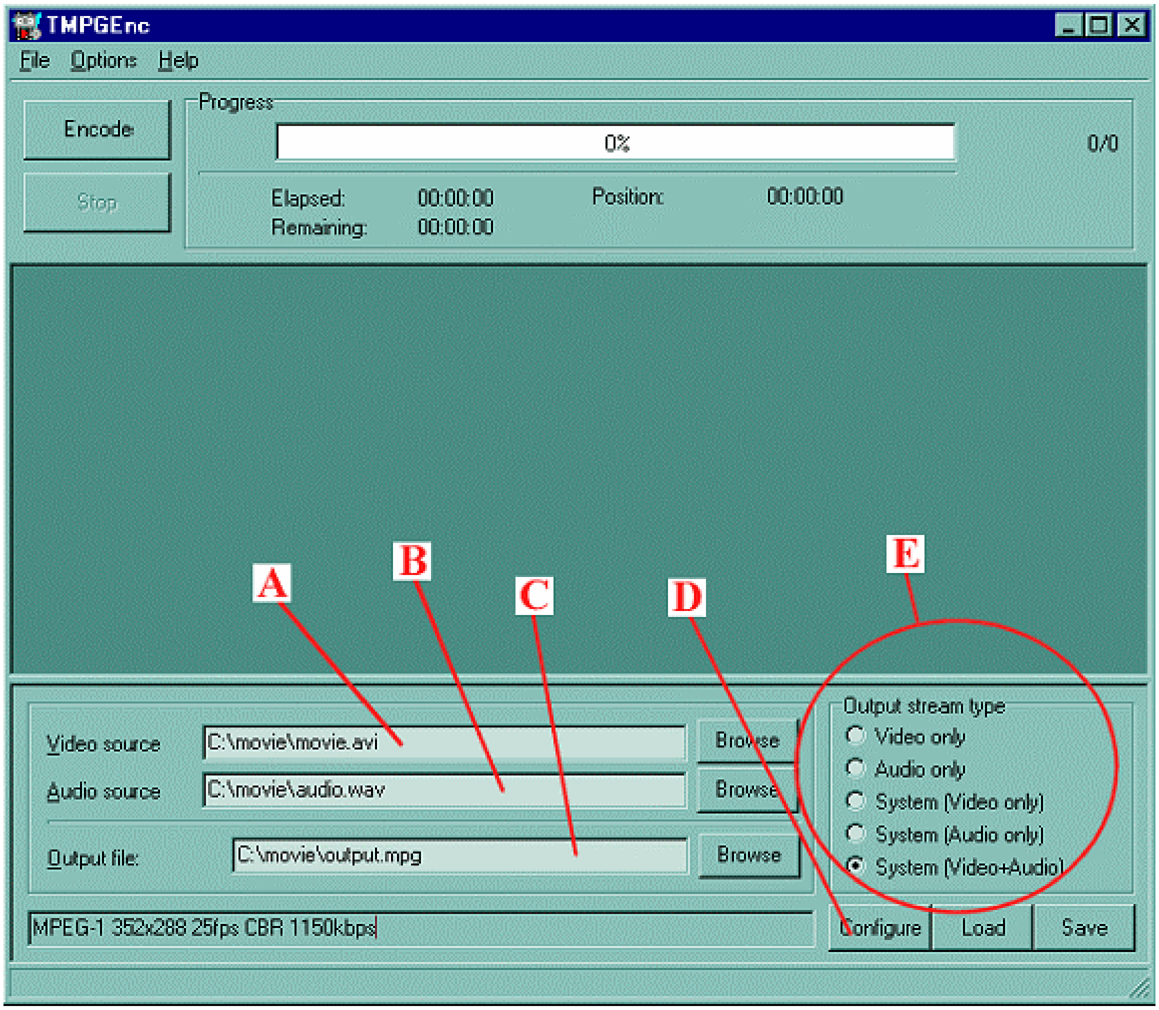
If you intend on converting a DVD into a VCD with TMPGEnc then you probably should frameserve it with Flask Mpeg. If you intend on just using the Ligos and WinDVD filters as explained breifly in my DVD to VCD guide, then you may need to extract the audio first with Graphedit and add it to the movie seperately or audio may not appear.

This guide basically assumes that you are converting an AVI file but all the same settings will apply to any file you can open in TMPGEnc or frameserve to it.

Before I start you will need:

TMPGEnc 12a

Load TMPGEnc. As usual any new utility looks hard to use at first but is actually not too bad. Take a look at the picture below.



0% 0/0

Elapsed: 00:00:00 Position: 00:00:00
Remaining: 00:00:00

Video source: C:\movie\movie.avi
Audio source: C:\movie\audio.wav
Output file: C:\movie\output.mpg

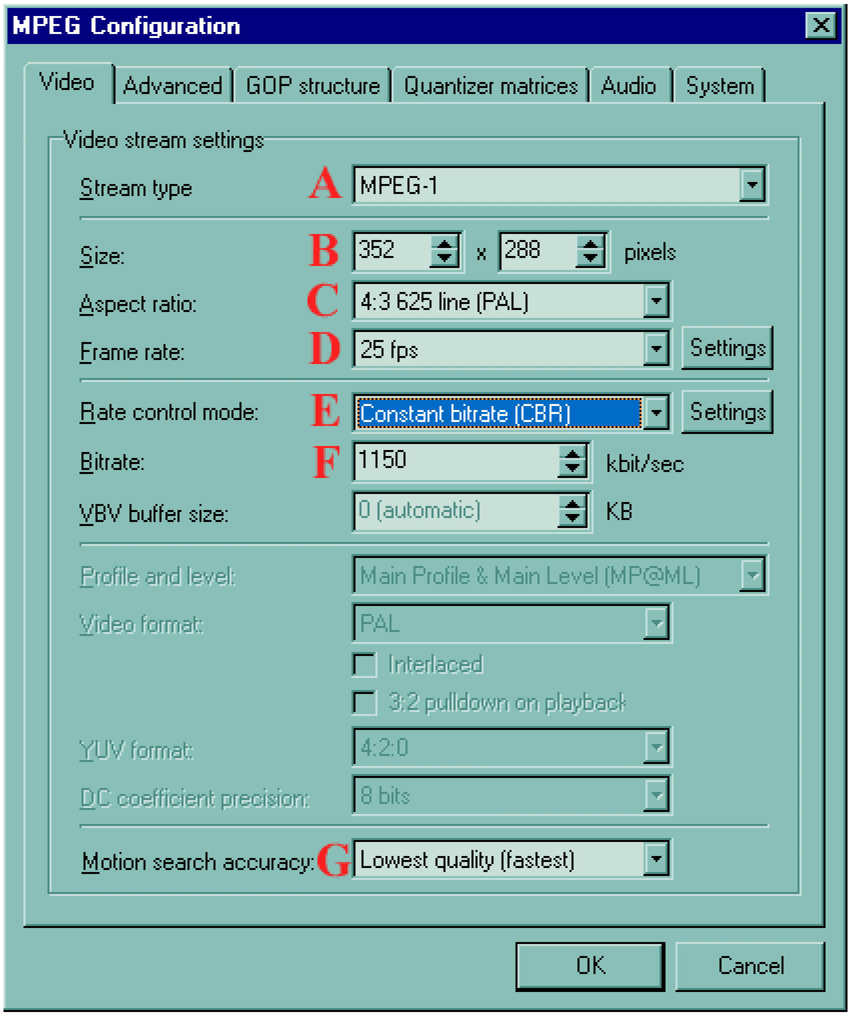
Output stream type
○ Video only
○ Audio only
○ System (Video only)
○ System (Audio only)
● System (Video+Audio)

MPEG-1 352x288 25fps CBR 1150kbps

First we will choose the AVI file we wish to make into a VCD. Do this by hitting the 'video source' Browse button (**A**); (If you are frameserving you will choose the *.avs file instead of the *.avi). The audio source (**B**) is not needed unless you have a seperate audio track that needs multiplexing to the video file. This is sometimes the case if you needed to normalize or edit the audio for any reason but keep in mind that TMPGEnc can also normalise audio for us. Finally we can choose where we wish to save our final VCD Mpeg file (**C**).

TMPGEnc is great because it can do everything we need. You can use it to convert just audio alone to Mp2 by not selecting any video. You can use it to convert just video to Mpg by saying 'Video only'. You can even use it to cut and join Mpeg files in any way you like. The selection boxes (**E**) allow us to choose what TMPGEnc will give us. Always choose: System (Video+Audio) if you want an Mpg with video and audio.

The Load and Save buttons of the 'output sytream type' just let you save all the settings you used to create your movies. This can be very useful if you need to set custom settings for your movies, but we will not need any in this guide. If you download anyone elses settings you can load them using these buttons too. Anyway, lets configure TMPGEnc to make our VCD. Press the configure button (**D**), and this box appears:



MPEG Configuration

Video stream settings

Stream type **A** MPEG-1

Size: **B** 352 x 288 pixels

Aspect ratio: **C** 4:3 625 line (PAL)

Frame rate: **D** 25 fps

Rate control mode: **E** Constant bitrate (CBR)

Bitrate: **F** 1150 kbit/sec

VBV buffer size: 0 (automatic) KB

Profile and level: Main Profile & Main Level (MP@ML)

Video format: PAL

□ Interlaced

□ 3:2 pulldown on playback

YUV format: 4:2:0

DC coefficient precision: 8 bits

Motion search accuracy: **G** Lowest quality (fastest)

The Video tab are the most important ones, so here is what the options are all about.

(**A**) Obviously since we are making a VCD we must select Mpeg-1.

(**B**) Then we must choose a Video CD standard size. No other sizes will work except: 352 x 240 for NTSC movies and 352 x 288 for PAL movies. Even if you are not making a perfect whitebook standard VCD then its a good idea to choose one of these settings anyway because its a nice standardized format :).

(**C**) Aspect ratio should be set to 4:3 PAL for PAL VCDs or 4:3 NTSC for NTSC VCDs.

(**D**) The framreate must be set to 25fps for PAL VCDs or 29.97fps for NTSC VCDs.

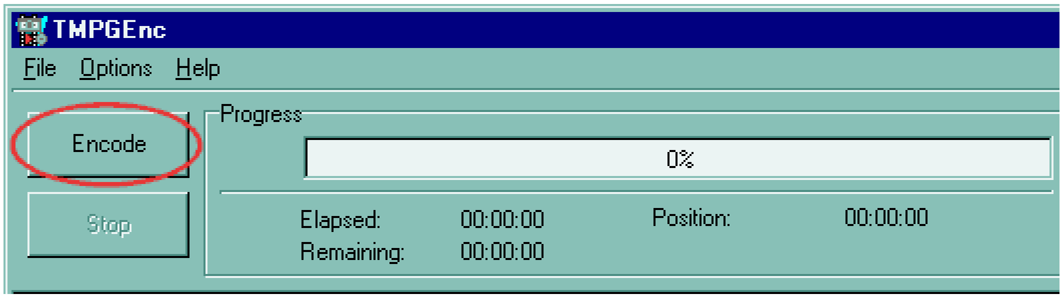
(**E**) You must use Constant bitrate (CBR) for VCDs able to play on standalones.

(**F**) You must always use a bitrate of 1150 kbit/sec.

The rest of the options are greyed out because they do not apply.

(G) For motion search accuracy I suggest you either use Normal or High Quality. It doesn't matter which one you choose apart from one obviously looking slightly better. If you are a quality freak you can set the thing on highest quality instead. Obviously the higher the quality the longer it will take to compress. There is not a huge difference between the normal and higher quality modes so experiment to see what you think is best before you do a full movie.

That should be all you need to set! Press OK to get out of the Configire section and press the large Encode button - Bingo!

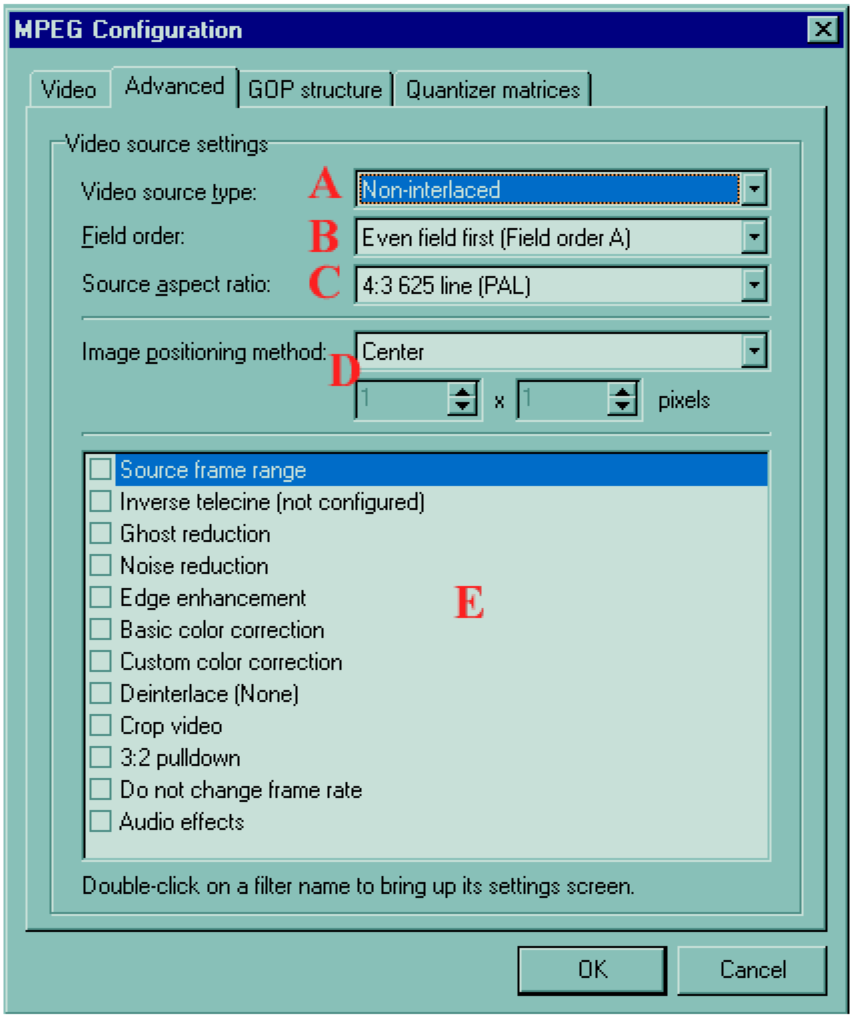


Elapsed: 00:00:00
Remaining: 00:00:00
Position: 00:00:00

ADDITIONAL SETTINGS

For a VCD you can usually keep all other setting default and everything should be fine. But these extra filters are absolutely wonderful and no other mpeg encoder I've seen has come close to the options offered with TMPGEnc! This includes the big names such as the Panasonic Mpeg Encoder, Xing and Ligos - they are total rubbish by comparison! Unfortunately to explain how to use everything would take many articles and a lot of time. So instead I will give a breif overview and let you play about with them yourselves.

Lets go back to Configuration and click on the next tab, the 'Advnced' settings:



Video source settings

Video source type: Non-interlaced

Field order: Even field first (Field order A)

Source aspect ratio: 4:3 625 line (PAL)

Image positioning method: Center

1 x 1 pixels

- □ Source frame range
- □ Inverse telecine (not configured)
- □ Ghost reduction
- □ Noise reduction
- □ Edge enhancement
- □ Basic color correction
- □ Custom color correction
- □ Deinterlace (None)
- □ Crop video
- □ 3:2 pulldown
- □ Do not change frame rate
- □ Audio effects

Double-click on a filter name to bring up its settings screen.

(**A**) Non-interlaced or interlaced, this option is only used to make the resolution larger or smaller. Just use non-interlaced.

(**B**) Field order. With an interlaced video the picture is made up in two parts one of all odd lines and the other of all even lines, see my interlace article Here for more details on this subject. Basically it shouldn't make any difference to your video, it will either do it in A field order starting with the top field first (i.e. AB AB AB), or B order first starting with the bottom field (i.e. BABA).

(**C**) This lets you change the Aspect ratio of the video to make it either widescreen or normal screen. You will always use 4:3 for VCD's. An aspect ratio of 1:1 means no change at all and 16:9 is usual for widescreen.

(**D**) The center option helps to centralize your video if it is has been resized or cropped previously.

(**E**) Here is where things get really interesting. There are a whole bunch of filter that can be applied to the video before the encoder transforms it into Mpeg. Many of these can be done to the AVI in VirtualDub, but isn't it nice of the programmer of TMPGEnc to include so many options :). To select them just tick beside the boxes, *but* to configure them you must *double-click* on the selection you picked or the options box wont appear. Again there are too many settings to explain in detail so I will just highlight a few of the most useful ones:

Crop Video: This is the one you will probably use most of all. It crops the sides of your AVI to the size you want it, its very easy to use. Double-click on it and up will pop a preview of your video. Then increase the numbers in the Top, Bottom, Left and Right boxes to cut the video to the sizes you need. It also gives you the option of keeping the size but just blacking out the edges like a letterboxed video.

Audio Effects: The main point of this is to make video that has low sound volume louder. It does this by normalizing the wave file before it converts it to MP2. Normalizing, as I'm sure you've read in my guides before, is a way of amplifying audio to go as loud as possible without 'clipping' off the scale and causing corrupted parts. I'd say always use a normalization of 90-100%; I always use 100. They've also thrown in the option to fade in and out at the start and end of the movie which is more useful for small video clips that tend to end suddenly.

Edge Enhancement: Another beautiful feature that allows you to control the general sharpness of the picture without significantly effecting the soft tones. If used correctly this can be set to produce some very nice mpeg videos that are both sharp and clear.

Basic Color Correction: This is useful if the original AVI has a green or reddish tinge to it. You can change the brightness and contrast too. The option to adjust the Gamma is usually for better PC viewing and may make the picture look too light on a Normal TV; you must try and play a clip on TV or whatever you intend to view it with before you decide on the best colour corrections to do. The gamma is different from the brightness and contrast in that it changes the midtones rather than how bright the colours start and stop. I will write more about colour correction in other guides and you can learn a bit about colour in my 'More Filter Tricks' section Here if you want.

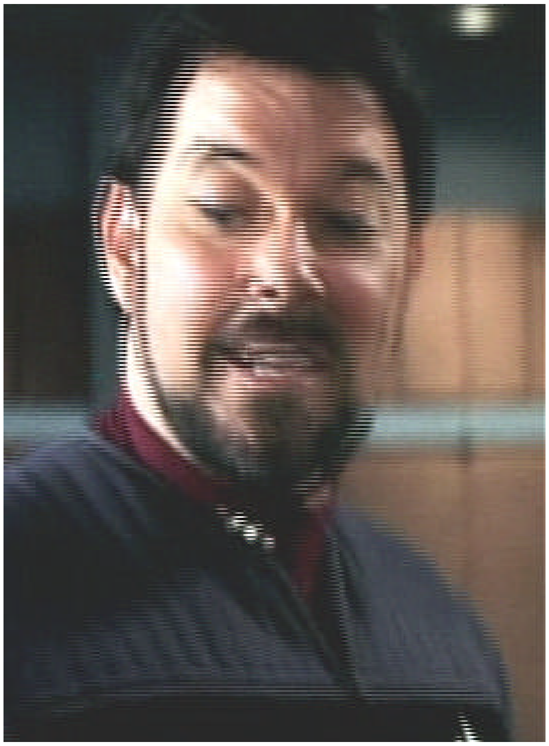
Custom color correction: This is harder to use but gives better control over image colour values. To start you have to press the 'Add' button to add an adjustment layer over the image or you cannot mess about with the settings.

Ghost reduction: This helps to prevent flicker and ghosting effects. You can either set it yourself or let it automatically guess the best one for your video clip.

Noise Reduction: Another excelent filter although hard to get good settings with so if unsure try the default settings. It has the option for temporal filtering or spatial filtering (if I'm using the term correctly). Just mess with the settings for a good result. Be very careful with this because it can actually cause ghosting effects. For details on how noise reduction works it may be useful to check out my guides on the VirtualDub filters Here.

Deinterlace: If you find that your video file looks like this picture below, especially in scenes with lots of action then you will probably need to use the deinterlace option.

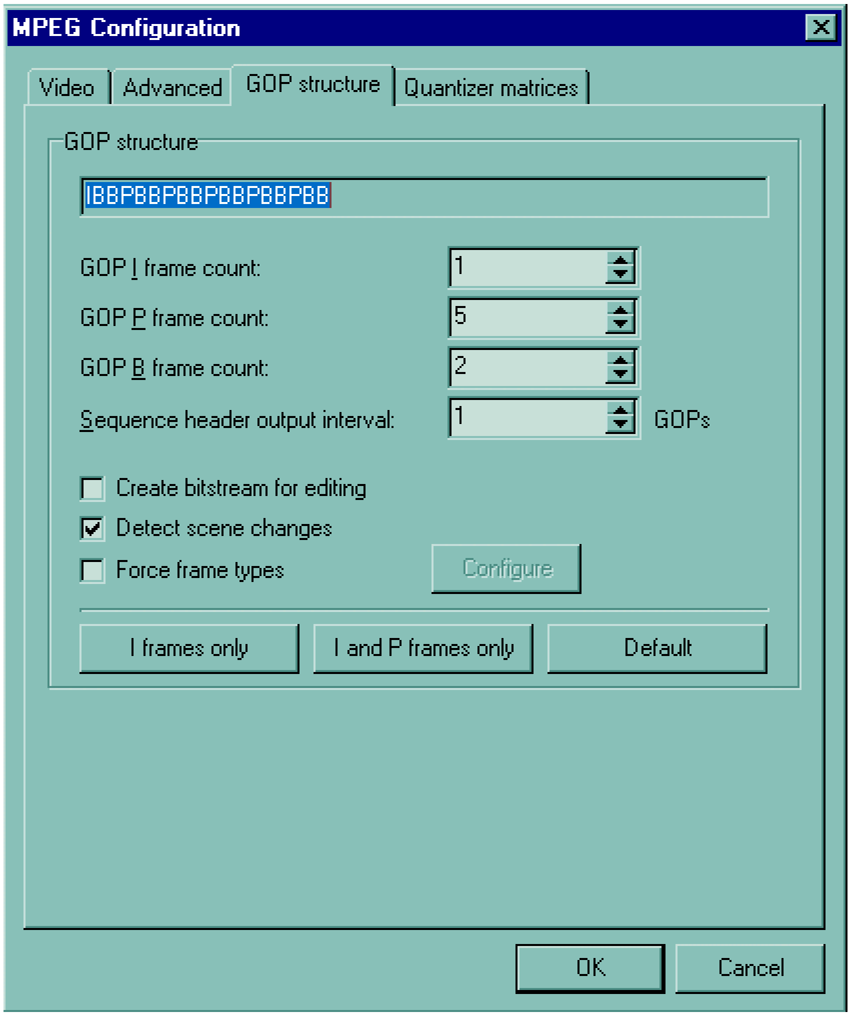
Usually Blend (adaptive) is the way to go. The theory is that you shouldn't get this problem with PAL video but you almost always do with video capture cards. With DVD's you will only sometimes get this problem with the extras and not the main movie.



Inverse Telecine: Inverse Telecine is a method that changes an interlaced 29.97 frames per second movie back into its original framerate of 23.976 frames per second. This is no use for VCDs or even SVCD's because the framerate must be either 29.97fps or 25fps. If you have problems with interlace artifacts you will be better off using the deinterlace filter. Its also the hardest possible thing to get right and a simple solution is beyond me!

GOP Structure

GOP is short for 'Group Of Pictures'. It sounds realy complicated but the idea is simple enough: I-frames are whole 'keyframes' in other words whole picture is like a photo. B-Frames represent just bits of the picture that are different from the previous frames (i.e. when there is a motion change). For an explanation of Keyframes and Delta check out my appendix on it Here. The I, P and B frames are arranged into GOPs that produce a good balance between good compression highest quality motion reproduction.



GOP structure

IBBPBBPBBPBBPBBPBB

GOP I frame count: 1

GOP P frame count: 5

GOP B frame count: 2

Sequence header output interval: 1 GOPs

□ Create bitstream for editing

☑ Detect scene changes

□ Force frame types

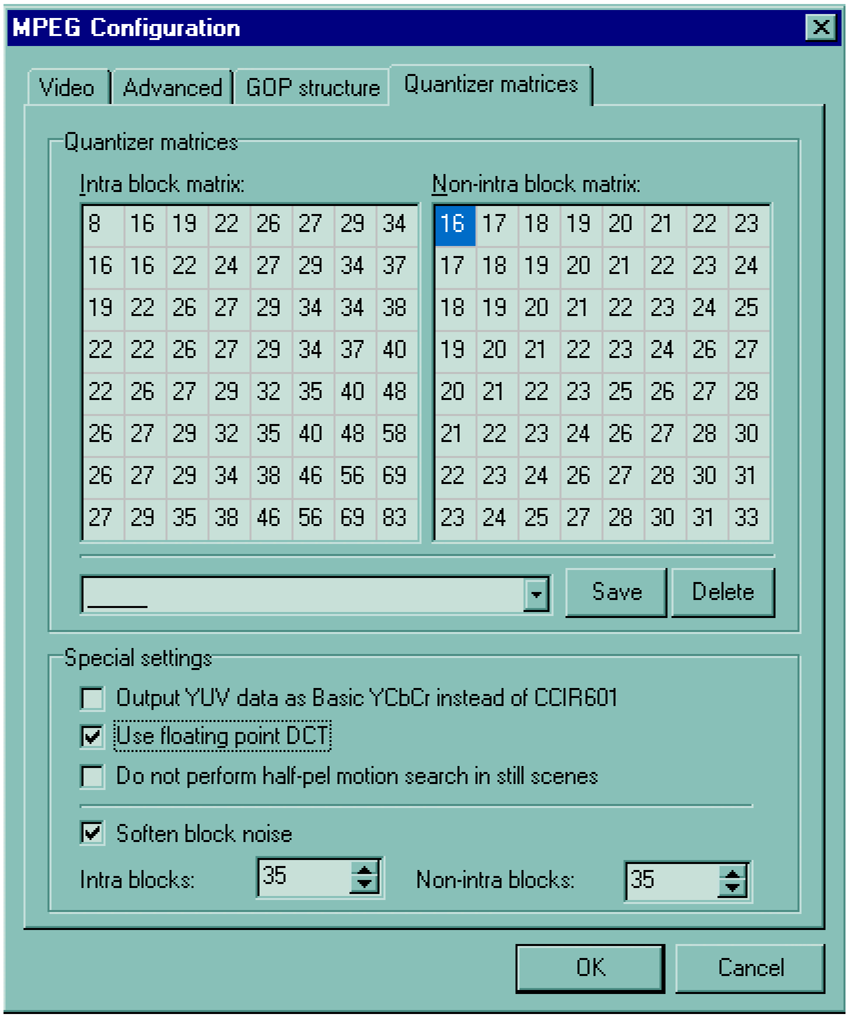
In short I think these settings should be left default for VCD's to avoid any problems. A lot of research has gone into the best settings by the Mpeg orgainazation and the default settings are great for just about anything.

I frames only: Using this option turns the MPEG into a sort of MJPEG movie without any half frames, only full frames. This eliminates motion macroblocks almost completely but will either increase the filesize or, in the case of VCDs because its a constant bitrate, may actually make much worse quality! The main reason people use I-Frame only is because they can cut and edit the video in VirtualDub or another video editing application frame by frame like they can with MJPEG.

I and P frames only: This is only really useful for custom very low birate movies. By skipping B frames you are able to produce video that is much smaller than normal MPEG files. The bad news is this will result in slighly jerky video that is not well suited for any kind of quality VCD and probably wouldn't be playable on a standalone VCD player.

Quantizer Matrixes

I'm not sure exactly what settings are best for Quantizer Matrices. They are designed as filters to prepare certain types of video for compressing. Generally I'd say stay with the defaults because they are tried and tested settings. Apart from being able to change the patten of each mpeg macroblock by setting different numbers in the table, there are a bunch of pre-defined settings that you can choose from the drop-down menu. You may, for example, find that the CG gives better results on animated images and MPEG on real life movies or vice versa.



Quantizer matrices

Intra block matrix:

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

Non-intra block matrix:

16	17	18	19	20	21	22	23
17	18	19	20	21	22	23	24
18	19	20	21	22	23	24	25
19	20	21	22	23	24	26	27
20	21	22	23	25	26	27	28
21	22	23	24	26	27	28	30
22	23	24	26	27	28	30	31
23	24	25	27	28	30	31	33

Special settings

☐ Output YUV data as Basic YCbCr instead of CCIR601
☑ Use floating point DCT
☐ Do not perform half-pel motion search in still scenes

☑ Soften block noise

Intra blocks: 35 Non-intra blocks: 35

Looking a little lower we have the 'Special settings' section. These again are for either special video types or for just increased or just different 'looks' for the final movie by encoding in a certain way.

Use Floating Point DCT: This option will spend more time making sure that the mpeg video remains true to the original image. You must remember that these settings will servely slow down encoding and you may not notice much difference in quality - if you are a quality freak just use it.

Soften Block Noise: Since the Mpeg format compresses blurry images better than it does sharp images, it has a prefilter designed to soften the image without loosing too much sharpness. Again this is a matter of taste; take a look and see the results. If you think the bluriness is not too much you will get less noticable macroblocks. If sharpness is more important then keep it as normal. To increase the blur just increase the block noise numbers, 35 is a good setting anyway though. Intra blocks and non-intra blocks are basically the same as saying 'keyframes' (i.e. intra) or 'non-keyframes'. You may find that more softening on non-intra blocks produces better results in action scenes or vice-versa - experiment!

Do not perform half-pel motion in still scenes: This option helps to get rid of slight macrolock-noise in non-action scenes where there is very little movement. This works quite well but can give the apperance of slightly unnaturel motion at certain points. Again I suggest you do some tests and see what you prefer, some notice it while others do not.

Subtitle Ripping Guide for DivX

Okay, I thought it would be nice to create a tutorial on how to rip subtitles from a DVD. I think people are shying away from doing this merely because it sounds difficult. So let me tell you now that it is laughingly easy to do! You can even make your own subtitles, but for now let's just stick to taking what already exists.

But why bother getting subtitles? Why not?! Subtitles take practically no memory on the DivX! In a two hour movie such as my following example 'The Mummy' DVD to DivX Rip, the whole of the subtitles will take a micro sized 50KB of extra space! I have never come across a DivX CD that didn't have many many times that space left on the CD after the film has finished. Also, its rather cool to be able to show your friends the DivX you made that has the option of subtitles.

There are only two DivX Player's that I use to play subtitles: MicroDVD and RedZ DivX Player. Therefore this guide only deals with the subtitle formats these use. However, on the digital digest download section you will fin utilities to convert to just about any subtitle format you need. You'll also find subtitles for most of the popular movies for download on DivX Digest too =).

I use Redz DivX player if I don't make MicroDVD menus and stuff for my DVD, because there is no thinking to it. You just open the movie in it, select the subtitles and it plays. Now, onto the conversion. I am going to make a DivX with subtitles of The Mummy as an example.

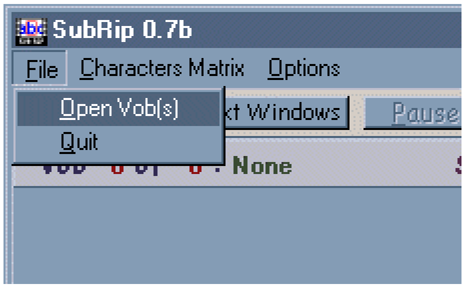
Before I start here are the things you will need:

SubRip (v0.8b or above)

SubConvert (0.9b1 or above)

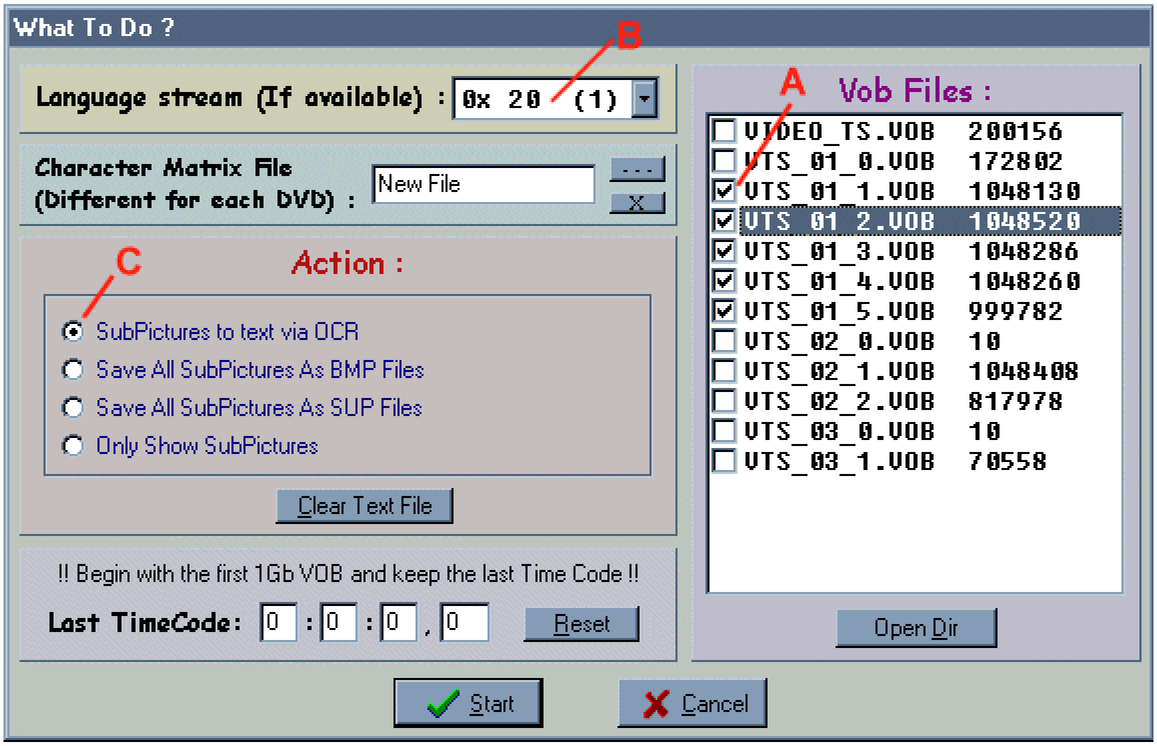
Rip the VOB files from the film to your Hard Disk as explained in my Flask Mpeg Guide. Open SubRip and select:

File > Open Vob



Up will pop the selection dialogue box seen below. Under the red Action heading choose the SubPictures to text via OCR (**C**). Click on the Open Dir. button and find the Vob's that you Ripped from the DVD. Put a tick alongside the VOB files you want to rip the text from (**A**). If you find that the default language is the one you do not want, for example, you may want German subtitles, but instead it gives you English. In that case choose another stream from the drop down stream list (**B**).

Once you have selected the Vobs and the correct stream (by trial and error) you are ready to go! Make sure the Vob files are the ones with the film on it, of course, and not something else on the DVD. Okay, click start!



What To Do ?

Language stream (If available) : 0x 20 (1)

Character Matrix File (Different for each DVD) : New File

Action :

● SubPictures to text via OCR
○ Save All SubPictures As BMP Files
○ Save All SubPictures As SUP Files
○ Only Show SubPictures

!! Begin with the first 1Gb VOB and keep the last Time Code !!

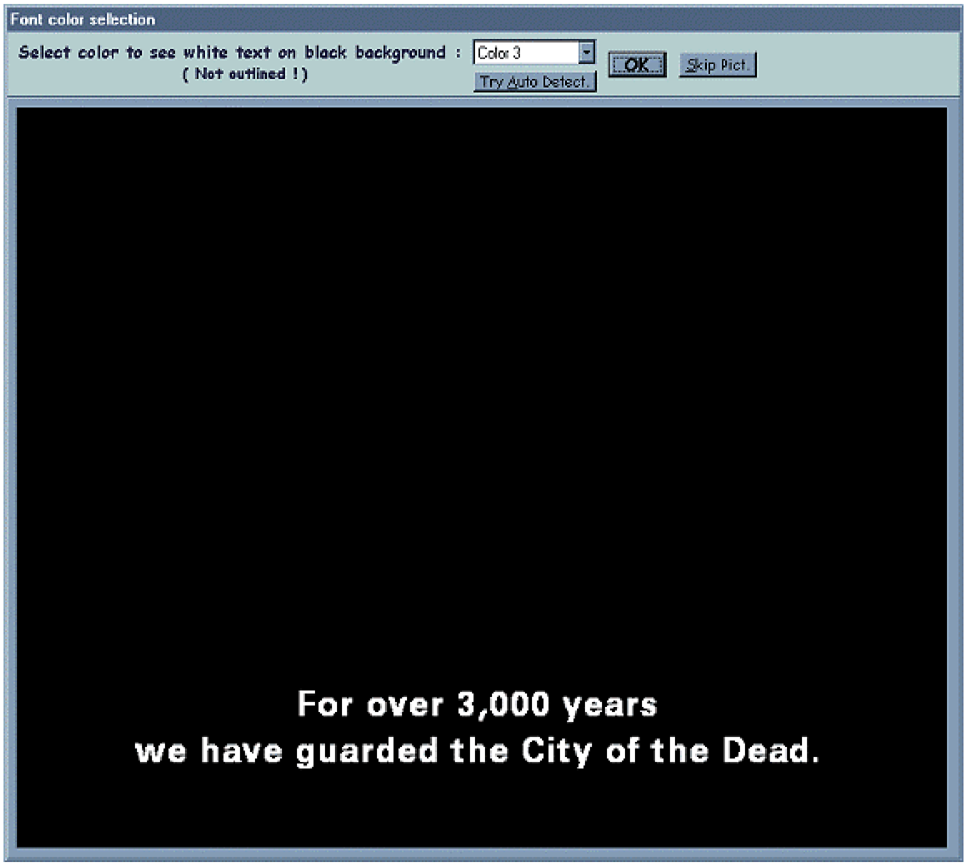
Last TimeCode: 0 : 0 : 0 , 0

Vob Files :

```
☐ VIDEO_TS.VOB   200156
☐ VTS_01_0.VOB   172802
☑ VTS_01_1.VOB   1048130
☑ VTS_01_2.VOB   1048520
☑ VTS_01_3.VOB   1048286
☑ VTS_01_4.VOB   1048260
☑ VTS_01_5.VOB   999782
☐ VTS_02_0.VOB   10
☐ VTS_02_1.VOB   1048408
☐ VTS_02_2.VOB   817978
☐ VTS_03_0.VOB   10
☐ VTS_03_1.VOB   70558
```

Note: *DVD's use actual picture files for their subtitles. This means that the computer has to use Optical Character Recognition (OCR) to convert the picture to a text file.*

The following window will appear next to make sure the picture is clear. If the text looks white on a black background click the OK button. If not, change the 'color 3' box (or whatever it is called) to another number; or you could try the auto detect button. When it looks correct click the OK button.



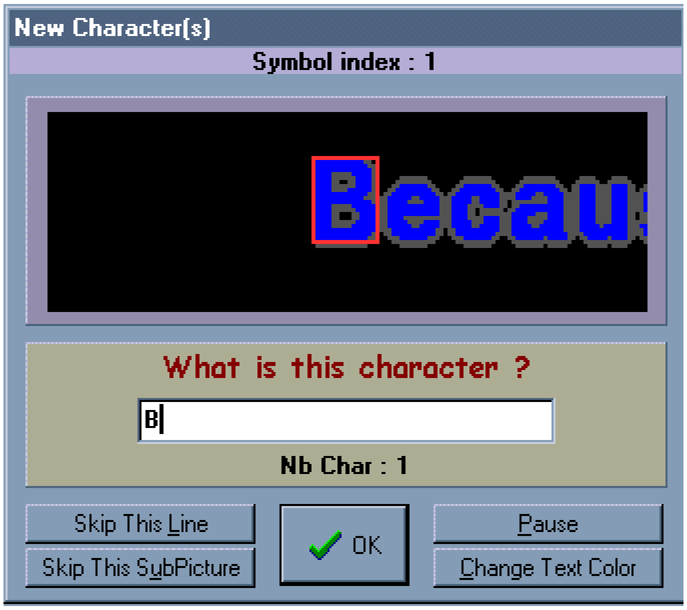
Font color selection

Select color to see white text on black background : (Not outlined !)

Color 3

For over 3,000 years
we have guarded the City of the Dead.

If you are running the program for the first time the computer must learn the letters. Be patient and answer its questions by typing in the correct letter as it asks. It is important not to put in the wrong letter and also to make sure that the capital and lower case letters are the correct. Don't worry, the reading process will get faster as you go along. After the first 30 or more letters the computer will almost completely take over.



New Character(s)

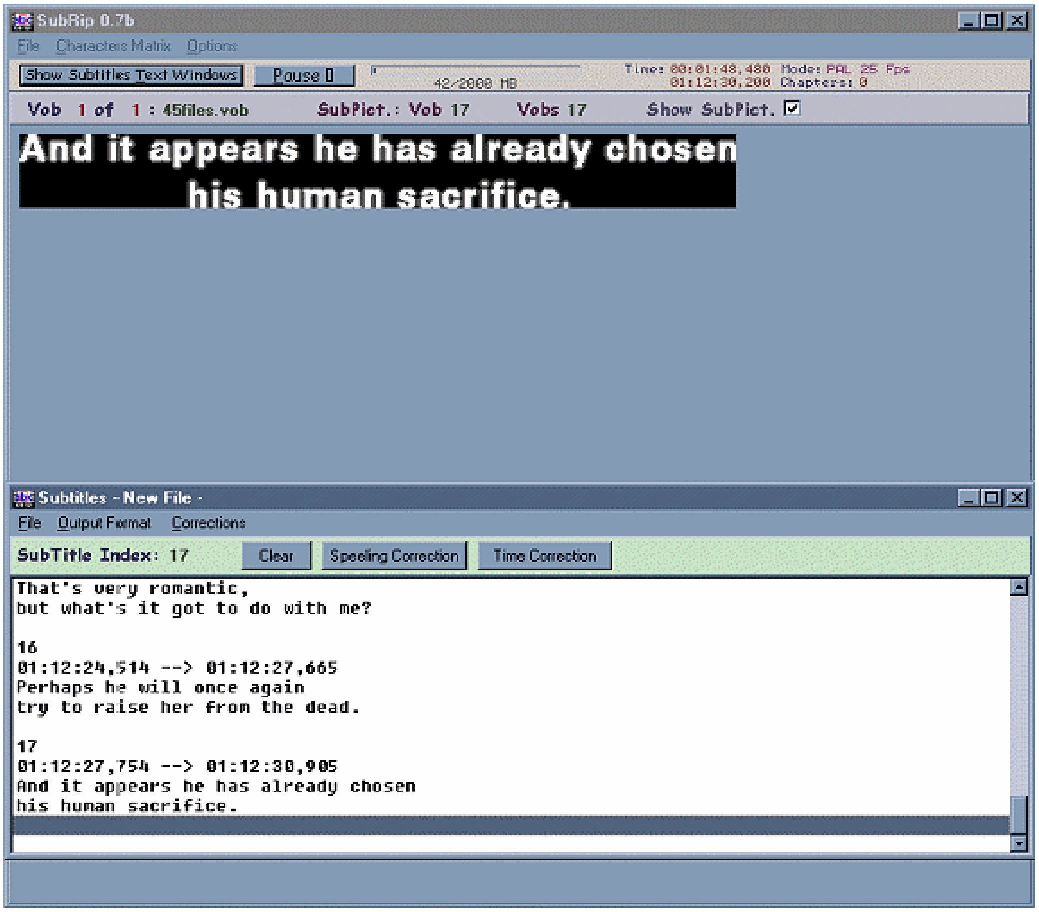
Symbol index : 1

What is this character ?

B

Nb Char : 1

The text that the computer has read is shown in a white box below. The picture that the DVD uses is shown above it. If you cannot see the white Subtitles box, click the 'show subtitles text windows' to see it. Now, sit back and wait until the whole file is ripped.



42/2000 MB

Time: 00:01:48,480 Mode: PAL 25 Fps
01:12:30,200 Chapters: 0

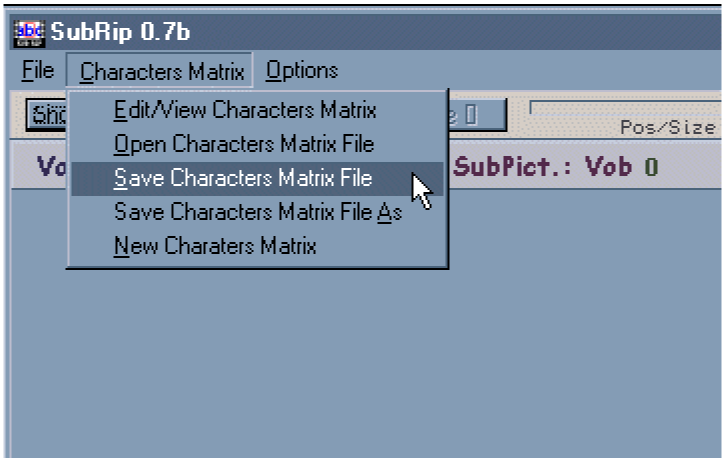
Vob 1 of 1 : 45files.vob SubPict.: Vob 17 Vobs 17 Show SubPict. ☑

And it appears he has already chosen
his human sacrifice.

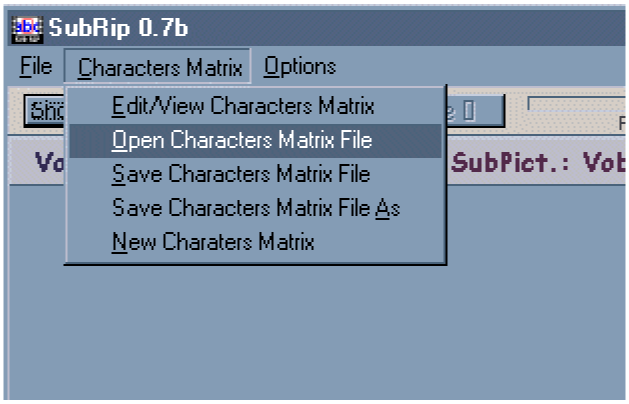
SubTitle Index: 17

```
That's very romantic,
but what's it got to do with me?

16
01:12:24,514 --> 01:12:27,665
Perhaps he will once again
try to raise her from the dead.

17
01:12:27,754 --> 01:12:30,905
And it appears he has already chosen
his human sacrifice.
```

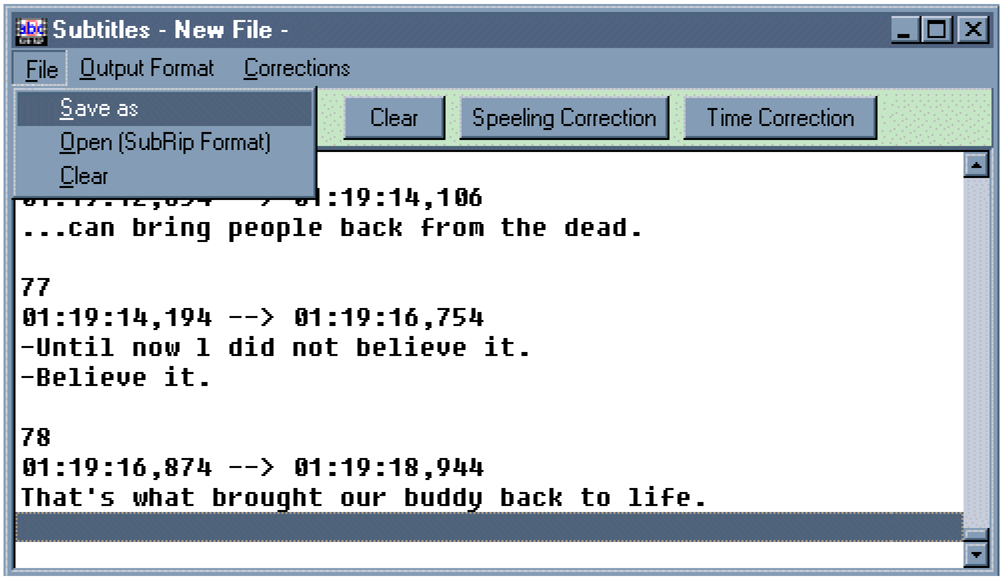
Now the computer has learned what the text looks like you can save this information to a file for future use. This is a good feature because you do not need to tell the computer what certain letters are every time you use it. To save this information, go to: Characters Matrix > Save Characters Matrix.


SubRip 0.7b
File
Characters Matrix
Options
Edit/View Characters Matrix
Open Characters Matrix File
Save Characters Matrix File
Save Characters Matrix File As
New Charaters Matrix
Pos/Size
SubPict.: Vob 0

When you choose your next Vob file to rip, you can go to Characters Matrix > Open Characters Matrix and load this information once again i.e.


SubRip 0.7b
File
Characters Matrix
Options
Edit/View Characters Matrix
Open Characters Matrix File
Save Characters Matrix File
Save Characters Matrix File As
New Charaters Matrix
SubPict.: Vob

Now the important thing to do is save the text you just ripped. Go to the white Subtitles window and select: File > Save as.

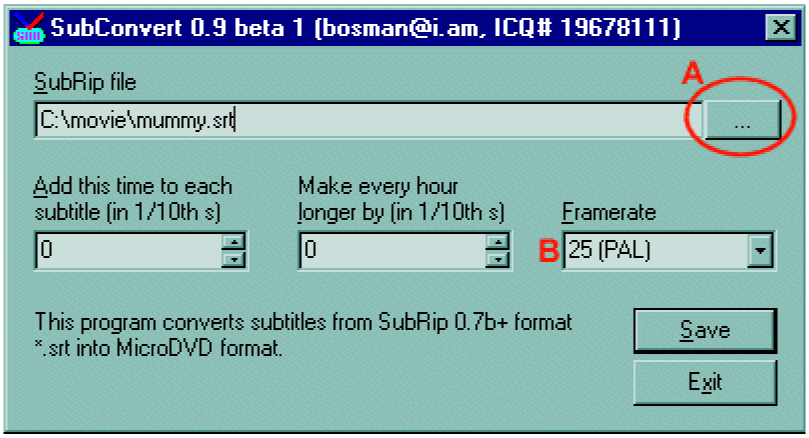


```
                            1:19:14,106
...can bring people back from the dead.

77
01:19:14,194 --> 01:19:16,754
-Until now l did not believe it.
-Believe it.

78
01:19:16,874 --> 01:19:18,944
That's what brought our buddy back to life.
```

The processed text is literally nothing more than a text file even though it may have the extension .srt or .sub or whatever. If you were to open your subtitles in wordpad they would look something like this:

```
1
00:01:04,274 --> 00:01:06,390
Thebes: City of the Living.

2
00:01:07,634 --> 00:01:10,068
Crown jewel of Pharaoh Seti the First.

3
00:01:13,194 --> 00:01:16,664
Home of lmhotep, Pharaoh's high priest.

4
00:01:17,474 --> 00:01:19,226
Keeper of the Dead.
```

MicroDVD can read the SubRip format, but you won't be able to use MicroDVD's more advanced features like setting font face, size, position, color for each subtitle line separately. So it is preferable to convert this to MicroDVD's native subtitle format first. To do this, open 'SubRip Convert'. In the frame rate box (**B**) select 25 if you use a PAL CD and 29.7 if you use NTSC or type in whatever you use. Find the input file (i.e. the one you just ripped) by using the browse button (**A**). Once done, click the 'Save' button and choose where you wish to save it.



SubConvert 0.9 beta 1 (bosman@i.am, ICQ# 19678111)

SubRip file: C:\movie\mummy.srt

Add this time to each subtitle (in 1/10th s): 0

Make every hour longer by (in 1/10th s): 0

Framerate: 25 (PAL)

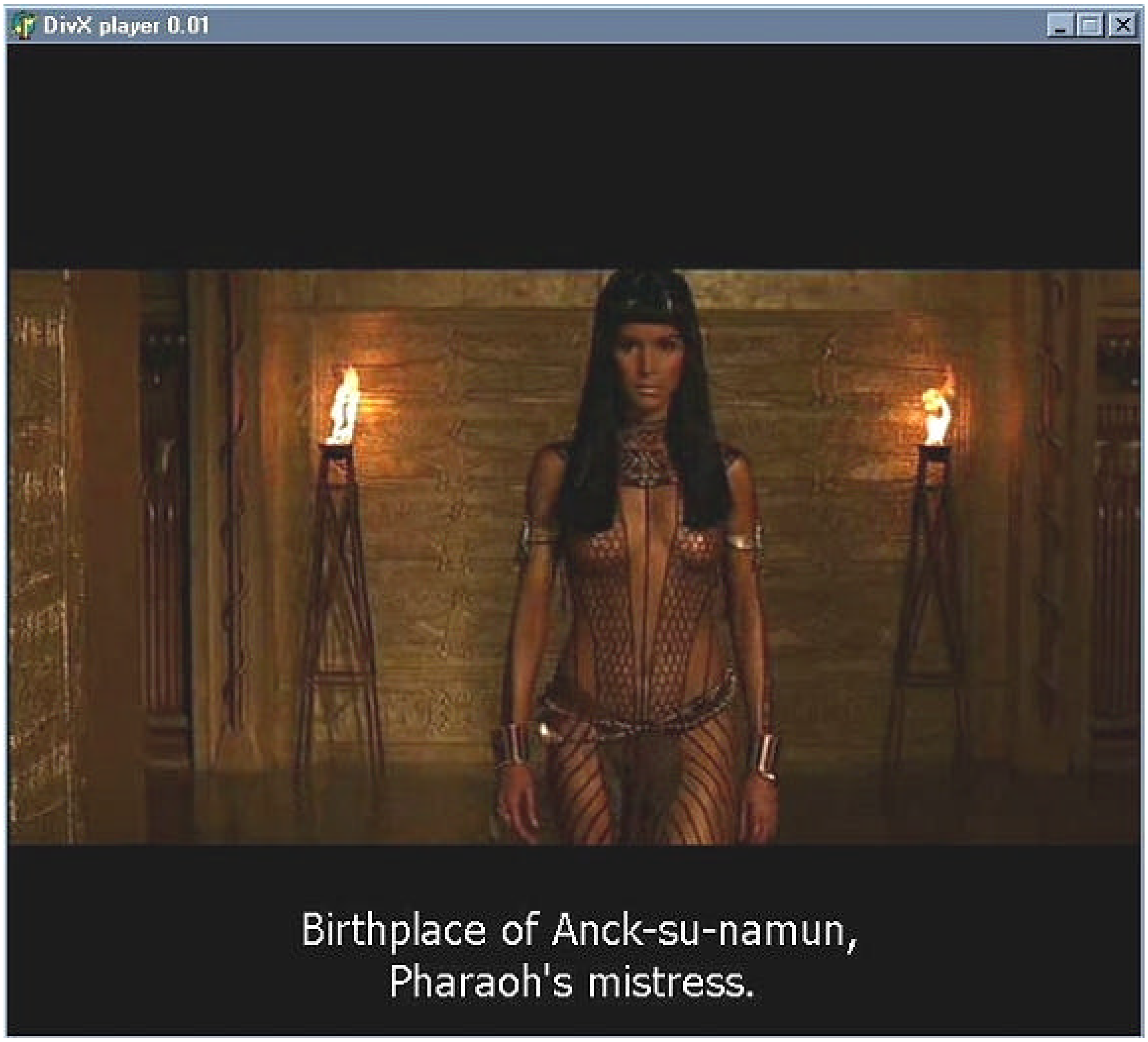
This program converts subtitles from SubRip 0.7b+ format *.srt into MicroDVD format.

The finished file should have a .txt extension unless you change it to something else. The converted file will look like this if you open it with a word processor:

```
{1606}{1659}Thebes: City of the Living.
{1690}{1751}Crown jewel of Pharaoh Seti the First.
{1829}{1916}Home of lmhotep, Pharaoh's high priest.
{1936}{1980}Keeper of the Dead.
{2029}{2108}Birthplace of Anck-su-namun,|Pharaoh's mistress.
{2126}{2190}No other man was allowed to touch her.
{2731}{2818}But for their love,|they were willing to risk life itself.
```

Ignore the bracketed numbers as they tell the DivX or MicroDVD player when to put the text on the screen. The bar lines (i.e. |) tells MicroDVD to put the text after the bar below the previous.

And here is how the finished DivX looks, cool huh!



Editing the Subtitles

If you get a really weird DVD such as a multi-angle DVD, you may have to select individual Vob files merge the subtitles together. This is also very easy to do since it is just text! Open the first of the text files you have converted in Wordpad (or another word processor). Then open Wordpad again and open the next converted file in that. Copy the text from the second text file and paste it into the first. Do the same again for each text file until you have just one single text file with the text of the movie in it.

You can open the text in any word processor such as Microsoft Word and spell check it too. Once you are happy with the finished file save it with the SAME name as the DivX movie and use the extension .sub. I called my DivX movie: 'mummy.avi' because it was a DVD rip of the film 'The Mummy'. So to save the subtitles text I will save the file as: "mummy.sub". Please include the quotation marks (" ") to force the word processor to save it with the .sub extension otherwise it will save it as something like mummy.sub.txt or mummy.sub.doc!

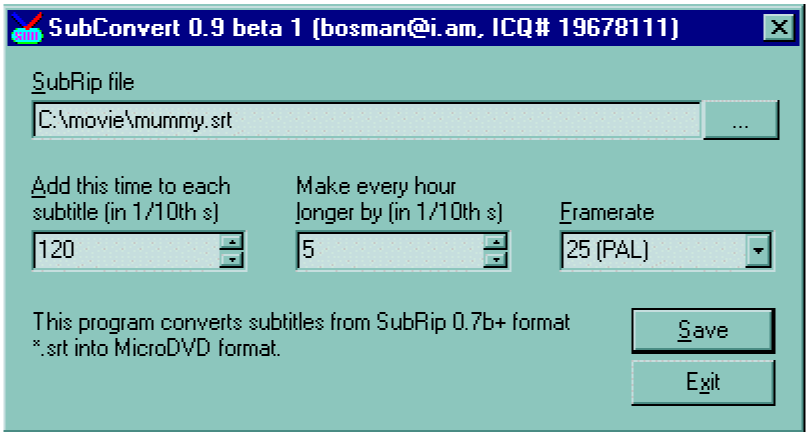
That's it! Save both the text file and the AVI file on the same CD and you are finished. Open the CD in Redz DivX Player and it will tell you that subtitles have been found. You can choose the font you want and the color too.

Synchronizng the Subtitles

Occasionally, the text and speech on the subtitles is not in tandem. This is usually a framerate problem because, for whatever reason, the frames do not match the text time codes. I suggest you don't get rid of your "original" subrip files in case this happens! The reason I say that is that the latest version of subrip converter has an option for automatically changing ALL the time codes on your subtitle text file before it converts it to the MicroDVD format that we all like ;^). This kills two birds with one stone so to speak.

Here is how it works. Open your newly converted subtitle file (something.srt) as though you were going to convert it to the MicroDVD format (i.e. something.sub). Now, everything works in tenths of a second, so 10 would represent 1 seconds delay. Usually the movie will be correct but need sliding either left or right to make it match the rest of the film. Where it says, "add this time to each subtitle", if you put 10 it would make the subtitles appear 1 second later than they do already. If you put -10 it would make the subtitles appear I second earlier than it used to appear, simple huh!

Where it says "Make every hour longer by.." it will stretch the subtitles out longer. So by putting 10 in it, it will gradually move the subtitle positions until they appear 1 second later by the time an hour of film is played. By choosing -10, you guessed it, it will gradually shorten the the time that each subtitle appears by 1 second.



SubConvert 0.9 beta 1 (bosman@i.am, ICQ# 19678111)

SubRip file: C:\movie\mummy.srt

Add this time to each subtitle (in 1/10th s): 120

Make every hour longer by (in 1/10th s): 5

Framerate: 25 (PAL)

This program converts subtitles from SubRip 0.7b+ format *.srt into MicroDVD format.

Weird Ass Foreign text such as Chinese subtitles!!

Sorry, subrip can only handle roman style text such as English, French etc. Your only option is to try Flask Mpeg's subtitle feature in DVD mode. It works most of the time so should for non-roman style text too.

MicroDVD INI Files

To make a CD that will tell MicroDVD player to offer the option of subtitles, you must make an .ini file. If you want to learn about writing these then go to the creating menu's part of my multimedia DivX section. For those of you who don't want the hassle I have designed the following ones for you to use. Just change the names and stuff to fit your own movie.

Just copy the text to notepad and rename the parts highlighted in red to your own movie files. For example, if my movie was titanic I'd put: Title=Titanic ; AVIName=titanic.avi. Lets say my subtitles were German, I'd put: 1=GER German ; File=titanic.sub. You get the idea!

CD

```
[Micro DVD Ini File]


[MAIN]
Title=The Mummy
ID=12345678
Delay=1

[MOVIE]
Directory=.
AVIName=mummy.avi

[SUBTITLES]
Directory=.
Format=0
Lines=2
EstimateDisplayDuration=0
1=ENG English
File=mummy.sub
```

Once changed, save it in notepad by calling it: "Mdvd.ini" (with quotation marks to force it to have the .ini filename). Just copy this file along with your movie and subtitles text to your CD-R (don't put them in folders) and everything should play fine!

A Double CD INI File

A two CD INI file is not difficult to do either. Just copy the FULL subtitle document (e.g. mummy.sub) onto both CD's. You do NOT need to split them when using MicroDVD! Then change the red highighted parts of the INI below. The only difference is you must tell MicroDVD how long each avi movie is in frames. So in my example the first part of the movie is CD1Frames=112000. Which means the AVI has 112000 frames in it. The amount of frames can easily be found by opening the file in VirtualDub and looking at File > File Information. It will then say how many frames where it says "# of frames (time)".

CD 1

```
[Micro DVD Ini File]


[MAIN]
Title=Movie Name
ID=1234678
CDNumber=1
Delay=1

[MOVIE]
Directory=.
AVIName=movie1.avi
AVI2Name=movie2.avi
CD1Frames=112000
CD2Frames=114000

[SUBTITLES]
Directory=.
Format=2
Lines=2
EstimateDisplayDuration=0
1=ENG English
File=movie.sub
```

CD 2

```
[Micro DVD Ini File]

[MAIN]
Title=Movie Name
ID=9101112
CDNumber=2
Delay=1

[MOVIE]
Directory=.
AVIName=movie1.avi
AVI2Name=movie2.avi
CD1Frames=112000
CD2Frames=114000

[SUBTITLES]
Directory=.
Format=2
Lines=2
EstimateDisplayDuration=0
1=ENG English
File=movie.sub
```

Multilanguage DivX

Yes, that's right! It is possible to have as many audio tracks as you like with your DivX or Video CD's. The only player that supports a second audio track is MicroDVD. BSPlayer is messing about with multi-tracks but we will stick with MicroDVD for the moment. With MicroDVD all you need to do is specify the secondard audio track in the ini file, and that easy enough to do.

Before I start here are the things you will need:

VirtualDub 1.4c

Getting the Second Audio

If you used FlasKMPEG to get your DivX movie you will be unfamiliar with how to grab just the audio from a DVD. This is important because it may only take half an hour to grab the secondary language audio track from a DVD with most programs, but the only way to do it with FlasKMPEG is to encode a whole Vob file into another DivX video! As you can imagine this will take ages! But if you have that kind of time to waste it is just as easy to use a Flask videos alternate soundtrack. All you'd need to do is extract the audio from your video with VirtualDub. Anyway, to learn how to extract any audio file from a Vob file *without* Flask you should read the audio extraction methods explained in my 'Ripping with Mpeg2avi' guide.

Compressing the Audio

By now you should have extracted an secondary language track from the DVD and converted it into PCM Wave. The trick now is getting it compressed and synched with the original movie. To do this follow exactly my multiplexing guide under the 'Ripping with Mpeg2avi' section. Don't worry if your original DivX movie has perfect audio already we are making a new DivX movie out of it. So at the end of all this you will have:

1. Your original DivX movie with English (or your main audio) soundtrack.
2. Our newly multiplexed DivX with a new soundtrack.

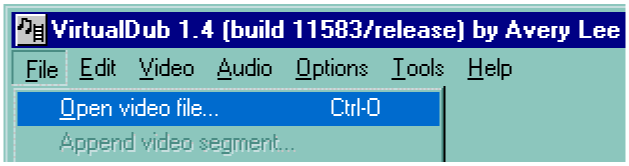
The reason for all this is only so we can know that the audio is perfectly synched and also it is the easiest way to compress it too. For those of you that want to use other programs such as Xing's Audio Catalist etc., that is fine too since we are not multiplexing the secondary audio back to the video. Why? Because there is no need and it plays better this way.

One or Two CD's?

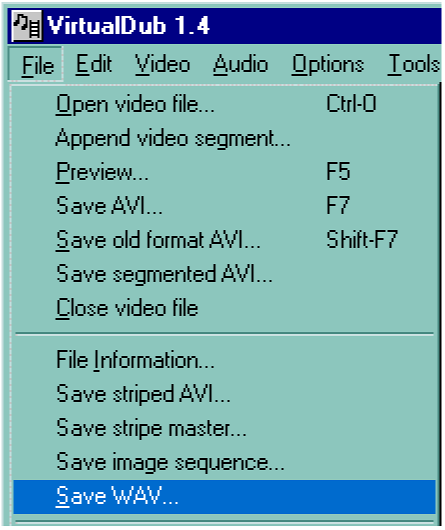
If you are making a 2 CD DivX then you must make it as a single whole movie first. Then add the secondary audio track, and then cut it in half with VirtualDub. Go to my cutting & joining AVI guides to see how to cut movies. The reason for this is that the secondary audio track must be split in the exact same place the original movie was.

Extracting our New audio Track

This is the easy bit. Open your second audio track movie/s in VirtualDub.


VirtualDub 1.4 (build 11583/release) by Avery Lee
File Edit Video Audio Options Tools Help
Open video file... Ctrl-O
Append video segment...

Select: File > Save WAV...



Don't worry, this will *not* extract the audio to PCM wave, it just calls it a wave. What you will get is your compressed audio such as the Mp3 file or the WMA audio but you can call it what you like, sound.wav, sound.mp3, sound.wma etc. You can extract just about any other audio formats from an AVI using this method too.

Okay, now you will have:

1. Your origainal DivX movie with your first audio track
2. Your new DivX with a secondary audio track
3. Your extracted secondary audio track

Now you can delete the DivX with the secondary audio track and end up with:

1. Your origainal DivX movie with your first audio track
2. Your new secondary audio track i.e. Something.wav

That's it! MicroDVD will allow you to select this secondary audio track at any point in your movie. But for information on creating DivX CD's for MicroDVD continue to read my 'Multimedia DivX' section.

How to Create a Multimedia DivX

Part 1: The Basics

So, you want to be really cool, huh! You want all your friends to see the amazing brand new DivX CD that you made yourself. Neither is this any ordinary DVD rip done by someone who has just learnt how to press that large GO button on FlasKMPEG =). No, this is something special! It looks just like a real DVD yet it fits two hours onto a single 650 MB CD! It lets you choose another language to listen to. It lets you choose as may subtitle languages as you like! It has a detailed chapter list, so you can jump to your favorite section just like in a commercial DVD! When you put the CD in it loads up an animated menu allowing you to select all these features. In fact, if someone didn't know any better they probably wouldn't even be able to tell that your film wasn't a real DVD! Well, if you do not already know how to make such a CD yet you're in luck, because I'm going to take you through the whole process step by step.

All of these things can be done with the most excellent MicroDVD Player. And this isn't just for DivX movies you can do it with just about any video format your computer can play, VCD, ASF, SVCD, MJPEG and so on. The only downside is you can only play them on your PC.

In actuality this is a simple "put it all together" kind of tutorial. I am assuming you already know how to rip a DVD to DivX and have one. I also assume you know how to get subtitles for your DVD and I already assume you can extract a secondary audio track for intergration into your multimedia CD. If not just follow the guides that explain how to do all this.

Before I start you will need:

MicroDVD & INI Editor

MicroDVD Player's INI Files

To make MicroDVD play your extra audio tracks, subtitles and everything else, you must make a text file with an .ini extension. MicroDVD will read this text file to know what to do with them. This .ini file is similar to the .ifo file of a normal DVD. Locutus has been kind enough to create a program called INI Editor which has some very nice features. But I find it less confusing to write it out as a text file first and then use the editors advanced features later. Either way, once I show you my method you can do it anyway you like.

Lets make an INI file! Much of the following is covered in greater detail in the instruction of MicroDVD Player so I'll just go over the main points briefly. A typical INI file looks like the following list. This is again the .ini from the film The Mummy and has only been set out to support the features I am using in this tutorial. If you don't use any feature just don't add it to the text file. All MicroDVD text files start with the title **[Micro**

DVD Ini File] in square brackets. Notice also that that there are other headings. **[MAIN]** deals with the CD opening info. **[MOVIE]** describes the movie location. **[LANGUAGES]** describes where each secondary sound file is kept. **[SUBTITLES]** describes where each subtitle file is kept. And **[CHAPTERS]** describes how the chapters are divided.

Just so you know, this is what a quite complex ini file looks like:

```
[Micro DVD Ini File]


[MAIN]
Title=The Mummy
ID=111
CD name Delay=1


[MOVIE]
Directory=.
AVIName=mummy.avi


[LANGUAGES]
Directory=.
MultipleAudioAVI=0
Primary=ENG English
1=COM Commentary
File=comment.wav


[SUBTITLES]
Directory=.
Format=0
Lines=1
EstimateDisplayDuration=0
1=ENG English
File=mummy.sub


[CHAPTERS]
1=0 Chapter 1
2=9251 Chapter 2
3=16228 Chapter 3
4=31294 Chapter 4
5=39061 Chapter 5
```

```
6=49578 Chapter 6
7=58944 Chapter 7
8=66349 Chapter 8
9=72366 Chapter 9
10=77658 Chapter 10
11=96475 Chapter 11
12=106561 Chapter 12
13=118049 Chapter 13
14=121468 Chapter 14
15=130295 Chapter 15
16=140406 Chapter 16
17=148010 Chapter 17
18=157690 Chapter 18
19=168499 Chapter 19
```

It looks scary, but its more or less just a list of stuff and where to find it. Once you know what to put you will be fine. For full details of all additional features you can have in an .ini, file please refer to the MicroDVD help pages provided with the program.

Writing Your Own INI File

Open notepad or any text editor and lets write your .ini file. Start with the first section:

```
[MAIN]
Title=The Mummy
ID=111
Delay=1
CDNumb er=1
```

Title= call this whatever you like, its not important. When you put your CD in this will be the title of your film.

ID= any number you like. Each CD must have an separate ID number so that MicroDVD can identify it.

Delay=1 Just use 1. This is how many seconds MicroDVD will wait to change chapters. A second is good to give it time to play correctly.

CDNumber=1 If it is the second CD of the film put =2 and so on. Each separate CD must have its own INI file to help stop confusion.

```
[MOVIE]
Directory=.
AVIName=mummy.avi
CD1Frames=67827
CD2Frames=80808
```

Directory= this is the location of the movie .avi file on your CD. If you just bundle everything on the CD put a full stop mark (.) as shown.

AVIName= this is the location of the movie .avi file on your CD. If you just bundle everything on the CD put a full stop mark (.) as shown.

AVIName= tell it the location of the main film ie. something.avi. If it is inside a folder on the CD call it AVIName=\foldername\something.avi.

CD1Frames= This is only needed if your movie is put on two or more CD's. You must put the number of frames from each movie after this option. It is needed to keep audio and subtitles in synch. Finding the frame number to enter is easy: just open VirtualDub, move to the last frame of each CD movie and write down the frame number!

```
[LANGUAGES]
Directory=.
MultipleAudioAVI=0
Primary=ENG English
1=COM Commentary
File=comment.wav
```

Directory= this is the location of the secondary audio track on your CD. If it is inside a folder on the CD call it Directory=\foldername\something.wav

MultipleAudioAVI=0 advanced function just leave it 0.

Primary= ENG English just keep this as English if your main .avi movie is in English, its the default language. If it was German, for example you'd put GER German. It actually doesn't matter what you put its just a name ;). The first three letters are the abbreviation for the whole word.

1=COM Commentary Each extra audio track is given a number and a name so '1=COM Commentary' is the audio track with The Mummy Director's commentary. It doesn't matter what you call it though. **File=** is the location of the file which I called comment.wav. If it was inside a folder on the CD call it File=\foldername\comment.wav. Each language should be numbered with the file location just under each number thusly:

```
1=COM Commentary
File=comment.wav
```

```
2=FRE French
File=\language2\french.wav

3=GER German
File=\language3\german.wav
```

```
[SUBTITLES]
Directory=.
Format=0
Lines=1
EstimateDisplayDuration=0
1=ENG English
File=mummy.sub
```

Directory= Same as with other examples. It is the location of the folder that the subtitles are kept in. If there is no folder simply put a full stop (.) as shown.

Format= Just keep it 0. MicroDVD supports three other kinds of subtitles SubMagic, SubRip and SubViewer by putting in 1, 2, or 3 you can select the format.

Lines= How many lines will the subtitles take up? One line at the bottom, two or three? Just put the amount here.

EstimateDisplayDuration=0 Keep this as 0 unless you have made your own subtitles which require a set time (see MicroDVD instructions for details of this).

1= and **File=** is exactly like the Languages section the number is the language name and the file directly below it is the location ie:

```
1=ENG English
File=mummy.sub

2=FRE French
File=french.sub

3=GER German
File=german.sub
```

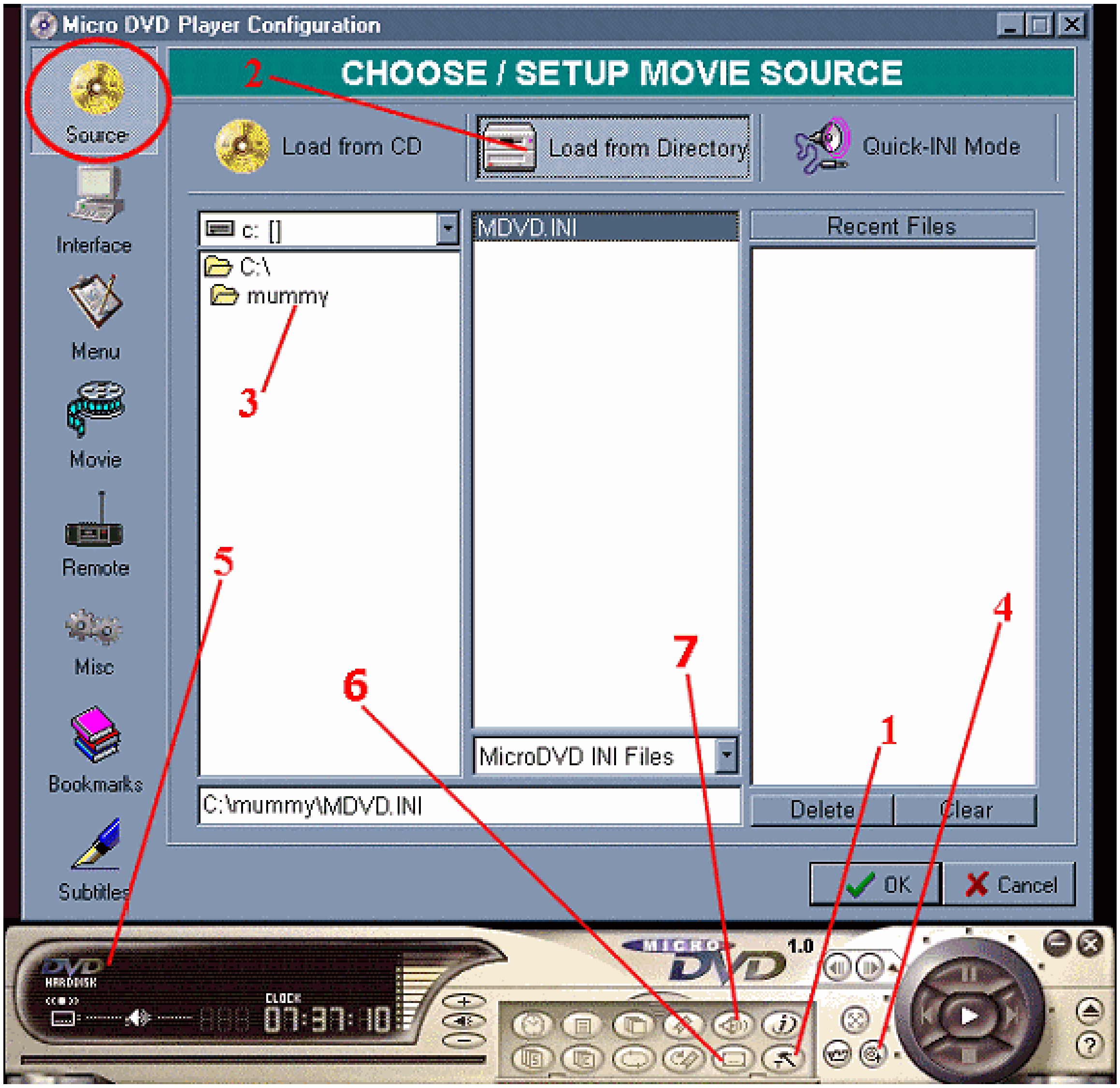
If you are testing your subtitles they will only appear in full screen mode in MicroDVD. Also, you cannot put the subtitle text files inside another folder if you want it to be compatible with Redz DivX Player.

2 CD Subtitles

Because there are tools designed to chop up subtitles and renumber the time codes people think this is nessasery for MicroDVD - it is not! Just copy the *full* subtitles text file onto both CD's. MicroDVD knows its the second CD and so will go to where it left off.

Testing the INI File

Okay, we are nearly finished ;) save your text file as "MDVD.INI" including quotation marks to force the computer to save it as .ini rather than .ini.txt! This name is important as it is the one MicroDVD looks for on the CD. If all has gone well this .ini will work on your new DivX or whatever CD. Lets test it now! Make a new folder on your hard drive for all the files in your movie, I called mine 'mummy' because it is a DVD rip of the movie The Mummy. Open MicroDVD player and press the configuration button (**1**). Up pops the following box. Select the 'Source' button at the top to bring up the directory selection. Select the 'Load from Directory' option (**2**) and select the file MDVD.INI. Close the configuration box.



c: []

C:\

mummy

MDVD.INI

MicroDVD INI Files

C:\mummy\MDVD.INI

Now click the spiral button (4) until the DVD HardDisk icon appears in the far left display (5). Now press the play button to preview your movie. Left click the mouse once so the MicroDVD player control appears but the film remains in full screen mode. Then press the subtitles button (6) and select your subtitles. Then press the alternate language to test that (7). If all goes well you should be smiling =), if not look over your notes again and check the MicroDVD instructions for a more detailed description on the settings of your .ini file.

Getting the Chapters from a DVD Vob File

The chapters for a DVD film are held in the .ifo file on the CD. The main film of a DVD looks something like this:

Vts_02_0.ifo

Vts_02_1.vob

Vts_02_2.vob

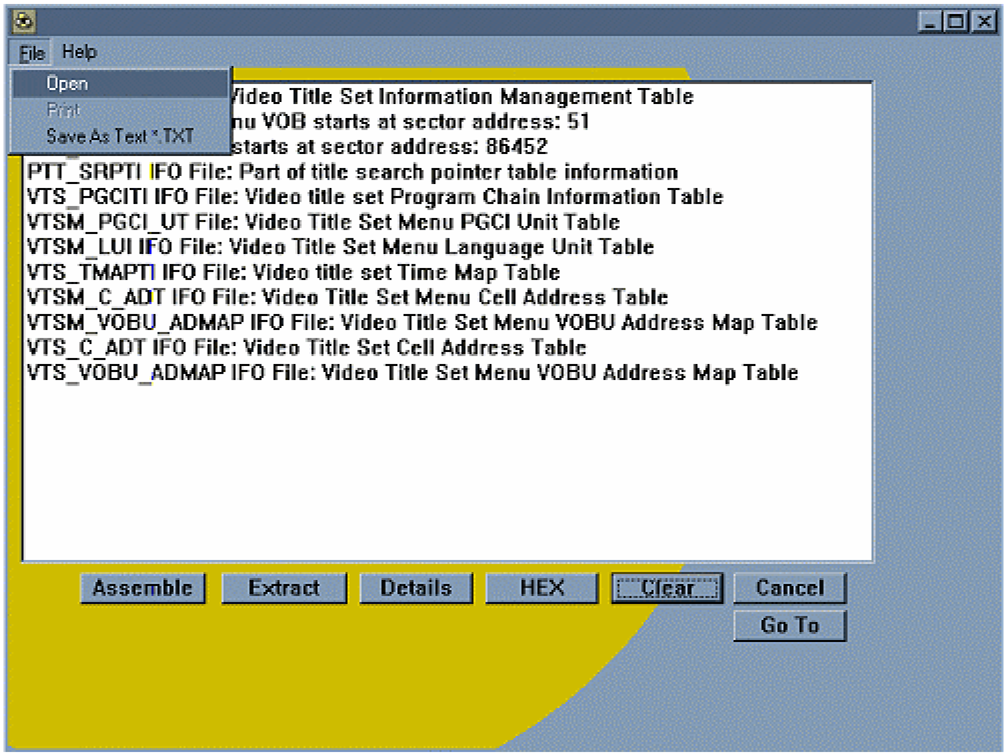
Vts_02_3.vob

Vts_02_4.vob

For details on DVD structure take a look at my basic DVD structure article in the appendix section.

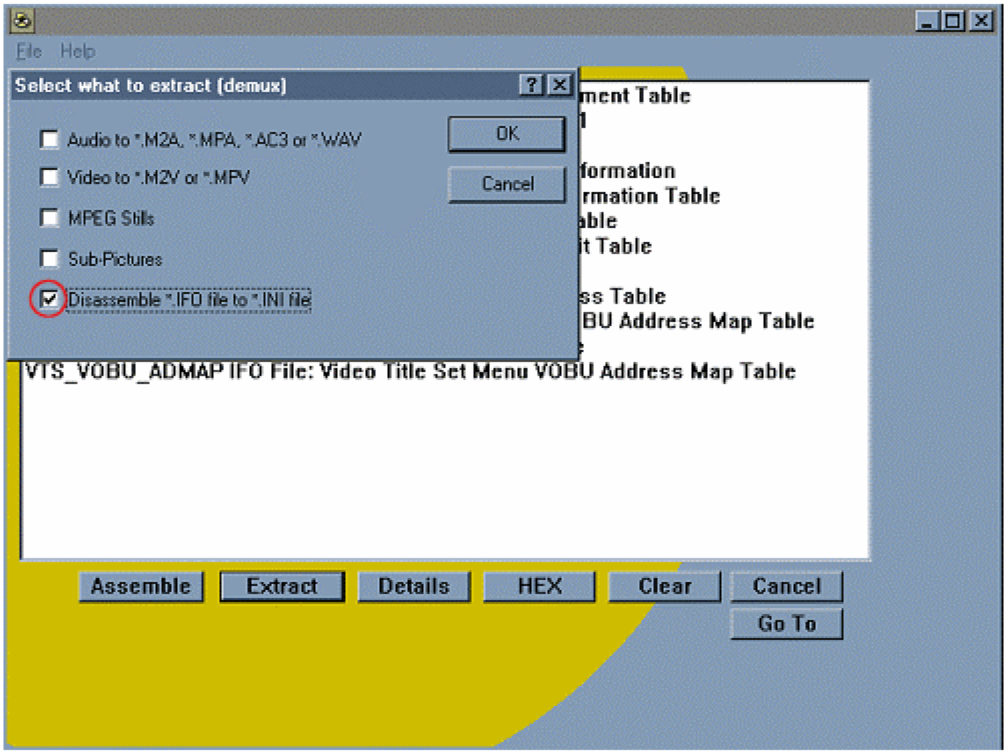
These have to be extracted and converted first by Vobsnoopy and then by MicroDVD's INI editor. Run Vobsnoopy and Open the first .ifo file of the movie list. There will be other .ifo files in the DVD but choose the one that matches the .Vob files of the movie; it will look something like Vts_0x_0.ifo.

Anyway, choose: File > Open.



```
/ideo Title Set Information Management Table
nu VOB starts at sector address: 51
starts at sector address: 86452
PTT_SRPTI IFO File: Part of title search pointer table information
VTS_PGCITI IFO File: Video title set Program Chain Information Table
VTSM_PGCI_UT File: Video Title Set Menu PGCI Unit Table
VTSM_LUI IFO File: Video Title Set Menu Language Unit Table
VTS_TMAPTI IFO File: Video title set Time Map Table
VTSM_C_ADT IFO File: Video Title Set Menu Cell Address Table
VTSM_VOBU_ADMAP IFO File: Video Title Set Menu VOBU Address Map Table
VTS_C_ADT IFO File: Video Title Set Cell Address Table
VTS_VOBU_ADMAP IFO File: Video Title Set Menu VOBU Address Map Table
```

Hit the extract button and only check the box that says 'Disassemble *.IFO file to *.INI File' and save it somewhere.

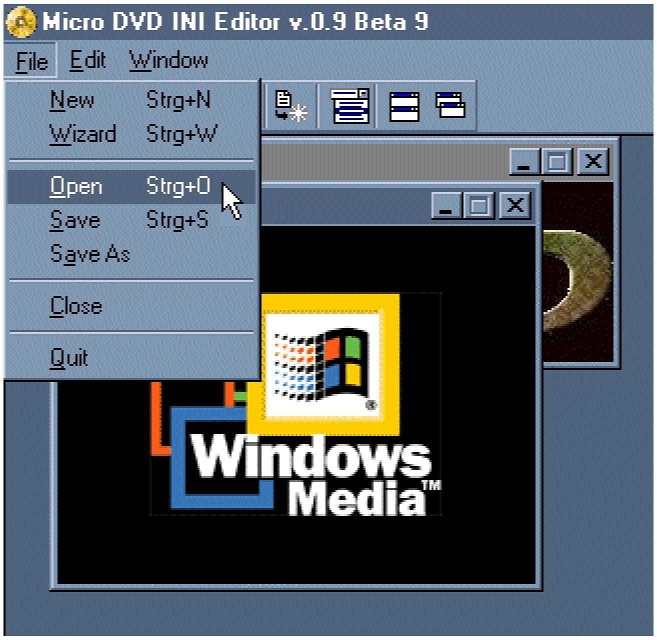


Select what to extract (demux)

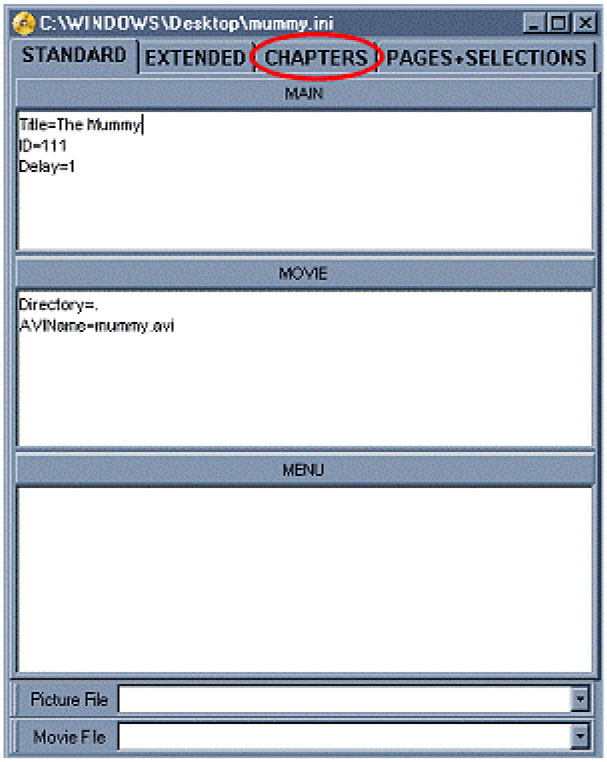
- ☐ Audio to *.M2A, *.MPA, *.AC3 or *.WAV
- ☐ Video to *.M2V or *.MPV
- ☐ MPEG Stills
- ☐ Sub-Pictures
- ☑ Disassemble *.IFO file to *.INI file

Now is the time to run the MicroDVD INI Editor. Just in case you missed it, its located in the MicroDVD folder when you downloaded it, its called: INIEditor.exe. If you haven't got it go to the MicroDVD website and get it.

Open the new .ini file that we have already previously made in notepad.



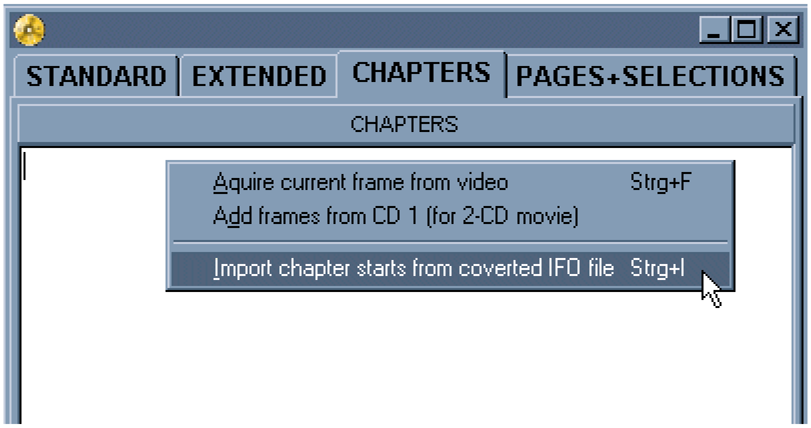
Up pops this box. Don't worry, it is little more than a normal text editor with helpful options. Everything you typed in notepad could easily have been typed in this editor. Try clicking on the various tabs to see how your file is broken up. Then click on the Chapters tab.



```
Title=The Mummy
ID=111
Delay=1
```

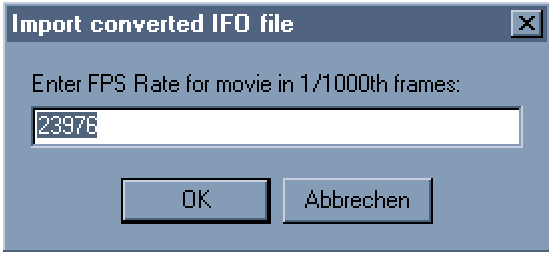
```
Directory=.
AVIName=mummy.avi
```

This section of the editor allows you to either create your own chapter positions or import them. We are going to import the file we have just extracted with Vobsnoopy. Right-click the mouse in the chapters window and select the option from 'Import chapter starts from IFO file'.


STANDARD
EXTENDED
CHAPTERS
PAGES+SELECTIONS
CHAPTERS
Aquire current frame from video Strg+F
Add frames from CD 1 (for 2-CD movie)
Import chapter starts from coverted IFO file Strg+I

Choose the frames per second of your DivX, VCD or MiniDVD in the import box. This is usually 23.976 for a NTSC DivX .avi file and 25 for PAL. You may also try 29.970.

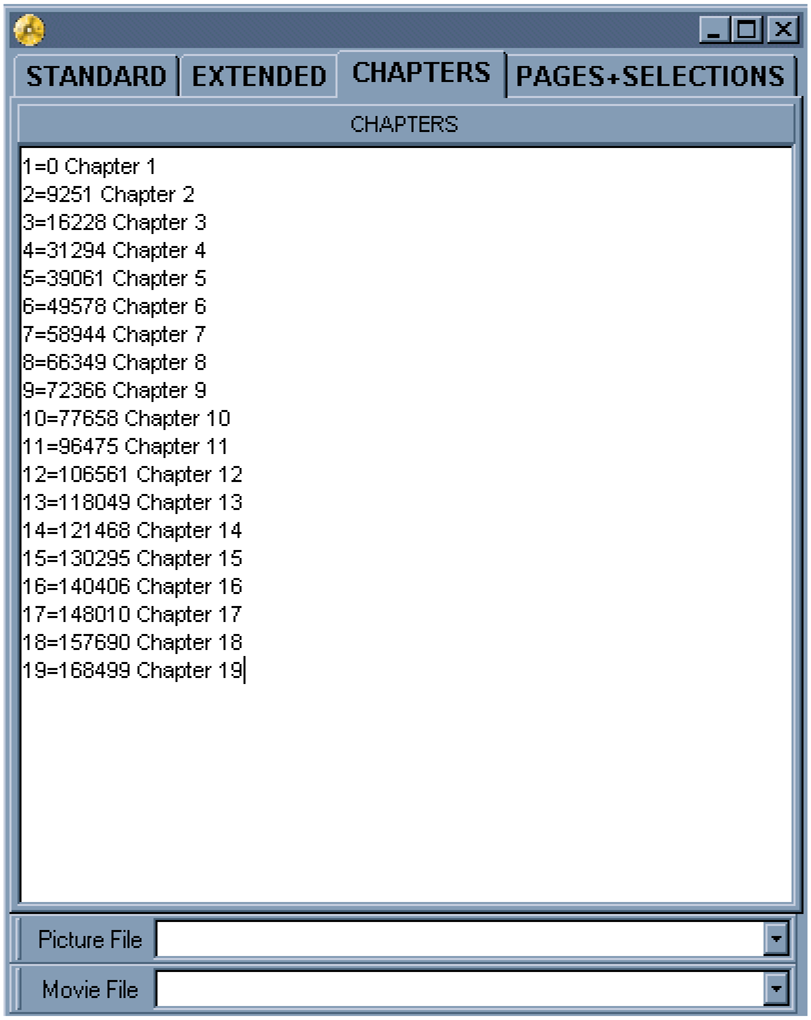
Basically, if the chapters don't quite start in the right place try grabbing them again with a new framerate. Once done press OK.


Import converted IFO file

Enter FPS Rate for movie in 1/1000th frames:

23976

Up pops your chapter time codes =). You can rename them here if you wish so that it describes the section of the film the chapter points.



```
1=0 Chapter 1
2=9251 Chapter 2
3=16228 Chapter 3
4=31294 Chapter 4
5=39061 Chapter 5
6=49578 Chapter 6
7=58944 Chapter 7
8=66349 Chapter 8
9=72366 Chapter 9
10=77658 Chapter 10
11=96475 Chapter 11
12=106561 Chapter 12
13=118049 Chapter 13
14=121468 Chapter 14
15=130295 Chapter 15
16=140406 Chapter 16
17=148010 Chapter 17
18=157690 Chapter 18
19=168499 Chapter 19
```

When you are finished, go to File > Save. That's it! Open MicroDVD again and check that it jumps to the chapters you have made and also so it lists them as designed.

That's all for this tutorial. If you are feeling brave you can go to the next one and learn to create menus and the such like. As I said before, the MicroDVD's help file is very well written so I do not need to spend too much time explaining each function in detail. But if you followed everything I said step by step you will now understand how it works.

How to Create a Multimedia DivX

Part 2: Creating Menu's

> *"Some believed we lacked the programming language to describe your perfect world. But I believe that, as a species, human beings define their reality through suffering and misery. The perfect world was a dream that your primitive cerebrum kept trying to wake up from. Which is why the Matrix was re-designed to this: the peak of your civilization".*
> *- Agent Smith (From the Matrix)*

If you want to download the Matrix menu example I have made, I have uploaded it to Myspace at:

http://www.myspace.com/Folders/12256546/

Password: menu

WARNING: If you can't download it its tough! Myspace is a Bunch of Crap! I'm looking for a better place now.

Unfortunately, you have to join myspace.

The zip file is 13MB and contains all the pictures and video animation, subtitles and INI files. It will play in MicroDVD player but when you click on the play movie or specials they wont play because I'm not uploading 650MB's of the Matrix just to show the menu....well, I could, no...I'd better not..he, he! Anyway, its chopped into two parts and needs to be rejoined with HJSplit which is also found in the myspace folder. I hope you like it.

Introduction

In my last tutorial I explained the basics of creating a DivX that had chapter listings, subtitles and alternate language options. All these extra features rely on the MicroDVD Player. Now I have decided to finish the job and explain how to make those cool menus that every DVD seem to have. These menu rips will almost completely match those of the original DVD's. In fact, if someone didn't know any better they probably wouldn't even be able to tell that your film wasn't a real DVD! MicroDVD Player is even able to present menu links over the top of moving film sequences just like the kind you would get in such films as The Matrix - truly amazing stuff!

I'm sure after this goes on line a whole host of tutorials will appear teaching the would be DVD ripper all they need to know about menus 'n' stuff, just like they have with many of my guides. It doesn't bother me so much because it still helps to teach others and will bring a higher percentage of good quality rips for everyone ;^)....But just remember one thing, you learnt it first on the Digital Digest website! Don't let anyone tell you different!

Obviously Locutus is a man who doesn't like to settle for second best =^). The MicroDVD Player has functions that are able to mimic a real DVD almost exactly in terms of features. But these features came at a cost, namely, that that we learn how to create a correct .ini file to use them :^(.

Actually, these menus 'are' all pretty easy to do. The only real problems are keeping track of what you are doing. You will be testing your newly created .ini file and the MicroDVD Player will come up with a vague error message in German! This is where it can get very frustrating! Usually it is just a minor mistake that was made, but if you are doing a very complex menu system with actor profiles and other special features etc., then it can take a long time to track down the problems! Sometimes a helpful message will pop up telling you which line in the .ini file has the error. In this case just count down the lines and check the settings in this area of the text file.

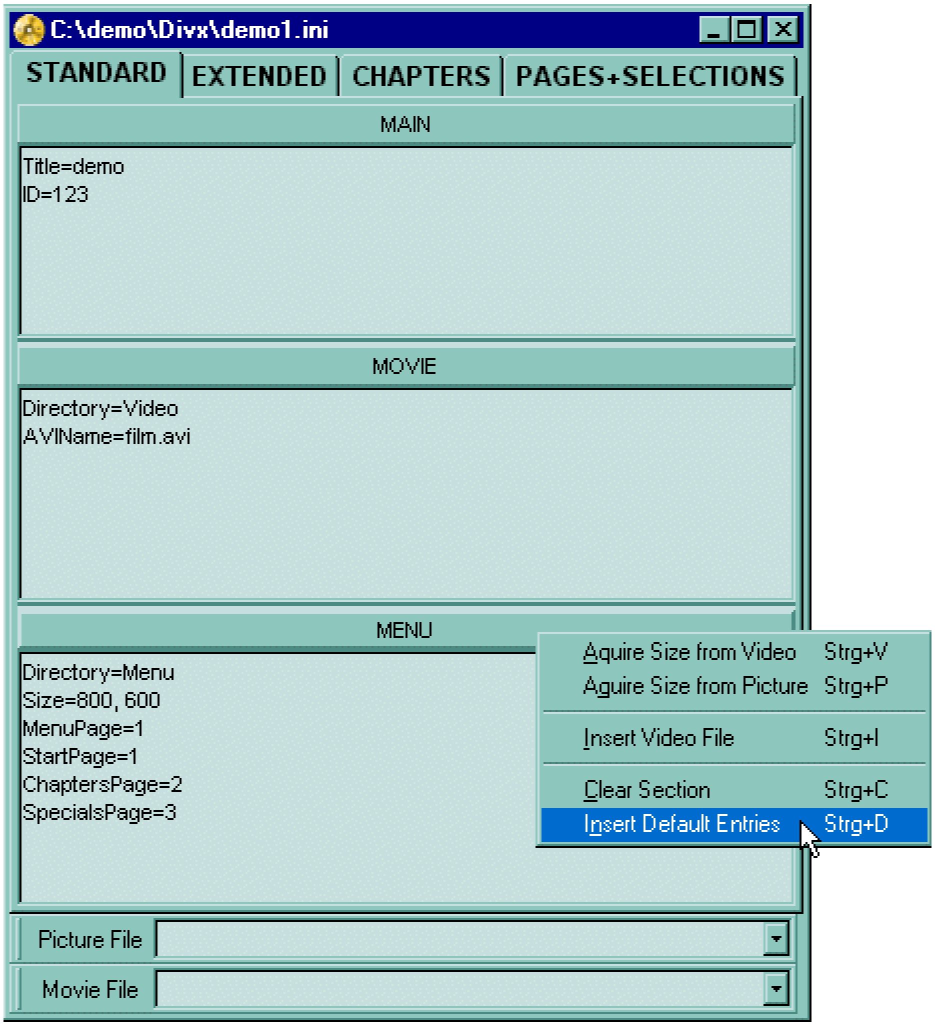
Note: I would recommend that you have the original MicroDVD documentation handy for quick reference. If you haven't already read through it, it may be an idea to have a quick browse for a while first to get the general idea of what it is capable of. That documentation explains in full detail ALL the features.

Before I start you will need:

MicroDVD & INI Editor

Basic Concepts

First we will restrict ourselves to a boring set of pages with no animated sequences. Open up the MicroDVD Ini Editor, like we did before, and lets get better acquainted with the controls. Press the new page button and up pops this windows again. Right-Clicking on any Menu area (ie. Main, Movie or Menu in this case) you will bring up a list of useful options. These are not always vital but are there to save you having to type out all the commands. As you can see here I have set up a basic demo .ini file with the Title 'demo'. The CD ID is just 123, use any numbers you like. I have put the Movie inside a folder on my CD and called it Video. So under the Movie section I have to tell it Directory=Video, duh! The actual file inside the video folder I called 'film.avi' so I had to tell it where that was too. All this you should already know by now though.



MAIN

```
Title=demo
ID=123
```

MOVIE

```
Directory=Video
AVIName=film.avi
```

MENU

```
Directory=Menu
Size=800, 600
MenuPage=1
StartPage=1
ChaptersPage=2
SpecialsPage=3
```

Menus are created just like web pages, so if you understand how to make a web page then you have a head start. The 'MENU' box at the bottom of this ini editor is only there to explain where the "web pages" are kept and a few other minor details. It has nothing to do with the actual set-up of the pages. You have to imagine this menu box as just a bunch on links to your 'real' pages. If you Right-Click and select 'Insert Default Entries' a whole list of things will be pasted into the box. This is like a questionnaire that needs to be filled out. Any settings that you do not want or need can be deleted in the same way you would in any word processor. Lets go through the options I have:

Directory=Menu. This is the Folder where I am going to keep all the Menu "web pages". You can, of course, just whack everything on the CD without any folders but I think keeping them in folders makes it easier to see what you are doing. Everything related to menus, all your film sequences, all your pictures and so on, must be kept in this single folder. If you have no folders it will assume they are just on the CD and not in a folder.

Size=800, 600. This is the size of the menu pages that will be displayed. I have selected 800 pixels across by 600 pixels down which is one of the PC standard resolutions. You can also try 640 x 480, 1024 x 768, 1152 x 864 or whatever you like. If you are using a movie sequence for your Menu you do not need to set the size because it defaults to the movie size.

StartPage=1. When you start your DivX playing, MicroDVD needs to know which menu page to show first of all. From this page the rest of the menu pages can be accessed. All pages you create are given a number so I have chosen page 1 as the start page. This can be likened to a websites Home Page.

MenuPage=1. The Menu page is the page that contains all the menu options. This is different from the StartPage only in that MicroDVD will go automatically to this page when you stop the movie or when it ends, whereas the StartPage will only appear when the CD is first inserted. I have chosen page 1 again; this means that both the MenuPage and StartPage are the same page, which will contain all the selections that both need.

AudioName=. For those boring DVD nav menus that have no animated pages and no sound, you can specify the name of an audio file (WAV, MP3 etc.) here that is played as background audio whenever the menu pages are running.

Note: The AudioFile is played only when the page being displayed is a still image. When a video page is active, the Audio file is paused and restarted when a still image page is active again.

ChaptersPage=2. Since MicroDVD has special chapter selection buttons it likes to know where these Menu pages are too ;^). No big deal, if you have a chapters page say which page number this is. If you don't have chapters just don't put this line in.

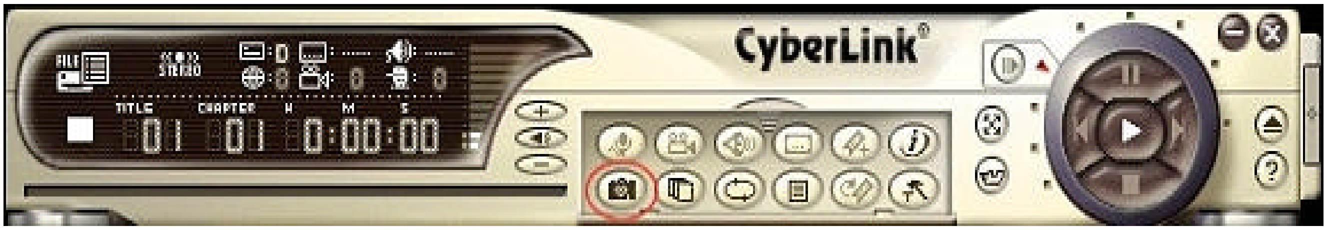
SpecialsPage=3. Specials are the additional movies you get on a DVD such as the Making of the Film or extra scenes that were cut out of it to fit it into a time slot. If you have any special features on the DivX then you can tell MicroDVD what page to display first to let the user select these options. Specials can also be where, like in the Matrix, you are given actor profiles and so on to read. These are all done as separate "web pages" just like anything else. If you don't have any Specials just don't put this line in.

Okay, that's it! If you always keep pages 1, 2, 3 & 4 as your start, menu, chapters, and specials pages then you will never go wrong ;^). So lets start making our pages.

Creating Menu Pages

Menu pages are literally nothing more than a picture (or film sequence) with links on them! If your original DVD has just pictures for its menus, instead of any animated stuff, I would strongly suggest you just rip those from it =^). The easiest way to do this is to navigate the menu using PowerDVD (or any other player that allows image capture) and hit the camera button to take a snap of the screen. Open a good Paint Package such as

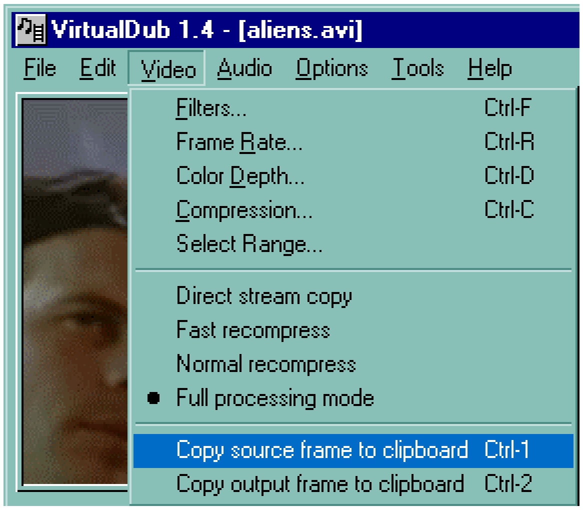
Adobe Photoshop, Paintshop Pro etc., and select Edit > Paste. Then save your grabbed page.



If you are not lucky enough to have these rather expensive Paint Programs there is a free program called IrfanView which does the job excellently check out my dowload section. I recommend that you save each page as highest quality JPEG to save memory unless the image sharpness is absolutely vital. If you don't have PowerDVD you could try hitting the "Print Screen" button on your keyboard and then pasting it.

I'll probably say this again but it is helpful if you know. If you have an animated menu AND use still pictures in your menu too, the pictures must be the same resolution. So if your animated menu movie clip is 640 x 480 then your still pictures must also be saved as 640 x 480 images.

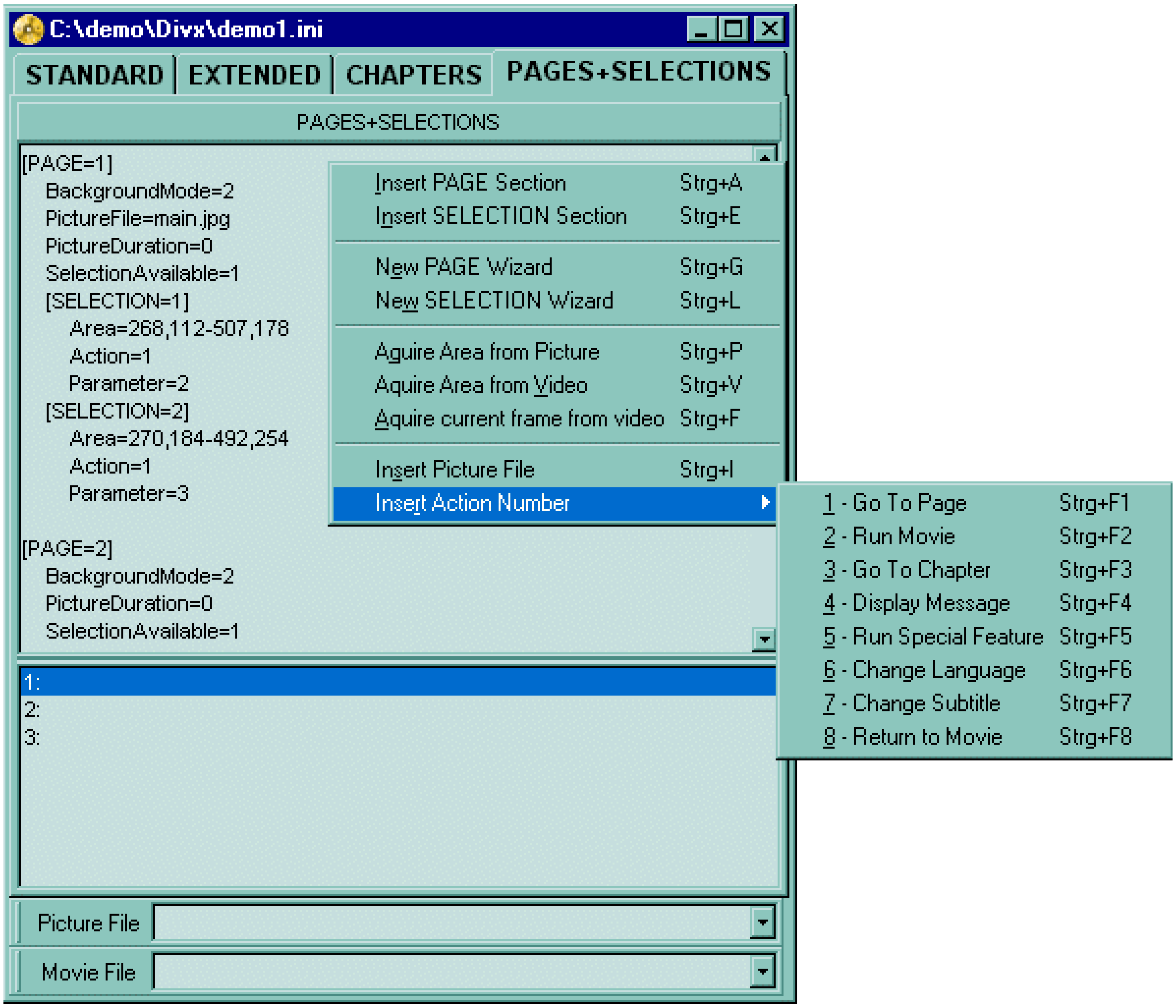
There are many ways to rip the menu's from a DVD. As you probably know, both the animated menu's *and* the still pictures are recorded as a single movie. For still pictures this movie will look rather like a slideshow when played back. The DVD .ifo file knows what parts to play back but all we get when we rip it is the movie. Anyway, this movie is usually the smallest Vob file on the DVD. If you just want to use still pictures encode it with uncompressed AVI or Huffyuv MJPEG for highest quality. If you want an animated DVD start up to your DivX, encode it like any other DivX movie. When done you can get the pictures by opening the DivX movie in VirtualDub. Then you can chop out the movie sequance as you like. Then to get still pictures, move to the picture you want in VirtualDub and choose Video > Copy source frame to clipboard. Then paste it into Paint Shop or whatever art package you are using.



I will explain how to do still picture menu's first.

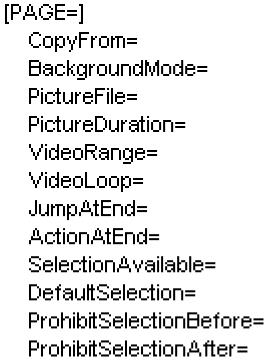
Building Menu Links

Lets take a gander at the ol' Pages+Selections section of the MicroDVD .ini Editor. Now don't get confused here, ignore all lower boxes. The Picture File selector Box and the Movie File Boxes are only there to help you quickly reselect stuff you have already made. The same goes for the larger box in the middle that has 1: 2: 3: in my example picture. That bit is only to help you jump from one page to another, you can just as easily scroll down to find the section yourselves. In short, the only box that concerns us now is the top one. Right-Clicking on the PAGES+SELECTIONS heading brings up the list of options seen on the right of this picture. They provide some very useful functions indeed. Ignore the New PAGE and SELECTION Wizards as they will soon get confusing and they will only do what we are now anyway.


C:\demo\Divx\demo1.ini
STANDARD
EXTENDED
CHAPTERS
PAGES+SELECTIONS
PAGES+SELECTIONS
Insert PAGE Section Strg+A
Insert SELECTION Section Strg+E
New PAGE Wizard Strg+G
New SELECTION Wizard Strg+L
Aquire Area from Picture Strg+P
Aquire Area from Video Strg+V
Aquire current frame from video Strg+F
Insert Picture File Strg+I
Insert Action Number
1 - Go To Page Strg+F1
2 - Run Movie Strg+F2
3 - Go To Chapter Strg+F3
4 - Display Message Strg+F4
5 - Run Special Feature Strg+F5
6 - Change Language Strg+F6
7 - Change Subtitle Strg+F7
8 - Return to Movie Strg+F8
1:
2:
3:
Picture File
Movie File

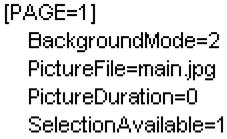
```
[PAGE=1]
   BackgroundMode=2
   PictureFile=main.jpg
   PictureDuration=0
   SelectionAvailable=1
   [SELECTION=1]
      Area=268,112-507,178
      Action=1
      Parameter=2
   [SELECTION=2]
      Area=270,184-492,254
      Action=1
      Parameter=3

[PAGE=2]
   BackgroundMode=2
   PictureDuration=0
   SelectionAvailable=1
```

Lets make our first page. Right-Click and select insert PAGE Section. A whole bunch of crazy crap will suddenly appear i.e:



```
[PAGE=]
   CopyFrom=
   BackgroundMode=
   PictureFile=
   PictureDuration=
   VideoRange=
   VideoLoop=
   JumpAtEnd=
   ActionAtEnd=
   SelectionAvailable=
   DefaultSelection=
   ProhibitSelectionBefore=
   ProhibitSelectionAfter=
```

Go to the [Page=] bit and tell it the page number you are creating, in this case it is page1 so you will put [PAGE=1]. For now, delete all other options except these vital ones:



```
[PAGE=1]
   BackgroundMode=2
   PictureFile=main.jpg
   PictureDuration=0
   SelectionAvailable=1
```

Background Mode=. This tells MicroDVD this page will display either: **1** a video sequence or **2** a picture. Since we are only dealing with pictures choose 2. So we put: BackgroundMode=2.

PictureFile=. Most importantly this tells MicroDVD what picture this page will display. So choose your main menu picture; mine was called main.jpg so I put PictureFile=main.jpg. If you cannot remember what you called the picture, Right-Click after the "=" part and choose the option 'Insert Picture File'. It will then let you browse for the picture you want. Select the picture and click Open. The correct name will be put in and you will notice that the .ini Editor also opens this picture to show you what it looks like. If the picture gets in the way just close it for now.

PictureDuration=. MicroDVD lets you set how long the current page will be displayed. This time period is counted in seconds, so if I put: PictureDuration=10 then the picture would display for ten seconds only. If you put: PictureDisplay=0 the picture will display forever until you choose a link on it. If you do choose to set a time bracket in which to display the picture you must add for your next line the page that the picture will go to after the time limit has run out. This is done by putting: JumpAtEnd=. The number after the "=" represents the page it will jump to. So to recap, to make a page 1 display for only 10 seconds and then jump to page 2, you put:

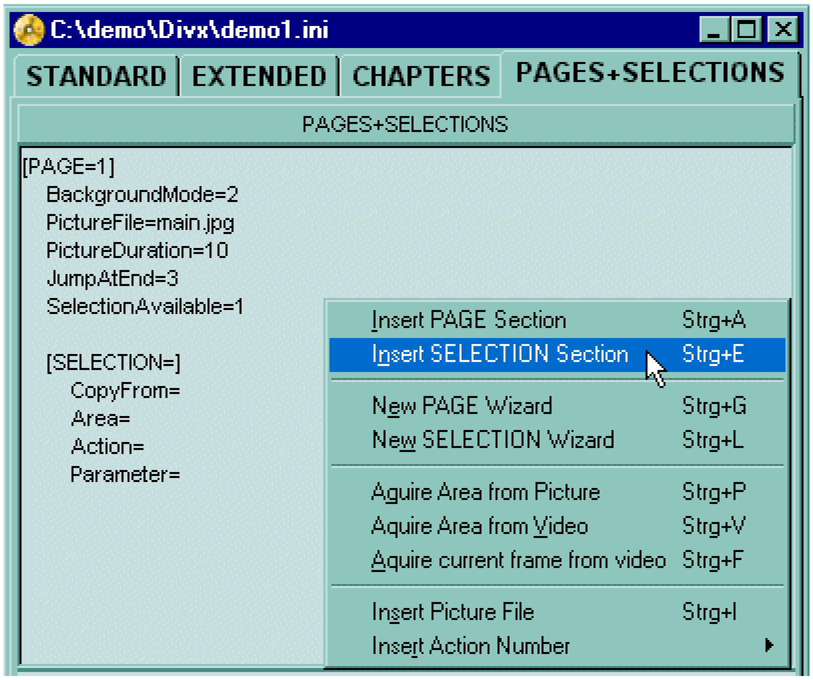
PictureDuration=10

JumpAtEnd=2

SelectionAvalable=. MicroDVD must know if this page contains any links that the user can click on. 1 means Yes, 2 means No! We want to add some links to this page so we will choose: SelectionAvalable=1.

Creating Page Links

So you have created your first page: the Menu Page. Now we have to add links on this page. Right-Click under the page text you just created and choose 'Insert SELECTION section' this will bring up another bunch'a'crazy crap =^) (see below picture). Number the selection heading by putting a 1 after the "SELECTION=" sign just like we did with the Page Section. In other words it should read [SELECTION=1] but any number will do providing you don't use the same number for another selection on the same page.



C:\demo\Divx\demo1.ini
STANDARD
EXTENDED
CHAPTERS
PAGES+SELECTIONS
PAGES+SELECTIONS
Insert PAGE Section Strg+A
Insert SELECTION Section Strg+E
New PAGE Wizard Strg+G
New SELECTION Wizard Strg+L
Aquire Area from Picture Strg+P
Aquire Area from Video Strg+V
Aquire current frame from video Strg+F
Insert Picture File Strg+I
Insert Action Number

```
[PAGE=1]
  BackgroundMode=2
  PictureFile=main.jpg
  PictureDuration=10
  JumpAtEnd=3
  SelectionAvailable=1

  [SELECTION=]
    CopyFrom=
    Area=
    Action=
    Parameter=
```

It is important to remember also that, although the [SELECTION=] heading looks like a separate page (because it has its own heading), it is actually still part of the [PAGE=] section above it. ALL [Selection=] details found underneath a [PAGE=] heading belong to the page above. To illustrate, look at the example below. In this example selections 1, 2 & 3 all belong to page 1 whereas selections 1 & 2 below it will belong to page 2.

```
[PAGE=1]
[SELECTION=1]
[SELECTION=2]
[SELECTION=3]
```

[PAGE=2]

[SELECTION=1]

[SELECTION=2]

A selection is basically another name for a Link. The commands that are put underneath each selection heading will either take you to another page, or perform some kind of action such as starting your movie. Under the Selection heading you will usually have the following options:

Area=. This is literally the position of the link on the screen. It is defined by four numbers representing the four points of an imaginary box surrounding the text or icons on the page. I will show you how to get these Area Positions very soon.

Action=. This tells MicroDVD what will happen when you click on this link on the page. Each action is given a number thusly:

1. Go to Page
2. Run Movie
3. Go to Chapter
4. Display Message (in pop-up box)
5. Run Special Feature (movie)
6. Change Language
7. Change Subtitles
8. Return to Movie

These options need no detailed explanation, but if you need more details on using them check out the MicroDVD documentation.

If you Right-Click after the "Action=" sign and select the 'Insert Action Number', it will let you pick which option you want to use as your action and paste it in.

Parameter=. This command is directly connected to the Action= function. If, for example, you chose Action=1 which means 'Go to a Page' the only way it will know what page to go to will be if you tell it the page number in the Parameter= part. So if I want a link to take me to page 2 I would put the following:

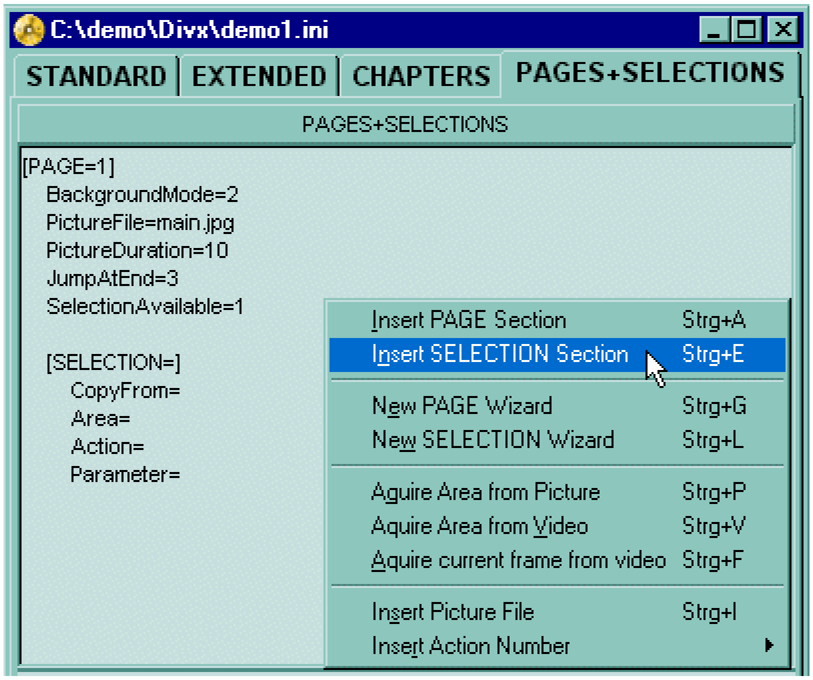
Action=1

Parameter=2

Note: CopyFrom= is an advanced option just delete it for now! What does it do? Well if you find that the settings for each page or selection are almost identical except for a couple of settings, you can then set CopyFrom= to the page number or selection number that is almost the same. MicroDVD will automatically use those settings as default for the CopyFrom page. Then any commands you put after the CopyFrom= page will override the ones that were copied. See the MicroDVD documentation for a full explanation.

The Link Position

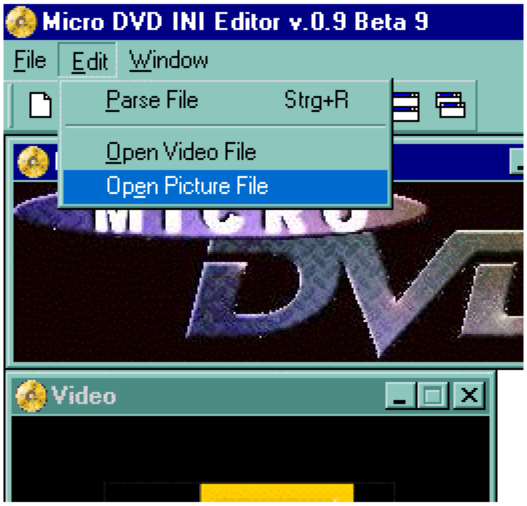
Lets define a link for page 1. We have already inserted our selection (link) and our page looks like this:


C:\demo\Divx\demo1.ini
STANDARD
EXTENDED
CHAPTERS
PAGES+SELECTIONS
PAGES+SELECTIONS
Insert PAGE Section Strg+A
Insert SELECTION Section Strg+E
New PAGE Wizard Strg+G
New SELECTION Wizard Strg+L
Aquire Area from Picture Strg+P
Aquire Area from Video Strg+V
Aquire current frame from video Strg+F
Insert Picture File Strg+I
Insert Action Number

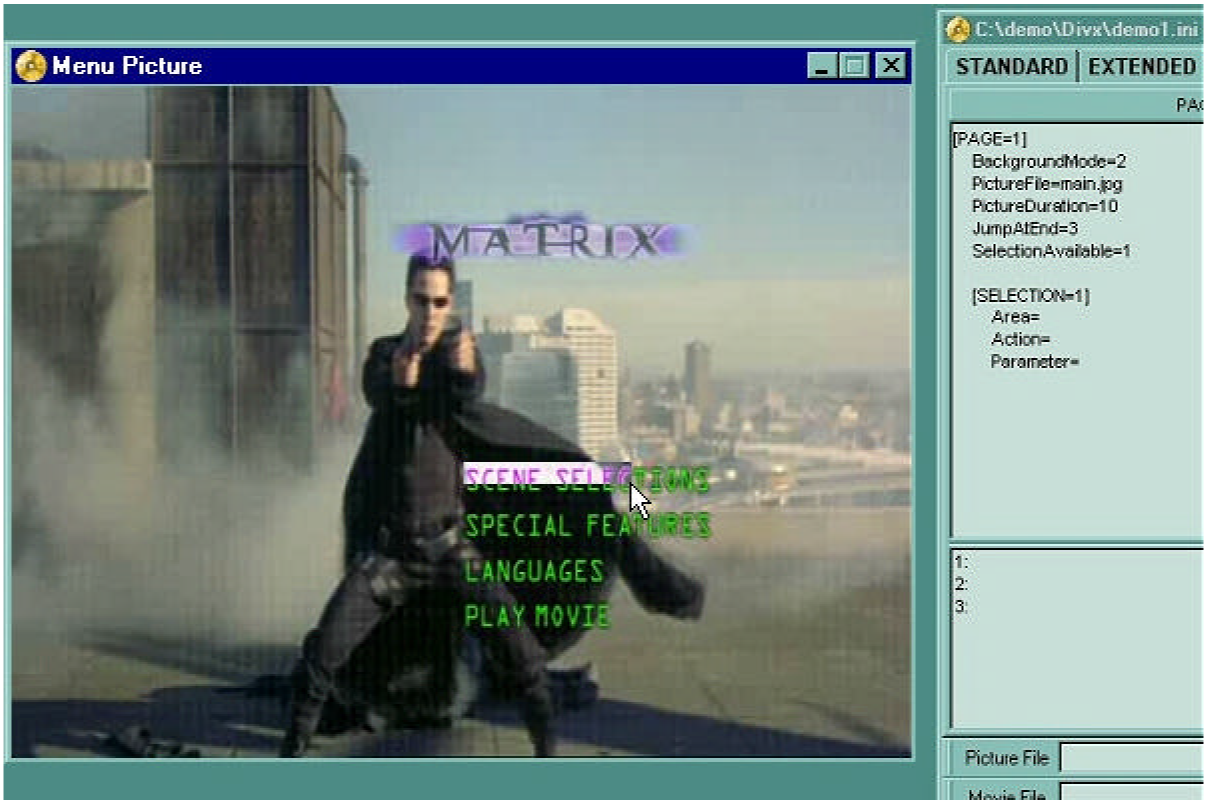
```
[PAGE=1]
  BackgroundMode=2
  PictureFile=main.jpg
  PictureDuration=10
  JumpAtEnd=3
  SelectionAvailable=1

  [SELECTION=]
    CopyFrom=
    Area=
    Action=
    Parameter=
```

Note: Don't forget the above is just one page, the selection part under it is just the properties of the first link on the page!

Okay, open the 'menu.jpg' picture by going to Edit > Open Picture File.



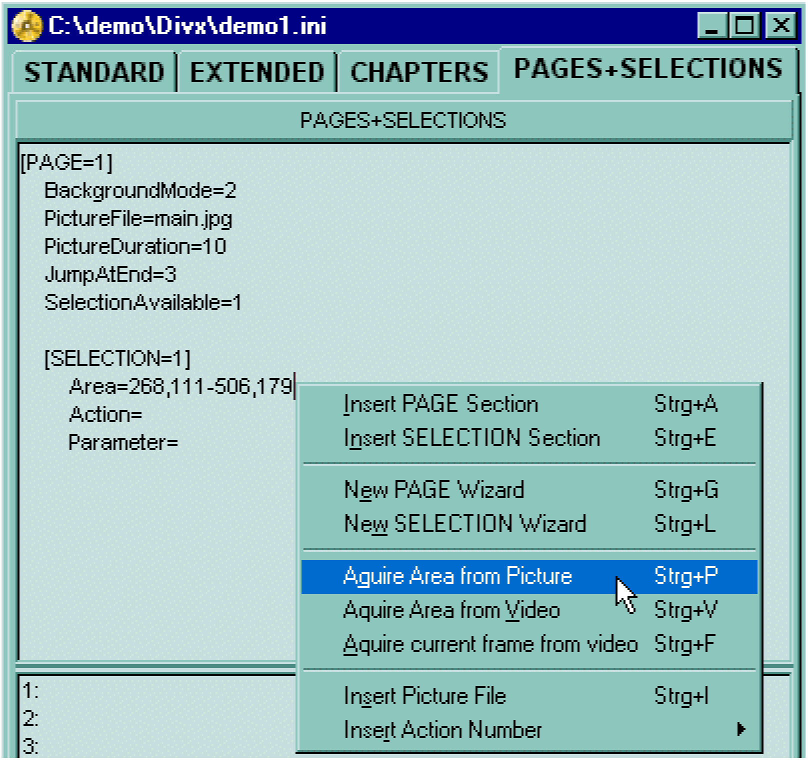
Up pops the Main Menu picture that you ripped from the DVD. We need to put the position of the first link (Chapters link) after the 'Area=' option. So use the mouse to drag a mask box across the word 'Chapters'.



```
[PAGE=1]
  BackgroundMode=2
  PictureFile=main.jpg
  PictureDuration=10
  JumpAtEnd=3
  SelectionAvailable=1

  [SELECTION=1]
    Area=
    Action=
    Parameter=
```

```
1:
2:
3:
```

Now, Right-Click just after the 'Area=' option in the .ini editor and choose: 'Acquire Area From Picture'. MicroDVD ini Editor will now paste the position you just highlighted:


C:\demo\Divx\demo1.ini
STANDARD
EXTENDED
CHAPTERS
PAGES+SELECTIONS
PAGES+SELECTIONS
Insert PAGE Section Strg+A
Insert SELECTION Section Strg+E
New PAGE Wizard Strg+G
New SELECTION Wizard Strg+L
Aquire Area from Picture Strg+P
Aquire Area from Video Strg+V
Aquire current frame from video Strg+F
Insert Picture File Strg+I
Insert Action Number

```
[PAGE=1]
  BackgroundMode=2
  PictureFile=main.jpg
  PictureDuration=10
  JumpAtEnd=3
  SelectionAvailable=1

  [SELECTION=1]
    Area=268,111-506,179
    Action=
    Parameter=
```

```
1:
2:
3:
```

In the picture above I have set the PictureDuration to 10 so it will jump to page 3 after 10 seconds. For a page that doesn't move choose PictureDuration=0 and don't add the JumpAtEnd= at all.

Note: If you are making a video sequance as a menu instead of a picture, the easiest way to find the positions of the text on the video is to capture it as a picture first (with VirtualDub not MicroDVD) and then open it in MicroDVD to get the position. But more about video menus later.

Since this link will take us to another page that contains the chapter listing, we will set its action to 1 (ie. Go to Page) and the Parameter will be 2 (2 is the chapters page number). It will look like this:

```
[SELECTION=1]
Area=268,111-506,179
Action=1
Parameter=2
```

Your Next Link

Now do the same again for your next link on this page ie:

1. Choose 'Insert SELECTION section' and number the selection ie: [SELECTION=2].

2. Highlight the area of the next link on the page and choose 'Acquire Area from Picture'. The next in my list is the 'Specials' "page".

3. The action will be Action=1 (go to a page).

4. And the Parameter=3 (it will jump to page 3 which is my specials page).

And do the same again for the next link on this page and the next and the next until all the links for your page have been done! Now that the links on your page are finished you can start on page two.

Page 2

Start page 2 the same way you created your first page...to recap:

1. Right-Click and choose 'Insert PAGE section' and put the number of the page ie [PAGE=2].

2. Set BackgroundMode=2 (picture only page).

3. Set the PictureFile=chaps.jpg (my chapters page picture is called chaps.jpg)

4. PictureDuration=0 (picture will stay on the screen until a link is clicked)

5. SelectionAvalable=1 (Yes, you want to be able to click on the Chapter links)

Once that is done you can add your links again as described in the "Your Next Link" bit above.

Continue to do this with every page you make just like you did before. Essentially that is all you need to know to create a very complex menu system with just pictures - so go for it!

Advanced Page Options

Okay, that's about it for pages, I'm not gonna spoon feed you on this one =^). Just mess about with all the settings until you make a working set of pages. But perhaps I had better briefly explain some of the more advanced options pertaining to page creation that you may have missed.

ActionAtEnd=. Remember I explained that the 'JumpAtEnd=' option will automatically jump to another page once a certain time limit is up. Well, the action at end is almost the same except instead of jumping to another page it will perform an action. You use the same action numbers that you would for a link action. So if you were to say:

PictureDuration=10

ActionAtEnd=2

This set-up would will wait for 10 seconds and then Run the movie from the start. Other actions are:

1. Go to Page
2. Run Movie
3. Go to Chapter
4. Display Message (in pop-up box)
5. Run Special Feature
6. Change Language
7. Change Subtitles
8. Return to Movie

Note: Links can also trigger multiple actions. For that purpose, the Action entry can hold multiple numbers, separated by a "#", e.g. "7#1" to change the subtitle and then jump to another page. The parameters also have to be separated by a "#". Remember that the parameter for action 4 (Display Message) not contain "#" characters as they would be interpreted as parameter delimiter.

Creating Animated Menu's

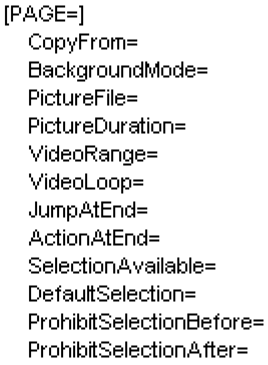
You would think that creating an animated Menu system would be much more difficult than just doing pages, but actually there is very little difference! All pages and links are done exactly the same, except instead of opening a picture to define where the links are found, you just open the video clip. You will select your Link areas just like you would on a normal picture =^). But instead of using the 'Acquire Area from Picture' option, you will Right-Click and use the 'Acquire Area from Video' option. Easy huh! Because of these similarities I will only explain those details that I haven't already explained concerning video backgrounds. If I don't do it this way my short article will turn into an entire book or something!

All video sequences used for the menu backgrounds must show the links *as part of the actual film*. This is important because MicroDVD doesn't let you display any text other than subtitles over the top of a video sequence (unless I've missed something?). So if the original DVD has movie clips with the links already pasted on top of the movie screen then you are in luck. Otherwise you will need to use a professional Video Editing package to edit your video sequences and place text over your video clips, and re-save them like that.

Another thing that is very important to remember is that *all* video sequences that you use in your menu backgrounds must be the same .avi file! In other words, if you have three different sequences and wanted to use one for each page, you must merge them together into one single file first. Let me just say that once again, you only get ONE video file to create all of your menu backgrounds with! This being the case, if you have more than one film sequence you want to use, you must take all the sequence avi files and merge them together using something like Peck Power Join. Power Join should work okay with these files because your intro sequences must be pretty small files anyway. If not, try merging the .Vob intro sequences together into one file with Vobmerger or whatever you prefer to use.

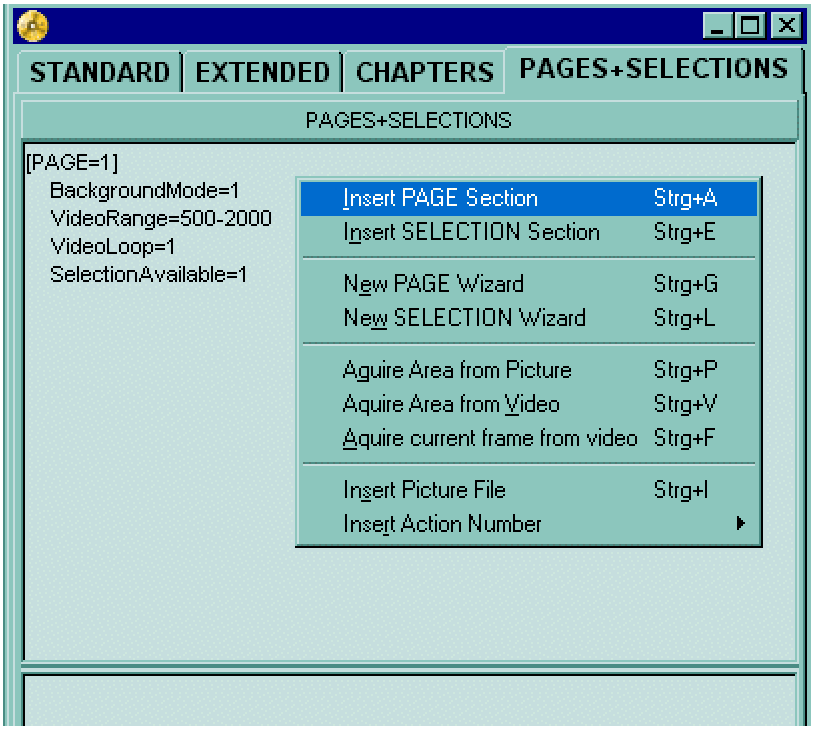
Setting Up the Animated Page

Okay, lets take you through making a single animated menu. Start to create your new page as you would a normal non-animated page. Right-Click and choose 'Insert PAGE Section'. Again you get loads of settings ie:



```
[PAGE=]
   CopyFrom=
   BackgroundMode=
   PictureFile=
   PictureDuration=
   VideoRange=
   VideoLoop=
   JumpAtEnd=
   ActionAtEnd=
   SelectionAvailable=
   DefaultSelection=
   ProhibitSelectionBefore=
   ProhibitSelectionAfter=
```

Just delete the ones you will not use. So delete the last three 'Default' and 'Prohibit' Selection options. Delete the 'PictureFile=' and 'PictureDuration' because you are not using a picture on this page. You can also delete 'JumpAtEnd', 'ActionAtEnd' etc., until you are left with the options shown below:

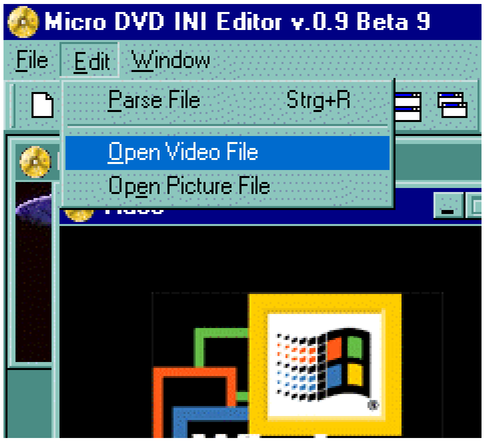

STANDARD
EXTENDED
CHAPTERS
PAGES+SELECTIONS
PAGES+SELECTIONS
Insert PAGE Section Strg+A
Insert SELECTION Section Strg+E
New PAGE Wizard Strg+G
New SELECTION Wizard Strg+L
Aquire Area from Picture Strg+P
Aquire Area from Video Strg+V
Aquire current frame from video Strg+F
Insert Picture File Strg+I
Insert Action Number

```
[PAGE=1]
   BackgroundMode=1
   VideoRange=500-2000
   VideoLoop=1
   SelectionAvailable=1
```

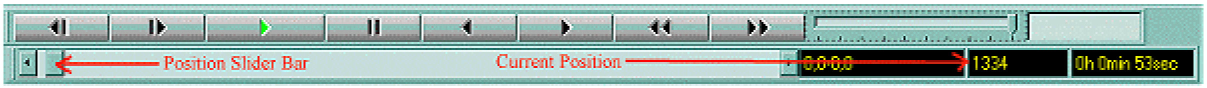
BackgroundMode=. We have set the BackgroundMode to 1. Remember I said that setting it to 2 tells MicroDVD the page is a picture, well setting it to 'BackgroundMode=1' tells MicroDVD it is a film sequence instead.

SelectionAvailable=. Again this is set to 1 as this film sequence has links pasted across the screen. So setting 'SelectionAvalable=1' means links are on this page.

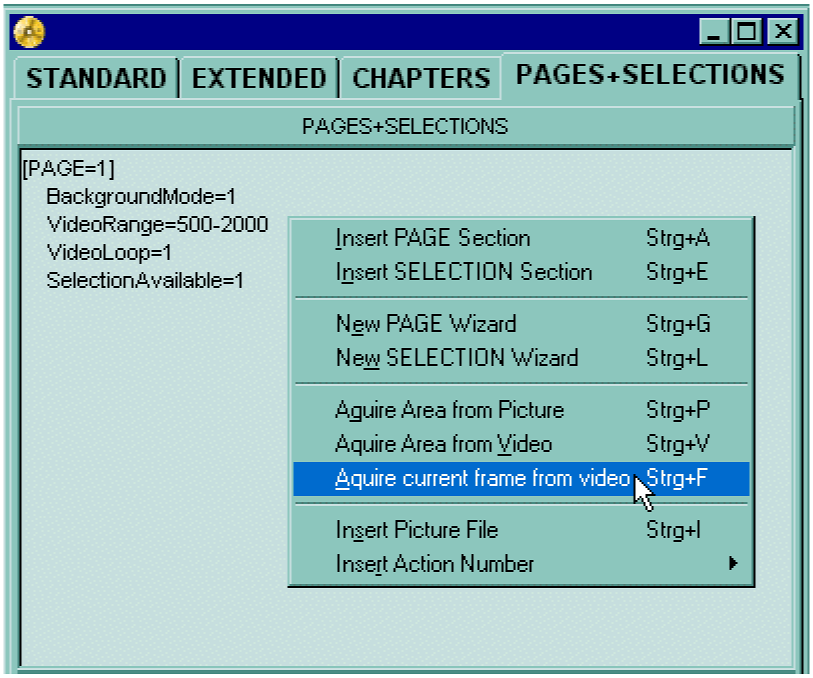
VideoRange=. We haven't seen this feature yet. It tells MicroDVD to start playing the film sequence from one position in the film to another. This is rather like making chapters or subtitles, you have a start frame number and an end time number. To make life easier for you, you can get the start and end points by opening the movie ie:



Using the Position Slider Bar (below) to position your film at the frame you wish to start from. You can also use the pause, fast forward etc., buttons just like any movie player. The important thing is your current position frames number (as indicated in my picture with the Current Position arrow).



Move up and click just after the "=" of 'VideoRange='. Now Right-Click and select 'Acquire current frame from video'. The .ini Editor will now paste the current film position for the first frame number of the Video Range. Use the Video Bar to move to the final position that you wish to play for your page and press pause again. Now go up to the .ini editor once more, select 'Acquire current frame from video' again and you will have the final position that your page will play to.


STANDARD
EXTENDED
CHAPTERS
PAGES+SELECTIONS
PAGES+SELECTIONS

```
[PAGE=1]
    BackgroundMode=1
    VideoRange=500-2000
    VideoLoop=1
    SelectionAvailable=1
```

Insert PAGE Section Strg+A
Insert SELECTION Section Strg+E
New PAGE Wizard Strg+G
New SELECTION Wizard Strg+L
Aquire Area from Picture Strg+P
Aquire Area from Video Strg+V
Aquire current frame from video Strg+F
Insert Picture File Strg+I
Insert Action Number

Make sure that the start and end positions are separated by a dash "-". In the example above the start position is frame 500 and the end position is frame 2000, so I put VideoRange=500-200.

Before we move on it is important to remember that we are not playing the whole .avi file as a background sequence. We are merely picking out clips from it which we will play on our page.

VideoLoop=. Unless you tell the video to loop it will play the frames you specified on your page but when it gets to the end the screen will just stop! You have a choice, either continue to loop this sequence over and over again until someone chooses a link, or get it to play once and then set an action at the end, such as jumping to another page. Setting **1**

mean Yes and **0** means No! By choosing VideoLoop=0 you are saying that you do not want the video to loop after it gets to the end frame. By choosing VideoLoop=1 (as in the above case) we are saying loop this frame forever or until someone clicks on one of the links.

That's it! This is all you need to know to make a repeating movie sequence in the background of your menu page! Creating links for a movie sequence page is EXACTLY the same as doing it on a page, do go to it! To recap:

1. Choose 'Insert SELECTION section' and number the selection ie: [SELECTION=1].
2. Highlight the area of the link on the page and choose 'Acquire Area from Video'.
3. Set the link 'Action=' (setting 1 means go to a page).
4. And add the Parameter= (which page you are linking to).

And do the same again for the next link on this page and the next and the next until all the links for your page have been done! =^)

Final Points

I think I have covered everything you need to know to get started on your own intro sequences with or without movie backgrounds. For detailed descriptions of all the functions MicroDVD can do just refer to the Documentation. But perhaps a few final tips to help you get going would be in order.

Video Links

When you are using the MicroDVD .ini editor to get area positions for a link on a video, you cannot actually see the area that you are selecting like you can with the picture files. Hence, I would strongly advise that you take a snapshot of the video clip first. With this picture you can map out the positions of your links exactly. The important thing is getting the positions right, it doesn't matter how you get them! Use VirtualDub on your menu movie and copy the frame you need.

Pictures & Video

Most DVD's use both still picture menus and a movie background for their menus. The important thing to remember is to make sure the pictures you use as pages are the same size as your movie menu. Otherwise the links will all be positined wrong!

Loops

If you do not want a video sequence to loop forever you can set it to jump to another page or perform another action when it reaches its end frame. Set the just set VideoLoop=0 and choose one of the action commands. These are done pretty much the same as I have already indicated:

Go to Page
Run Movie
Go to Chapter
Display Message (in pop-up box)
Run Special Feature
Change Language
Change Subtitles
Return to Movie

So lets say I want my DivX to open with an intro sequence and then jump to the menu page. Also the menu page has a single link on it that plays the movie. I would make something like this:

```
[PAGE=1]
BackgroundMode=1
SelectionAvailable=0
VideoRange=1-500
VideoLoop=0
JumpAtEnd=2

[PAGE=2]
BackgroundMode=2
SelectionAvailable=1
PictureFile=main.jpg

[SELECTION=1]
Area=268,111-506,179
Action=2
```

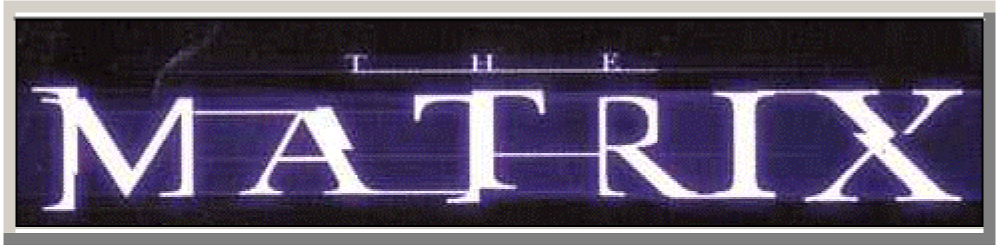
Notice that I have no parameter after my action. This is because Action=2 is to run the film and so I don't need one. But lets say, for example, I wanted to do the same thing but this time I wanted the link to to go to page 3 instead, I would put:

```
[SELECTION=1]
Area=268,111-506,179
```

Action=1

Parameter=5

That's it! I hope you have had as much fun as I did designing these freaky little menus =^). And finally a word from our sponsor:



WHAT IS THE MATRIX?!

"It's that feeling you have had all your life. That feeling that something was wrong with the world. You don't know what it is but it's there, like a splinter in your mind, driving you mad, driving you to me. But what is it?" - Morpheus

Streaming Mpeg-4 from Your Website

Some people asked if it was possible to stream DivX or WMA audio from a website and the answer is yes! In fact, that was why the format was created in the first place. But to do so effectively we must stick with the original ASF format Micro$oft have developed. There is no reason to use Real Media anymore who make us pay for their software anyway. And it doesn't look any better than Mpeg-4 either. I know this is pandering to the Microsoft mentality of 'we give it free with Windows so we can squash the competition', so switch to Linux then - you wanna stream it or not?!

The new streaming formats Microsoft are using are called WMV (Windows Media Video) and WMA (Windows Media Audio). There is practically no differnce between the old ASF and an WMV other than the name (i.e. something.**wmv** instead of something.**asf**) but I think that is an important point to say because the name change is confusing if you don't know.

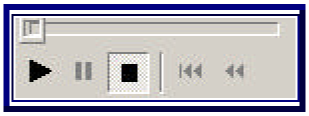
Before I start here are the things you will need:

Windows Media Encoder

Windows Media Encoder is a free utility that can be downloaded from Microsoft. It only really likes to convert AVI and Wave files, but if you have problems just follow any of my guides to convert your video or audio clips into a format Media Encoder can handle.

ADDING STREAMING AUDIO

Click the play buttons below and you will see what you will learn to do:

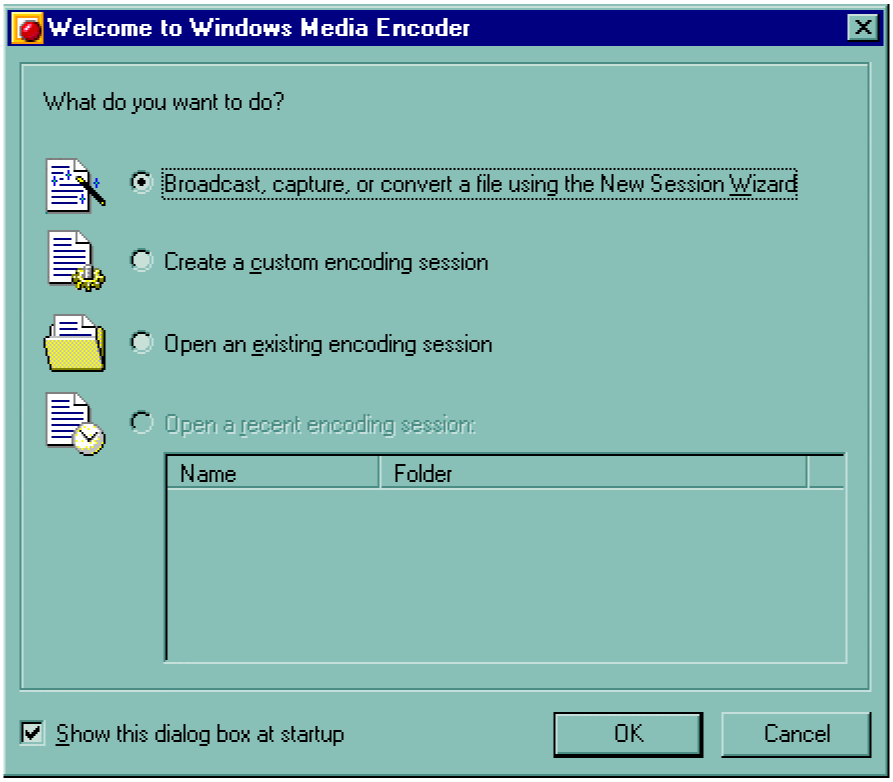


The Matrix Soundtrack - Rage against the Machine

The Matrix - Clip

ENCODING THE AUDIO

This is the easiest part. Load Media Encoder and select the broadcast, capture or convert option. Press OK.



Welcome to Windows Media Encoder

What do you want to do?

● Broadcast, capture, or convert a file using the New Session Wizard

○ Create a custom encoding session

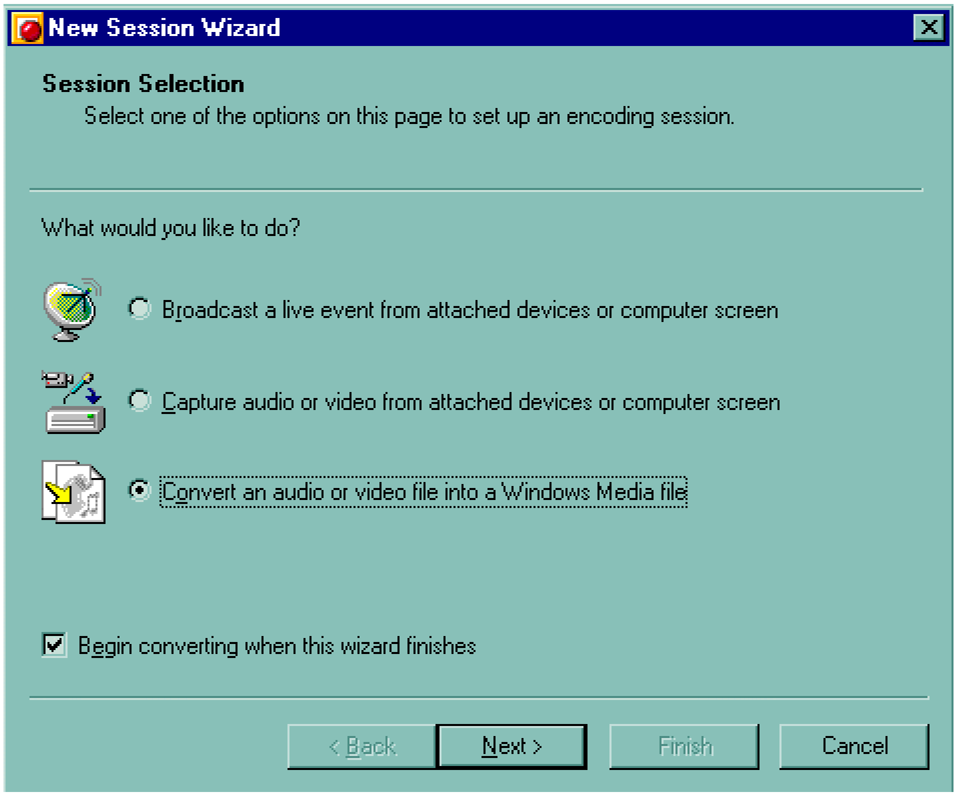
○ Open an existing encoding session

○ Open a recent encoding session:

Name	Folder

☑ Show this dialog box at startup

Choose convert audio or video. Press Next.



Session Selection

Select one of the options on this page to set up an encoding session.

What would you like to do?

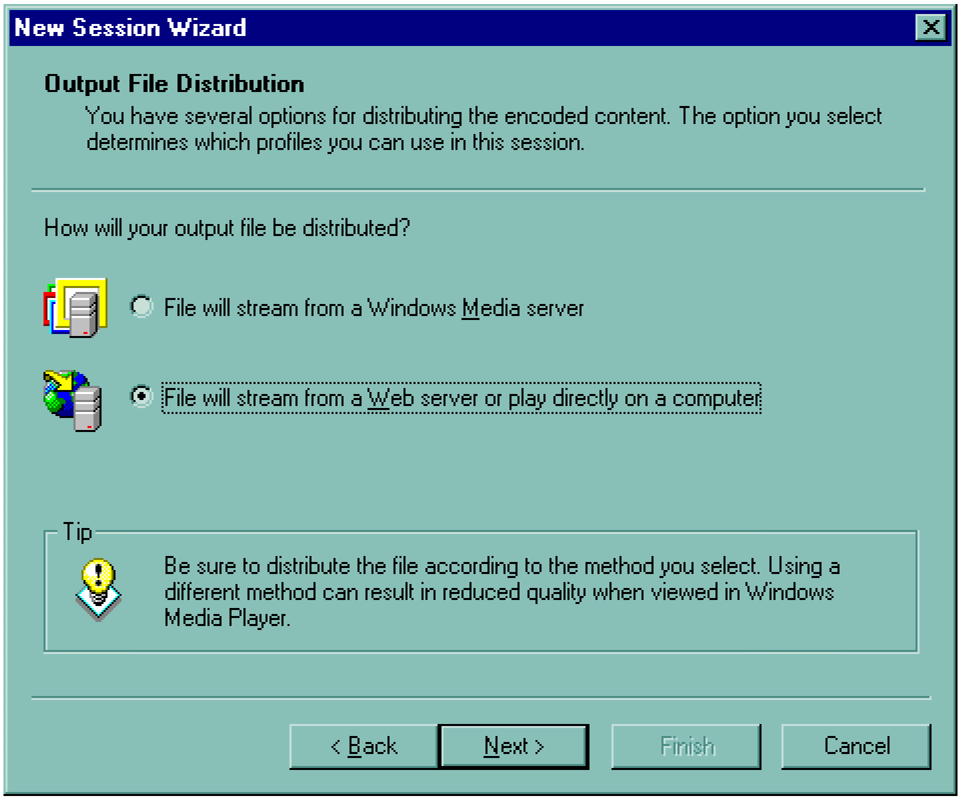
○ Broadcast a live event from attached devices or computer screen

○ Capture audio or video from attached devices or computer screen

● Convert an audio or video file into a Windows Media file

☑ Begin converting when this wizard finishes

Choose file will stream from web server. Press Next.



Output File Distribution

You have several options for distributing the encoded content. The option you select determines which profiles you can use in this session.

How will your output file be distributed?

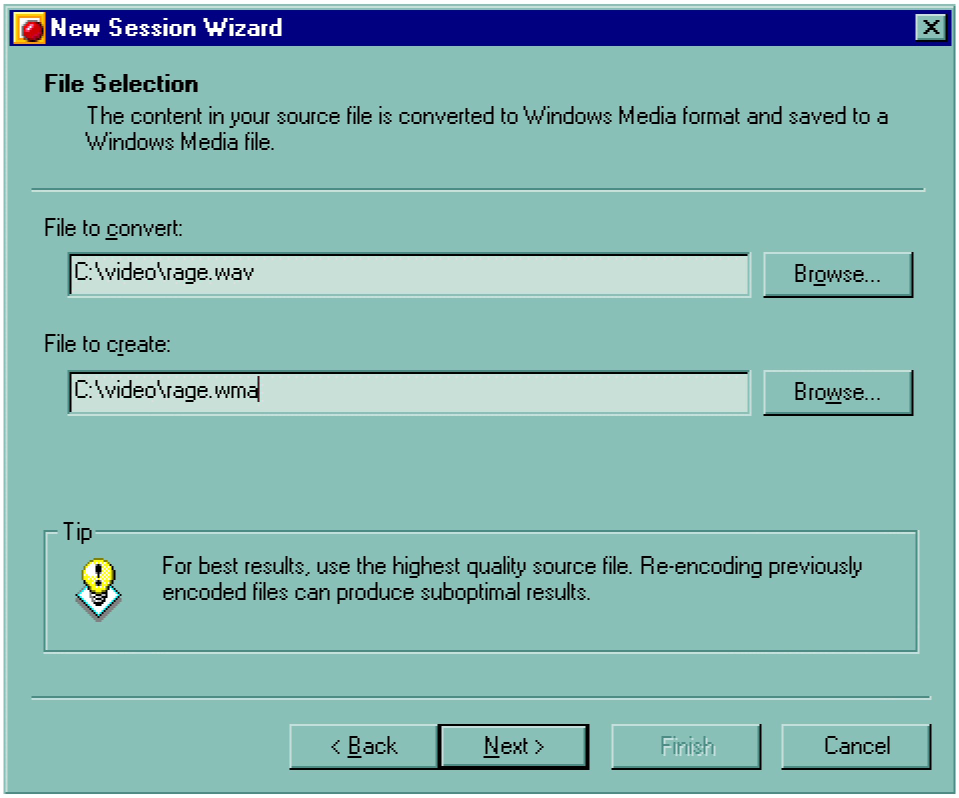
○ File will stream from a Windows Media server

● File will stream from a Web server or play directly on a computer

Tip

Be sure to distribute the file according to the method you select. Using a different method can result in reduced quality when viewed in Windows Media Player.

Choose the AVI or Wave file you wish to convert to mpeg audio or video. Press Next.


New Session Wizard

File Selection

The content in your source file is converted to Windows Media format and saved to a Windows Media file.

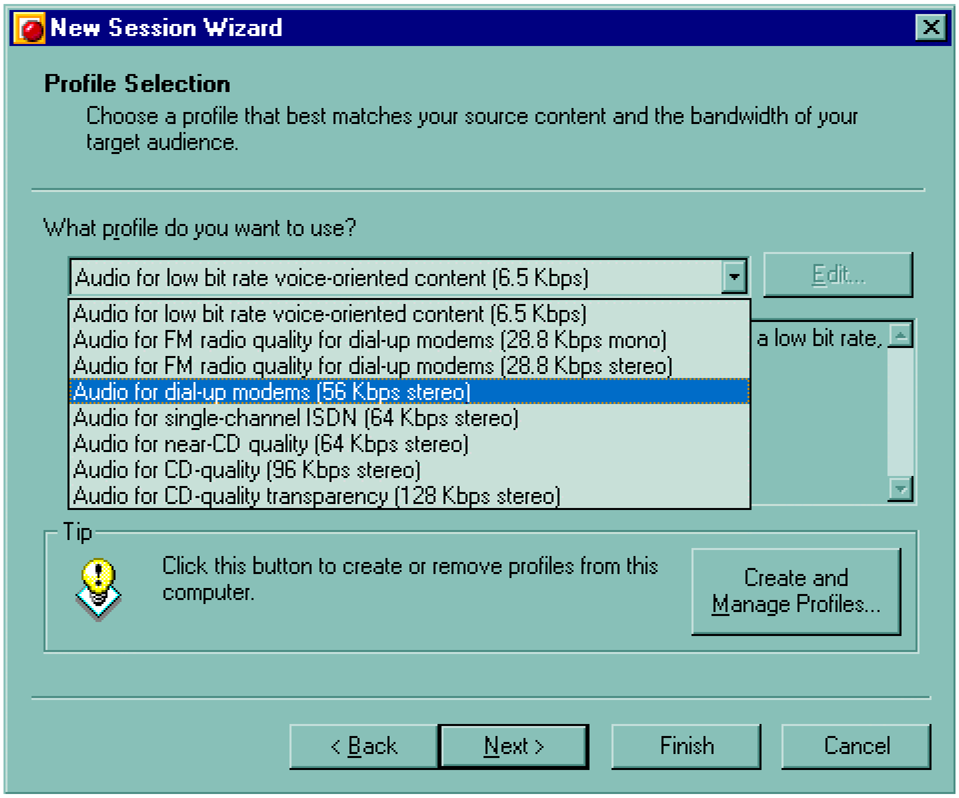
File to convert: C:\video\rage.wav

File to create: C:\video\rage.wma

Tip

For best results, use the highest quality source file. Re-encoding previously encoded files can produce suboptimal results.

Choose the bitrate. The best bitrate to choose is based on the audiance that is going to view the streaming media. If you want 99% of the internet to see it correctly choose 56 kbps modem. If you want only people with Cable modems, DSL or T1 etc., connections to see it, you can go much higher quality and choose ISDN, LAN connection or something like that. Press Next.



Profile Selection

Choose a profile that best matches your source content and the bandwidth of your target audience.

What profile do you want to use?

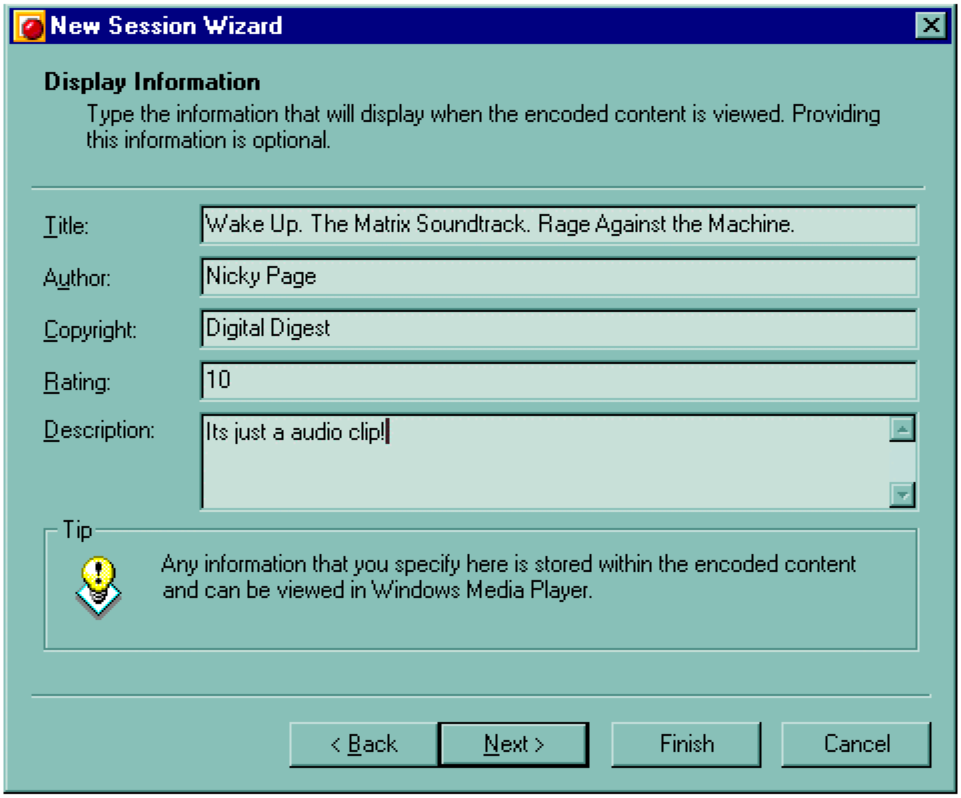
Audio for low bit rate voice-oriented content (6.5 Kbps)

- Audio for low bit rate voice-oriented content (6.5 Kbps)
- Audio for FM radio quality for dial-up modems (28.8 Kbps mono)
- Audio for FM radio quality for dial-up modems (28.8 Kbps stereo)
- Audio for dial-up modems (56 Kbps stereo)
- Audio for single-channel ISDN (64 Kbps stereo)
- Audio for near-CD quality (64 Kbps stereo)
- Audio for CD-quality (96 Kbps stereo)
- Audio for CD-quality transparency (128 Kbps stereo)

a low bit rate,

Click this button to create or remove profiles from this computer.

If you want information about the video to be displayed by media player when prople view or download the clip you can enter it here. Press Next.



Display Information

Type the information that will display when the encoded content is viewed. Providing this information is optional.

Title: Wake Up. The Matrix Soundtrack. Rage Against the Machine.

Author: Nicky Page

Copyright: Digital Digest

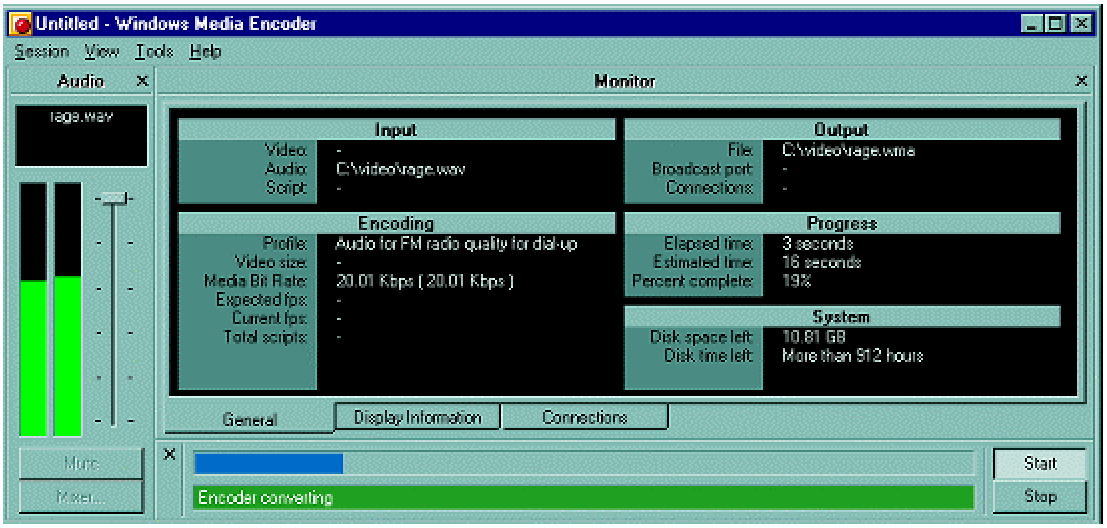
Rating: 10

Description: Its just a audio clip!

Tip

Any information that you specify here is stored within the encoded content and can be viewed in Windows Media Player.

The encoder gets to work, and before you can say: "Hey! That encodes faster than Mp3" its done!



Input

Video: -
Audio: C:\video\rage.wav
Script: -

Output

File: C:\video\rage.wma
Broadcast port: -
Connections: -

Encoding

Profile: Audio for FM radio quality for dial-up
Video size: -
Media Bit Rate: 20.01 Kbps (20.01 Kbps)
Expected fps: -
Current fps: -
Total scripts: -

Progress

Elapsed time: 3 seconds
Estimated time: 16 seconds
Percent complete: 19%

System

Disk space left: 10.81 GB
Disk time left: More than 912 hours

Encoder converting

That was easy enough, huh! Now you have a WMA audio or WMV video which is the same format as ASF's media format.

PUTTING IT INTO YOUR WEBPAGE

If you already have good web design skills it will be easy to add these streams. To embed the video stream you must use Active X Scripting. You can also use Javascript etc., to make nice buttons etc., but this is the easiest way to do it and I'm not going to teach HTML or scripting, to be honest I only understand them very very basically myself!

The Bare Essentials

If your worried its gonna be hard to do, trust me, I'm gonna simplify it for you. The truth is you can more or less just upload your video or audio clip and 'cut & paste' the commands I will show you into your webpage and you will be done! In fact just linking to these *.WMV or *.WMA files, like you would any file, will make them play when someone clicks on the link. But what we really want to is to stick them inside the website so visitors can click on them like a virtual VCR. This is what I will explan now.

All WMV or WMA files have and identification name to tell your internet browser what they are. But instead of calling it 'Mr. Streamiee' Microsoft decided to use the name:

clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95

As I'm sure you know all HTML should be put inside triangular brackets (i.e: < >). So to tell your webpage you have a streaming media clip to embed inside your page you say the following:

```
<object classid="clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95">
```

Then we must tell it what video or audio is being embeded into the page so we must give it the parameter name and location of the file, like this:

```
<object classid="clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95">
<param name="Filename" value="Http://www.webhost.com/yourname/filename.wma">
```

The above is the whole web address, but if the streaming media is stored in the same location as the HTML page we can just use its name such as:

```
<param name="Filename" value="filename.wma">
```

Finally we can end the scripting by using the close object command i.e: </object> like this:

```
<object classid="clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95">
<param name="Filename" value="Http://www.webhost.com/yourname/rage.wma">
</object>
```

That was the bare, bare essentials and will make the clip automatically play as soon as the webpage loads.

Customising Your Streaming Media

Now you know basically how its done we can add on our understanding with some extra commands. Let's add the line:

```
<object classid="clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95" height="352" id="Player"
width="240">
```

The blue part says to tell the web browser to insert the video where ever you paste the text. The green 'id="player" part merely tells it to use a video control pannel. The red parts let you to define how large the video will appear. This means even if you made a clip that was 50 pixels by 50 pixels, if you said the video screen should be 352 x 240 then that is what will be displayed. This is a useful feature becasue you can often make a video look more impressive by making a small resolution clip and then resizing it a third larger on the screen. Try that!

Next come many 'parameters names' that can be added to control the video display. These are just added in a long list underneeth the video. For example, take a look at this rather more complex one below!!

```
<object classid="clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95" height="40"
id="Player" width="139">
<param name="AudioStream" value="-1">
<param name="AutoSize" value="0">
<param name="AutoStart" value="0">
<param name="AnimationAtStart" value="0">
<param name="AllowScan" value="-1">
<param name="AllowChangeDisplaySize" value="-1">
<param name="AutoRewind" value="0">
<param name="Balance" value="0">
<param name="BufferingTime" value="5">
<param name="ClickToPlay" value="-1">
<param name="CursorType" value="0">
<param name="CurrentPosition" value="-1">
<param name="CurrentMarker" value="0">
<param name="DisplayBackColor" value="0">
<param name="DisplayForeColor" value="16777215">
<param name="DisplayMode" value="0">
<param name="DisplaySize" value="5">
<param name="Enabled" value="-1">
<param name="EnableContextMenu" value="-1">
<param name="EnablePositionControls" value="-1">
<param name="EnableFullScreenControls" value="-1">
<param name="EnableTracker" value="-1">
<param name="Filename" value="filename.wma">
<param name="InvokeURLs" value="-1">
<param name="Language" value="-1">
<param name="Mute" value="0">
<param name="PlayCount" value="1">
<param name="PreviewMode" value="0">
<param name="Rate" value="1">
<param name="SelectionStart" value="-1">
<param name="SelectionEnd" value="-1">
<param name="SendOpenStateChangeEvents" value="-1">
<param name="SendWarningEvents" value="-1">
<param name="SendErrorEvents" value="-1">
<param name="SendKeyboardEvents" value="0">
<param name="SendMouseClickEvents" value="0">
<param name="SendMouseMoveEvents" value="0">
<param name="SendPlayStateChangeEvents" value="-1">
<param name="ShowCaptioning" value="0">
<param name="ShowControls" value="-1">
<param name="ShowAudioControls" value="-1">
<param name="ShowDisplay" value="0">
<param name="ShowGotoBar" value="0">
<param name="ShowPositionControls" value="-1">
<param name="ShowStatusBar" value="0">
<param name="ShowTracker" value="-1">
<param name="TransparentAtStart" value="0">
<param name="VideoBorderWidth" value="0">
<param name="VideoBorderColor" value="0">
<param name="VideoBorder3D" value="0">
<param name="Volume" value="-80">
<param name="WindowlessVideo" value="0">
</object>
```

You'd need a whole bunch of articles to explain every feature so I am not gonna go through them all. Instead I will explan what some of the useful ones do and then you can play about with the rest to see how they work. For more information just visit Microsofts website.

The most important thing to know about these commands is that when it asks for a value it usually will be 0 or 1. In other words its asking Yes (1) or No (0). So if you use the parameter: `"ShowCaptioning" value="0"` its asking the question "Show captioning?" "Answer= no!" which tells it not to show any captioning. Anyway lets continue:

<param name="AutoStart" value="0">

This command is vital because it decides if the video starts playing automatically or not. If you have lots of clips on the same page then you will need this option, otherwise everything will play at the same time. Just set the value to zero and the video will only play when the visitor clicks on it.

<param name="AutoRewind" value="0">

This tells the video to start from the begining once its done. Its not a loop to replay the video it only resets it.

<param name="ShowStatusBar" value="0">

This makes the little black bar we see at the bottom of media player, showing the play length, and so on.

<param name="ShowTracker" value="1">

This is the media player slider bar. With most audio files you can move to parts of the song but this doesn't work too well with video files. **0** means turn the slider bar (tracker bar) off, and **1** turns the trackerbar on.

<param name="AutoSize" value="0">

If this value is set to 1 then the video will automatically resize to the original size of the WMV file and not what you define.

<param name="AnimationAtStart" value="0">

If this is set to **1** it will show the media player logo before the video plays. It doesn't look very proffesional to me so I always turn the baby off by setting value **0**.

<param name="TransparentAtStart" value="0">

Normally media player will give a black background before the video starts. If you set this to value **1** then it will start with the same colour as the webpage.

<param name="ClickToPlay" value="1">

This gives the option for the user to click on the video screen to start it playing. This is useful if you don't want any play buttons stuck at the bottom of the video. The visitor can just click on the video and it will play.

<param name="ShowAudioControls" value="1">

Shows the volume control or not.

<param name="ShowDisplay" value="0">

Shows the author details i.e. made by blokie etc.

That all you need to know. If you come across a website with a Windows Streaming Media file but does something you don't know how to do just look at the HTML. To examine it in Internet Explorer just go to: View > Source. Then look for the embeded media file and see what parameters are set.

COMPATABILITY

Only Internet Explorer really supports Active scripting correctly so the above won't always work on Mac or on Netscape browsers. But the good news is there are some final tweeks we can add to our Streaming Media that will allow at least some cross browser compatability. This is the HTML that Microsoft recomend:

```
<!-- BEGIN GENERIC ALL BROWSER FRIENDLY HTML FOR NETSHOW V3 -->
<OBJECT ID="MediaPlayer" width=320 height=240 classid="CLSID:22D6F312-B0F6-11D0-94AB-0080C74C7E95"
codebase="http://activex.microsoft.com/activex/controls/mplayer/en/nsmp2inf.cab#Version=,1,52,701"
standby="Loading Microsoft Windows Media Player components..."
type="application/x-oleobject">
<PARAM name="FileName" value="http://myserver/mypath/myfile.asf">
<PARAM name="whatever" value=""
<PARAM name="whatever" value=""
<PARAM name="whatever" value=""
<EMBED type="application/x-mplayer2"
pluginspage="http://www.microsoft.com/Windows/Downloads/Contents/Products/MediaPlayer/"
SRC="http://myserver/mypath/myfile.asf"
name="MediaPlayer"
width=320
height=240>
</EMBED>
</OBJECT>
```

As you can see it is exactly the same as we did before except it downloads stuff from Microsoft if it hasn't already got what it needs installed. Now this is the final step then, just copy the above for every file you want to stream and make the changes you have already learned about as needed for your perfect streaming video. Good Luck!

ADVANCED TECHNIQUES (Chapters & ASX Files)

Before I start here are the things you will need:

Windows Media 7 Resource Kit

Okay, now you have the basics lets show you how to do some more smart stuff, huh! I honestly couldn't find any websites that explained how to do this kind of thing with ASF, so as usual this guide will be one of the first you ever read. You'd think there would be lots of guides too, because what I'm gonna show you is such as easy thing to do! In my search I could find only the very bare bones of some ASX stuff on the Microsoft website, and to be honest I doubt anyone, but the most persistent, would actually find it. Searching through the Microsoft Website is like searching through a maze!

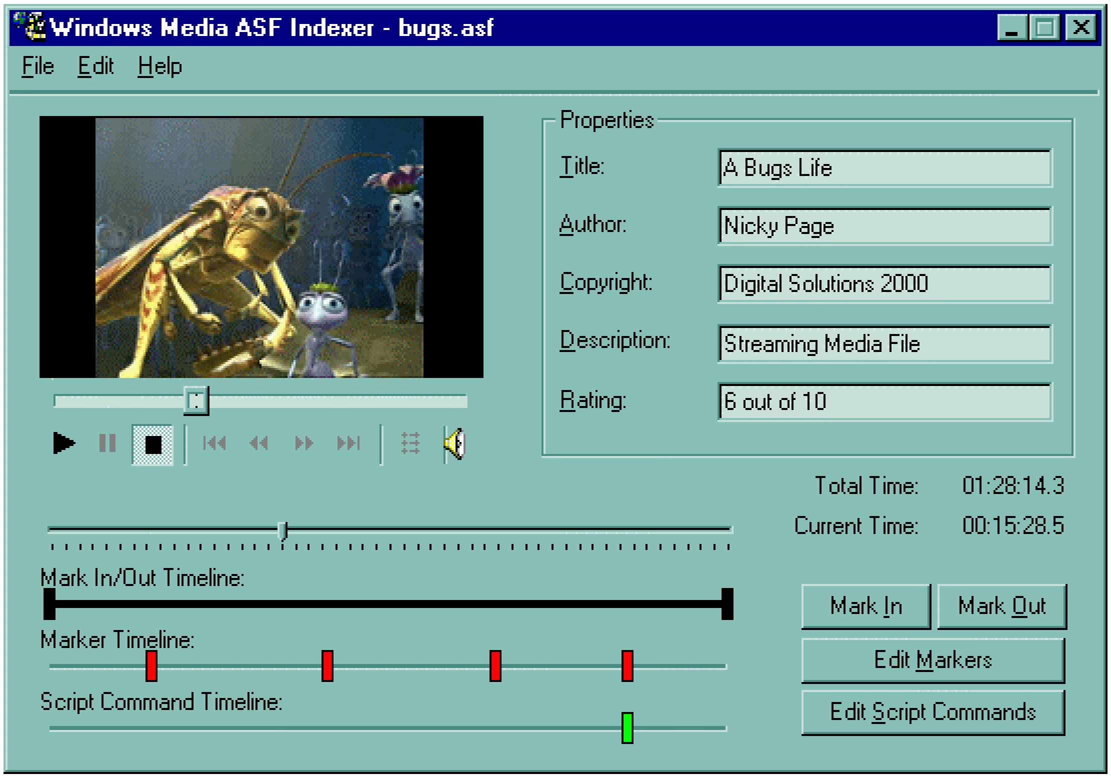
Anyway, download Windows Media 7 Resourse Kit and install. You'll probably find it a pain to download if your on a 56k modem because its one of those direct install from the internet thingies. But the only important utility we need in this pack is Windows Media ASF indexer, you may find the other stuff useful too though. The following guide applies to both to ASF, WMV and WMA files.

CREATING CHAPTERS

Since its hard to move to an exact point in an ASF file by using the slider bar the ASF indexer can be used to set as many chapters as you like, just DVD chapters.

Opening the File

Start the ASF Indexer and go to File > Open... to browse for your file. If your video is called something.**WMV** instead of something.**ASF** just type in the browse box *.wmv and press Enter. Then you should be able to see the WMA file and open it.



Title: A Bugs Life

Author: Nicky Page

Copyright: Digital Solutions 2000

Description: Streaming Media File

Rating: 6 out of 10

Total Time: 01:28:14.3

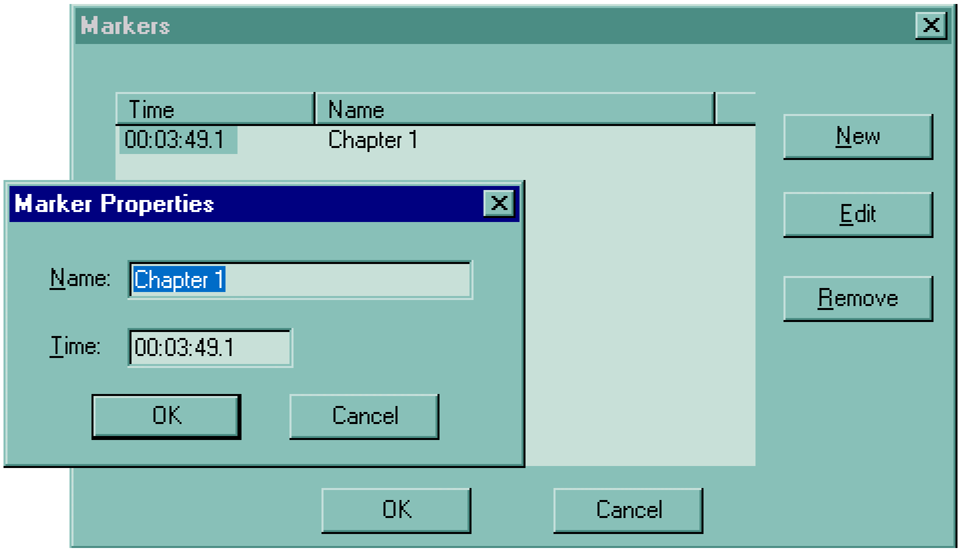
Current Time: 00:15:28.5

As you can see from the above image we have opened our streaming video. The Video has the usual play, stop, pause and volume buttons as any player. Feel free to move about in the video and play it. On the right you can enter any author details as you like and these will automatically be shown once you have saved your final ASF.

Next we have the Mark In and Mark Out buttons. These are only used if you decided you needed to cut your video into a smaller movie. These chopping controls work pretty much in the same way as VirtualDub or TMPGEnc's. Press Mark In to say where the start of the movie will be and then use the Mark Out to say where the movie will end. Then just save the movie and you will find it has been chopped down to the size you wanted.

Adding Chapters

The Edit Markers button has one job only and that is to add chapters. Use the video player and slider bar to move to the exact place you want your first chapter to start and then press the Edit Markers button, up will pop the following box:



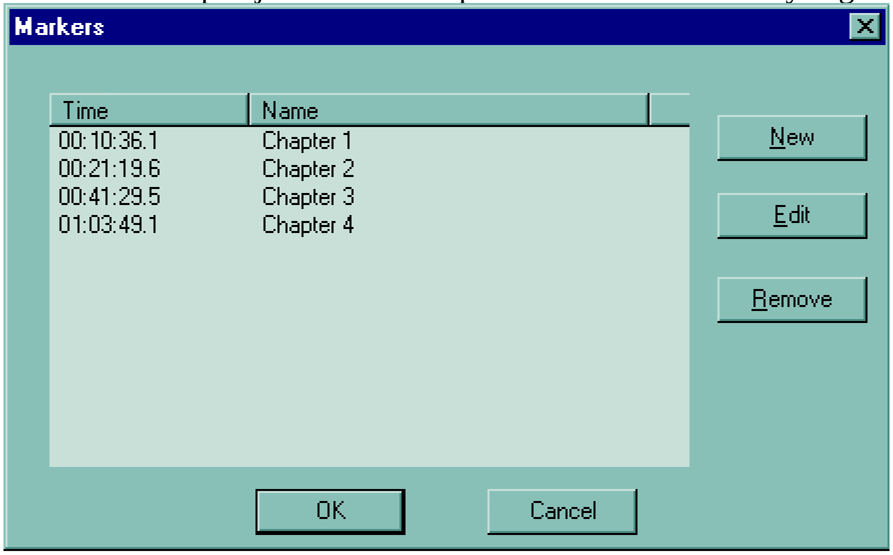
Time	Name
00:03:49.1	Chapter 1

Name: Chapter 1

Time: 00:03:49.1

Press the New button and put in the name of the chapter you want (I used the name Chapter 1 but you can put whatever you like). Notice also the Time box below, this is the position you are in the movie at the moment, so you do not need to change the time now unless you know the exact time you want this chapter to start.

Thats it! Press OK and you have done your first chapter! Now just move to the next part of the movie and add chapter 2, then chapter 3 and so on. A list of chapters will be recorded as seen in the picture below (i.e. chapters 1-4). If you decided you wanted to change any of the chapter times or names just click on the chapter and choose Edit. If you don't want a chapter just select it and press the Remove button - you get the idea!

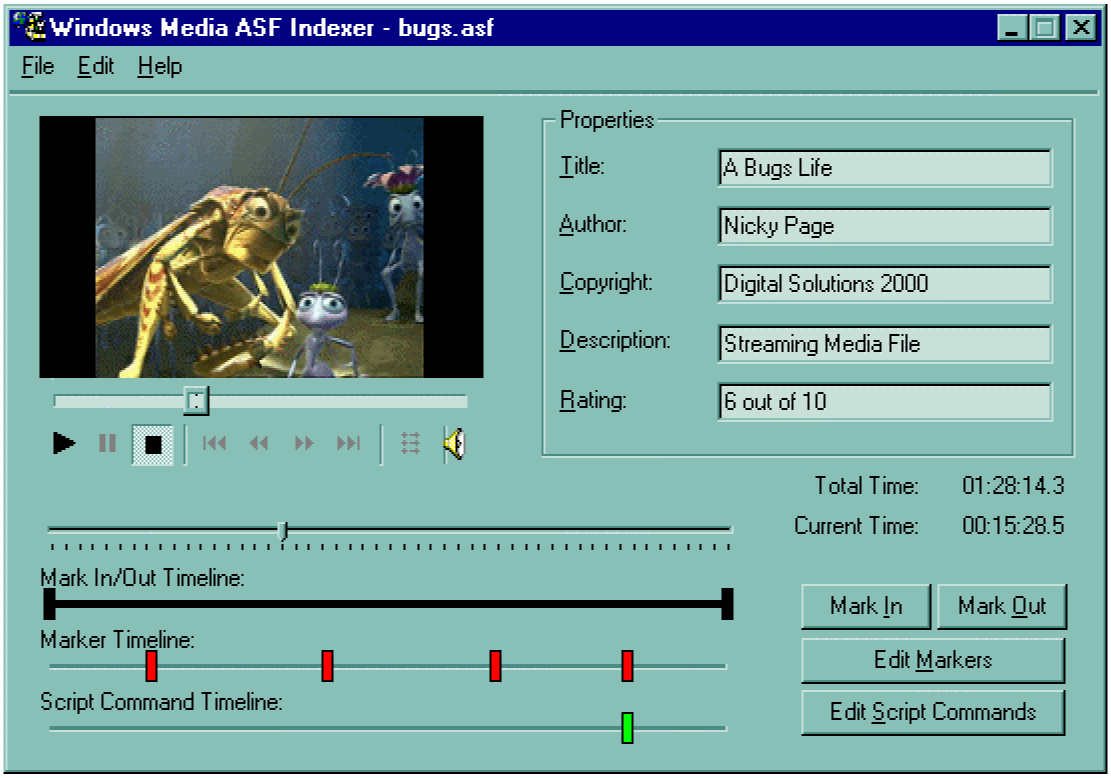


Time	Name
00:10:36.1	Chapter 1
00:21:19.6	Chapter 2
00:41:29.5	Chapter 3
01:03:49.1	Chapter 4

When you are happy with your chapters save the file as an ASF and be amazed!

ADDING URL CHANGES

Lets assume you wanted to make a website that presented something like a new product for your business. Maybe you would like to have your business spokesman explain about it with a super-cool ASF movie? So wouldn't it be great if as he changed from one subject to the next a new webpage could load up with all the details and pictures needed to illustrate the process?! The good new is that's exactly what the Edit Script Commands button does!



Windows Media ASF Indexer - bugs.asf

Properties

Title: A Bugs Life

Author: Nicky Page

Copyright: Digital Solutions 2000

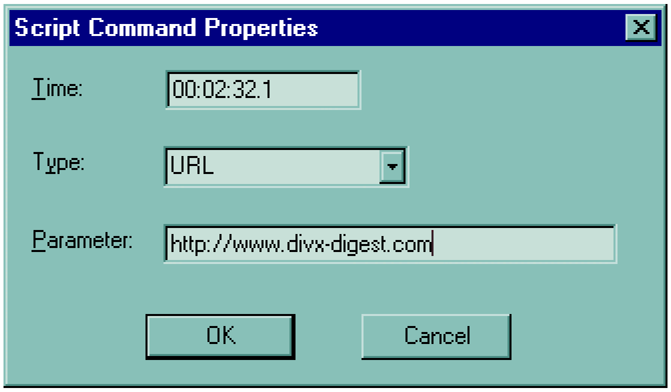
Description: Streaming Media File

Rating: 6 out of 10

Total Time: 01:28:14.3

Current Time: 00:15:28.5

This works exactly the same way as adding chapters. Find the position in the movie where the spokesman is telling them about a certain thing and press the Edit Script Commands button. Select New and the following box appears:



Script Command Properties

Time: 00:02:32.1

Type: URL

Parameter: http://www.divx-digest.com

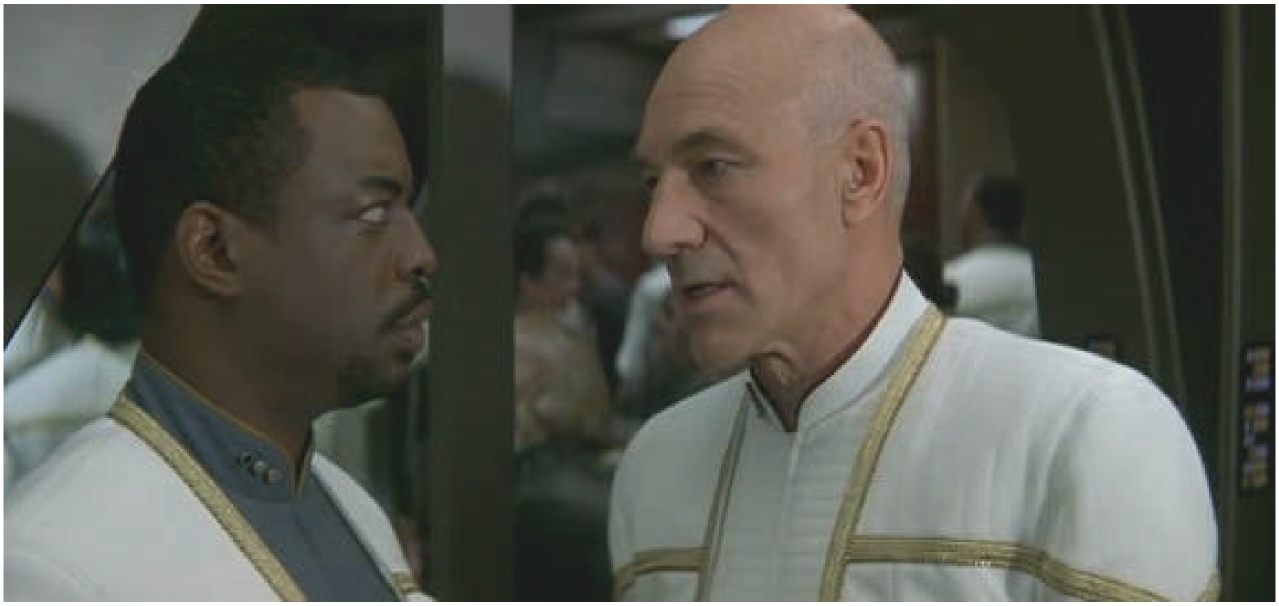
Keep the Type option set to URL and type the webpage you would like to load up at this part of the movie in the Parameter box - Bingo! As soon as your video reaches this point it will automatically load the new webpage without stopping the spokesman from talking - result!

WRITING ASX FILES

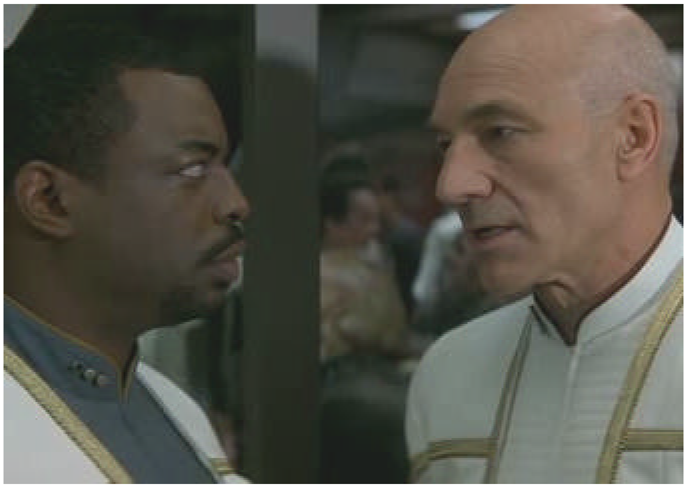
To be continued.....sorry, I'm too tired, I'll finish this one a bit later.

Anamorphic DVD's & Aspect Ratios

4:3, 16:9, Anamorphic, Widescreen, CinemaScope, Panavison, Pan & Scan, Letterbox - the list goes on! But what do all these alien names mean? When we go to the cinema to watch a movie we will see a picture like the one below:



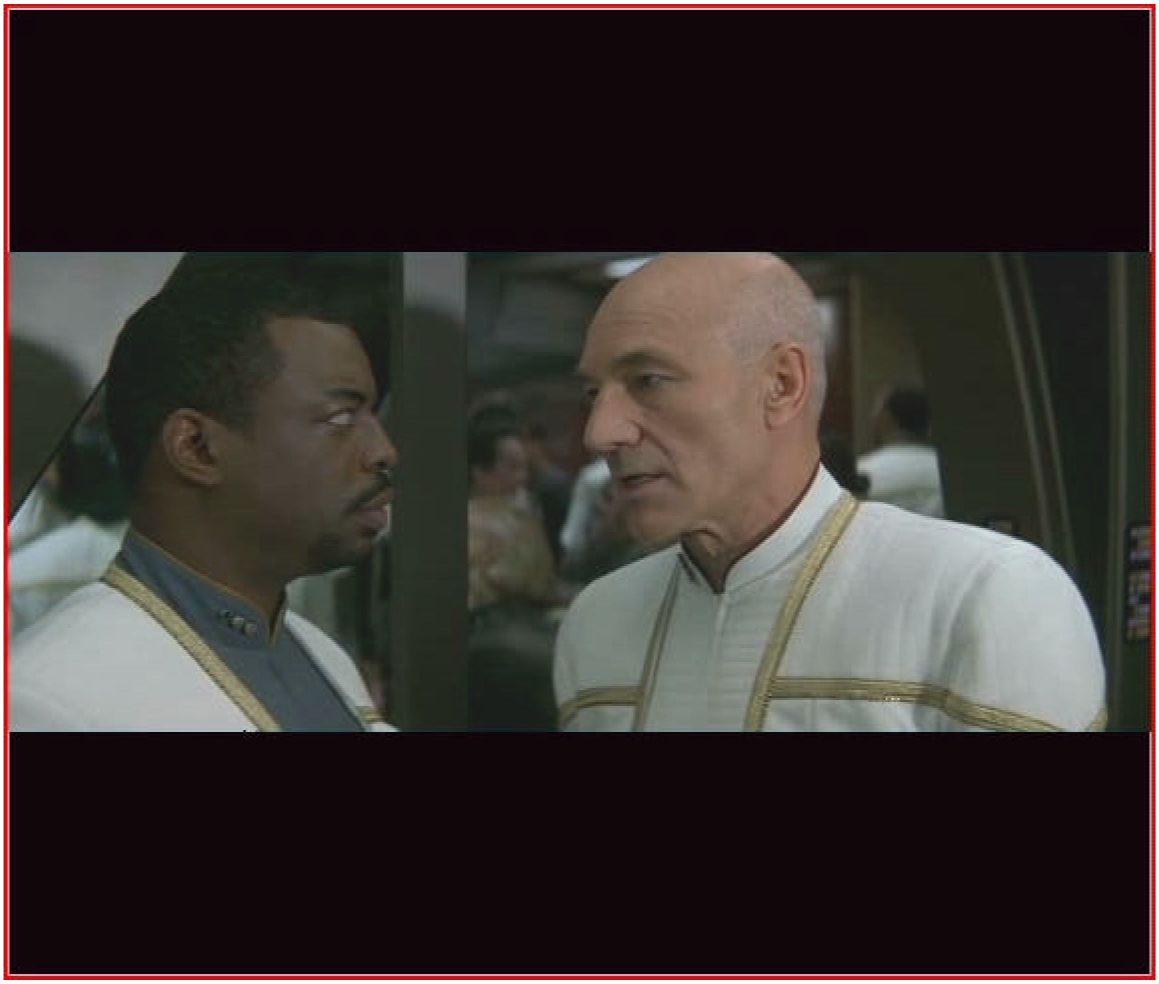
But when we watch the same movie on VHS video we may get something like this:



It is a terrible loss of detail, I know. In the past TV screens could not show wide pictures. So the film industry thought up ways they could convert widescreen movies so they can be viewed on our home TV's.

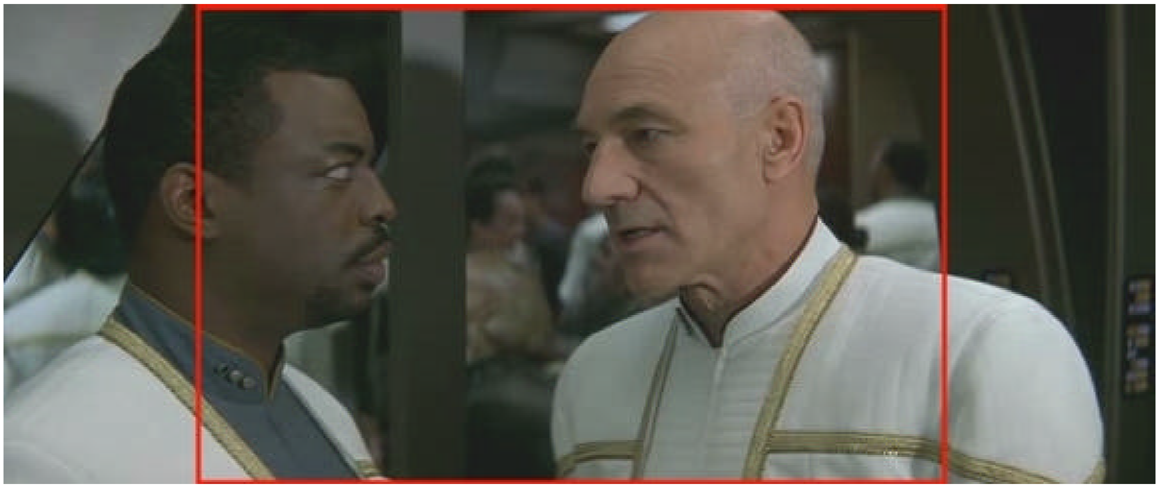
LETTERBOXING

In the picture below the red outline represents the size of your TV. With Letterboxing the picture is shrunk to fit the whole thing in and black bars are added at the top and bottom to fill the gaps.



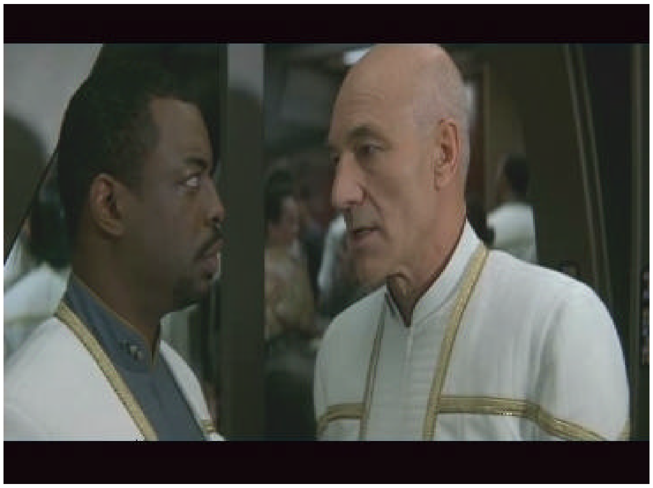
PAN & SCAN

This method is called Panning and Scanning because the the movie is re-recorded from the cinema screen, but the camera pans left and right to scan only for the parts of the movie that are vital. Most old VHS movies are Pan & Scan and, as a result, many of the fine details of the movie were lost. In the picture below the red square represents the only parts of the movie that would be recorded to video with Pan & Scan.



ANAMORPHIC RESIZING

Anamorphic resizing is the royal magic trick of the DVD revolution. The whole movie is recorded at full resolution but the image is literally squashed out of shape and no fine details are lost! It is only with the relatively new introduction of wide screen TV's that it has been possible to view anamorphic movies at full resolution. A DVD player has a built in function that allows it to re-stretch an Anamorphic movie out to full size again. This way anamorphic DVD are able to take full advantage of whatever extra resolution specifications advances in TV technologies make. If your TV is not wide enough to take an anamorphic image, then the image will just be Letterboxed and look like any widescreen movie.



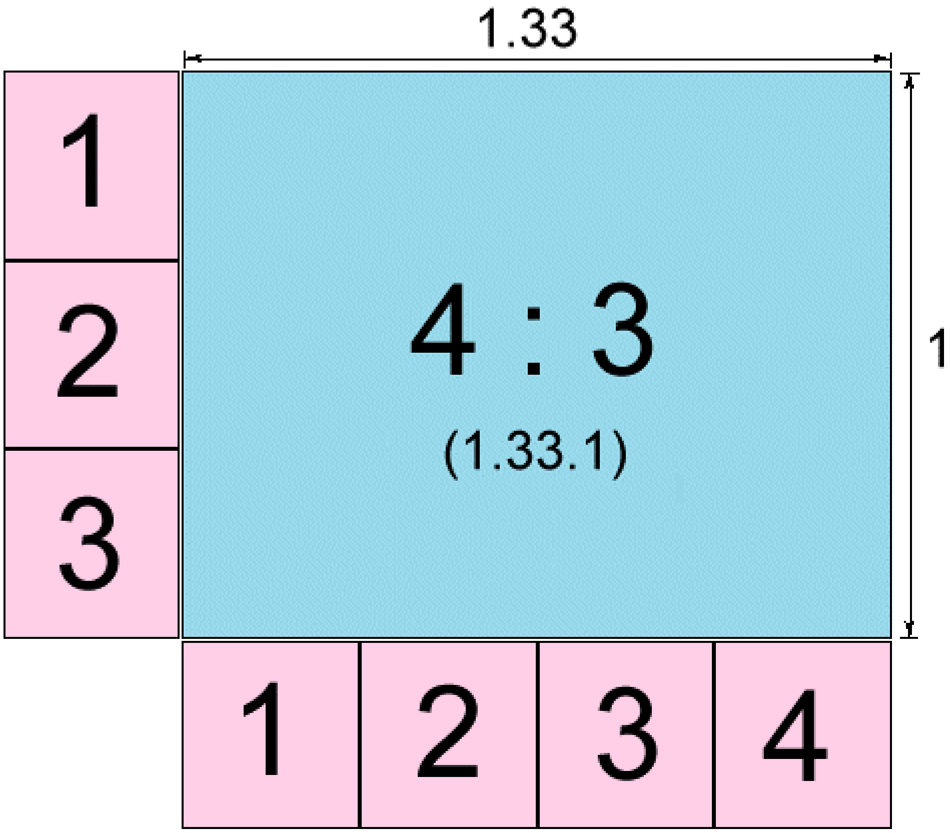
If you have already decoded a Widescreen DVD to your computer, you will probably find that the picture is still squashed. It is not an error with the ripping process, but rather the computers inability to resize the image as would be done on a normal DVD player. The above picture is a 720 x 576 PAL rip, but when opened in a DVD player it will be resized to something like 852 x 480!

If you find that the DVD of your favorite movie doesn't say widescreen, it may be an idea to wait a bit before you buy it. Sometimes they are replaced by the widescreen versions later on, and you will end up buying the movie twice, no doubt to the joy of the movie industry =).

Aspect Ratios

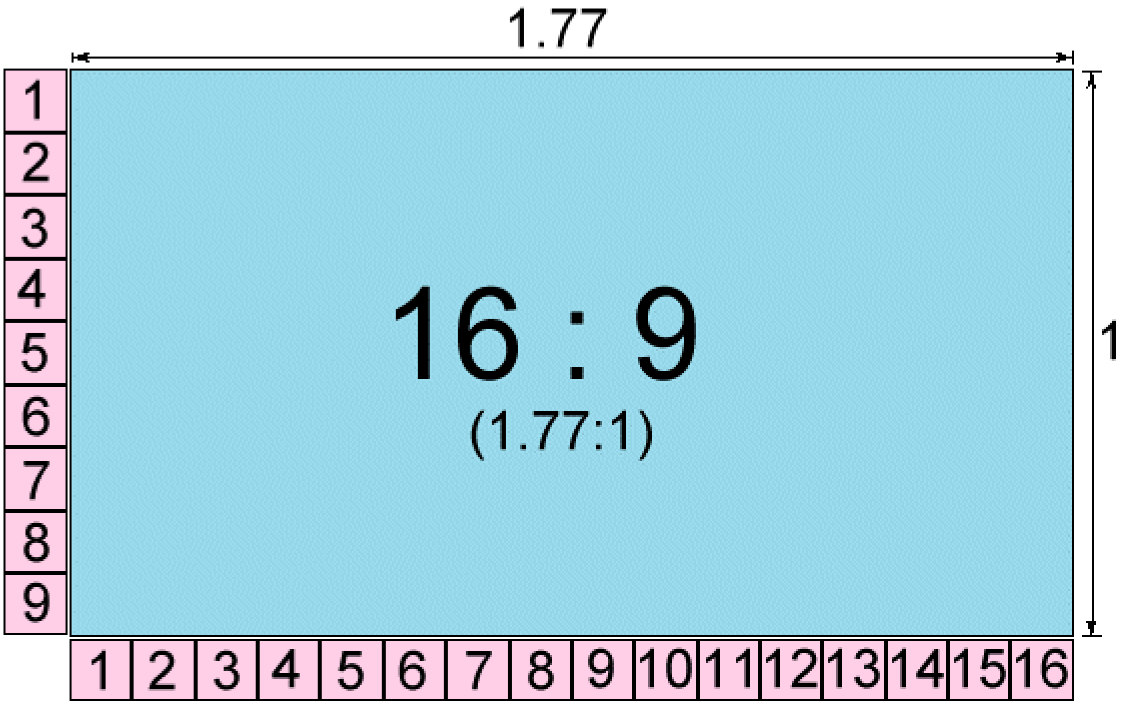
Yes, many of you have looked on a DVD box and read Anamorphic Widescreen 16:9 2.35:1 and thought 'what the Fu*k!' Nevertheless, most of you will be familiar with the idea of ratios even if you *think* you don't. Imagine, for example, you are mixing a cocktail drink. It is four parts vodka to three parts of soda. That would be a ratio of 4:3. It doesn't matter if you used four gallons of vodka to every three gallons of soda, or four egg cups of vodka to every three egg cups of soda, it would still be a ratio of 4:3!

Aspect ratios work exactly the same way. Interestingly enough, your TV screen is probably an aspect ratio of 4:3 too =o). That is, three parts down by four parts across. But to say a picture is always four by three parts makes it hard to make exact measurements. Instead an alternative system is used; we measure the height and the width. For example, in the image below we have a ratio of 1.33:1 (or 1.33 to 1). The 1 can be anything. Imagine, for example, the blue rectangle below was 1 meter high. The width would then be 1.33 meters across, or, if you like: 1 meter and 1 meter 33 centimeters wide.



This is why aspect ratios are used on DVD's. It doesn't matter what measurements we use, it could be millimeters, centimeters, pixels, points, whatever. It always remains in perfect shape if we resize it by the aspect ratio it was originally. There are a couple of 4:3 aspect ratios, for example 1.25:1, 1.36:1, but DVD's that say a ratio of 4:3 will usually use 1.25:1.

A second very often used aspect ratio is 16:9. That is, sixteen parts to every nine parts. This is basically what you will get from a widescreen TV.



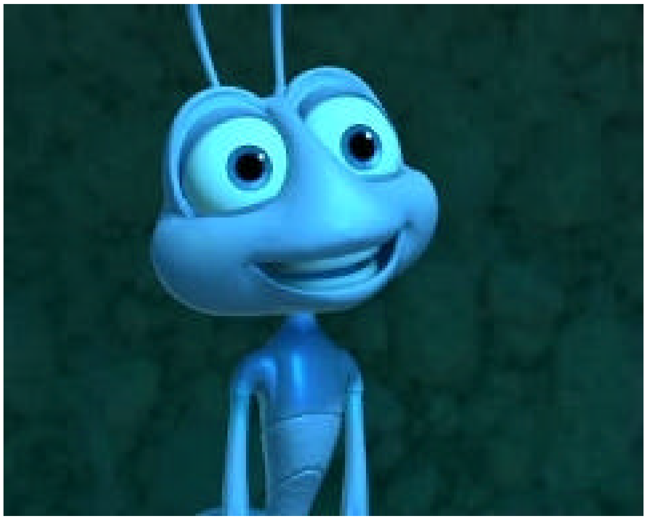
There are a myriad of possible ratios for widescreen. But probably the most used of them for DVD's will be 1:85:1 (American), 1.66:1 (European), 1.77:1 (British) and especially CinemaScope at 2.35:1 (worldwide). CinemaScope are also sometimes called Panavision or just WideScreen. Of course, it could be a mistake to say any aspect ratio is just British or just American etc., because studios just use the ratios they feel are best for their audience.

Resizing DVD's with Correct Aspect Ratio's

Some people are real sticklers for getting perfect aspect ratio. But it's almost impossible to get a perfect aspect ratio when we are limited to resizing by 16 pixel jumps every time! It makes cropping awkward and it also limits us to weird sizes. I have to admit that I haven't always followed perfect aspect ratios.

AN ASPECT RATIO OF 4:3

The picture below is taken from a 4:3 DVD. If your DVD is an aspect ratio of 4:3 then there's nothing to it! All PAL (European) DVD's use 720 x 576 pixels and all NTSC (north American) DVD's use 720 x 480 pixels.



And, since we must resize in blocks of 16 x 16 pixels, this makes the following sizes the *only* sizes we can use:

PAL	720 x 576
Size	**Ratio**
640 x 512	1.25:1
480 x 384	1.25:1
400 x 320	1.25:1*
320 x 256	1.25:1
240 x 192	1.25:1*

Note that 560 x 488 is sometimes suggested, but it is actually a ratio of: 1.147540983606556:1

NTSC	720 x 480
Size	**Ratio**
672 x 488	1.50:1
624 x 416	1.50:1*
576 x 384	1.50:1
528 x 352	1.50:1*
480 x 320	1.50:1
432 x 288	1.50:1*
384 x 256	1.50:1
336 x 224	1.50:1*
228 x 152	1.50:1*

**the resolutions marked with stars indicate sizes that may have trouble playing on a TV-out. This is because they are not multiples of 32 pixels. Of course cropping may cause the same problems with any of the sizes.*

For DivX rips I'd recommend something like 480 x 320 for NTSC. For PAL I'd say 480 x 384 is a good equivalent. I've tended to change my mind a lot on what I consider to be the best sizes to use. But I shy away from very large sizes because they cause too many artifacts. Also, full DVD resolutions will very often cause jerky playback. This will happen as often as the Divx hits a keyframe. For my 500Mhz CPU the DivX codec seems to give an optimum smooth playback with under 1000 blocks of 16 x 16 pixels before problems. Therefore a full 720 x 576 resolution DivX with 1620 blocks is no good for me. If you intend on making DivX's that will play on just about anyone's machine, then I suggest you consider this fact.

ANAMORPHIC ASPECT RATIOS

Here is where life gets tricky. Firstly, many DVD's will say something like: 2.35:1 Anamorphic (Approx.) the "approx" part doesn't inspire confidence in me =). If you have read the section on Anamorphic DVD's you will know that they rip at 720 x 576 (or 720 x 480) and will give a squashed picture like this one:

For those of you who use Flask Mpeg the rest of this information may not be needed. Because, if you check the maintain aspect ratio option, it will stretch the movie to the correct shape for you. But you still may have problems with those god accursed NTSC movies! Hence, the following information will be useful for getting correct aspect ratios in Flask when you keep the aspect ratio option turned off.

STRETCHING 16:9 (2.35:1)

2.35:1 WideScreen is probably the most used anamorphic size in Europe. But because we are limited to 16 x 16 blocks (because of the limitations of the mpeg format), the only size that is exactly 2.35:1 is 640 x 272 pixels. Actually, its an aspect ratio of 2.35294117647059:1 if you really want to impress your friends =).

These are the closest sizes:

672 x 288 (2.33:1)

640 x 272 (2.35:1)

528 x 224 (2.36:1)

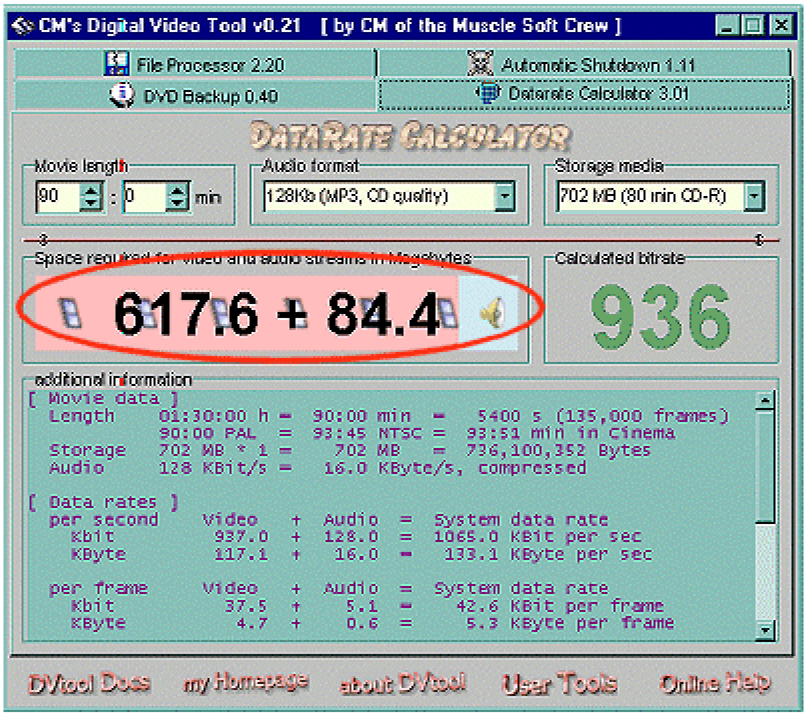
But that's a bunch of bunk anyway! If you resize an anamorphic DVD to 2.37:1 you'll see the incredible pancake show! This is basically because even anamorphic DVD's are still recorded with black bars at the top and bottom of the image.

Roughly speaking, resizing a anamorphic NTSC DVD by an aspect ratio of **1.73:1** will give the correct resolution, not 2.35:1. For PAL DVD's I've worked it out to something like **1.80:1**. These are all just estimated, so if anyone knows of a more exact method e-mail me.

USING CM'S DIGITAL VIDEO TOOL

I will give some detailed listings of aspect ratios a the end of this article, but I cannot list every possible situation. Instead, there is a great utility called CM's Digital Video Tool. It can do many things, but its most useful feature is its ability to calculate aspect ratios for us. Get the application Here.

Lets assume you want to resize your movie to be somewhere between 550-600 pixels across in size. It is an anamorphic 2.35:1 movie so we must use the aspect ratio of 1.80:1 to put it right. Right, we open Digital Video Tools, select the 'Datarate Calculator' tab and click on the bit I've encircled in red.



Movie length: 90 : 0 min

Audio format: 128Kb (MP3, CD quality)

Storage media: 702 MB (80 min CD-R)

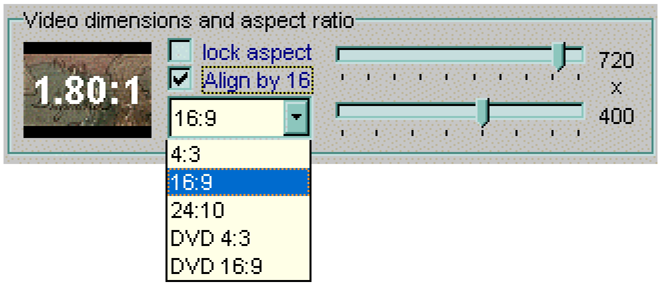
Space required for video and audio streams in Megabytes: 617.6 + 84.4

Calculated bitrate: 936

additional information

```
[ Movie data ]
  Length     01:30:00 h =  90:00 min  =   5400 s (135,000 frames)
             90:00 PAL  =  93:45 NTSC =  93:51 min in Cinema
  Storage    702 MB * 1 =    702 MB   =  736,100,352 Bytes
  Audio      128 KBit/s =   16.0 KByte/s, compressed

[ Data rates ]
  per second      Video   +  Audio   =  System data rate
    Kbit          937.0   +  128.0   =  1065.0 KBit per sec
    KByte         117.1   +   16.0   =   133.1 KByte per sec

  per frame       Video   +  Audio   =  System data rate
    Kbit           37.5   +    5.1   =    42.6 KBit per frame
    KByte           4.7   +    0.6   =     5.3 KByte per frame
```

It will change to the following display:



□ lock aspect

☑ Align by 16

16:9

720 x 400

From the drop down menu you can choose some standard settings such as 4:3 or 16:9 etc. But you can ignore this drop down menu completely if you like! The most important thing is to make sure the 'Align by 16' box is ***always*** ticked and the 'Lock aspect' box is *not* ticked until I tell you :). Then follow these three simple steps:

Step 1.

Move the slider bars left and right until you get the *correct* aspect ratio you need. It will appear in the box to the far left of the picture (in the picture above it is: 1.80:1). Forget the size settings on the right (in the picture above are 720 x 400). The important thing is to get the aspect ratio correct. If you cannot get it exactly then get it as close as possible. So, for example, if you cannot get 1.80:1 you could use 1.82:1 and it would still look almost the same.

Step 2.

Once you know the aspect ratio is correct tick the 'Lock aspect' box to keep it always the same.

Step 3.

Finally, move either slider bar until the movie is the resolution you want it. For example, I moved it down to 576 x 320 and this gave me a perfect size and aspect ratio of 1.80:1.

That's it! Now you should be able to figure out any aspect ratio you want. But here are some helpful guidelines to save some time:

ANAMORPHIC RESIZING (ASPECT RATIO 16:9)

PAL	720 x 576
Size	**Ratio**
720 x 400	1.80:1
704 x 384	1.83.1
688 x 368	1.87:1
762 x 368	1.83:1
656 x 352	1.86:1
640 x 352	1.82:1
624 x 336	1.86:1
608 x 336	1.81:1
592 x 320	1.85:1
576 x 320	1.80:1
576 x 304	1.89:1
560 x 304	1.84:1
544 x 304	1.79:1
528 x 228	1.83:1
512 x 228	1.78:1
496 x 272	1.82:1
480 x 256	1.88:1
464 x 256	1.81:1
448 x 240	1.87:1
432 x 240	1.80:1
416 x 224	1.86:1
400 x 224	1.79:1
384 x 208	1.85:1
352 x 192	1.83:1

NTSC	720 x 480
Size	**Ratio**
720 x 416	1.73:1
704 x 400	1.76:1
688 x 384	1.79:1
672 x 384	1.75:1
656 x 368	1.78:1
640 x 368	1.74:1
624 x 352	1.77:1
608 x 336	1.81:1
592 x 336	1.76:1
576 x 320	1.80:1
560 x 320	1.75:1
544 x 304	1.79:1
528 x 304	1.74:1
512 x 288	1.78:1
496 x 272	1.82:1
480 x 272	1.76:1
464 x 256	1.81:1
448 x 256	1.75:1
432 x 240	1.80:1
416 x 240	1.73:1
400 x 224	1.79:1
384 x 208	1.85:1
368 x 208	1.77:1
352 x 192	1.83:1

ANAMORPHIC RESIZING (ASPECT RATIO 24:10)

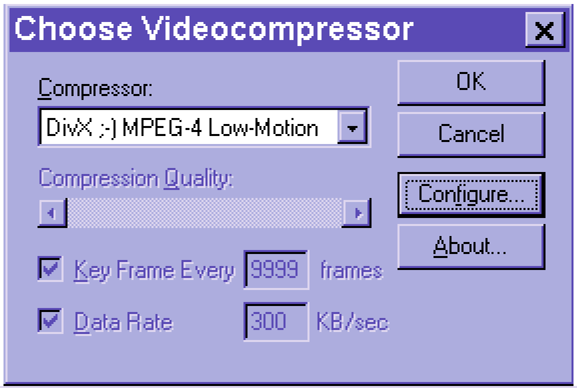
NTSC / PAL	
Size	**Ratio**
720 x 304	2.37:1
704 x 288	2.44:1
688 x 288	2.39:1
672 x 272	2.47:1
656 x 272	2.41:1
640 x 272	2.35:1
624 x 256	2.44:1
608 x 256	2.38:1
592 x 240	2.47:1
576 x 240	2.40:1
560 x 224	2.50:1
544 x 224	2.43:1
528 x 244	2.36:1
512 x 208	2.46:1
496 x 208	2.38:1
480 x 192	2.50:1

DivX Quality & Bitrate Guide

Everyone is looking for the ultimate settings! The ones that will make their DivX movie have the very best possible picture quality for the smallest possible file size! I too have spent much time searching this dark road. And now I feel its about time I gave everyone a few pointers from my experience =). These are just a handful of tips that I have made and should not be taken as though I am to telling you what settings you should be using. Rather, I hope to save you some time and help you to make good choices for yourselves.

INSIDE THE DIVX CODEC

Before we can decide on the best setting to use for our movies, it would be good idea to start by looking at what all those settings on the DivX codec do and what the differences are between the Fast Motion and Low Motion codecs.



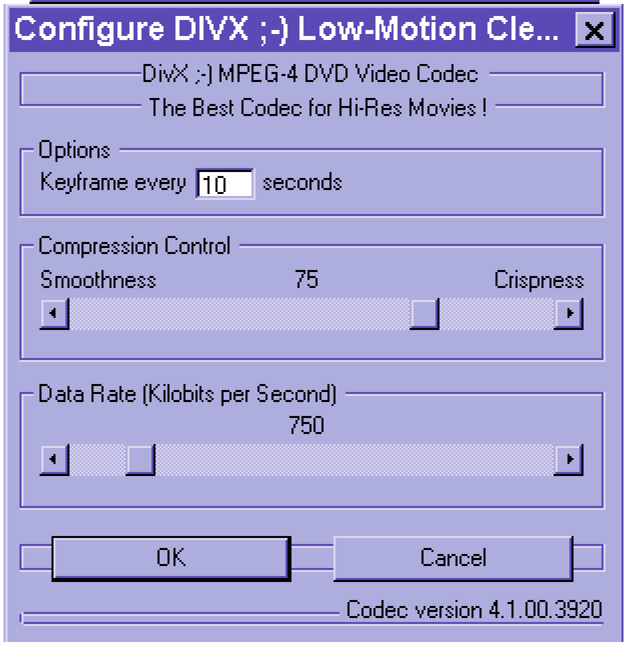
Choose Videocompressor

Compressor: DivX ;-) MPEG-4 Low-Motion

Compression Quality:

☑ Key Frame Every 9999 frames

☑ Data Rate 300 KB/sec



Configure DIVX ;-) Low-Motion Cle...

DivX ;-) MPEG-4 DVD Video Codec

The Best Codec for Hi-Res Movies !

Options

Keyframe every 10 seconds

Compression Control

Smoothness 75 Crispness

Data Rate (Kilobits per Second)

750

Codec version 4.1.00.3920

KEYFRAMES

Most video formats that use compression will have keyframes. These help the video player to seek to various parts of the movie easily and to keep the picture quality good. They also take up the most amount of memory of any video frame. Most codecs will use one keyframe between every 5-10 seconds. The default setting on the DivX codec is 1 keyframe every 10 seconds. This is fine most of the time and changing it will not really increase the image quality enough to notice. In the past people started making DivX's with a keyframe every 9999 seconds to keep down the file size. The result was that you couldn't resume watching the video from where you left off! Every time you played it you had to watch it from the start again!

BITRATE

There are two kinds of bitrate setting used on video compression. Constant Bitrate (CBR) and Variable Bitrate (VBR).

Constant Bitrate: constant bitrate represents how much memory per second a codec will use to encode a movie. Obviously, the higher the bitrate the better the picture quality. Say, for example, one second of video was 25 frames. If I set a bitrate of 1000 Kilobits per Second (kbps) and encode 1 second, each frame would use 40 Kilobits of memory each. If I used a bitrate of 2000 kbps each picture could use 80 Kilobits. Obviously an 80 kilobit frame will look much better than a 40 kilobit frame.

Notice too that it doesn't matter what resolutions we use! Lets say we encoded one second of video at 1000 kbps at a resolution of 352 x 288 pixels - the final filesize would be exactly 1000 kilobytes! Again, if we encoded one second of video at 1000 kbps, but this time we use a cinema resolution of 5000 x 5000 pixels; the final movie clip would *still* only be 1000 kilobytes in size! I know it sounds strange, but we set only '1000 kilobits' for one second of video and that is *all* it will use!

Variable Bitrate: a variable bitrate codec sets its own bitrate in accordance with the movie. To illustrate, mpeg compresion works basically by saving only the differences between sequances of frames. In non-action scenes there are very few differences between frames. Hence, if the total amount of differences are "small" the kilobits needed to store them will also be small. On the other hand, if there are lots of differences between frames it will take "lots" of kilobits to store them.

It is impossible to know how much memory a Variable Bitrate encoded movie will take because it all depends on the amount of Fast Motion and Low Motion scenes it has. Nevertheless, most variable bitrates codec have limits. For example, DVD's are often encoded with a certain maximum and minimum bitrates that will make sure they don't use too much memory and go over the target filesize or use so little memory that it gives a bad looking picture.

SMOOTHNESS & CRISPINESS SETTINGS

One trick people use on AVI files to reduce the filesize is to use a smaller framerate. For example, NTSC movies use a framerate of almost 30 frames per second (fps). If we only used 15 fps then we have effectively halved the filesize! This doesn't look too bad either especially on cartoons which are usually only produced at 15fps anyway!

But although this worked for uncompressed AVI's, no Mpeg format could do this! Say, for example, we tried to encode an Mpeg-1 file at 15 fps. The mpeg codec would automatically add extra frames to make the video at least 23.976 fps which is the speed captured with a motion picture camera! Then it would add a code at the start of the movie to tell the video player to only play it back at 15fps! In other words, the movie would still take the same amount of space as a 23.976 movie but play back at 15 frames a second! When I first wrote my previous quality guide I thought this rule must also applied to DivX, but this was not quite correct. The DivX codec is able to "drop" frames in order to save space!

And here is where the DivX smoothness and crispiness settings come in. The smoothness and crispiness settings refers to framerate preservation. The smoother you set it the less frames are dropped. If we set the crispiness high then it will tend to drop frames in order to save space, but the playback will start to become more jerky! But before you start panicing, this frame dropping effect only happens at very very low bitrates. At the bitrates we would normally encode a DivX, dropped frames are not a big problem. Nevertheless, while I used to say just use 100% crispiness for best quality, now I think keeping it to about 75 is safer. My tests have convinced me that you should not suffer from any dropped frames or jerky playback this way.

Framerates and Bitrates: Finally this brings me to a strange fact. If we encode a DivX at 30fps it will actually end up smaller than the same DivX encoded at 25fps! I cannot really explain this, but I have a theory. A 25fps movie encoded at 1000 kbps would (at a constant birate) use 40 kilobits per frame. But a 30fps movie encoded at 1000 kbps will use about 33 kilobits per frame. 30fps movies are basically 24fps movies with "repeat" frames added to take up space. Now, since, for the most part, Mpeg-4 only records the differences between frames the total differences between one movie of 25fps and the same movie at 30fps is zero! But the codec is now still only using a bitrate of 33 per frame. This is possibly what makes the smaller filesize, which in turn, gives a lower picture quality.

WHATS THE DIFFERENCE?!

"Hi Folks !

Always the same question, always the same answer... The LowMotion codec is a hack from version 4.1.00.4920 of the M$ MPEG4v3, the HighMotion codec is a hack from version 4.1.4917 of the M$ MPEGv3.

I really don't know the internal difference between the low-motion and the high-motion. I NOTICE the difference and decide to make two version, the low-motion which came from the beta version and the high-motion witch is the version 4.1.00.3917, the newer builds looks like be an low-motion style... The low-motion and the high-motion are the same file but with differents builds ! As you know I'm not the coder of this thing so I could not help you more... I mainly use the low-motion which produce better half-toning results, and the high-motion when the picture is very moving.

The final bitrate is extremly related to the CONTENTS of the video, a fast moving, very detailled picture is harder to code than a almost still, very clean picture... I think that the requested bitrate is a MAXIMUM bitrate, and the differences are in the way the encoder try to match this bitrate, the low-motion seems to allocate more bit to color and is 'tighter' to the requested bitrate, the high-motion codec allocate more bit to the luminance but is not as 'tight' as the low-motion is.

Gej

http://divX.ctw.cc"

As you probably guessed the above quote is from Gej the person who made the DivX codec! All Mpeg-4 codec's are just hacks of the Micro$oft ASF codec. This includes SmR (nAVI) and even the Angel Potion codec (according to Avery Lee) which claims to be the first one that isn't a hack. To be fair the Angel Potion does seem to perform a little differently from the rest and is perhaps a highly altered ASF. But either way, at the time of writing this article it was filled with bugs and so is not, in my opinion, a good choice to use. When I say they are "hacks" the only real differences are they allow just about any program to use the codec to encode Mpeg-4 files. There was no alteration to the codec at all - its just like making a hack for a time limited trial, the hacked program itself remains unchanged. So there is no real difference in quality between various Mpeg-4 hacks - all look basically the same quality!

This brings us to the big question of which is better, ASF or DivX! And why bother to hack it anyway? Well, Micro$oft decided no one could use Mpeg-4 unless it was encoded with their crappy encoder. ASF files were only permited to be produced at very small resolutions. ASF's when compared to pure Mpeg-4 files would add almost 100MB's of extra data to the average 650MB movie! This was due to bulky overheads and all the stuff designed to make them stream over the inernet. ASF's didn't allow MP3 audio and almost always ended up with serious audio synchronizatioin problems. Later versions of Media Encoder only allowed usage of the Fast Motion codec which produced bad image quality for low motion scenes! In short, DivX solved all these problems and anyone who says DivX is just a joke and ASF's look better just doesnt know what they are talking about!

FAST MOTION VS LOW MOTON CODECS

Gej was correct in his observations of the Fast and Low Motion codecs. BOTH codecs are variable bitrate which means the final filesizes are very hard to predict.

Fast Motion Codec

The Fast Motion codec is the hardest codec to predict a final filesize. You could encode one movie at 6000 kbps and another at 900 kbps with the Fast Motion codec and still end up with two movies of almost the same size! Or the one at 6000 kbps could end up double the size of the 900 kbps one! There is just no way to tell! When we set the Fast Motions bitrate we are setting the *Maximim* bitrate! This means it will always use an average bitrate of about 300 kbps until it reaches a high action scene and only then will it increase the bitrate to the maximum level! This is why people get confused, because it doesn't use the bitrate they put in until it finds an action scene!

Low Motion Codec

The Low Motion codec, on the other hand, uses a *Minimum* bitrate. This means that it will hardly ever go higher than what we set it. So if we set it to 800 kbps it would use 800 kbps on most scenes and only use a very little more or less depending on the action. This means the Low Motion codec is a lot more predictable.

COMPARING THE TWO CODECS

The following examples will make clear the strengths and weaknesses of the two codec's.

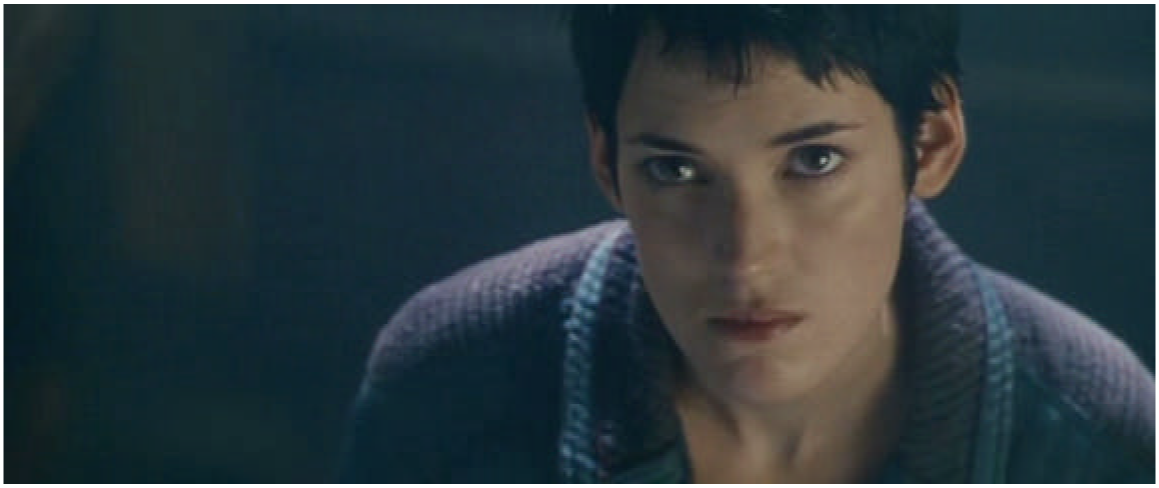
Fast Motion codec

Here is a high-action scene compressed with the Fast Motion codec at 6000kbps:

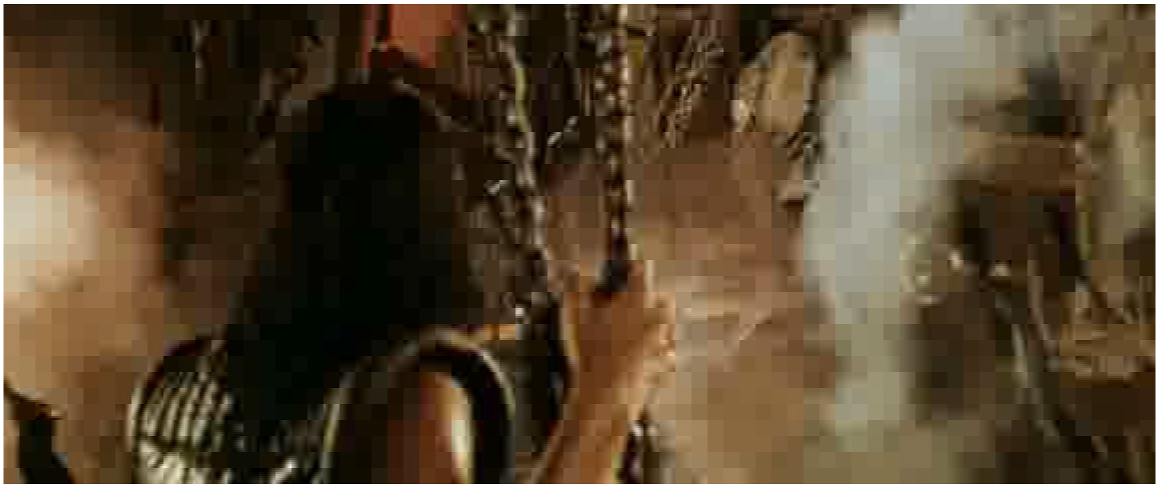
Here is a non-action scene compressed with the Fast Motion codec at 6000kbps:

Low Motion codec

The following is a non-action scene encoded with the Low Motion codec at 600 kbps:



Here is a high-action scene compressed with the Low Motion codec at 600kbps:



From the above examples it is clear that the Low Motion codec always looks better on non-action scenes even if it uses a bitrate as low as 600 kbps. It is noteworthy too, though, that when the Low Motion codec goes below 600 kbps it will not look significantly better than the Fast Motion codec unless the Fast Motion codec is set to the same bitrate too.

The Fast Motion codec also has an upper limit of about 2000 kbps. It doesn't seem to make much noticable difference to quality whether we choose Fast Motion at 2000kbps or 6000kbps! It does, however, make significant demands on filesize.

Finally, as we can see from the examples below, when the Low Motion codec is set to between 1000-1500 kbps it starts to look overall better quality than the Fast Motion codec no matter what we set it to!

Fast Motion 6000 kbps

Fast Motion 2000 kbps

Low Motion 1500 kbps

CONCLUSION

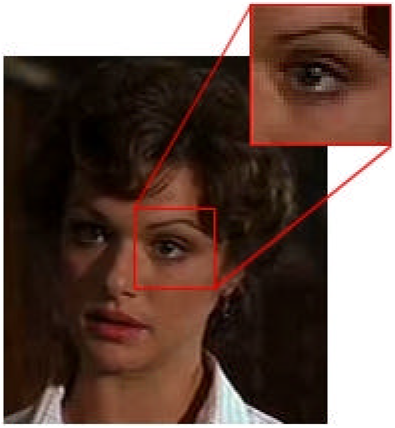
The Fast Motion codec saves the most space and does very well on high action scenes. If you are going to use it for movies always set it to 2000 kbps. There is little point using any other setting and it has the best quality to size ratio. You might consider using the Fast Motion codec for putting long movies on a single CD.

The Low Motion codec set at 600 kbps will do better job than the Fast Motion codec on just about all low action scenes. It cannot compete with the Fast Motion codec on action scenes until we set a bitrate of between 1000-1500kbps. After which, it starts to produce much better results than the Fast Motion codec.

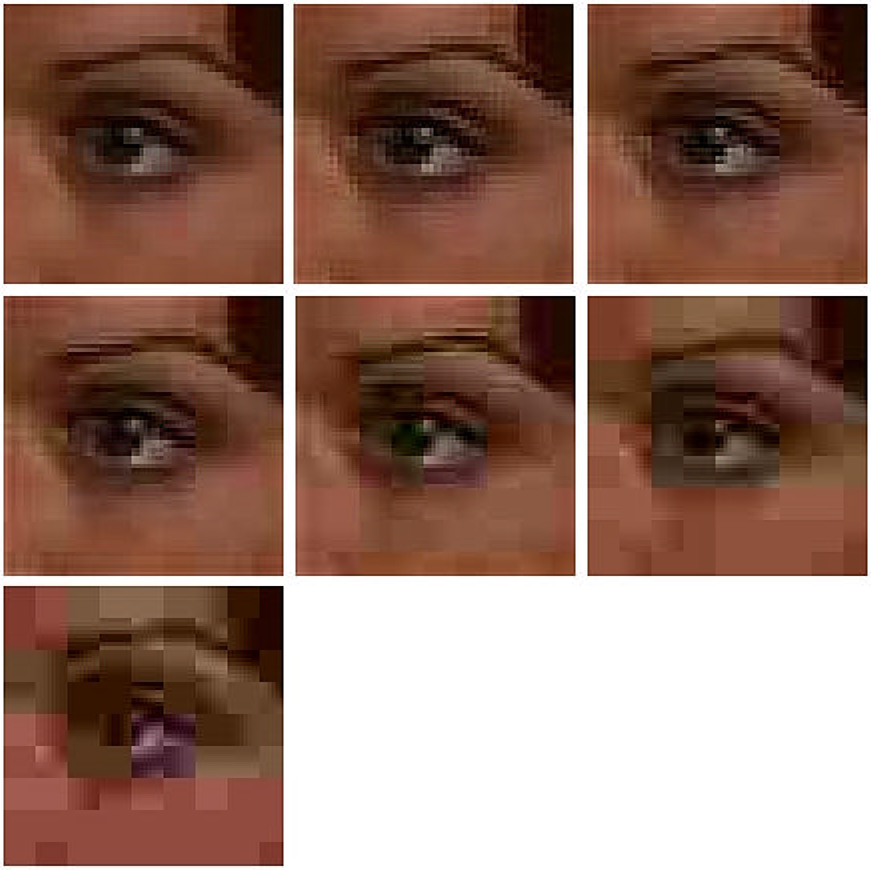
Final Note: At very small resolutions such as 320 x 240 the Fast Motion codec does a very bad job. It is probably better to use the Low Motion codec all the time in such cases. But, as always, do some testing and see what you think.

QUALITY AND RESOLUTION

All Mpeg formats, including DivX Mpeg-4, break up the picture into 16 x 16 blocks called macroblocks. Each block is allocated a certain amount of memory based on the bitrate we enter into the codec. The more the bitrate the better the quality. Lets see what happens to the macroblocks when we allocate less and less bitrate to them. The best way to see this is to enlarge the video so we can see better, like this:



Take a look at the images below. The first one was given the highest bitrate and the lowest was given the least bitrate. As you can see, the less bitrate we give it the more simplified the macrblocks get!



Ideally we would love to encode all movies at 720 x 576 full DVD resolutions. But at this resolution the image is broken up into 1,620 macroblocks. It has an effective resolution of 45 x 36 blocks.

Lets say we have decided that to fit a movie on a single CD as DivX we needed to use a bitrate of 800 kbps. But when we look at the picture at 640 x 480 we can still see many of these macroblock artifacts. We cannot increase the bitrate or it will not fit. Many think if we make the image big (i.e. full sized) the blocks will be smaller. This is wrong! Larger images means more blocks which means less memory is given to each block. When each block has less memory allocated to it, it becomes simplier and simplier until in the end most of the blocks become nothing more than blank colours!

The only solution is to make the image smaller. Smaller images means less blocks which means more memory is allocated to each block and each block looks more like the original picture. A resolution of 480 x 384 would give us 30 x 24 macroblocks, which is 900 less macroblocks. This can make a big difference in quality!

The above methodology is nothing new to DivX, the same has always applied to all Mpeg formats since they were designed. Because of the way it is compressed the picture resolution is not the key factor in determining the image quality. Smaller pictures actually look better than larger ones at the same bitrate. The technique for making the best quality image is simple mathematics:

1. Use the highest bitrate you can.

2. If the image shows too many macroblocks shrink the resolution a little.

3. Check it again. If there are still too many macroblocks shrink it a little more until they are not noticable. Its unlikely you will get rid of all macroblocks on a single CD DivX but you can get rid of most of them.

Commercial Video CD's (VCD) are considered by experts to be almost VHS quality video. Yet they only use a resolution of 352 x 240! This *is* possible even though a TV resolution is something like 576 lines. It is hard to compare an analog image to a digital image by mere resolution. For one thing VHS uses signal compression, yes, that's right, VHS is also a compressed format! But the response curve of VHS places -3 dB at around 2 MHz of analog luminance bandwidth is equivalent to 200 samples / line. VHS chroma is considerably less dense in the horizontal direction than MPEG source video. And from a sampling density perspective, VHS is superior only in the vertical direction, but when taking into account interfield magnetic tape crosstalk and the TV monitor Kell factor, not by all that much. Well, that what I read anyway! But the conclusion of the matter is using small resolutions than TV lines can still produce video's almost as good as VHS.

I have only a couple of rules I always follow for deciding video resolutions. I never make it larger than 640 x 480 (which is higher than TV resoloutions) and I never go lower than 240 pixels high unless absolutely nessasery.

CROPPING THE VIDEO

Considering the previous facts about macroblocks, it makes sense that cropping out the black bars found at the top, bottom and sometimes at the sides of a movie would allow the codec to allocate more memory to image quality. But the amount of memory allocated to a pitch black area is quite negligable. So the most vital thing to remember for optimal compression is to always crop a few pixels into the image so as to delete all of the black bars. Mpeg compression works best on blurry images; so if a hard black line is seen at the edge of your cropped movie it will not compress as effectively. In fact, if you cannot crop into the image then I'd say don't bother cropping at all, because the memory saved is so small.

Warning on Cropping

It is only fair to warn you that, athough in my opinion cropping improves the image quality of the DivX rip by a lot, before you decide that cropping is the best way to go, you must consider these four facts:

1. A cropped movie is sometimes harder to convert into another format. This is because you may need to re-add the black bars to the top and bottom of the movie first or the movie may be stretched out of shape.

2. A cropped DivX movie will play at the wrong aspect ratio in PowerDVD and some other Video players. On the other hand, Media Player, MicroDVD and many other players will play back a cropped movie perfectly.

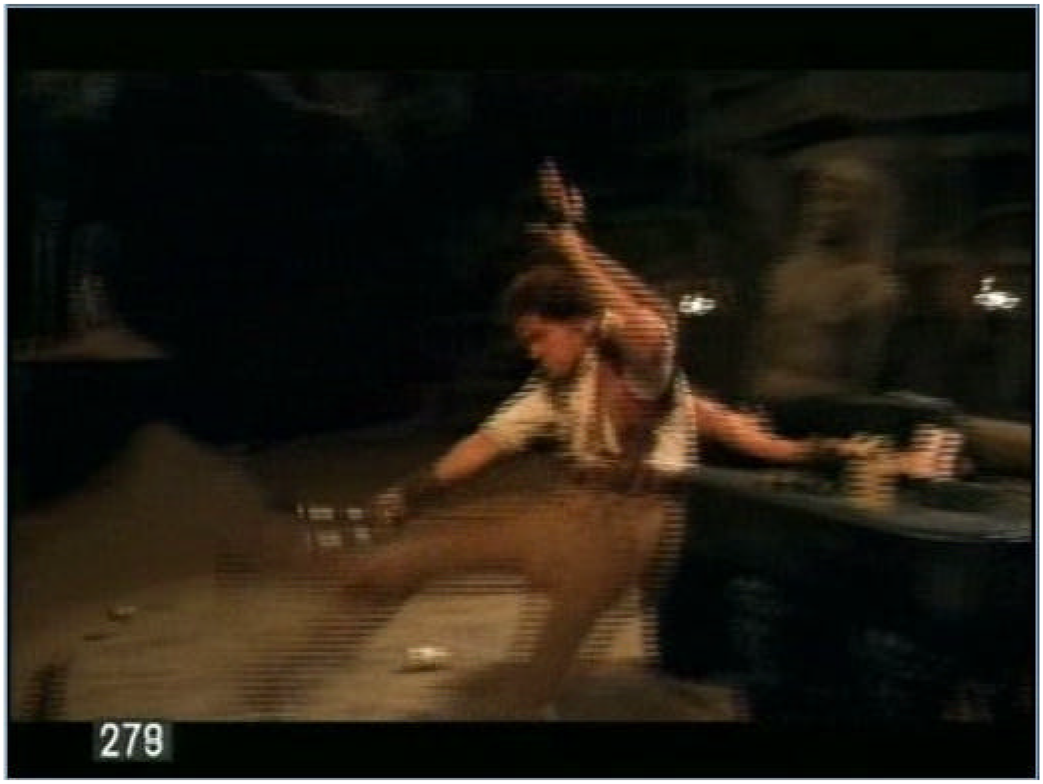
3. VCD's and SVCD's cannot be cropped if they are to be played in a standalone DVD player because it will not accept them.

4. Finally, Mpeg-4 files (and DVD files for that matter) have problems playing back on some hardware if they are not encoded in sizes that can be divided by 32. This means the Matrox G400 or the Nvidia GeForce would probably have problems outputing it to TV. This TV out problem is associated with the Mpeg-4 codec and does not apply to most other codecs.

As an example:
A 528 pixels wide size divided by 32 = 16.5. This is not a multiple of 32 and so may have trouble.
But a 576 pixel wide size divided by 32 = 18. This is a multiple of 32 and will play back perfectly.

DON'T BLAME IT ON THE BITRATE!

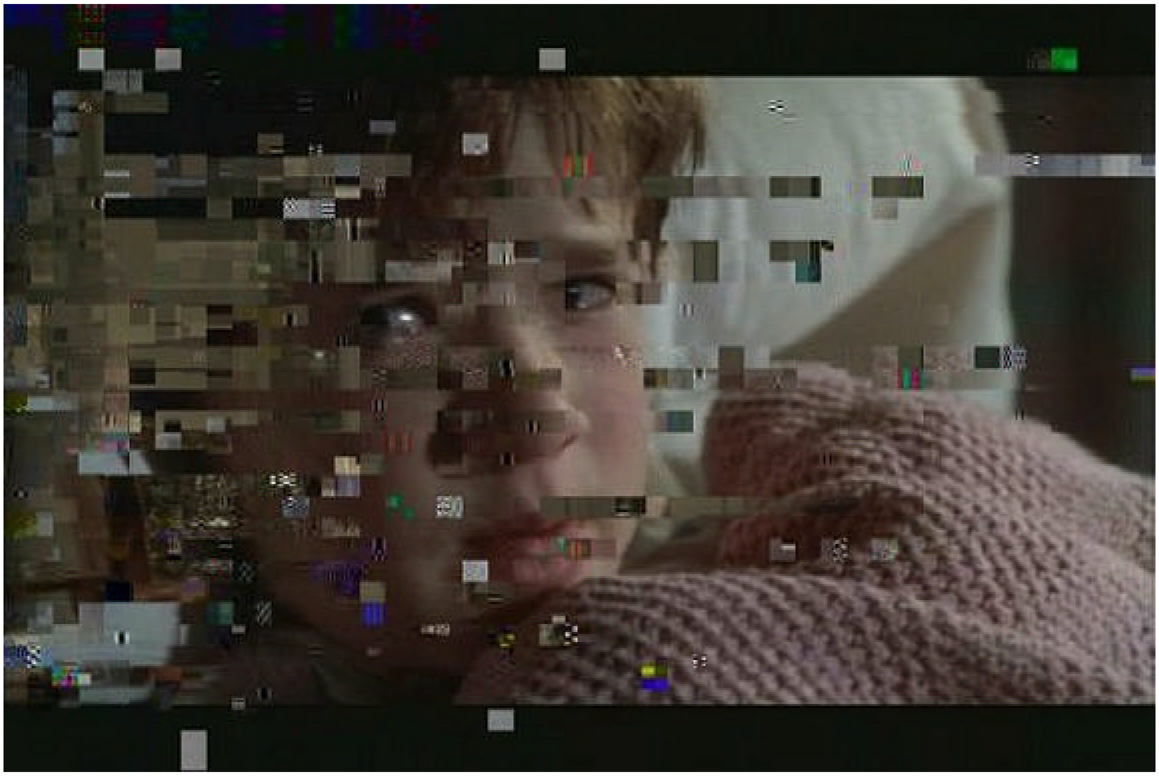
If you find you start to encode a movie and it looks like the picture below it is nothing to do with the compresion. It wouldn't matter if you used 6000kbps or 10kbps it wouldn't get rid of these lines. The problem is that Flask, or whatever decoder you are using is unable to decode your DVD correctly!



Since TV screens are built out of lines, the DVD has a code inside it that tells it how to output those lines to the screen and in what order. If the order of the "fields" is incorrect you will get the annoying combing effect we see above. So far there is not complete solution to this. You can try resizing the picture smaller or you can use Flask Mpeg's deinterlace filter. There is also a super slow annoying way to fix this using DVD2AVI. For more information this interlace subject check out the information in my appendix called: "Video Formats: NTSC & PAL / Telecine".

DeCSS ARTIFACTS

As you know, DVD's have a content scrambling system (CSS) which makes it difficult to copy. These require code keys to decrypt. If you use the wrong key you will end up with a corrupted file. Most ripping softare automatically finds and uses these keys now, but it is always a good idea to check your decoded DVD by playing it on your computer DVD player, just to make sure it is correct before you convert it to something else. You will know if it's corrupt because you will get loads garbage or green / pink blocks like in the picture below:



As you can see, it is not only 'dead people' this kid sees =o).

Video Comparisons (VCD / SVCD / DivX)

Its nice to know what kind of compression and qualiy we will get from the various video codecs. Thats one of the most common questions I get asked in fact. The three most popular compression formats are VCD, SVCD and Divx There are, of course, other formats that preserve the image quality better, such as uncompressed AVI or MJPEG. But these do not compress enough to store a movie on two 650MB CD's! And this is what we are interested in.

There are many issues to consider when choosing a storage format. For example, how much detail is lost during compression. How does it handle action scenes and low motion scenes. How much space does the final file take. How smooth is the playback and how much CPU power does it take to play smoothly. We must also consider how we want them played. If we wish to use only our PC's then any format may be fine. But if we wish to play our movie on a standalone video player, then we must restrict ourselves to a format that is compatible with that player.

Trying to figure out a fair way to compare them hasn't been easy. First I had to decide on the method I would use to encode them. After a while of testing, the programs that did the highest quality were: BBMpeg, TMPGEnc, Ligos LSX Encoder, Panasonic Mpeg Encoder, Xing Mpeg Encoder and Cinema Craft encoder. For Divx any encoder seemed to look basically the same quality, so in the end I choose VirtualDub because it made life easier. Anyhoo, after long deliberation I decided to go with TMPGEnc. The quality looks about the best and what is more it is a free program so anyone can use it. The only downside is it is painfully slow compared to, for example, Xing Mpeg Encoder which zipped through a VCD in seconds =).

HOW DID I COMPARE?

I took a DVD and extracted the Vob files to Hard Disk. It was an anamorphic DVD, which means the picture is slightly squashed, just in case you are wondering why. I wanted the best source video to use as a starting point. This means I didn't want to resize the video or add de-interlace filters etc. Luckily, since my PAL DVD was 720 x 576, I merely cropped the edges until they were 480 across. This gave me my final size of 480 x 576 which is supported by all the formats I am testing. I then used FlasK Mpeg to convert it to uncompressed AVI ready for encoding in the software encoder of my choice; I did not add sound at all!

For compression I used the highest quality settings I could find in TMPGEnc and turned them all up to full. As a result the VCD and SVCD's took an inordinately long time to compress. Please do not disregard these formats merely because it looks like they spent too long to compress. With lower settings they look almost as good and compress much much faster. For Divx you do not have all those fancy pre-filtering settings, so to keep it fair, I didn't mess with TMPGEnc extra filters either.

THE CLIP

I used a scene of 60 seconds (approx.) in length. I made sure I choose one that had both scenes of very low motion and also very fast motion. This was so I could see how the formats coped better.

THE SIZES

All three format specifications allow the encoding of the same size, namely, 480 x 576. But a VCD really needs to be 352 x 288 for best quality. So in the end I opted to do an xVCD (eXtended VCD) resolution and a normal VCD resolution so you could see both; all other settings were exactly the same. Just be warned that the xvcd will not play on a standalone VCD player.

THE BITRATE

The most vital vital setting for movies are the amount of bitrate we use to encode. Although SVCD's allow us to go as high as 2600 kbps, this is not a reasonable amount to use to fit it on 2 CD's. The VCD Mpeg-1 format can fit 74 minutes of video on a single CD. This requires a constant bitrate of exactly 1150 kbps. It doesn't matter what compression format we use, to fit it onto 2 CD's we must keep very close to this amount. So this is the touchstone I have decided to try and follow.

But the situation gets a little more complicated. The VCD format is a constant bitrate (CBR), which means it will always end up about the same size in the end. The Divx and SVCD specifications work best by incorporating a variable bitrate (VBR) which change the final filesize depending on the amount of action contained in the movie. How could I honestly compare a CBR movie to a VBR movie when the CBR movie, at 1150 kbps, turns out to be 10MB but the SVCD, because of its VBR, turns out to be 15MB?! Thats half the size again! If I encoded a movie of 650 MB and then another at 975 MB and said 'see, the SVCD at 975 MB looks much better than the 650 MB because I used such and such compression' you'd think I was crazy!

Anyway, to solve this problem I had to 'cheat' a little. I started off by setting both the VCD, Divx and SVCD with a bitrate of 1150 kbps and encoding them. As it turned out the Divx and SVCD's ended up slightly smaller than the CBR VCD, so I fiddled with the bitrates of the Divx and SVCD until it was almost the same filesize as the VCD. It was impossible to get them to match exactly but I managed to get it down to well within a megabyte. I also allowed the SVCD to have the full 2600 kbps as its maximum bitrate and gave it 300 kbps as a minimum bitrate to allow it enough room for variability.

To summarize, I used the same movie clip, the same resolution and highest quality settings I could for all formats. I encoded them so that the final movies were all almost exactly the same filesize! Because of this I did not use exactly the same bitrates.

FORMAT	RES	SETTINGS	SIZE	ENCODING

				TIME
VCD	352 x 288	1150 kbps (CBR) Floating point DCT	9.44MB	35 mins
xvcd	480 x 576	1150 kbps (CBR) Floating point DCT	9.44 MB	35 mins
SVCD	480 x 576	2 Pass (VBR) Average**:** 1200 kbps Max: 2600 kbps Min: 300 kbps Floating point DCT DCT Precision: 10 Bits	10.2 MB	1 hour 3 mins
Divx	480 x 576	Low Motion Codec Bitrate: 1310 kbps Keyframe every 10 secs Smoothness 75%	9.75 MB	9 mins

CONCLUSIONS

As was expected the latest codec's seemed to perform the best and the earlier the worst. All final movies played back relatively smoothly. The VCD and xVCD's still looked very very slightly smoother but this is almost always CPU related and doesn't reflect badly on the other formats abilities at all. The VCD would obviously take the least amount of CPU power, then the xvcd perhaps slightly more, then the SVCD and finally the Divx For low motion scenes you cannot easily beat SVCD's! The colour matching and halftones are almost identical with the original. For action scenes Divx wipes the floor with absolutely everything! There are hardly any noticeable macroblocks at all!

Perhaps its something I did wrong with encoding, but I am a little disappointed with how the SVCD handles fast motion scenes. I set the maximum bitrate to 2600 kbps hoping that this would allow it get rid of the blocks but it did not help much. The xVCD's had the most problems with macroblocks and the Divx had the least.

WHEN PLAYED FULL SCREEN

My quality test is to sit about 5 feet away from my computer screen and play each movie full-screen to see which one I thought gave the overall best look. No one watches a movie from a foot away from the TV screen, so I think such a test is just as useful at determining overall quality as examining the picture up-close.

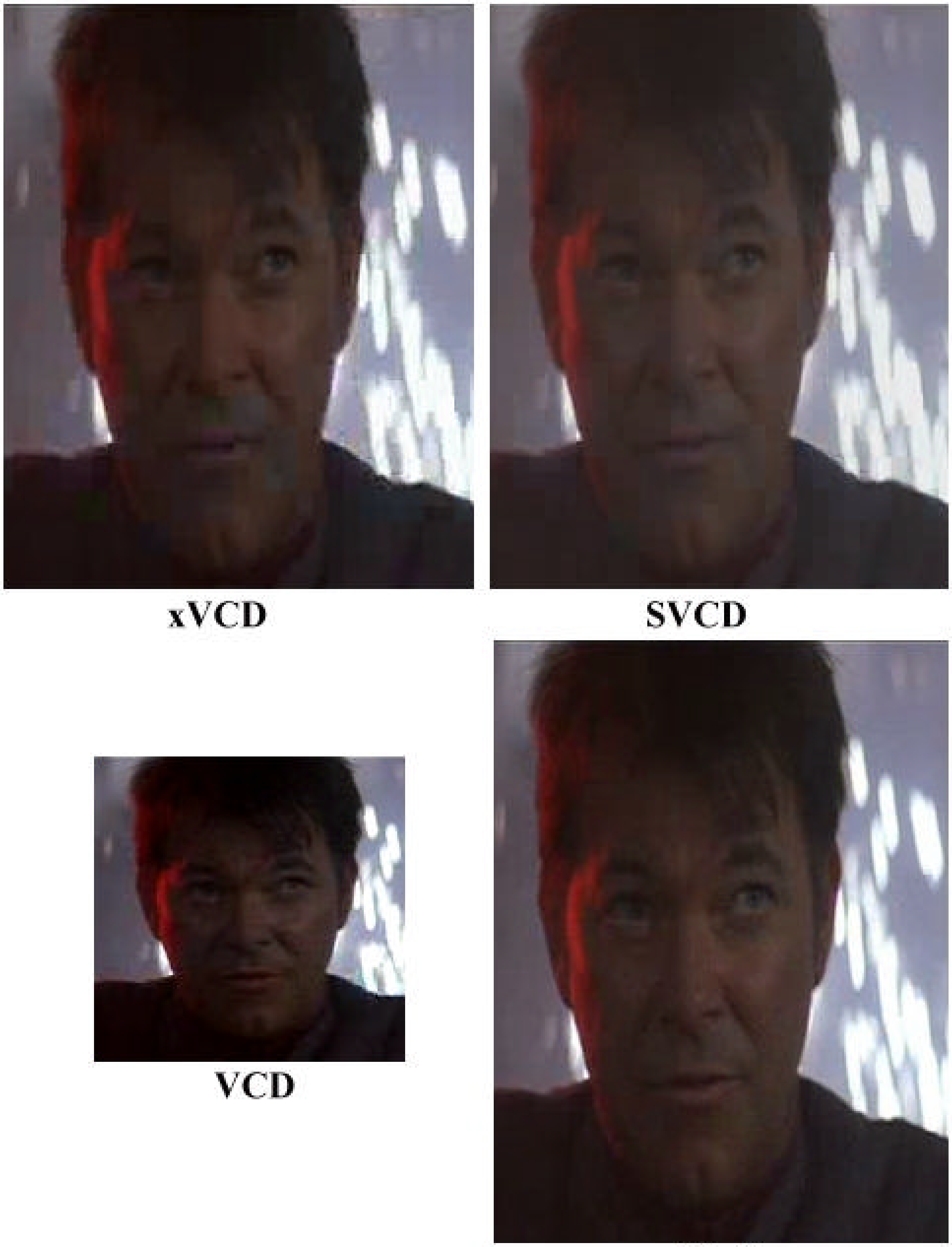
I'd have to say the Divx looked overall best full screen. The image was sharp, smooth and didn't break up into blocks too much. The main annoying thing about the Divx was the dark colours used! The SVCD looked bright and clear and would have been the best quality if it had not been for its bad macroblocks. As for the VCD, yes it also looked dark and slightly blurred, but you couldn't really notice any macroblocks like you could on the SVCD! So in some ways a VCD may be the better choice over SVCD for action movies and the SVCD for drama movies. The xvcd looked the worst of all! It was dark, blurred and blocky!

THE SNAPS

No comparison article would be complete without detailed examples, so here they are =). All images were captured with either VirtualDub or PowerDVD and have been saved as Jpeg's at the lowest possible compression. You may need to turn your monitors brightness up a little to see the full details of the samples.

HIGH ACTION

VERY HIGH ACTION



xVCD

SVCD

VCD

DivX

VERY LOW MOTION

xVCD

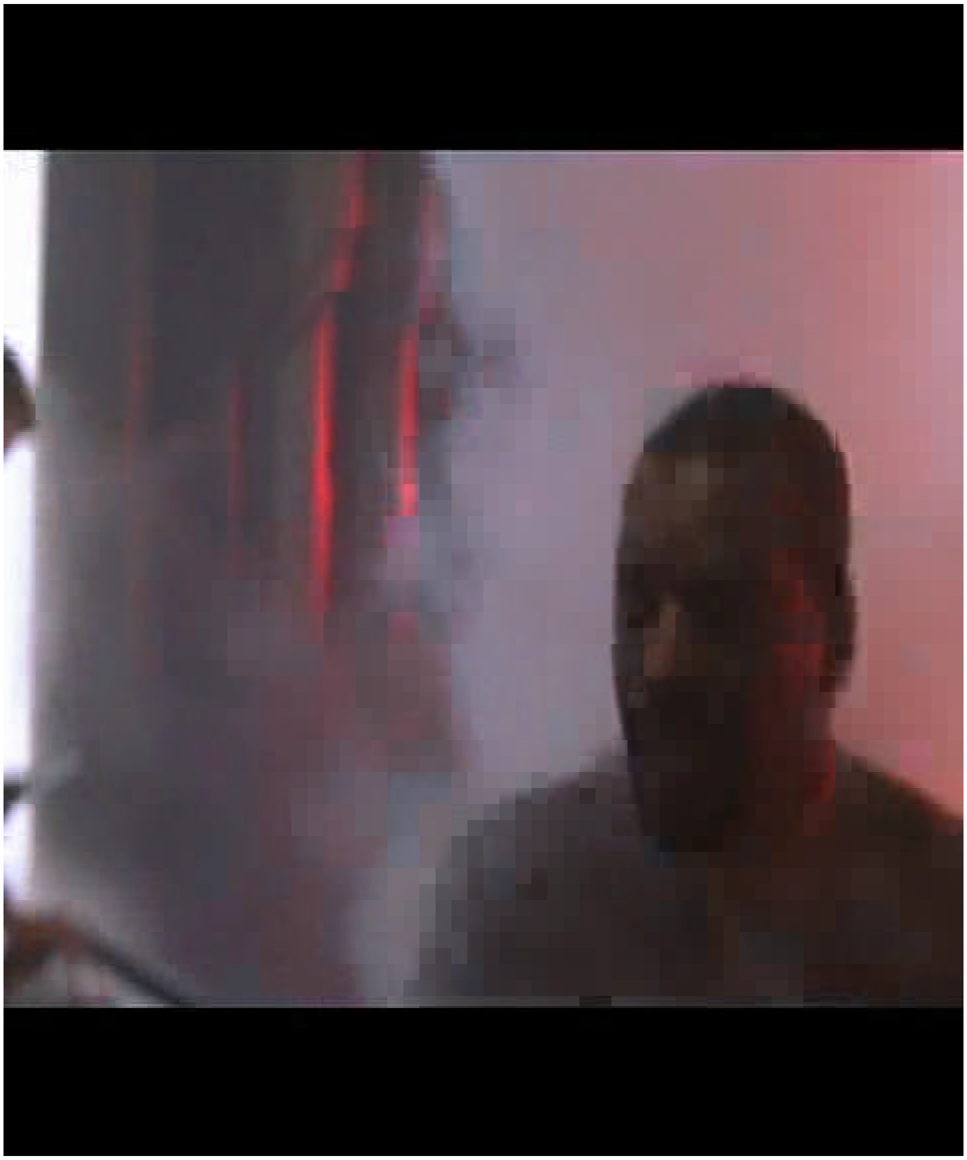
SVCD

VCD

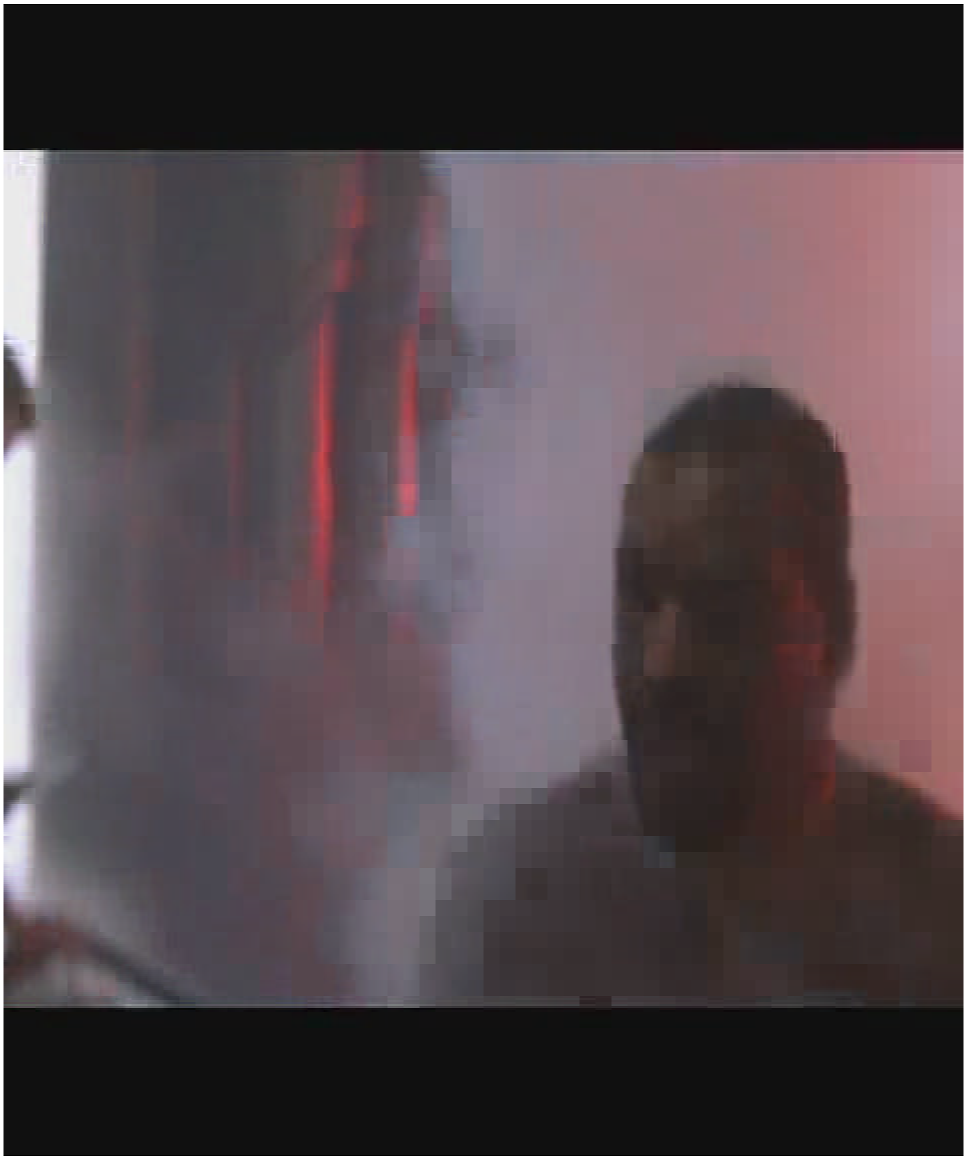
DivX

THE FULL SIZE EXAMPLES

HIGH ACTION SCENE



xvcd

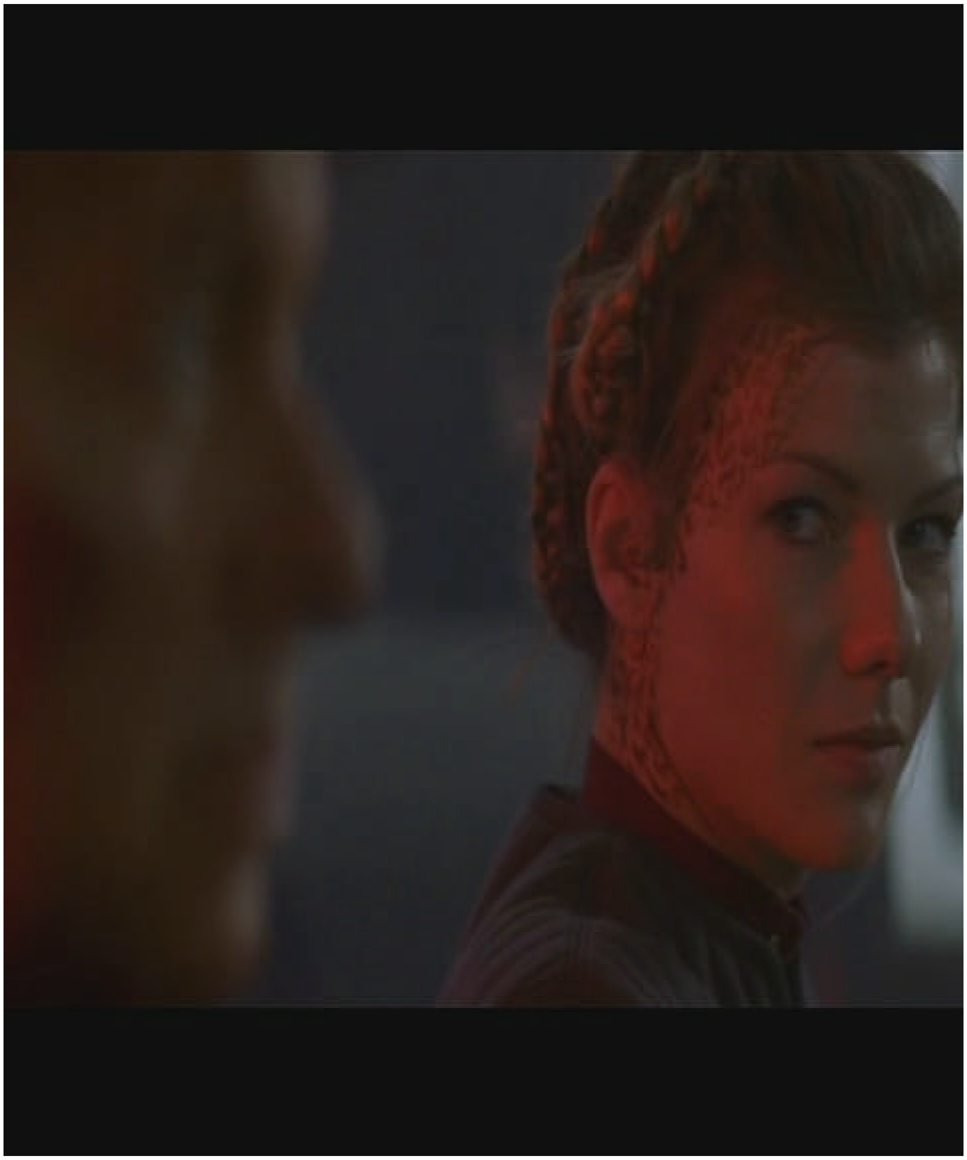


SVCD

DIVX

VERY LOW MOTION SCENE

xvcd



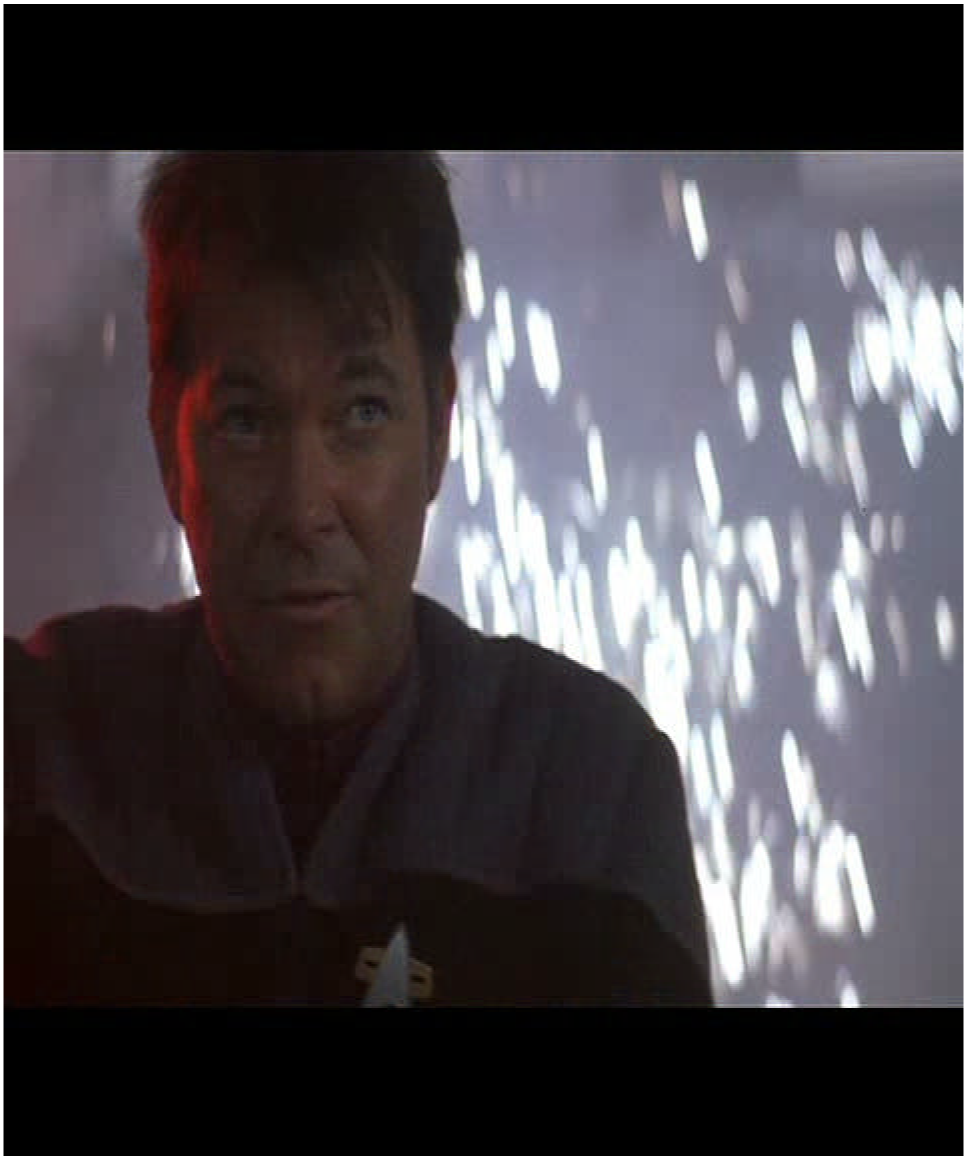
SVCD

DIVX

VERY HIGH MOTION SCENE

xvcd

SVCD



DIVX

MEDIUM / LOW ACTION SCENE

xvcd

SVCD

DIVX

Windows Media Encoder 8 Tests!!

Well I guess everyone is wanting to know if Microsoft are telling porky pies about the Media Encoder 8! If you haven't heard about the new codec check it out here. Micro$oft claim:

1. Near-VHS Quality at 250kbps Content encoded in version 8 can now deliver near-VHS quality (320×240 @24fps) at the low end of DSL/Cable connection speeds.

2. Near-DVD Quality Video at 500kbps Content encoded in version 8 can now deliver near-DVD quality (640×480 @24fps) at typical DSL/Cable connection speeds.

3. Best Video for Downloadable Movies For download-and-play movies, Windows Media 8 offers two-pass and “true” variable bit-rate encoding to guarantee video quality over an entire feature length film.

According to my tests its a bunch of crap!! How can they claim near-DVD quality at 640 x 480 and at only 500kbps! AND near CD quality audio at 64kbps!! That means they claim to be able to fit a 2 hour 30 mins of near-DVD quality video with near-CD quality audio on a single 650MB CD-R!

To be fair I think the 64kbps audio is near CD quality and I also think that its almost beyond doubt that Mpeg-4 is the best format for Internet streaming there is! I don't want to hear any Quicktime or Real Media crap from anyone about that - Microsoft won the battle as usual, live with it! But I think for Microsoft to claim such wild things about their new codec is just plain crazy! Lets take a look at my initial tests on it anyway and you can decide:

***Note:** These following tests are based on the Beta version of Media Encoder 8 so it is of course possible that the full version will be much better quality etc.*

THE TEST SETUP

Yes, yet again Digital Digest come up with the test examples before anyone else, so don't forget it baby! But I must be fair and give full credit to my pal PurpleMan who seems to have been the first to create a GUI for the newest Media Encoder. He was also the first to explain how to encode a DVD with it by frameserving. There are a couple more GUI's out now but I haven't had time to test them and no new GUI will change the following test results anyway.

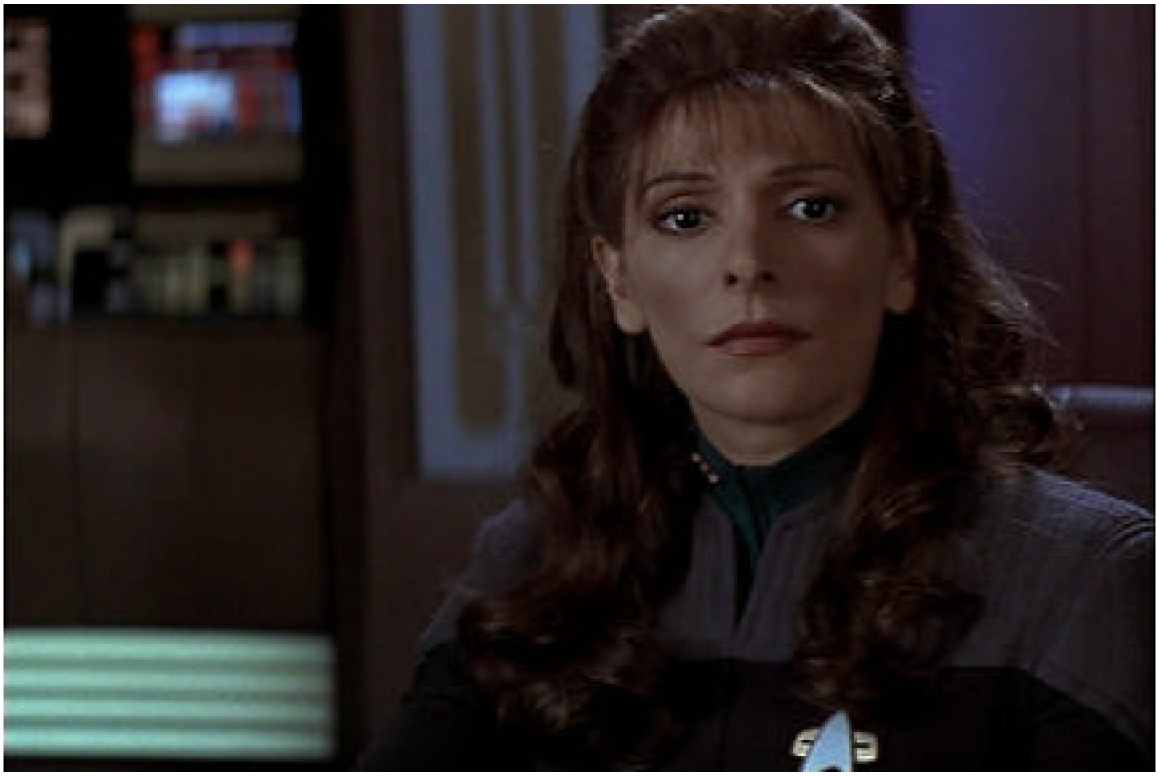
First I downloaded Purplemans GUI and Windows Media Encoder Beta 8 from Microsoft Then I converted two DVD clips one at 432 x 288 and the other at 640 x 480. Both were converted to uncompressed AVI's with uncompressed PCM audio using Flask Mpeg. This means there is no loss of quality and the initial video is almost perfect and of course much better than you could get using any video capture card! Each clip was 60 seconds long and had both low motion scenes (when people are not moving) and high action scenes (where things explode etc). I think testing it this way is the most fair because I'm using a scene that has both action and calm scenes to test how it handles all aspects of video. As I'm sure you know the old Mpeg-1 VCD's always broke up into blocks on action scenes so we must give it both.

LOW MOTION SCENES

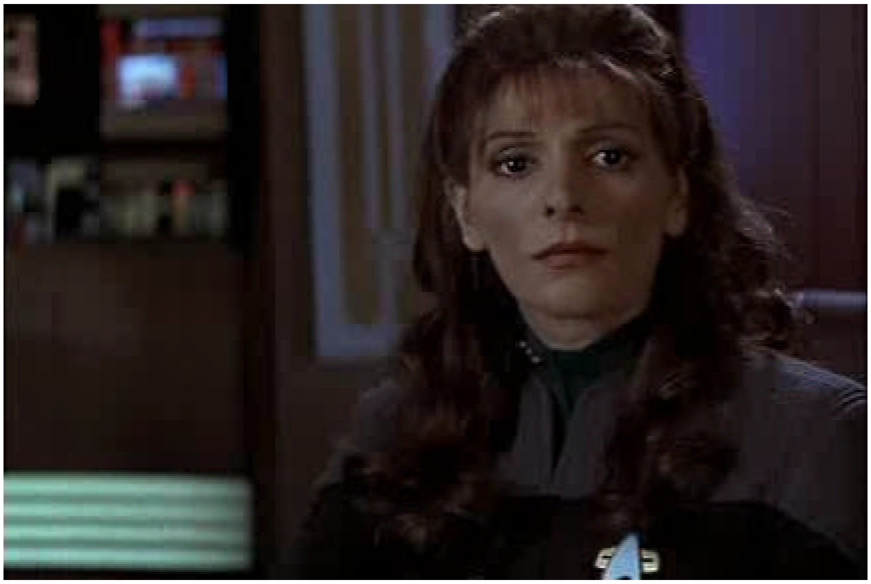
TV / VHS Resolutions (432 x 288)

Obviously I'm comparing the new Mpeg-4 codec's against the old DivX Mpeg-4 codec's (which beat both SVCD and VCD for quality see here for examples of that). All of the following tests are done using the highest quality Variable Bitrate settings of the WM8 codec.

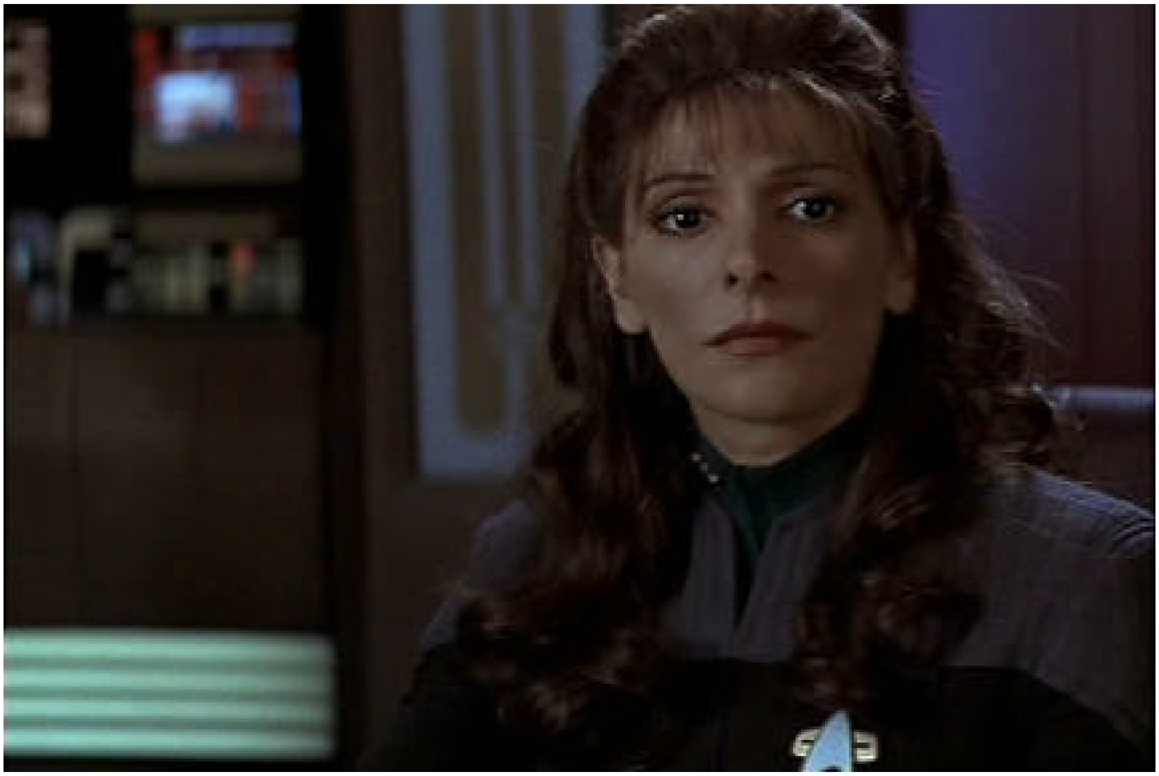
Here is the original uncompressed AVI of a low motion scene:



Here is the same one encoded with the Windows Media 8 codec using its 2 Pass Variable Bitrate (VBR) at 800kbps:



That may look quite cool but first look at how the Divx Low Motion codec does this same scene at only 500kbps!:



How does the Windows Media 8 (WM8) codec deal with the same scene at 2000kbps? Take a look:

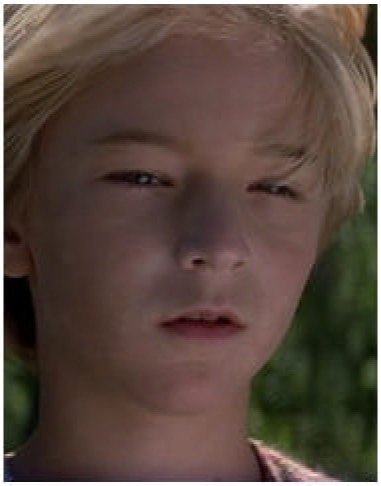
Here are the filesizes for the VHS video test:

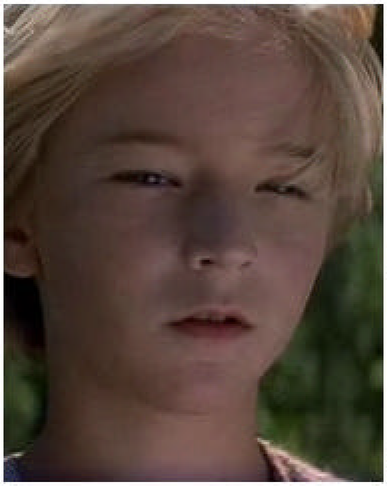
Bitrate / Codec	**Video Size**	**Length**
WM8 Codec (800kbps)	6.13 MB	60 seconds approx.
WM8 Codec (1150kbps)	9.26 MB	60 seconds approx.
WM8 Codec (2000kbps)	13.4 MB	60 seconds approx.
Divx Low Motion (800kbps)	6.27 MB	60 seconds approx.
Divx Low Motion (1150kbps)	8.16 MB	60 seconds approx.
Divx Fast Motion (800kbps)	4.74 MB	60 seconds approx.
Divx Low Motion (2000kbps)	9.11 MB	60 seconds approx.

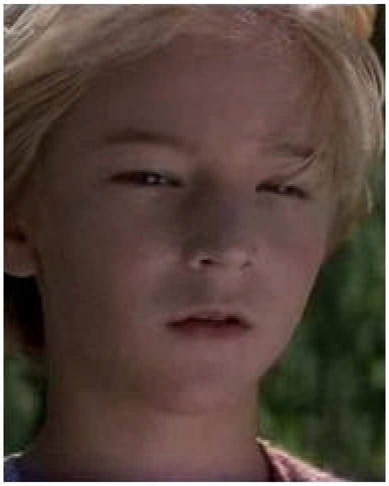
Note: this clip has 64kbps audio too

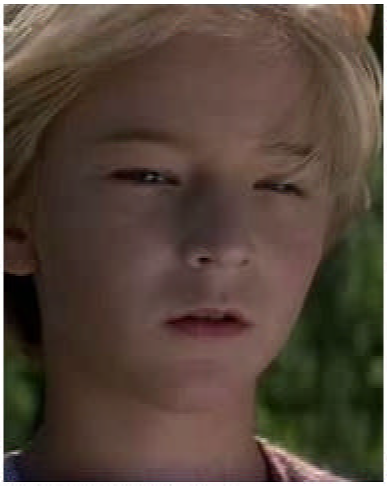
Near DVD Resolutions (640 x 480)

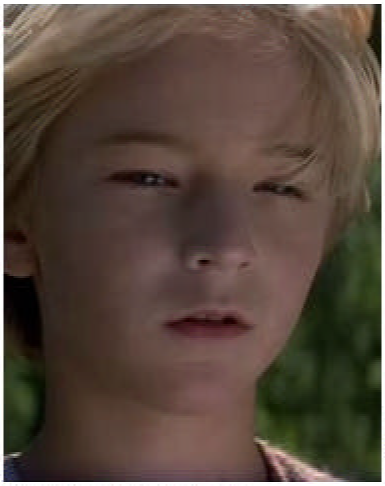
At higher resolutions the newer WM8 codec seems to perform a little better on Low Motion scenes. I think this is due to the fact that at higher resolutions disruptions of smooth tone are less obvious - either way it does a fairly good job of it.

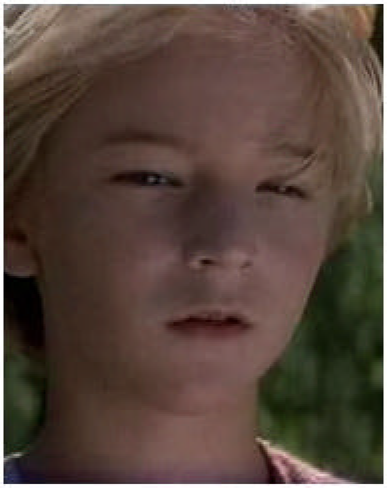

DVD Image


WM8 at 2000kbps


WM8 at 1150kbps


DivX Low Motion at 600kbps


DivX Low Motion at 1150kbps


WM8 at 800kbps

In my opinion the Divx Low motion codec still does a better job even at lower bitrates but it looks as if you can get away with it if you use a bitrate of about 1150kbps on the WM8 codec.

Here are the filesizes for the DVD video tests:

Bitrate / Codec	Video Size	Length
WM8 Codec (500kbps)	3.57 MB	60 seconds exactly

WM8 Codec (800kbps)	5.78 MB	60 seconds exactly
WM8 Codec (1150kbps)	8.26 MB	60 seconds exactly
WM8 Codec (2000kbps)	12.8 MB	60 seconds exactly
Divx Low Motion (600kbps)	4.62 MB	60 seconds exactly
Divx Low Motion (800kbps)	6.14 MB	60 seconds exactly
Divx Low Motion (1150kbps)	8.79 MB	60 seconds exactly
Divx Low Motion (2000kbps)	13.1 MB	60 seconds exactly
Divx Fast Motion (2000kbps)	5.36 MB	60 seconds exactly

Note: this clip was done without audio

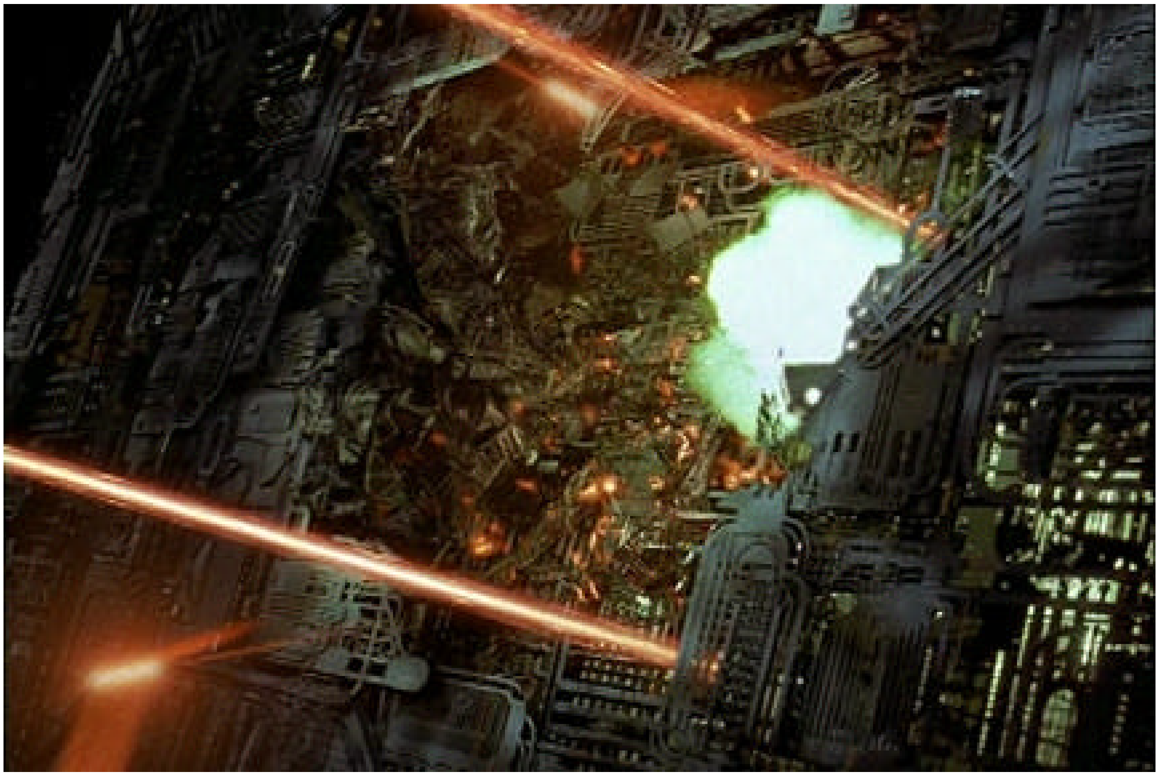
Conclusion

For low motion scenes the WM8 codec doesn't do quite as well as the old Divx Low Motion codec. This is especially true at VHS resolutions. At DVD resolutions this effect is less noticeable but the bitrate needs to be set to about 1000kbps or higher to get a smooth image! Generally speaking for non-action scenes the WM8 codec gives an overall sharper image with very slightly better colour matching, but there are still too many macroblock artifacts where there should be *smooth* tones!

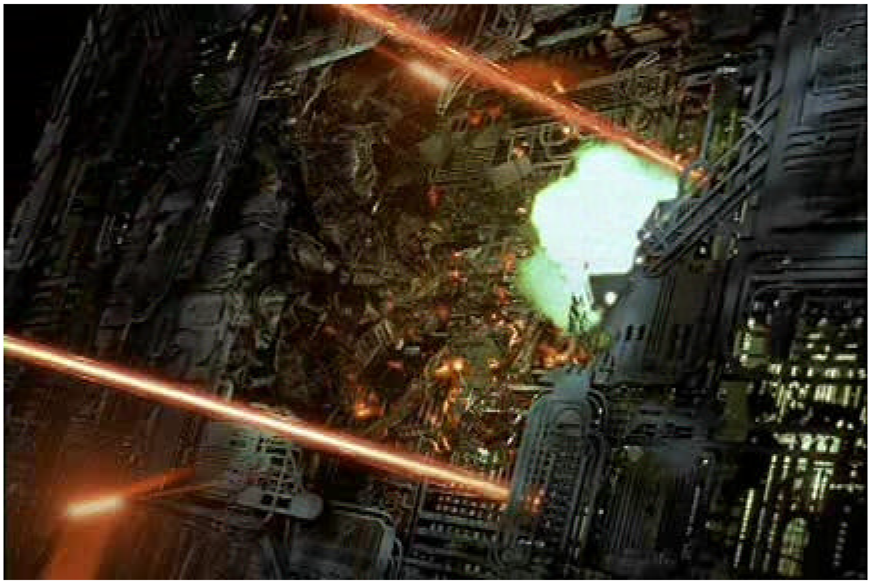
FAST MOTION SCENES

TV / VHS Resolutions (432 x 288)

The clips I'm using are the same, just a different parts of the same movie clip. Here is how the original high action clip looked before compressing.



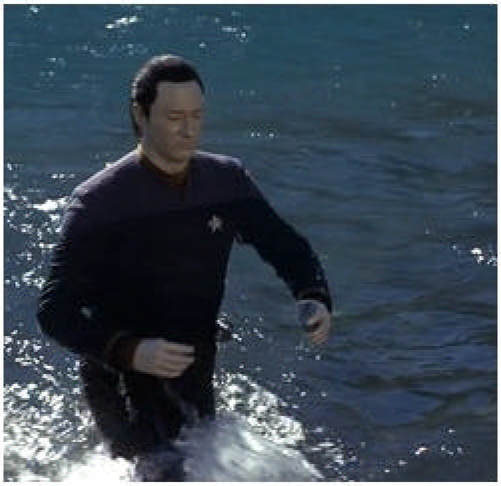
And this is the same scene rendered with the WM8 codec at 800kbps:



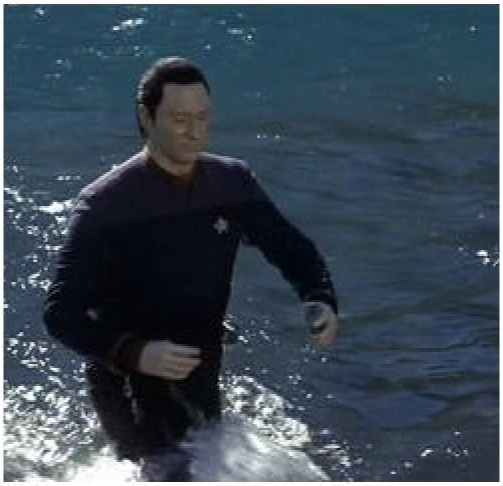
The final results of all my other pictures all looked so similar that I could see no point uploading them. In short, with any bitrate below about 800kbps I could see blocks on the WM8 codec, but anything above 800kbps they were not really noticeable But don't be disappointed the next near-DVD sized examples are much more enlightening.

Near DVD Resolutions (640 x 480)

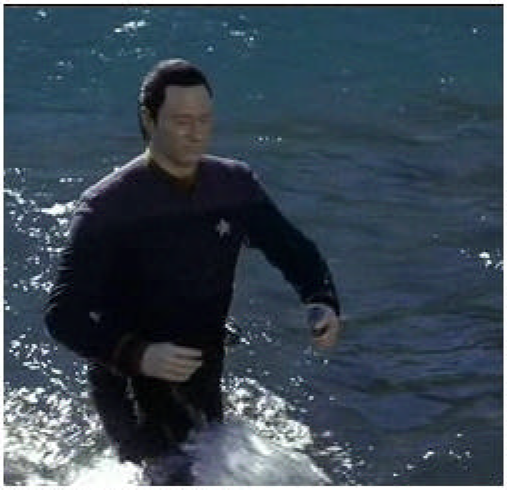
This is the interesting part :). The new WM8 codec has improved significantly over the old Windows Media 7 codec when it comes to Fast Motion scenes. The scene below is a *very* fast motion scene even if it doesn't look like one. This is because the whole screen is moving with the water and the waves are very sharp requiring lots of shape details. First off we can see that at 2000kbps the WM8 codec is getting close to DVD quality. In fact there is little change as low as 1150kbps which is the average 2 CD VCD bitrate.



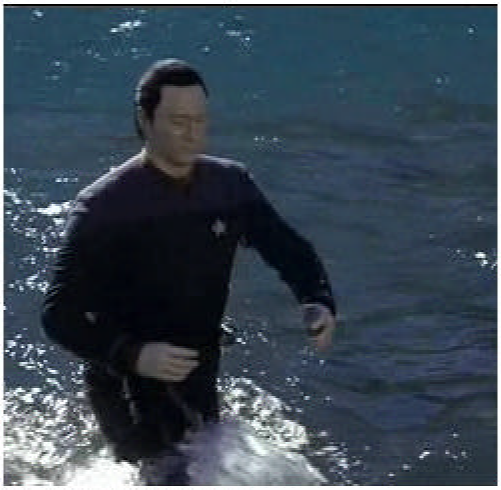
DVD Image



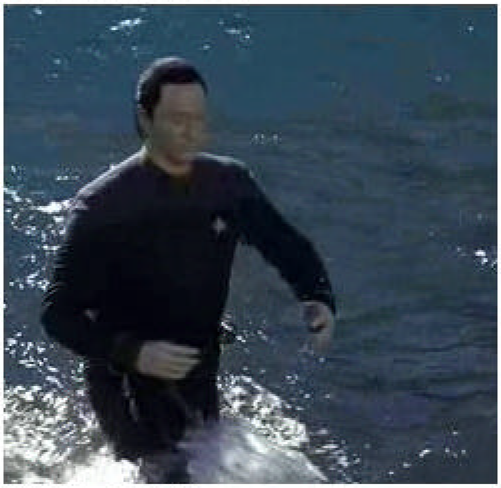
WM8 at 2000kbps



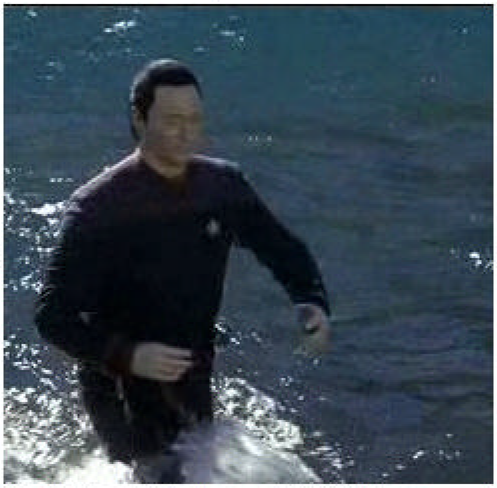
WM8 at 1150kbps



WM8 at 800kbps



WM8 at 600kbps



DivX Fast Motion at 2000kbps

Even when the WM8 codec is set to 600kbps it can get a quality very close to the old Divx Fast Motion codec when it was set to 2000kbps! The same applies to the previous

WM7 which is almost exactly the same as the Divx Fast Motion codec. Please don't let this confuse you, though. The final file size of the WM8 movie at 600kbps is still larger than the Divx Fast Motion at 2000kbps.

Conclusion

For Fast Action scenes the newest WM8 codec beats all others designed so far. But you need to be careful when using it because it is very unpredictable and will quickly eat up too much space if you let it!

IN A NUT SHELL

Does the new WM8 codec give Near-VHS quality at 250kbps as Microsoft claim?

Noooo! No chance in hell! At bitrates of about 600-700kbps you will get near VHS quality video depending on the amount of action in the movie. To do this you must use the Variable Bitrate (VBR) option. But this means that the final file size is still a bit unpredictable. In other words, even if you set a bitrate of 700kbps it is quite possible that you will end up with a final movie of 750MB or more!

Does the new WM8 codec give Near-DVD quality at 500kbps at resolutions of 640×480 as Microsoft claim?

Nope! Were you really expecting it to? Even at an average VCD bitrate of 1150kbps it barely reaches the standard needed. This is especially true when it comes to Low Motion scenes. Such non-action scene's are where the new WM8 codec fails most of all and where it has always needed some improvement.

Final Notes

The new WM8 codec is certainly much better than the old WM7 codec and also much better than the Divx Fast Motion codec. And so I think its quite a good replacement! I'd recommend using it at a bitrate of at least 600-700kbps for a single CD movie. Going higher probably isn't a good idea because it just get too large. Just imagine that the 600kbps WM8 is the same as the 2000kbps Divx Fast Motion codec and you won't be far wrong. For best quality on a double CD movie use a bitrate of between 900 to 1200kbps. Again higher bitrates will give slightly better quality but anything over 1500kbps will probably increase the filesize far too much. And finally, resizing the movie down a bit smaller than 640 x 480 will still give even better results.

While I wait for the final release

I guess I've developed the reputation as a Microsoft hater but actually that's not quite true. And, no, Bill Gates isn't standing next to me with a large gun to my head, honest :-P. It's become quite fashionable to hate Microsoft, there's even a 'tongue in cheek' song entitled 'Bill gates must die' flying about the Internet. Its probably not even Mr. Gates' fault that Windows is so problematic. I mean, he didn't even write the thing, its the product of teams of programers. I doubt he interviews them all personally and if the final product is substandard who do you fire? The whole of the Microsoft Development team?! If things were different and another company had been signed up to write the IBM Disk Operating System the situation would probably still be exactly the same. Only instead of everyone hating Bill Gates we'd all be hating Fred Bloggs or whoever was in charge!

What I'm getting at is that I actually do like Microsoft in some ways. I mean what other company in the world would give us a free utility like Windows Media Encoder so we could make streaming videos for our personal websites and interactive Video CD's?! My dislike of them comes from the fact that even with their huge wealth and resources they cannot seem to create a stable operating system that doesn't crash all the time or use up all the resources of your machine and leave you with practically nothing! If it wasn't for that I'd be sitting here with my 'Microsoft Rules' T-Shirt on :).

Yes, this is leading up to something that I wanted to rant about. The only reason people ever hacked the Microsoft codecs to make the DivX cones were because they wanted to use them like any other codec in their favorite encoding software. But Microsoft wouldn't let them. It wasn't because they hated Microsoft or wanted to bring them down by stealing their new codec and give it away to everyone like some have done with other Microsoft products. What I don't get is why Microsoft restricted use to begin with? They owned the copyright, they let everyone encode the video clips for free using their own encoder. So why not just release the codec so everyone could have used it in the first place?! I know many restrictions are gone now, but why did they wait until there were three or four hacks out before this happened. I think if they'd done it right from the very start people would be saying something like 'Real Media? who were they?' or 'Quicktime? What's that when its at home?'. There shouldn't be any competition because nothing I have seen so far can compete with Mpeg-4. Anyway, perhaps I'm missing something, I never did understand business tactics very well.

Wish List

You never know, perhaps someone in charge of the development of the WM8 codec will see this and implement some additional things that we have been wanting from the beginning. So this is my wish list of what I'd like to see in the final release of the codec...although I'm sure they already have a good idea about what is needed anyway =o).

Just give out the separate audio and video codecs. Let us download and install them. Its a pain forcing us to use just Media Encoder all the time. So to also make it usable in any encoder such as VirtualDub or Adobe Premiere etc., would be great. Media Encoder is fine for easily making Streaming Media but it would be nice if all the settings were in the codec itself. Or failing that give us the options I describe next with the Windows Media Encoder 8 and frameserving support so we can open Directshow and *.avs files in it!

Let us set the bitrate thresholds. 2 Pass variable bitrate is great! But it would be even better if we were allowed to choose the minimum, maximum and average bitrates in the same way we can with Mpeg-2. Lets face it, the main improvement of the WM8 is really a reordering of the bitrate thresholds to slightly different settings.

The usual settings. Obviously we still want to set the amount of keyframes and the smoothness settings etc. It would also be nice if we could force it *not* to drop *any* frames!

Noise Reduction / Prefilters. It's not number one on my list but it would be nice if we could set the quantizer matrix and prefilters etc., that were built into the new WM8 codec, such as the ability to soften block noise and so on. Some video and TV captured footage is very grainy and gives bad converted results. Other captured footage can be very clear but, for whatever reason, needs to be kept very sharp. A GUI that shows a preview of the video clip would be nice. I'm not asking for new filters just for control over the built in settings. I'm not sure what variables are used for filtering with the WM8 but I'm imagining this would work a little like how the Adobe Photoshop unsharp mask or smart blur functions do.

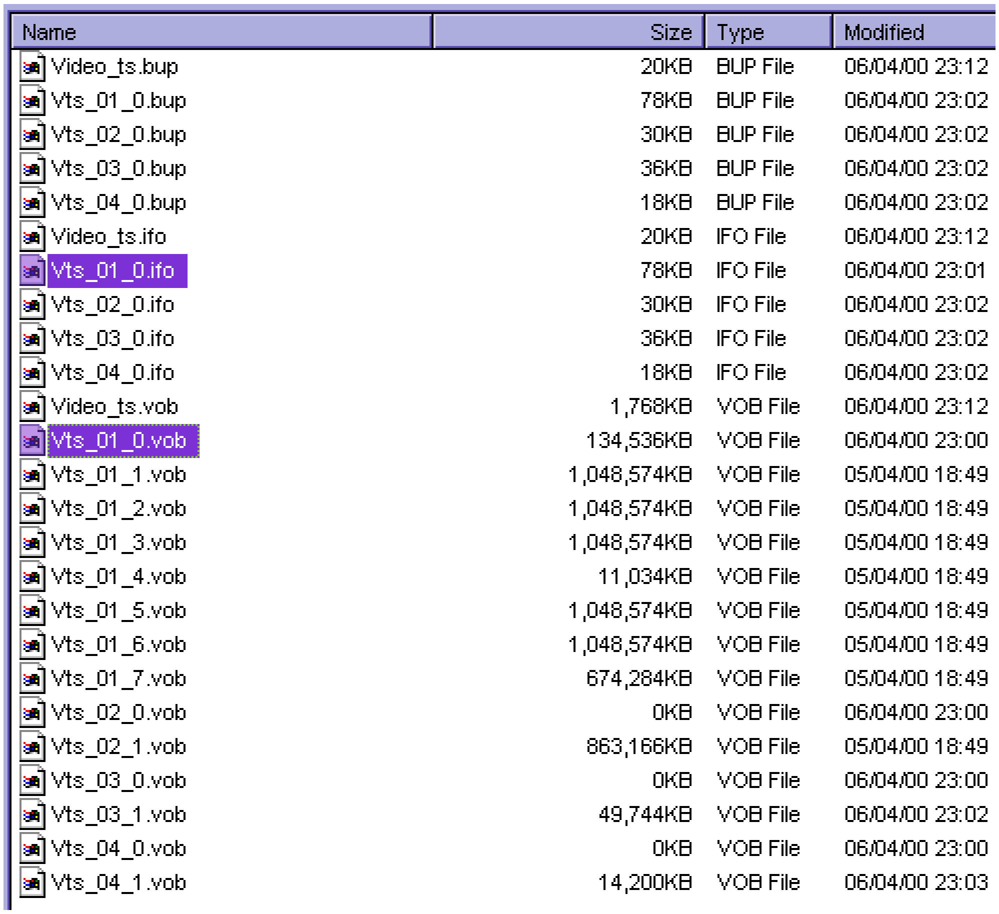
To Stream or Not to Stream. Any additional data or settings needed for streaming should ideally be a separate option. Sometimes we do not want any streaming data because the video clip is meant for just downloading or putting on a CD-Rom. Such additional info bloats the filesize too much.

WMA Audio Timing. None of the old WMA codec could keep very good timing on long runs. I haven't tested the new WM8 audio but this needs to be corrected if it hasn't already. Finally I couldn't see a joint stereo option. I'm sure this would save lots more space on stereo files that need to be very small.

That's all. Any additional support for automatic aspect ratios built into the header or filters to perform IVTC would also be very useful for Media Encoder, but that's nothing to do with the codec itself.

Basic DVD Structure

If you intend on ripping DVD's it helps if we have some idea of what we are actually ripping. When you rip a whole DVD to your hard drive you will see something like the picture below:



Name	Size	Type	Modified
Video_ts.bup	20KB	BUP File	06/04/00 23:12
Vts_01_0.bup	78KB	BUP File	06/04/00 23:02
Vts_02_0.bup	30KB	BUP File	06/04/00 23:02
Vts_03_0.bup	36KB	BUP File	06/04/00 23:02
Vts_04_0.bup	18KB	BUP File	06/04/00 23:02
Video_ts.ifo	20KB	IFO File	06/04/00 23:12
Vts_01_0.ifo	78KB	IFO File	06/04/00 23:01
Vts_02_0.ifo	30KB	IFO File	06/04/00 23:02
Vts_03_0.ifo	36KB	IFO File	06/04/00 23:02
Vts_04_0.ifo	18KB	IFO File	06/04/00 23:02
Video_ts.vob	1,768KB	VOB File	06/04/00 23:12
Vts_01_0.vob	134,536KB	VOB File	06/04/00 23:00
Vts_01_1.vob	1,048,574KB	VOB File	05/04/00 18:49
Vts_01_2.vob	1,048,574KB	VOB File	05/04/00 18:49
Vts_01_3.vob	1,048,574KB	VOB File	05/04/00 18:49
Vts_01_4.vob	11,034KB	VOB File	05/04/00 18:49
Vts_01_5.vob	1,048,574KB	VOB File	05/04/00 18:49
Vts_01_6.vob	1,048,574KB	VOB File	05/04/00 18:49
Vts_01_7.vob	674,284KB	VOB File	05/04/00 18:49
Vts_02_0.vob	0KB	VOB File	06/04/00 23:00
Vts_02_1.vob	863,166KB	VOB File	05/04/00 18:49
Vts_03_0.vob	0KB	VOB File	06/04/00 23:00
Vts_03_1.vob	49,744KB	VOB File	06/04/00 23:02
Vts_04_0.vob	0KB	VOB File	06/04/00 23:00
Vts_04_1.vob	14,200KB	VOB File	06/04/00 23:03

I guess you are wondering what all these files are there for, huh. Well to summarize:

vts_01_1.**vob**	Video Transport Stream (VOB = Video Objects)
vts_01_0.**ifo**	Navigational information (IFO = Information)
vts_01_0.**bup**	Navigational info backup (BUP = BackUP)
video_ts.**ifo**	Secondary navigational information
video_ts.**bup**	Secondary Navigational information backup

For the most part the only files we are concerned about are the **IFO** and **VOB** files. The rest are needed more by a DVD player for error correction and seeking requirements.

When I refer to a "title" in my articles I mean a feature of a DVD. For example, DVD's usually contain the main movie. That is the 1st "title". Then they will have the main trailer, 2nd "title". Then they may also have the short trailer 3rd "title". Then they will have the specials (if any) such as the making of the movie 4th "title". Finally, they will have a small movie clip used for the options and selection menus 5th title. Even still picture menus are made into a slideshow movie so there is rarely any additional files. If you do find additional files they are probably meant to be installed on a PC, like in the Matrix DVD which has an interactive presentation on it.

Vob Files

DVD Vob files contain lots of things bundled together. Imagine each Vob as a box containing various items. The two most important items inside it are the Mpeg-2 video and the Ac3 (or Mp2 / PCM) audio streams. A Vob file can contain many audio tracks at the same time. One can be English, another French, German etc., then you can have alternate music scores and director comments and so on all as separate tracks. A Vob file can also contain subtitles which are usually in the form of transparent pictures that are displayed over the top of the played DVD like a slideshow.

If you play a Vob file back it will usually play everything together i.e. you will hear both the French, German, English and the Directors comments all going at once! This is because they are Multiplexed as one single file. This basically means every file is chopped up into thin slices and stacked together like a pack of cards.

Vob files may also contain repeated sequences. It may have one second of one scene in English, then one second *of the same scene* but in German, and then French and so on. If you were to play it back you wouldn't be able to watch it as a whole movie. This is a big waste of space, but on small movies and cartoons it is still done. Other movies have different playbacks for people of different ages. This means the whole movie is all there but extra scenes are spliced into the movie for people who are older. This method is also popular for cut movies that give the option to play both the cut and uncut versions. In this case you will get both scenes played one after each other. Then we have multiangle movies where you will get the movie played for so long from one camera view, and then the same scene played again from another angle. This is done so the viewer can choose which view to watch as they watch the movie.

It is obvious from the above picture that the largest VOB files on the DVD will be the main movie, the next largest files will be the special features, the next largest files will be the trailers and finally the smallest files the menus. But almost all DVD's will have VOB files that are split into 1 GB files each. So a single movie will usually be built of 2-7 or more 1 GB VOBs. The trailers will just be a single VOB or sometimes just stuck on the end of the movie. And any special features may be built out of one or more Vobs. These Vobs are not separate chapters but are all part of a single title. This can be likened to a book, you will have many pages to a book but only one book. In the same way there are many Vobs to a DVD movie, but there is no pause between them they are played, they are read as one single file.

Every movie or title on your DVD has its own special name. For example, the main movie in the picture above is called:

Vts_01

All Vob files that are part of that title will be numbered in order like this:

Vts_01**_01**
Vts_01**_02**
Vts_01**_03**
Vts_01**_04**
Vts_01**_05**
Vts_01**_06**
Vts_01**_07**

So you will know that Vts_02_xx is not part of the same title or feature as Vts_01_xx.

IFO Files

As you can see from the previous descriptions of the Vob files, it would be impossible to play most Vob files back in a DVD player without some sort of guidance. If you did, you may get many languages playing at once. All subtitles appearing on screen and repeated scenes all over the place. Then when it reached the end of the first 1 GB Vob file it would stop because it wouldn't know what to play next!

Luckily few DVD's are quite this complex. But nevertheless, they do need information on how they should be played back. And this information is contained in the IFO files. Each IFO file contains a "play list" which tells the DVD player exactly what scenes to show, what subtitles to display and what audio track to use for each selection we make.

Again, each IFO file has is own special name which is always the same as that of the DVD. So going back to our example: the main movie on this DVD is called:

Vts_01

So the IFO file that describes how to play that movie is also called:

Vts_01

If we wanted Flask Mpeg to read this movie like a DVD player would we would open it in DVD mode and choose:

Vts_01_01.ifo

Then Flask would know that it is supposed to play back the Vobs

Vts_01**_01**

Vts_01**_02**

Vts_01**_03**

Vts_01**_04**

Vts_01**_05**

Vts_01**_06**

Vts_01**_07**

If you opened:

Vts_02_1.ifo

You would probably end up with the special features of the DVD and opening:

Vts_03_1.ifo

Would give you a movie trailer

Well, that's about all you need to know about the structure of a DVD. Hopefully this will help you with your decoding of them to DivX.

NTSC, PAL & Interlace Explained

The Motion Picture Camera & Cinema

Motion picture cameras are based on photographic film just like your everyday hand held photographic camera. Hollywood movies use 35mm film but professional camera men often use 16mm and the home enthusiast will usually be content with 8mm. To record a movie, motion picture film is spun around a big reel inside a camera and exposed 24 times a second. As a result it will capture 24 photographs, or what we call 24 "frames" every second (fps). Each frame is one complete photograph, it is not digitally stored or compressed - you could almost literally cut each one out and stick it in your family photo album if you wanted! Once the movie is made, the film is developed, placed onto a projector, and projected onto the cinema screen.

Resolution

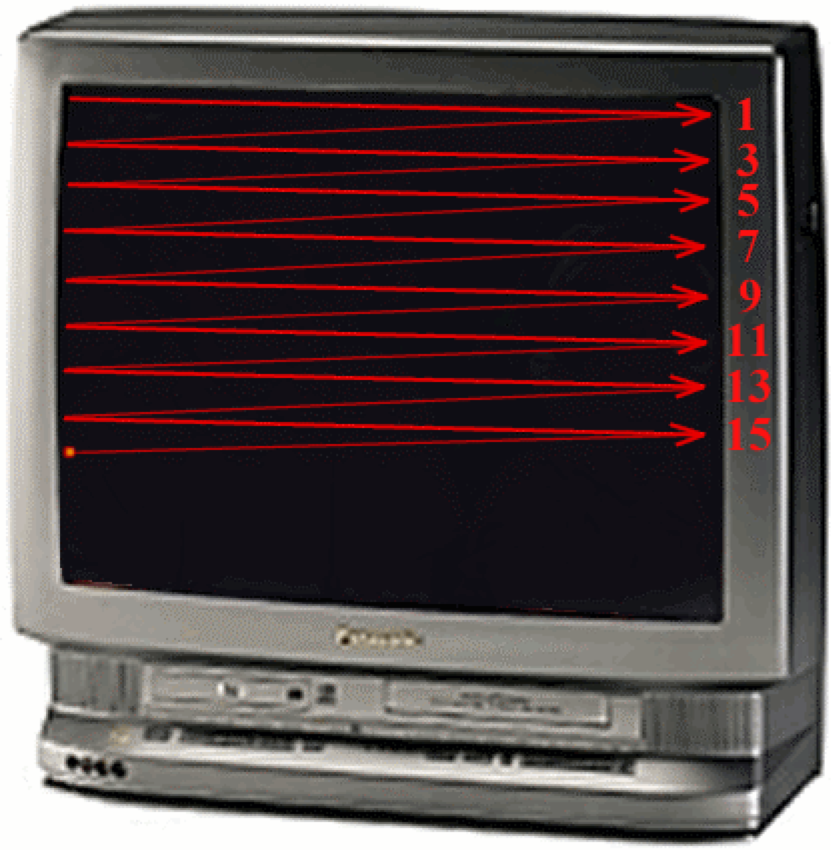
In terms of resolution its not really possible to compare a 35mm film to a VHS or VGA resolution because, like any photographic film, its resolution is based on a myriad tiny light sensitive crystals embedded into the film. When these are struck by light they change colour to match the light that has hit them producing a photo. But a 35mm film, based on average crystal size would be about 5000 x 5000 pixels. This is also the resolution Photoshop artists such as Craig Mullins use to create movie backdrops for the cinema. Nevertheless, the human eye can barely see the equivalent of 3000 x 3000 pixels of such a small area. So when a 35mm movie is scanned into a computer to try and get its full resolution for digital editing, it will be scanned in at 4096 horizontal pixels, also known as 4K.

Television

Television, on the other hand, is a whole other ball of wax! As you probably know, a TV screen is basically a empty glass box (or tube) with all the air sucked out of it. Inside the front of this glass box it is covered with a mesh of red, green and blue phosphor dots. At the back of the tube it has three devices (called electron guns) that shoot three beam's of electricity at these phosphor dots. When the electricity hits the dots they glow and a colour picture is produced. Increase the beam strength and you can brighten the amount of red, green or blue light produced at any part of the screen. This, in effect, allows the colours to mix into just about any colour and brightness imaginable. You might compare this to mixing coloured paints together to form new colours. Whatever way you look at it, this produces a colour picture that looks almost like real life.

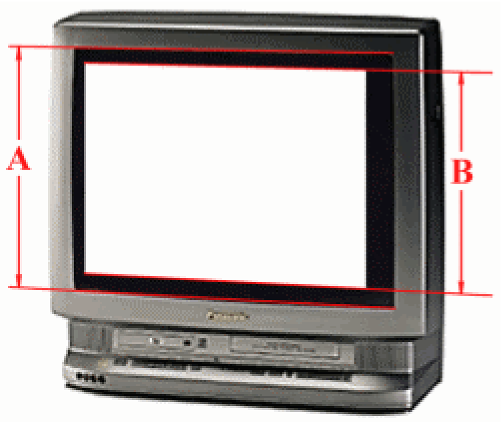
Interlace

Next is the important point! To produce a picture, these electric beam are controlled by electromagnets to scan from side to side across the TV screen (as illustrated in the picture below). The beam fly across the screen in the same motion our eyes use when we are reading a book. They start from the left, finish one line and then shoot back to start the next line.



When TV's were invented in the 1920s the type of phosphor used to produce the colours did not respond very fast. This meant it was impossible to get a picture in one shot; instead we would get a flickery strobing effect moving down the screen! To solve this they decided that instead of putting the lines on the screen one at a time (i.e. lines: 1, 2, 3, 4, 5) they would put them on every other line in one pass (i.e. 1, 3, 5, 7, 9) and then in-between the previous lines on the second pass (i.e. lines: 2, 4, 6, 8 etc.). This allowed a whole picture to be produced in two very fast scans and allowed enough time for the slow phosphor dots to recover. This, then, prevented any strobing effects from appearing - success! This process is called interlacing!

Resolution



An analog TV's resolution refers to the number of horizontal lines displayed on the screen. This is broken up into the active and non-active areas. The non-active or blanked (**A**) area is not used for the actual television picture and is basically always 'blanked'. The extra signal information that would have been put here is often used for closed captioning, synch info or other information such as VITC. But obviously the bit we are interested in is the active part which refers to where the actual picture will appear (**B**).

NTSC

The TV industry is dominated by two main standards for TV design: PAL and NTSC. NTSC is one of my pet hates basically because of it's rather low quality and use of weird framerates. NTSC stands for the **N**ational **T**elevision **S**ystems **C**ommittee, it is the colour video standard used in North America, Canada, Mexico and Japan. Some engineers have said it should stand for **N**ever **T**wice **S**ame **C**olor because no two NTSC pictures look alike :). Due to the electric system used in the US it was decided to scan the lines across the NTSC TV screen at about 60Hz (or 60 half frames per second) which produced 30 whole pictures every second. NTSC resolution is about one sixth less than that of PAL - about 89 lines less. This may not seem so bad, but divide a sheet of paper into six even parts and chop one off of the bottom and you will have a lot of detail lost. NTSC uses 525 horizontal lines of which only 487 make up the active picture.

PAL

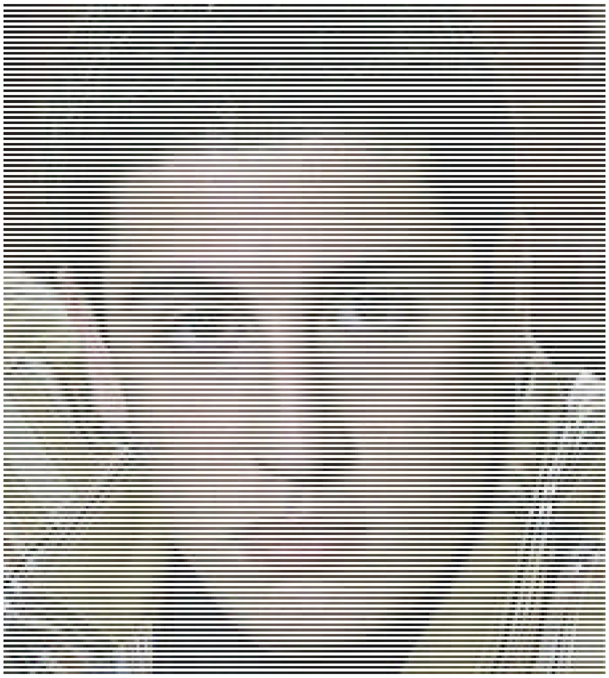
PAL stands for **P**hase **A**lternating **L**ine, it is the TV standard used for Europe, Hong Kong and the Middle East. It was a new standard based on the old NTSC system but designed to correct the NTSC colour problems produced by phase errors in the transmission path. PAL resolution is 625 horizontal lines but only 576 of these are used for the picture. PAL is higher quality than NTSC, it keeps a sharper picture and remains closer to the original format produced by motion picture cameras. Due to the European electric standards it was decided to interlace PAL lines every other line at 50Hz producing 25 whole frames every second.

TELECINE

This is the bit you've all been dying to read. Unfortunately I have not written this with a bunch of amazing solutions in mind. The idea is more to help you understand what is going on with your video so you can decide how you will process it better.

Just so you don't get confused you should be clear on what the difference is between a frame and field. A 'field' is basically every other scan line of a picture. Two fields stuck together makes a single frame on a TV set! In the picture below only one field is displayed on the left. Its hard to see because only every other line is displayed. The picture on the right is a whole frame. It is produced when we stick both fields together.

THIS IS ONE FIELD (OR A HALF FRAME)



THIS IS TWO FIELDS (OR A WHOLE FRAME)

Single fields that start from line 1 of the TV screen are called 'odd' because they go in odd numbers (i.e. 1, 3, 5 etc.). Fields that start from the second line to fill the gaps of the first are called 'even' because they go in even numbers (i.e. 2, 4, 6 etc.). Fields that start from line 1 are more often also called "Top" fields because they start from the first "top" line on the screen. Whereas single fields that start from the second line down are called "Bottom" fields. Okay, now everything you read should make perfect sense =)

TELECINE

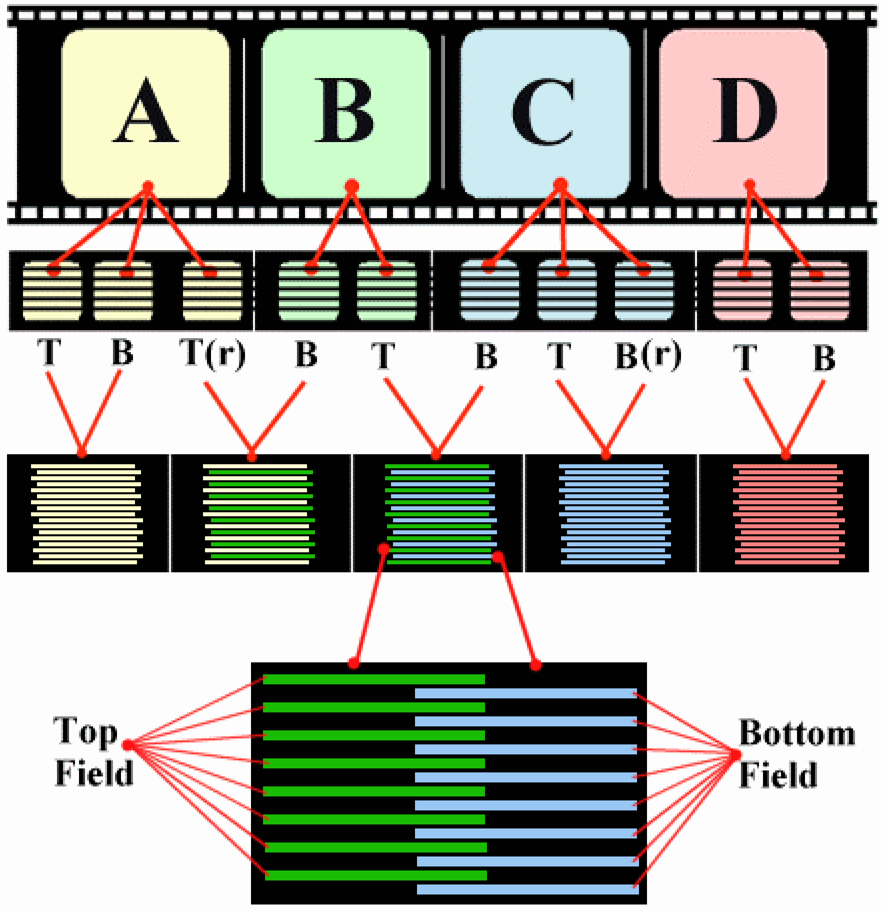
As I have already mentioned, a motion picture camera captures its images at 24 frames every second. Each frame is a full image. An NTSC television, however, must play 30 frames per second, and these frames must be interlaced into two fields both top and bottom! So basically what we are saying is we must play 60 half frames (or fields) every second. The only way we are going to be able to play a 24 fps motion picture on NTSC television is to change it from 24 fps to 30 fps and interlace these frames into two fields making 60 half frames per second. This transformation process is done with a machine called a Telecine. A Telecine machine does something called pulldown, which, in its simplest explanation, "pulls down" an extra frame every fourth frame to make five whole frames instead of four!

3:2 Pulldown NTSC

3:2 pulldown is a name that confuses people basically because the term "pulldown" is rather ambiguous - in other words, it's not really *pulling down* anything! The process sounds complex but its really quite straightforward and I have designed a picture to illustrate. The top row in the picture below represents four frames from a motion picture camera. These are full frames and not yet interlaced they are represented as A, B, C, D.

Now look at the second line in our picture below. The Telecine machine takes the first whole frame A and splits it into three fields (stop reading and take a look now). For the first field it uses the top field (**T**) which means it takes lines 1, 3, 5 etc., from the original digitized picture. The next field taken from A is the bottom field, so it will take lines 2, 4, 6 and so on. The third file we see labeled (**Tr**) is just a copy of the first field again (so I labeled it **T(r**) to mean: *top repeated*').

Now the Telecine machine goes onto the next frame B. This time it just takes the top and bottom fields. Then we move on to the third frame C; it splits it up into three fields, bottom (**B**) top (**T**) and a repeat of the bottom one again (**Br**). Finally, the forth frame D is split into the top and bottom fields. Thats it, that is *all* a telecine machine does!



In short, this results in a field order of 3 fields, 2 fields, 3 fields, 2 fields! Or, if its easier to understand, our picture above shows it as: 3 yellow, 2 green, 3 blue, 2 red.

So that is why it is called 3:2 pulldown, it goes in a sequence of 3, 2, 3, 2 and so on. It can be said to "pull down" a whole frame and split it into three fields and two fields. Finally, after the Telecine machine has finished the forth frame D, it will start the process all over again with the next four pictures of the movie.

In short we end up with:

At **A**b **A**t / **B**b **B**t / **C**b **C**t **CB** / **D**t **D**b

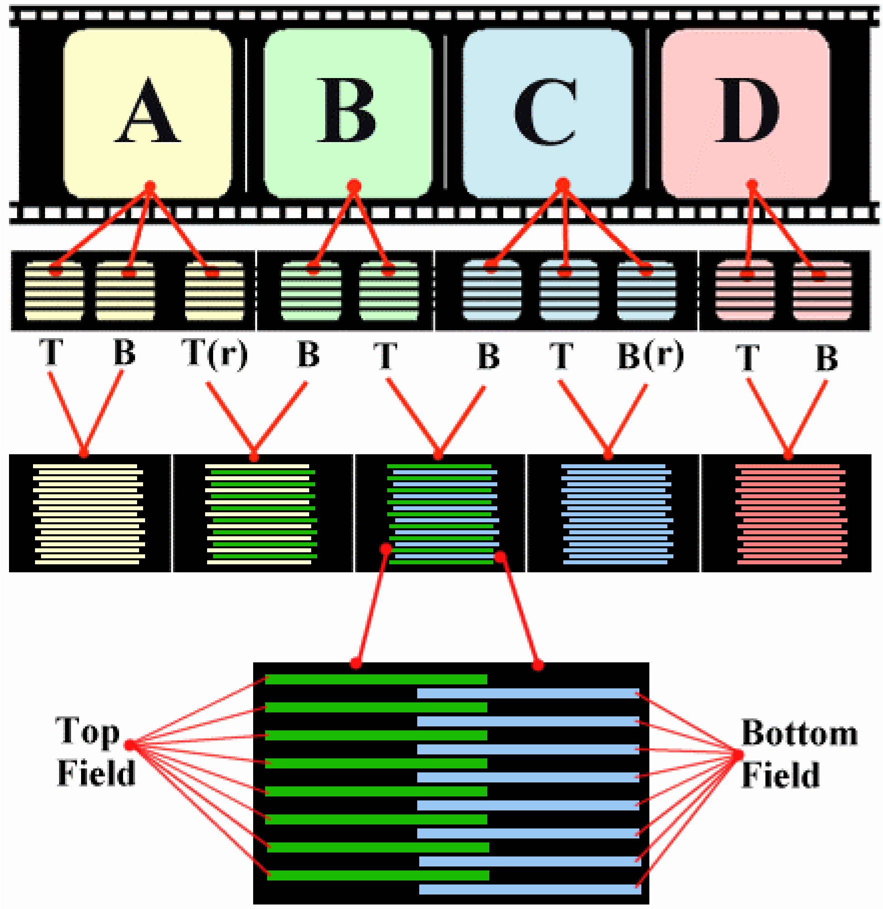
But because it always goes: top, bottom, top, bottom, top, bottom etc., we would just say it without indicating top or bottom fields. So instead of the above we would describe it as:

AAA BB CCC DD

Whatever way you look at it in the end you end up with 5 whole frames instead of 4. This turns a 24 fps movie into a 30 fps movie!

Interlacing the picture back together

Lets look at the picture again. Look at the third line down. Here we can see how these fields would be woven back together to produce a whole picture again, as we would see on a TV or computer screen. The top field of frame A is woven together with the bottom field of frame A. Then the *repeated* top field of frame A is woven together with the bottom field of frame B. The top field of frame B is woven with the bottom of frame C. The top field of frame C is woven together with the top *repeated* field of frame C. And finally, the top field of frame D is woven together with the bottom field of frame D.



That's quite a mouthful to explain in words but examine the picture, it should really explain itself. Since each frame is stuck together instead of describing telecine by saying it uses top, bottom, top, bottom in the order:

AAA BB CCC DD

We would say:

AA AB BC CC DD

The change is only how we group the letters of course and means nothing more.

A Weird Framerate

This is not quite the end of the saga. The old black and white TV's used to play back at a perfectly round 30 fps. But as usual NTSC found a way to destroy that perfection! With the introduction of color TV it was decided (because of technical reasons which I don't understand) that the movie must be played back at 29.970 fps (59.94Hz) which is basically only 99.9% of its full speed. As a result NTSC movies still have the same amount of frames they did when they were telecined, but they are played back at a fractionally slower rate.

2:2 Pulldown PAL

PAL movies also get telecined but not in the same way an NTSC movie does. A Telecine machine will use what is sometimes called 2:2 Pulldown! This basically turns every frame into two fields so they can me played on a standard PAL television. This makes 25 frames into 50 field which when played on a TV set at 50Hz will produce 25 whole frames per second. So instead of going 3, 2, 3, 2, 3, 2 it will go 2, 2, 2, 2, 2, 2! This produces the fields:

At **AB / B**t **B**b **/ CT CB / DT dB**

Or just:

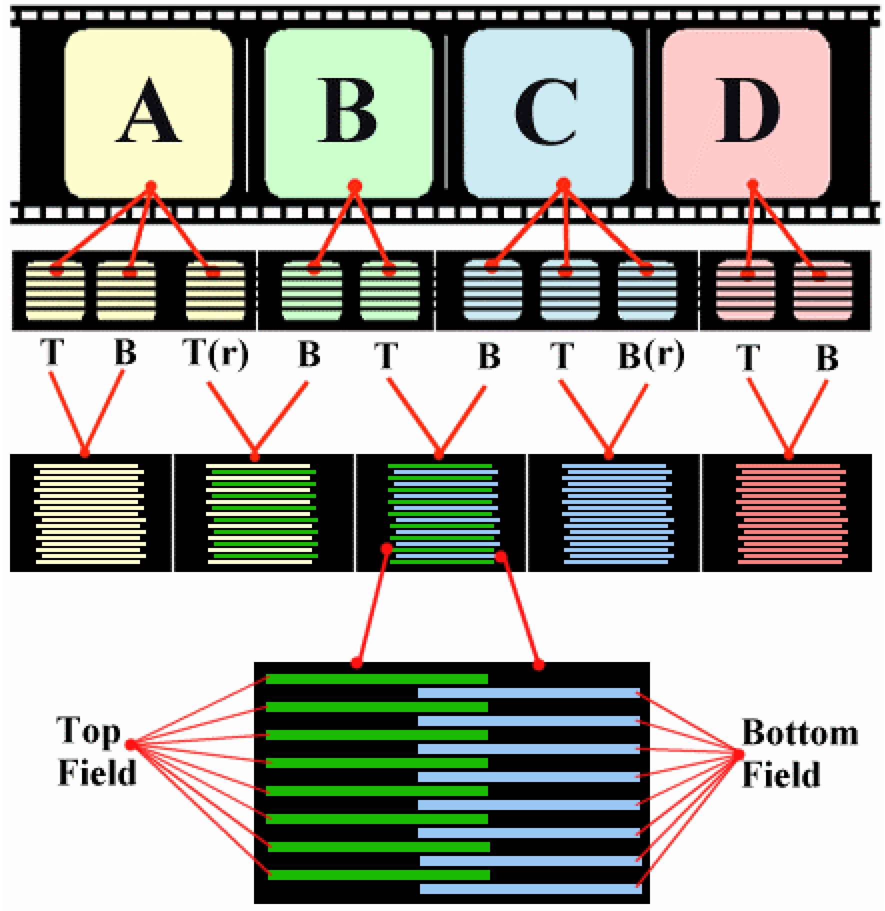
AA BB CC DD

Again, a PAL movie will contain all the frames from a 24fps film with *no* additional ones, but it will still play those frames back faster at 25 fps. In a way of speaking it is just as correct (or wrong) to say that a PAL movie is 24 fps because no frames have been added to it, they are just played back faster.

INVERSE TELECINE (IVTC)

I think I'm correct in saying that there is no such thing as an Inverse Telecine machine :). But, as the name suggests, inverse telecine is a process that turns a 30 fps movie back into a 24 fps movie. Basically what it does is take out all those extra fields that were added to the movie to make it 30fps. Its about now that I start spluttering because this is an awkward subject and I can't find any information on *exactly* how Inverse Telecine is performed! So instead I will describe what "looks" like should be done based on how it was telecined in the first place.

Lets go back to our picture! As you can see from the second row down, to turn the 24fps movie into 30fps we have to separated the pictures into 10 single fields (or half frames) by adding two fields that shouldn't normally be there. Counting from left to right, all we would need to do to turn or 10 fields back into 8 fields (to turn 30 fps into 24fps) is to delete fields 3 and 8. Remember we are talking fields here *not* frames.



But taking out fields 3 and 8 would produce a movie that had a field order of: top, **bottom, bottom**, top, bottom, **top, top**, bottom! Since you cannot weave together two *bottom fields* or two *top fields* we would need to swap them around. So imagine the order of the numbers as:

1, 2	3, 4	5, 6	7, 8
T, B	**B, T**	**B, T**	**T, B**

To get the correct order we must change them to:

1, 2	4, 3	6, 5	7, 8

T, B	**T, B**	**T, B**	**T, B**

Which gives us an order of: 1, 2, 4, 3, 6, 5, 7, 8 which should theoretically fix everything.

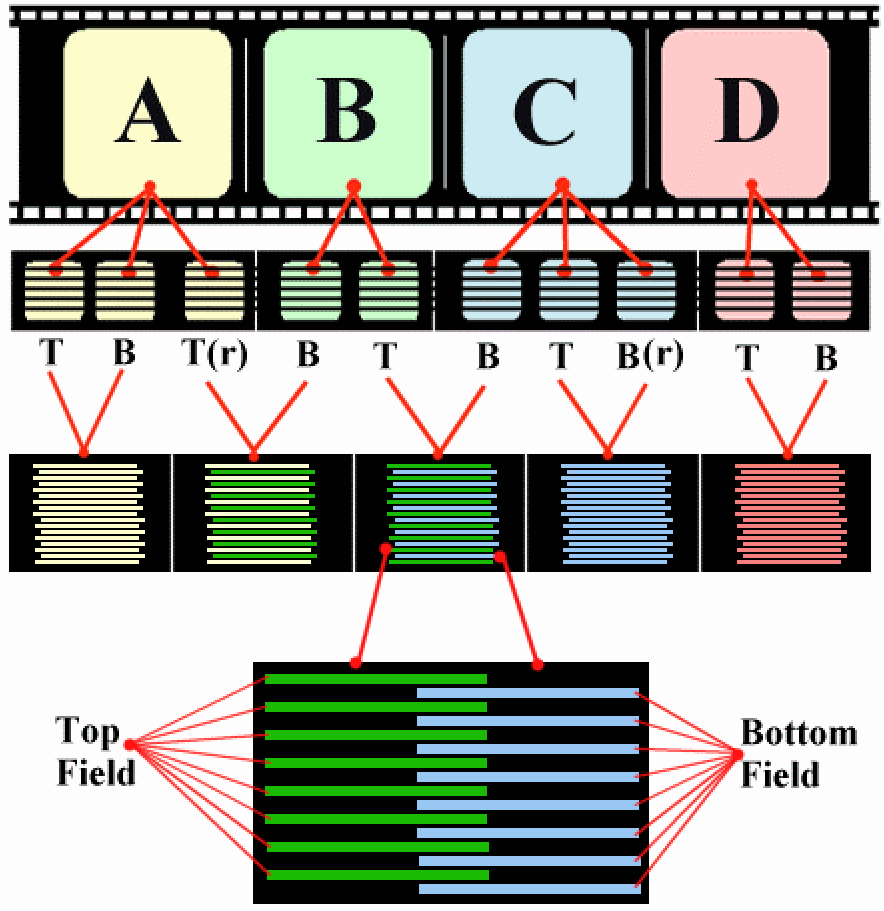
<u>The Framerate Mystery Unraveled! 23.976 / 24 / 29.970 / 30</u>

If the only framerates we use are 24, 25 and 29.97 then why to people speak of using 23.976? This is to do with how the movie has been created. A 25 fps movie still has the same amount of frames as a 24 fps movie because none have been added. But nevertheless a PAL television chooses to play them back at 25 fps. This makes the PAL movie play back at a slightly shorter length and means the audio will be out of synch slightly. To compensate for this, when a movie is telecined they apply to it what is called a 'pitch-correction' which speeds up the audio to match the playback speed, in the case of PAL this means they perform a pitch correction of about 4%.

The amount of frames an 3:2 pulldown telecined movie has is 30 fps. But an NTSC television will play them back slower at 29.970 fps (59.94Hz). The amount of actual frames hasn't changed, none have been added or taken out! Here is where the 23.976 part comes in! If we inverse telecine a 30 fps movie we would end up with 24 fps. But if we inverse telecined a 29.970 fps movie, because it has a slightly slower speed, instead of getting 24 fps as we should, we will end up with the slightly slower rate of 23.976 fps.

PROBLEMS WITH INTERLACED MOVIES

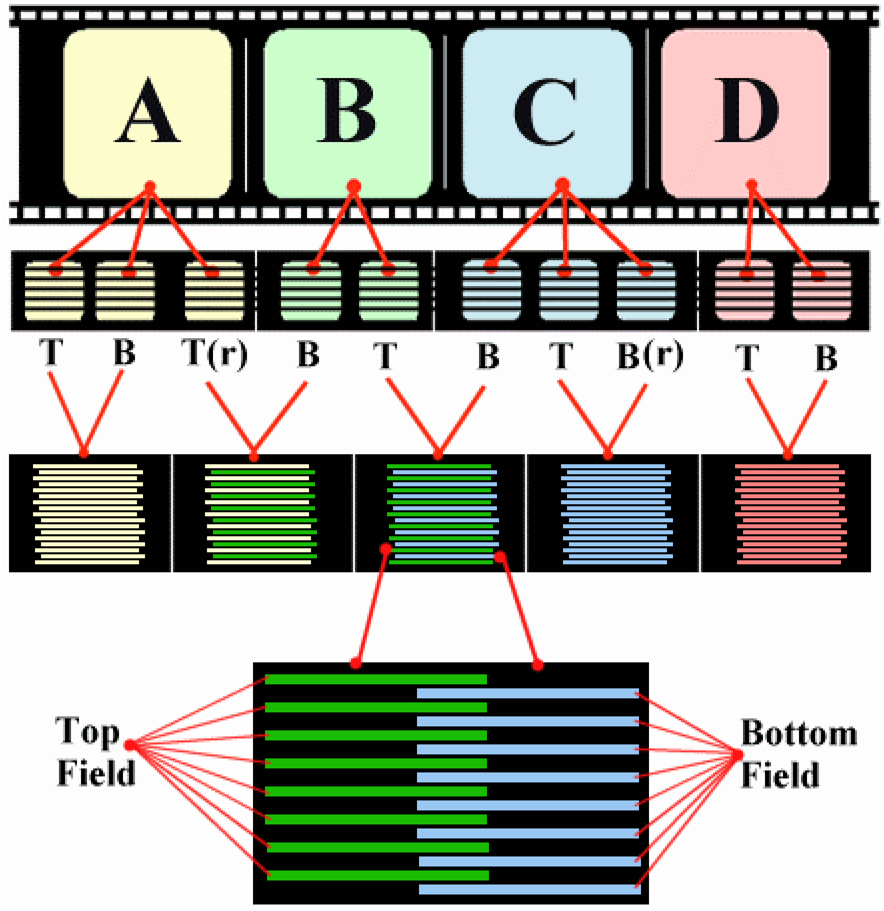
Interlaced movies look fine on a standard TV but for some unknown reason they appear terrible on a PC monitor!? Lets take a look at our example one last time to see why. Look at the last row where it shows how the top field of frame B is interlaced together with the bottom field of frame C.



We are getting the top and bottom fields from two completely different frames!! Imagine taking half of one picture and half of the next and trying to put them together into a single picture, its impossible! On a PC this produces what we see below. Here we have Star Treks William Riker walking across the room from left to right. Notice that the top field from the previous frame shows him a little to the left and the bottom field of the next frame shows him a little to the right. This is what produces this combing effect and no amount of shifting the lines to the left or the right will fix it!

Inverse Telecine Troubles

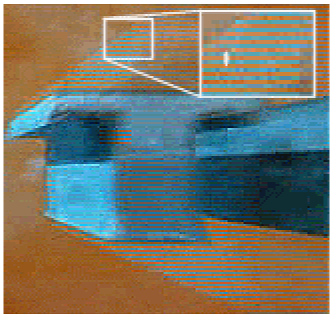
Look at our illustration one more time. A 3:2 pulldown movie can also be encoded as 2:3 which produces *exactly* the same result but is done backwards - instead of getting 3, 2, 3, 2 we will get 2, 3, 2, 3! But this doesn't matter because since a 3:2 pulldown movie can be cut and edited after it is made the very first frame doesn't always start with the top of field of **A** anyway! It could, for example, start with the next one across - the bottom field of **A**. In fact, it could start with *absolutely any* of the 10 fields in the sequence!



Hence as far as I can see there must be at least 10 ways to perform inverse telecine. Five assuming the first field is top and five assuming the first field is bottom. Let me know if you know exactly how IVTC works and I'll update this article to explain it better.

OTHER ISSUES

Some of the specials and extra features of a DVD seem to have been recorded from a telecined 29.970 fps source! This means that the interlaced picture is actually edited as an interlaced picture on a computer and then reinterlaced again! There is absolutely no way to fix such a problem because the lines are literally a part of the original picture now. For example, I have taken this frame from the trailer of one DVD and separated the fields into two. When I squash all the lines together from *one single field* I get the following picture:



Of course, I could be completely mistaken about this, but that is what appears to be the case.

Capture Cards

Most of the Graphics Cards, TV Tuners and Video Capture hardware we use to record video to the PC will not perform any kind of IVTC. Neither do they seem to give a damn what order they whack the TV fields together. This means regardless of if you use PAL or NTSC, if you want to capture any video footage at above 240 pixels high (for NTSC, PAL is 288) you will get at least some interlace problems! When you are capturing below 240 pixels the capture card will only use one field and hence interlace problems will be almost impossible. If your capture card can get larger picture without problems check the instructions to see how its doing it! You may find to your horror that it is actually just capturing at 240 and enlarging the picture *after* it has been captured. This is obviously a serious waste of space!

Deinterlace filters

Since to perform inverse telecine (IVTC) to make a 30 fps movie back into a 24 fps movie is so awkward there are a few alternatives that have been designed to work on just about any movie. There are only two types that I know:

Bobbing: To Bob basically means to enlarge each field into its own frame by interpolating between the lines. So from one field we are producing a full frame. Because the top fields are a line higher than the bottom the image may appear to "bob" but this is usually fixed by nudging the while frame up or down a pixel. You are only really getting half the resolution with bob but the interpolation is usually very good quality. If you are stuck for a way to bob your video my AVISynth guide offers a bob feature, check it out Here.

Blending: Flask Mpeg's deinterlace filter look for the parts of a picture where the two fields do not match and blends the combing effect together. The lower the threshold the more the two parts are blended and the less of a combing effect appears. The problem with this method is that the final picture can quite often end up a bit more blurry.

DVD & TELECINE

DVD's offer a strange twist to the whole Telecine and 3:2 pulldown business. Almost all DVD's will have the movie stored as whole pictures at 24 fps. This is the original format of the film with no Telecine. At the start of every Mpeg-2 DVD file there are certain header codes that tell it how to play back the DVD. Since it is stored digitally it can give the fields or frames from the DVD and to the hardware or software in any order it likes. It can split the movie into two fields and perform telecine instantly. To do this has three flags that can be applied to the header code: RFF (repeat first field) TFF (top field first) and FPS (frames per second).

For a PAL DVD the FPS flag can be set to 25 and the DVD will send the picture information to the hardware at 25 fps instead of 24 fps as is stored on the DVD.

For NTSC DVD's the movie needs to be 29.970 fps so the FPS flag is set to 29.970. But this looks odd because the movie is over far too soon. Imagine it like playing cards, if you throw 4 cards on the floor every second the whole pack will be finished in half the time than if you threw 2 cards onto the floor. The solution is to telecine the movie with 3:2 pulldown to increase the amount of "cards" we have to start with. To do this it uses the RFF and TFF flags are set in the header code. By setting the DVD to Repeat the First Field again you make the video display the fields in the order 3, 2, 3, 2. By setting the TFF flag you set the DVD to start from the top field so the order always goes: *top, bottom, top, bottom*.

Theoretically then, it should be possible to patch the header code of a DVD's Mpeg-2 file and make it play back at 24 fps instead of the 29.970 fps! In fact some people have made patches to do this, but so far, for another unknown reason they are very unreliable and the video turns out just as bad!

Progressive and Interlaced together!

I don't think I have mentioned what a progressive image is yet? A progressive image is a whole frame that it is not interlaced. Motion picture camera's capture images that are progressive. They are not telecined or split into separate fields. Computer monitors do not need to interlace to show the picture on the screen like a TV does it puts them on one line at a time in perfect order i.e. 1, 2, 3, 4, 5, 6, 7 etc.

Many DVD's are encoded as progressive pictures, with interlaced field-encoded macroblocks used only when needed for motion. Flask Mpeg tries to take advantage of this fact, because if you set it to 24 fps (or 23.976) it will give the option to reconstruct progressive images. This does not perform any deinterlacing on the video but ignore all the flags and just reads the DVD one progressive image at a time.

This is another confusing issue for me. I have no idea how a DVD movie can be both interlaced and progressive other than by the fact that a progressive movie can be played back as interlaced due to control flags. If I learn any more about this I will update my articles accordingly.

VHS, VCD & DVD

To finish, perhaps it would be nice to say a few words about the video formats too. It wasn't long after TV that VHS video recorders appeared on the scene and a yet a while latter when the Video CD-Rom's did. Of course, there were other video formats, but VHS (Vertical Helical Scan) and MPEG (Moving Picture Experts Group) won the battle, at least as far as home video was concerned. This is a little strange really because Sony's Betamax video was probably the better quality! Anyway, all video formats to date have required one form of compression or another to be able to record the huge quantities of information needed to store full motion video.

VHS

VHS video is stored just like audio on a reel of plastic tape impregnated with ground up iron. This plastic tape is spun in front of an electromagnet that replicates the strength of the TV's electric beam as they would appear scan across the screen. This caused magnetic 'kinks' in the iron parts of the tape that are almost identical to the original TV signal. A reversal of this storage process would produce the image back on the TV screen. The signal is simplified before it reached the tape therefore making it take up less space.

As anyone who has ever used video tape knows it soon looses quality. It appears grainy, looses colour accuracy and starts to produce white glitches and audio waver - a better solution was needed.

MPEG-1

As computer technology advanced CD-Rom video formats became popular and the Moving Picture Experts Group designed a compression format that could store over an hour of VHS quality video on a single CD-ROM This soon become very popular in the east but never truly caught on anywhere else. This was due to the fact that recording it was difficult and slow and the quality was not really any better than normal VHS anyway. The big big advantages of Mpeg-1 video was that it was almost impossible for it to loose picture quality like a VHS videotape! It could last perhaps over 100 years of use without *any* noticeable degradation of image quality!

MPEG-2

Since (at the right bitrate) Mpeg-1 was able to produce TV quality pictures superior to VHS, the Mpeg organization decided to design another version that allowed Mpeg-1 back with interlaced images so it could be used for TV broadcasts. This format was called Mpeg-2. Other features were added to Mpeg-2 to make it compress slightly better and higher quality, but the main difference was the addition of interlace support.

Since Mpeg-1 VideoCD's showed that a CD based digital video was not only a viable option, but also a very preferable that is one if the storage space was enough. When CD-ROM designs were upgraded to be able to store 4.38 gigabyte or more of information, it was decided that these new CD's would be the new storage media for video. It was called DVD to mean Digital Video Disc although it was later changed to mean Digital Versatile Disc because it was 'versatile' enough to hold other data besides video.

Resolutions

Resolutions are an important issue for amateur video enthusiasts who want to capture their video at full TV quality. Professional video editors are told to capture at 640 x 480 pixels for highest quality. But a PAL TV resolution is 576 lines. Then we have the Mpeg group saying that 352 x 288 is the *full* VHS video resolution! The problem seems to lie in the fact that its hard to equate a TV resolution with a computer image. The TV is built up of lines but the dot definition is rather "fuzzy" looking. So rather than me rattling on about the pro's n cons here I will merely end this article by quoting what the Ligos corporation (the creators of the LSX Mpeg-2 encoder) say in regard to this subject:

> "The resolution of computer video, however, doesn't generally equate to the video world of televisions, VCRs, and camcorders. These devices have standards for resolution that are generally focused on the horizontal resolution (the number of scan lines from top-to-bottom that make up the picture). Here are some numbers for comparison:

Video Format	**Horizontal Resolution**
Standard VHS	210 Horizontal Lines
Hi8	400 Horizontal Lines
Laserdisc	425 Horizontal Lines
DV	500 Horizontal Lines
DVD	540 Horizontal Lines

> With these numbers in mind, it is important to remember this rule when bringing the worlds of computer and video together: the quality of an image will never be better than the quality of the original source material.

> We suggest capturing at a resolution that most closely matches the resolution

of the video source. For video sources from VHS, Hi8, or Laserdisc, SIF resolution of 352x240 will give good results. For better sources such as a direct broadcast feed, DV, or DVD video, Half D1 resolution of 352x480 is fine. There are other advantages to following these guidelines. Your files will be smaller, consuming less space on the hard drive or on recordable media like CD-R and DVD-RAM. You'll also be able to encode more quickly".

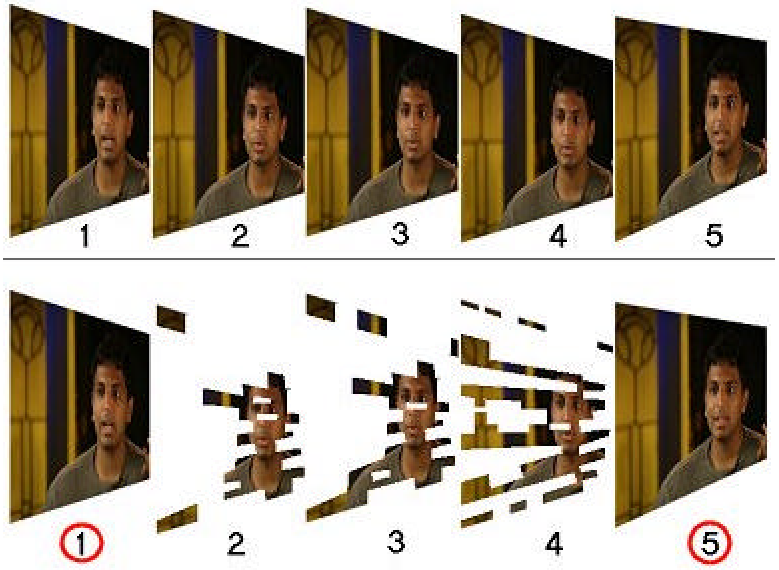
AVI 4GB Limit Explained

Be careful when dealing with large files! The Windows 95/8 file system cannot handle AVI files larger than 4 gigabytes! There has been much confusion regarding this issue. In short, the old Windows File Allocation Tables (FAT16) cannot store more than 2 gigabytes per file. Windows 98's FAT32 allows a 4 gigabyte storage capacity per single file. So where does the 1 and 2 gig storage limit come in? Most AVI parsers use something called signed arithmetic. This forces a storage limit of 2 gigabytes for .avi files on Windows 98. But the multimedia system in Windows 95 cannot cope with RIFF files (such as .avi files) bigger than 1 gigabyte! And this is why people will say there is either a 1, 2 or 4 gigabyte limit on single file storage. This is why DeCSS, DOD Power Ripper, Vob Merge, Peck Power Join and all other programs like this are incapable of storing more than 4 gigabytes on most peoples computer systems! I think this is the chief reason for so many unexplained error messages when using these programs too!

There is a partial way around this that is used extensively by Virtual Dub that is similar to OpenDML. These settings can be used to create very large .avi files by grouping smaller files together so they are forced to play in sequence. This solution means that the finished files can only be played with Windows Media Player or other programs that support OpenDML type files. For more details on this consult the VirtualDub help files.

Keyframes & Delta Frames Explained

All Mpeg movies are built out of 16 x 16 squares. To save space the squares that are "almost identical" to the same squares in the next frame of the movie are discarded. This makes a very high compression ratio, because in a scene where two people are talking and not moving very much the only squares that need to be copied across one frame to the next are those around the mouth. But because only squares are carried across from one frame to the next you do not have a complete picture, just squares! For example, to view frame five you must load frame 1, 2, 3 & 4 and stick all the squares together to make frame 5! Many ASF's and some of the early DivX movies were made almost completely like this. The only problem with this method is that you couldn't select where you wanted to watch the movie from. You couldn't, for example, watch half a movie and then come back later and fast forward through to the part where you left off. You had to watch it from the beginning! This was because to fast forward Media Player (or any player) had to examine EVERY frame before it could reconstruct the movie at the point you left off!



(picture above) The top row represents frames 1-5 in an uncompressed AVI file. Notice how each frame is a complete frame. Now look at the frames below from an Mpeg file. Frames 1 and 5 are Keyframes showing a complete picture, but frames 2, 3 & 4 contain only the bits of information (delta frames) that are different from the previous.

Here is where Keyframes come in! A keyframe is added every so many seconds to keep track of the position of the movie. It also provides a perfect picture on which the half frames (delta frames) can be based. Mpeg encoding methods call keyframes I-Frames or Intra-frames. The half-frames for mpeg come in two kinds: B-frames or the backward frames and P-frames the predicted frames. This sounds complicated but both do very similar things and are designed to only store the difference between the frames in front and behind the keyframe as small blocks.

Most compression software use 1 keyframe every 5 to 10 seconds. Because keyframes, as full pictures, hold so much more information than partial frames they will increase the file size quite a bit. So obviously the fewer you have the better the compression. If you chop an mpeg file "in-between" keyframes the player will not be able to reconstruct the movie frames until it reaches the next keyframe frame in the list! Some smart mpeg software are able to reconstruct the final keyframes from the parts given and get around this problem, but you have to watch out for those who don't.

VirtualDub can usually fix a film without keyframes if it’s opened in repair mode, but it cannot yet cut in-between keyframes. This means we must cut *on* the keyframe.

With this in mind is useful to know that, provided you use the VirtualDub keyframe buttons to move forward and backward around the parts of the film you intend to cut and join you cannot possibly cut in-between a keyframe because you are moving only by keyframe jumps =^)