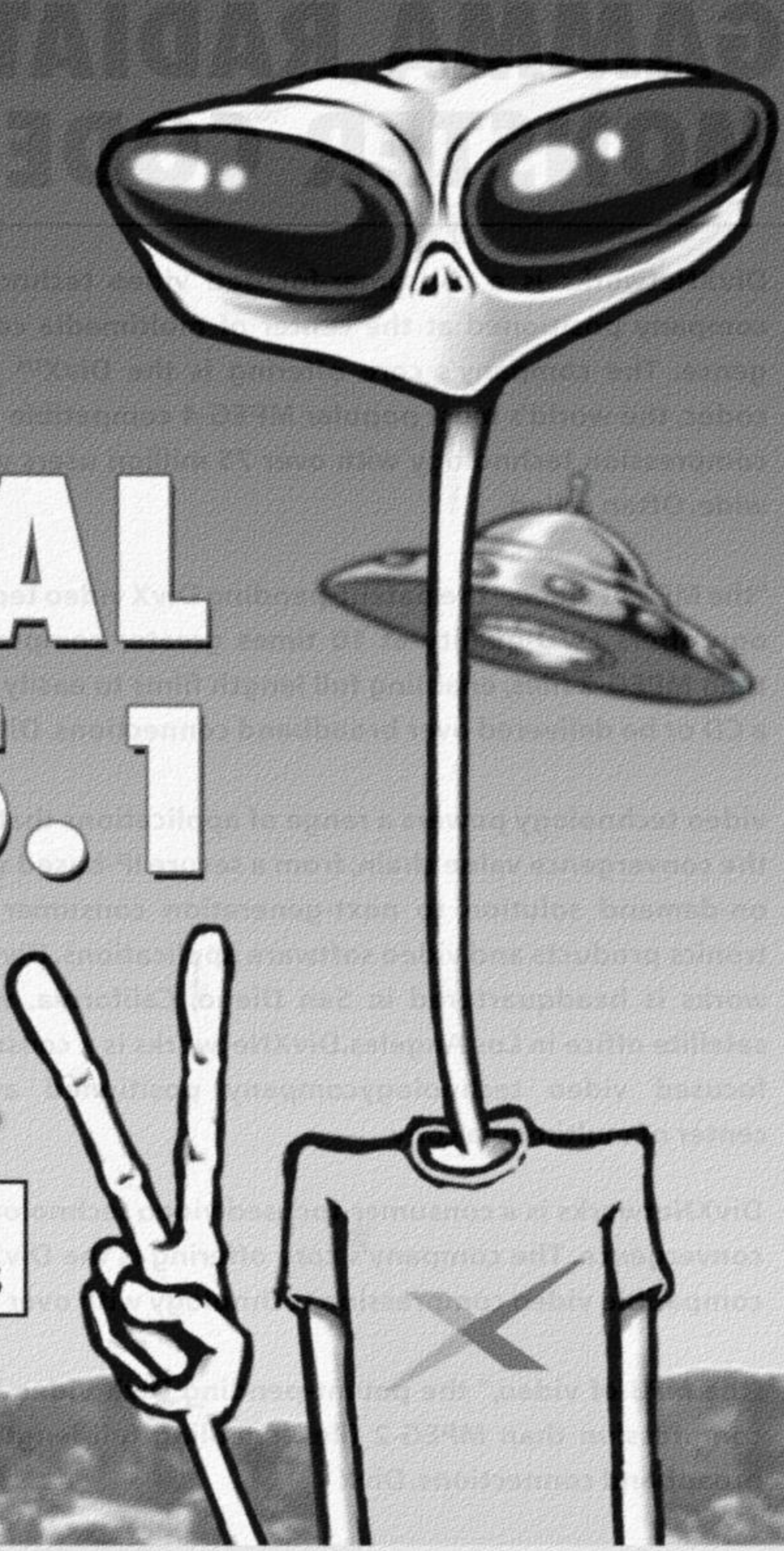
NEW VERSION 5.1

EXCLUSIVE UNDISCLOSED DIVX 5.1 CONTENT INSIDE

DIVX®

THE OFFICIAL DIVX 5.1 GUIDE



Discovered Alongside Remaining Dead Sea Scrolls

EVIL BRAIN KEPT ALIVE IN MAYONNAISE JAR

0 12245 62890 5

Revision 1.0

Contents

Index

Introduction

Quick start guide

Forward

Bitrate mode

Performance/Quality

Psychovisual Enhancement

Source pre-processing

Crop and Resize

MPEG4 Tools

Advanced

Interlacing

Video buffer verifier

Profiles

DivX Certified™ Program

Electrokompressiongraph

DivX Decoder

Acknowledgements

Introduction

Introduction

Welcome to the guide

Welcome to the official guide to DivX® 5.1!

With this guide we shall be demystifying the black-art of DivX encoding - leading you step by step through everything from creating your very first DivX video to exploiting the many powerful advanced features that DivX has to offer.

This guide covers all of the features available in both the DivX and DivX Pro encoders in detail. You will learn how to optimize video quality via the encoder and where to use the EKG for even further enhancement.

We will discuss the DivX Certified Program and how you can create videos compatible with the growing range of DivX Certified DVD players that are set to take the consumer market by storm in 2004. We will also show you how to configure the DivX decoder to improve the playback of DivX videos on your personal computer.

Before we begin it is worth noting the outstanding pedigree of DivX video technology. What follows will bring you up to date on over four years of DivX history.

The origins of DivX® video

In 1999, French video engineer Jérôme Rota (better known by the alias “Gej”) began experimenting with a powerful new video technologies.

The “MS MPEG4” codec from Microsoft, a video compression technology then new to market, was limited to storing video data inside their proprietary *ASF* (advanced systems format) container file. Gej wanted to use a high-quality video codec in combination with standard *AVI* (audio video interleave) containers, which would enable nearly all video editing applications to benefit from new compression.

By September 1999 Gej had achieved his goal of using a high performance codec in combination with the AVI container, and *DivX ;-)* (the emoticon a jibe at a now failed protected DVD rental system launched by *Circuit City*) was unleashed upon the Internet. Boasting compression ratios over five times superior to the technologies used in conventional DVDs, DivX ;-) would allow entire movies to be transferred to single compact discs or distributed over high speed connections in a matter of hours rather than days.

DivX ;-) gained a cult following amongst the internet underground almost overnight. Within days of the first DivX ;-) release it became possible to download entire feature-length videos online for the first time, and DivX ;-) was well on the way to becoming the MP3 of video.

But DivX ;-) was just the beginning. Intrigued by the possibilities of video compression and not ready to retire and rest on his laurels, Gej quickly dropped the winky face and began the long process of creating a new codec entirely from scratch.

In May 2000 Gej became co-founder of *DivXNetworks*, together with CEO *Jordan Greenhall*, Director of Product Management *Joe Bezdek*, R&D director *Darrius Thompson* and Vice President of Operations *Tay Nguyen*.

Soon thereafter, DivXNetworks covertly launched a website known as *ProjectMayo* that in early 2001 would become the stage for creating *OpenDivX* - an open-source development project that invited developers and video engineers worldwide to become part of a collaborative effort to create a free, open version of DivX. This would ultimately become the foundation of the XVID project. Meanwhile, DivXNetworks secured approximately $5.6M in funding from investors and began internally developing the future of DivX technology.

By August 2001, DivX 4 was born. A proprietary codec based on the international MPEG-4 standard, DivX 4 became the first in a long line of DivXNetworks releases, culminating in the commercial release of DivX 5.0 in March 2002.

Since that time DivX has gone from strength to strength - a patent-pending technology that is constantly evolving in terms of the performance, quality, and features. DivX 5.1 sees the ultimate culmination of this work.

What is DivX?

Firstly, let’s address what DivX is not. DivX is not a stand-alone encoding application; that is when you install DivX you are not installing a program that will allow you choose a file you wish to process and then proceed do the conversion automatically. For such an application, look to *Dr. DivX*.

Instead, DivX is a video codec (*CO*mpressor/*DEC*ompressor) accessible via most video applications supporting either:

- ►*Microsoft Video For Windows* for applications running under *Windows*
- ►*QuickTime* for applications running under *Mac OS*

In short this means DivX makes possible the export of high quality video from virtually any software of your. With the right software it becomes possible to take any video source - be it live capture, DVD, DV, MPG, MP2, AVI or other - and export them to DivX video.

The DivX 5 series encoder is an implementation of the *MPEG4 Video Standard*, supporting both *simple profile* and *advanced simple profile encoding*. Essentially the *Moving Pictures Expert Group* creates standards and methods for achieving various objectives, one of which is video compression. Defined within the MPEG4 Video Standard are various profiles representing feature sets that a codec complying with the standard must fully implement.

Because it is based upon the MPEG4 Video Standard, DivX naturally supersedes both MPEG1 (as used for Video CD) and MPEG2 (the standard for DVD video and Super Video CD) for low-bitrate encoding. Putting this into perspective, not only does DivX achieve video quality rivaling DVD with file sizes less than one fifth as large, but because DivX 5 is fully standards compliant your DivX videos can be played in any of a growing range of DivX Certified hardware players*.

* Different MPEG4 hardware players tend to support the MPEG4 Video Standard to varying degrees. For guaranteed playback capability and performance only *DivX Certified* players are recommended.

Why use DivX?

There are many codecs available these days, so why choose DivX to export your video?

►**Quality**
DivX is a professional grade video codec. The *Pro* version available to consumers for less than $20 is exactly the same program licensed for use in commercial production environments. While other MPEG4 encoders do exist, no other encoder includes the extensive optimizations to both performance and patent pending algorithms that enable DivX to deliver the highest possible quality while maintaining full standards compliance.

►**Certified video**
DivX produces video guaranteed to play back flawlessly on a completely new range of DivX Certified™ devices. The DivX encoder includes a simple wizard for selecting profile settings that allow you to encode for players with differing capabilities, ranging from the handheld profile for palm-top computers right through to high-definition profile for players that can decode for HDTV. Simply match the logo shown in the encoding wizard to the logo on your player and you can't go wrong! The DivX Certified Program already includes over 30 companies and DivXNetworks rigorously test every single certified player for guaranteed performance.

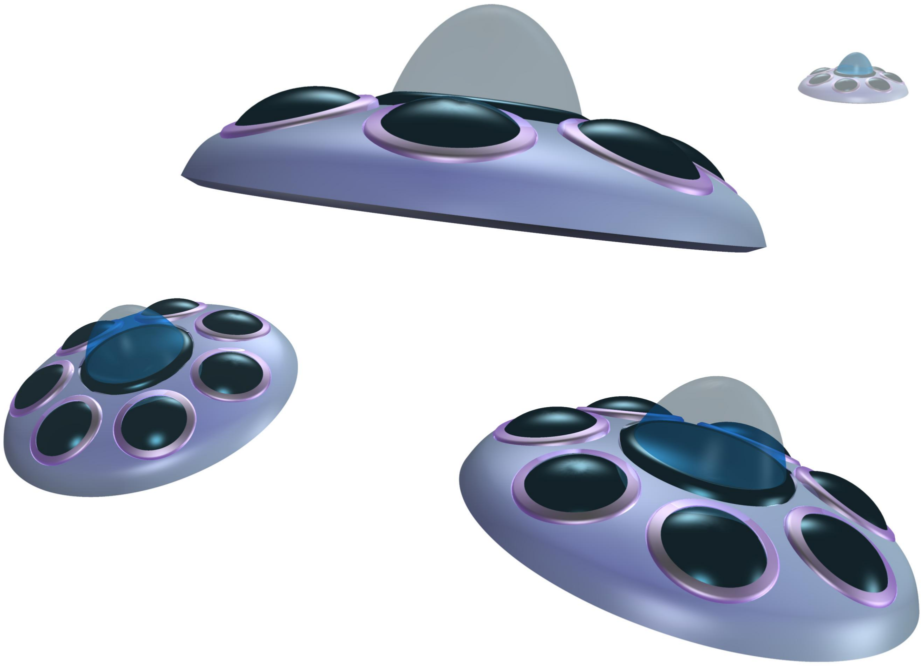
►**Home movies**
With DivX you can export your home movies in unprecedented quality then burn them to CD and share them with friends at next to no cost. Not only that, but because DivX movies take up so little file space you no longer have to worry about cutting down the running length and missing out on all those golden moments. Use DivX now to create clips you can be proud to show your relatives for years to come.

►**Backup DVDs***
DVDs are expensive, and one little scratch is sometimes all it takes to turn your favorite disc into a worthless coaster. By converting your DVDs into DivX format you can make inexpensive CD backups that will save your original discs from wear and tear. Let DivX help you to enjoy your collection while protecting it from damage.

* It is legal in many countries to make a single backup for personal use only. Local laws may apply. DivX is not intended as a tool for unlicensed copying.

►**Internet distribution**

It has not always been easy to share your videos with friends and colleagues via the Internet. Historically, video transferred over the net has been small, jumpy, blocky and suffering from poor color definition. DivX breaks down the barriers to transferring high quality video via the Internet. Using a regular broadband connection it is possible to transfer the equivalent of a feature length DVD in just a few hours and shorter clips in a matter of minutes.



Quick Start Guide

The OFFICIAL Guide

Quick Start Guide

Introducing VirtualDub

When working with video it is likely that you will require a variety of tools to achieve the best possible results. There are some processes for which dedicated-purpose tools can be invaluable in the world of digital video. Most notable of these tools by far is *VirtualDub*.

VirtualDub is, as its name suggests, a dubbing application. Although it does not incorporate all of the functionality typically associated with commercial video editing applications (such as easy DVD conversion, programmable digital VCR, video transition effects etc.), it specializes in dubbing operations. With it you can cut, splice, dub, and convert video between formats - as well as performing numerous handy filtering effects. VirtualDub is an essential tool in anyone's arsenal - but not only is it powerful, it is also remarkably simple to use.

In this quick start guide you will be learning the basics of VirtualDub while using it to create your very first DivX videos. Until you have completed this part of the guide you should simply follow the instructions precisely and resist the temptation to explore other features.

Before we begin you will need to download the VirtualDub software as it is not included with this guide. Don't panic! One of VirtualDub's many virtues is that it is an open-source project and available at no cost (although the author does accept donations). The download is also very small, you can find it at this address:

http://virtualdub.sourceforge.net

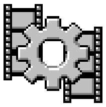
Download the VirtualDub .zip file and extract its entire contents to a new folder on your hard drive. No formal installation procedure is required. For this guide we shall be using VirtualDub version 1.5.4, however you will likely find any later version to be almost identical in terms of the interface.

You will also need a pre-existing sample .avi or .mpg file to work with in this guide. We recommend a short file taken from a reliable source (such as CD-ROM) that is unlikely to be corrupt or otherwise damaged.

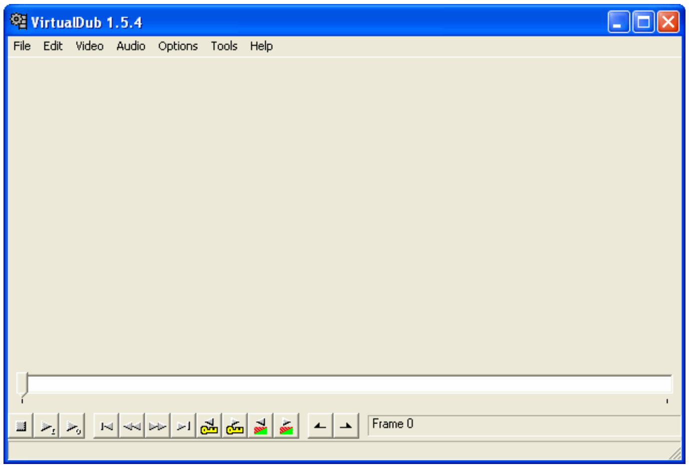
Your first DivX

To begin we will use VirtualDub to open an MPEG 1 video file and convert it to DivX format. For the purposes of simplicity our video will not include an audio track at this point.

1. Load VirtualDub by double-clicking on the VirtualDub program icon.



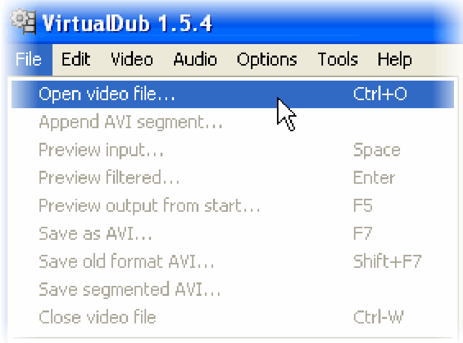
VirtualDub Program Icon



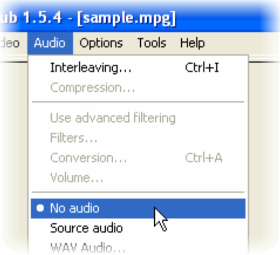
VirtualDub Main Window

In the VirtualDub main window (from top to bottom): *Program menus, Video display area (currently empty), Timeline and seek control, Timeline control buttons, Tool tips bar*.

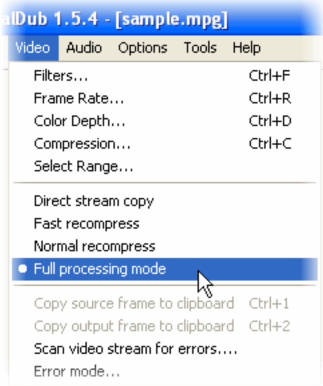
From left to right the timeline control buttons are: *Stop, Play input, Play output, Go to start, Backward, Forward, Go to end, Previous key-frame, Next key-frame, Find last scene change, Find next scene change, Set mark-in, Set mark-out*.


VirtualDub 1.5.4
File Edit Video Audio Options Tools Help
Open video file... Ctrl+O
Append AVI segment...
Preview input... Space
Preview filtered... Enter
Preview output from start... F5
Save as AVI... F7
Save old format AVI... Shift+F7
Save segmented AVI...
Close video file Ctrl-W

2. From the *File* menu choose *Open* and select your sample file.



3. From the *Audio* menu set *No Audio*. This instructs VirtualDub that even if the source file contains audio data we do not want to process it or copy it into our output file.

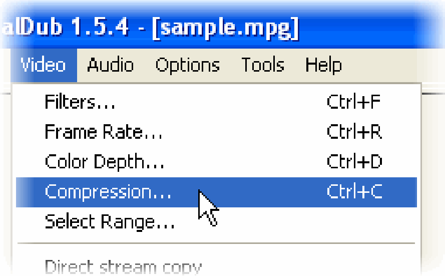


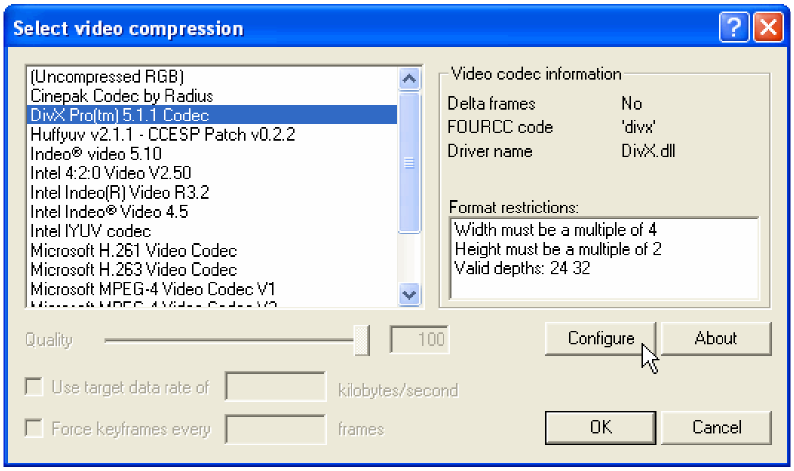
4. From the *Video* menu set *Full Processing Mode*. In this mode VirtualDub allows us both to recompress the video (here we will be going from MPEG1 to DivX) and also apply any filters should we choose to do so.

Because we will not be applying filtering here *Normal Recompress* might give us better performance. *Fast recompress* gives the best performance but may not work in all circumstances.

For the present time leave the video mode set to *Full Processing Mode*.

5. Now we want to configure DivX as the video compressor. Select the *Compression* option from the *Video* menu to open the compressors list.





(Uncompressed RGB)
Cinepak Codec by Radius
DivX Pro(tm) 5.1.1 Codec
Huffyuv v2.1.1 - CCESP Patch v0.2.2
Indeo® video 5.10
Intel 4:2:0 Video V2.50
Intel Indeo(R) Video R3.2
Intel Indeo® Video 4.5
Intel IYUV codec
Microsoft H.261 Video Codec
Microsoft H.263 Video Codec
Microsoft MPEG-4 Video Codec V1

Video codec information

Delta frames	No
FOURCC code	'divx'
Driver name	DivX.dll

Format restrictions:
Width must be a multiple of 4
Height must be a multiple of 2
Valid depths: 24 32

Quality 100

□ Use target data rate of kilobytes/second

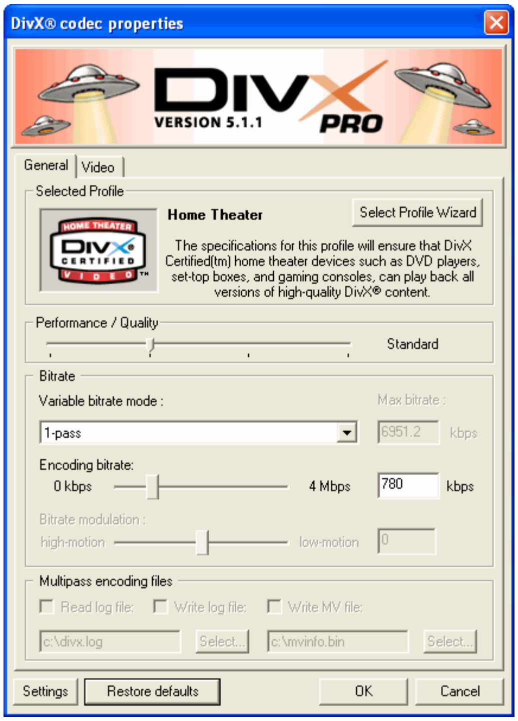
□ Force keyframes every frames

Select *DivX Pro(tm) 5.1 Codec* and click the *Configure* button to bring up the encoder configuration dialogue.

6. Congratulations!

If you have made it this far then this will be the first time you have seen the DivX encoder. At this point we simply want to configure the encoder with the default settings using the *Restore Defaults,* button but you might want to take this opportunity to explore the encoder configuration dialogue before doing so.

When you are ready, click *Restore Defaults* and then *OK*.



Selected Profile: Home Theater

The specifications for this profile will ensure that DivX Certified(tm) home theater devices such as DVD players, set-top boxes, and gaming consoles, can play back all versions of high-quality DivX® content.

Performance / Quality: Standard

Bitrate

Variable bitrate mode : 1-pass

Max bitrate : 6951.2 kbps

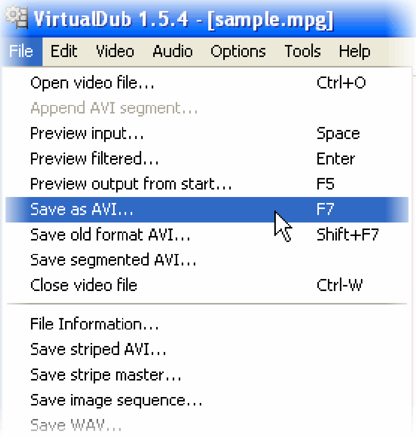
Encoding bitrate: 0 kbps – 4 Mbps: 780 kbps

Bitrate modulation : high-motion – low-motion: 0

Multipass encoding files

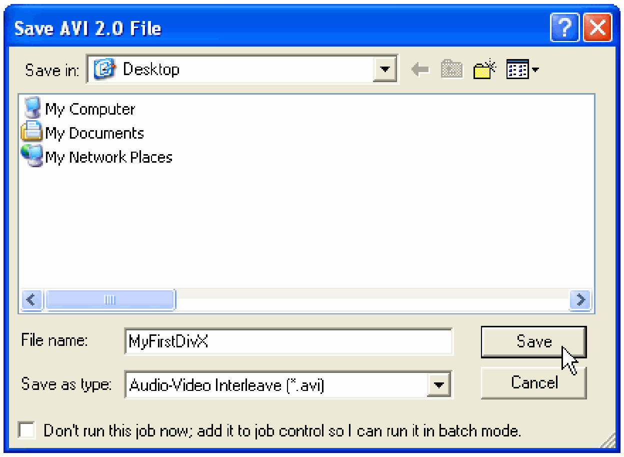
□ Read log file: □ Write log file: □ Write MV file:

c:\divx.log Select... c:\mvinfo.bin Select...



7. All that remains is to instruct VirtualDub to save the AVI file. When you do this VirtualDub will begin processing the video and writing the encoded data to file.

Enter a suitable filename and click *Save* to commence processing.



Save in: Desktop

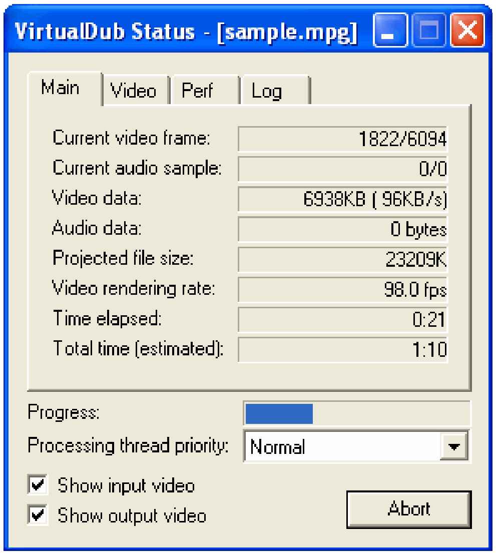
File name: MyFirstDivX

Save as type: Audio-Video Interleave (*.avi)

□ Don't run this job now; add it to job control so I can run it in batch mode.

8. The VirtualDub Status window should pop up and allow you to monitor the encoding progress.

Here amongst other things you can see the progress bar, how much video and audio data has been encoded so far and a projection of the final file size. The projection should be considered a *very* rough estimate because DivX may choose to vary the bitrate throughout the file.



VirtualDub Status - [sample.mpg]

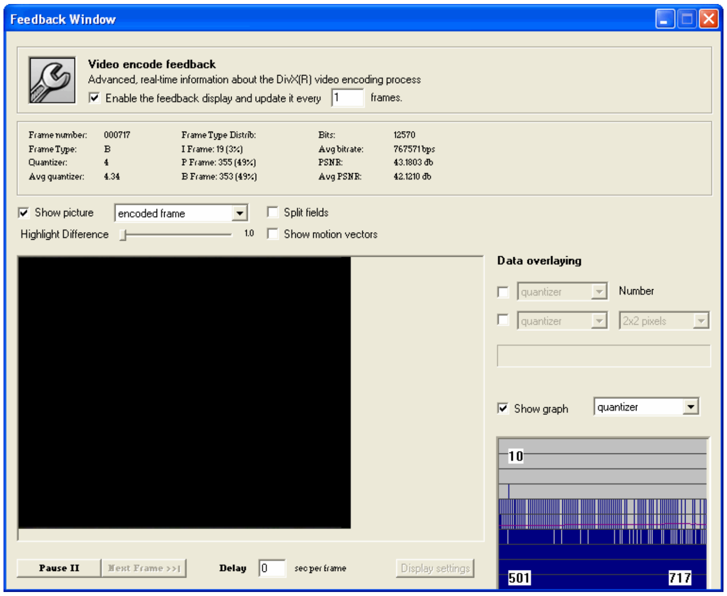
Main | Video | Perf | Log

Current video frame: 1822/6094
Current audio sample: 0/0
Video data: 6938KB (96KB/s)
Audio data: 0 bytes
Projected file size: 23209K
Video rendering rate: 98.0 fps
Time elapsed: 0:21
Total time (estimated): 1:10

Progress:
Processing thread priority: Normal

☑ Show input video
☑ Show output video

If the estimated time is quite long and you would like to work in other applications while encoding set *Processing thread priority* to *lowest*. This will make Windows share more CPU time with other applications but will also result in longer encoding times.



Feedback Window

Video encode feedback
Advanced, real-time information about the DivX(R) video encoding process
☑ Enable the feedback display and update it every 1 frames.

Frame number: 000717
Frame Type: B
Quantizer: 4
Avg quantizer: 4.34

Frame Type Distrib:
I Frame: 19 (3%)
P Frame: 355 (49%)
B Frame: 353 (49%)

Bits: 12570
Avg bitrate: 767571 bps
PSNR: 43.1803 db
Avg PSNR: 42.1210 db

☑ Show picture: encoded frame
□ Split fields
Highlight Difference 1.0
□ Show motion vectors

Data overlaying
□ quantizer Number
□ quantizer 2x2 pixels
☑ Show graph: quantizer

Delay 0 sec per frame

Pro

New to version 5.1 is the *Feedback Window*. This window lets you monitor the inner-workings of the DivX encoder in real-time.

You will see the feedback window unless it is disabled under the *Settings* button in the lower-left hand corner of the encoder configuration dialogue.

You might like to explore some of the display options shown here while your video encodes, but for now avoid altering the *Bitrate, pvLumaFlat,* and *pvLumaTexture* options in the bar at the very bottom of the window.

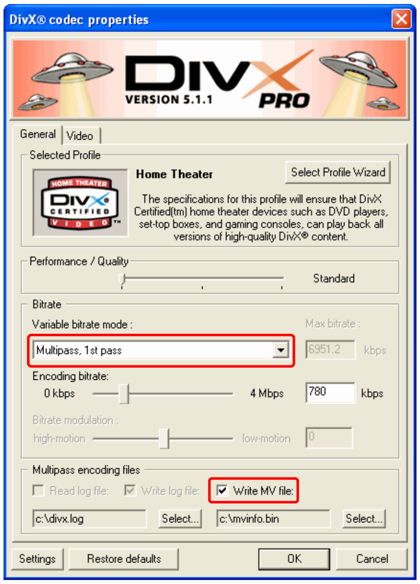
With regards to performance the feedback window may reduce the rate at which your video is encoded. To reduce overhead increase the update interval so that the feedback display is not updated every single frame.

Once your encoding has completed both the feedback and status windows will close automatically. You should then be able to play your first DivX movie using *The DivX Player* (bundled free with DivX) or another media player application on your system.

If you would like *The DivX Player* to open your file by default as opposed to any other player you can rename the AVI file after encoding so that the file extension reads “*.divx*“ instead of “*.avi*”. *The DivX Player* automatically registers the “*.divx*” file extension when it is installed.

Your first Multipass

While 1-pass mode is fast and particularly suited to certain situations, Multipass encoding offers more consistent quality by allowing the encoder to first analyze the entire video before encoding it.



Selected Profile: Home Theater

The specifications for this profile will ensure that DivX Certified(tm) home theater devices such as DVD players, set-top boxes, and gaming consoles, can play back all versions of high-quality DivX® content.

Performance / Quality: Standard

Bitrate
Variable bitrate mode : Multipass, 1st pass
Max bitrate : 6951.2 kbps
Encoding bitrate: 0 kbps – 4 Mbps: 780 kbps
Bitrate modulation : high-motion – low-motion: 0

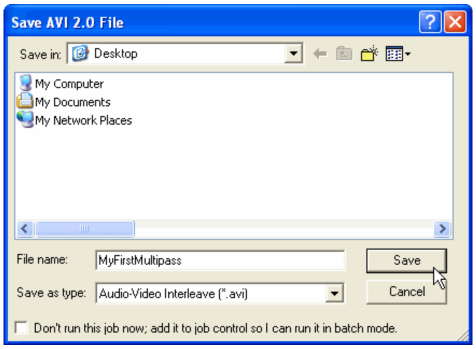
Multipass encoding files
☐ Read log file: ☑ Write log file: ☑ Write MV file:
c:\divx.log Select...
c:\mvinfo.bin Select...

1. Load VirtualDub and configure it as you did in *Your first DivX* steps 1 through 5 but this time set the *Variable bitrate mode* to “*Multipass, 1st pass*”.

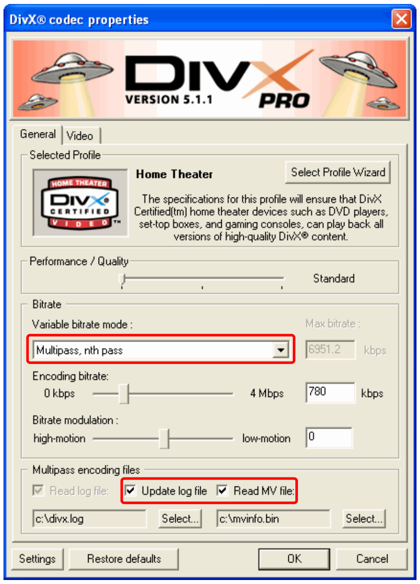
When you save a video in *Multipass, 1st pass* mode the encoder will generate a log file that contains an analysis of the video but no actual video output will be generated.

Enabling *Write MV file* allows the encoder to make use of a technique that accelerates successive Multipass passes.

2. Save your AVI file using a suitable file name. You don’t need to use a separate file name for every pass that you perform, so as before simply choose a name that fits the video.



Save in: Desktop
File name: MyFirstMultipass
Save as type: Audio-Video Interleave (*.avi)
☐ Don't run this job now; add it to job control so I can run it in batch mode.



Selected Profile: Home Theater

The specifications for this profile will ensure that DivX Certified(tm) home theater devices such as DVD players, set-top boxes, and gaming consoles, can play back all versions of high-quality DivX® content.

Performance / Quality: Standard

Bitrate
Variable bitrate mode : Multipass, nth pass
Max bitrate : 6951.2 kbps
Encoding bitrate: 0 kbps – 4 Mbps: 780 kbps
Bitrate modulation : high-motion – low-motion: 0

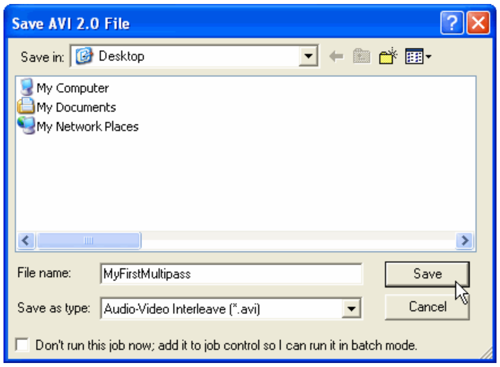
Multipass encoding files
☑ Read log file: ☑ Update log file ☑ Read MV file:
c:\divx.log Select...
c:\mvinfo.bin Select...

3. Without closing VirtualDub re-configure the encoder by setting *Variable bitrate mode* to “*Multipass, nth pass*” and ensure “*Update log file*” and “*Read MV file*” are enabled.

When you save a video in *Multipass, nth pass* mode the encoder will use the analysis log from the previous pass to help it encode the video with a consistent quality but still targeting the bitrate you have specified.

Because we have selected *Update log file* the encoder will record an analysis of this nth pass with respect to the decisions it makes based upon the previous analysis from the log file. In this way we can feed the analysis log from one nth pass into successive nth passes and actually refine the encoding process towards the optimal quality level.

Enabling *Read MV file* allows the encoder to make use of the accelerant technique previously enabled for this pass.



Save AVI 2.0 File

Save in: Desktop

File name: MyFirstMultipass

Save as type: Audio-Video Interleave (*.avi)

☐ Don't run this job now; add it to job control so I can run it in batch mode.

4. Save the AVI file using the same file name as you used for the 1st pass. It is normal practice in Multipass encoding to overwrite the output from the previous pass every time you save.

After an nth pass encoding has been completed you can play the AVI file to see the output.

5. Without changing any settings try saving the AVI file one more time, again choosing to overwrite the previous file. The encoder will use the log file we updated in the last nth pass to create an even higher quality video.

Although you might be tempted to continue running nth pass after nth pass there is of course a limit on the quality attainable at any given bitrate. Typically 98-99% of the optimal quality will be realized after 3 passes (1st, nth, nth) or less.

Notes on audio

DivX is a video codec. As such it does not handle audio in any way and we will not cover the very broad topic of audio in detail. However, you will almost certainly want to include audio alongside DivX video and so we will briefly cover a few important points.

1. There are many audio formats and codecs that can be used in combination with an AVI container, each with different properties and features. DivX AVI files generally use Constant Bitrate MPEG1-Layer 3 Audio, or *CBR MP3* for short. CBR MP3 is the recommended audio format for DivX video and has a number of benefits:

►MP3 audio is supported natively by most operating systems

►MP3 audio is supported by DivX Certified hardware players

►CBR MP3 audio is least likely to introduce de-synchronization between audio and video.

The MP3 codec distributed with Windows is the *Fraunhofer IIS MPEG Layer-3 Codec (advanced)*. This codec is capable of decoding high-bitrate MP3 format audio but is limited to low-bitrate encoding.

VirtualDub works only with audio codecs designed for Microsoft's *Audio Compression Manager*, or *ACM*, and there are two high-bitrate enabled ACM MP3 codecs that are commonly used for DivX encoding.

<u>LAME ACM</u>

Builds of the open-source LAME project have included an ACM version of the encoder since version 3.92. LAME binaries are widely regarded as some of the highest quality MP3 encoders available.

For information regarding the LAME project see:

http://lame.sourceforge.net/

For encoder binaries see:

http://mitiok.cjb.net/

Radium MP3

The Radium MP3 codec is a modified* version of the *Fraunhofer IIS MPEG Layer-3 Codec (professional)*. It is easy to install and simply replaces the *Fraunhofer IIS MPEG Layer-3 Codec (advanced)* that ships with Windows.

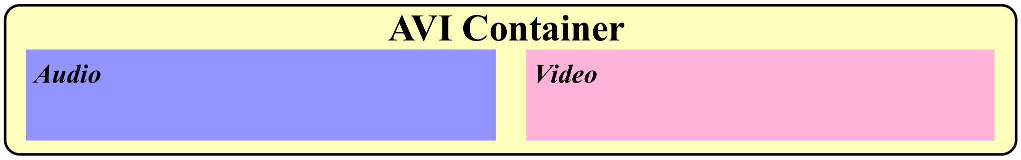
The Radium MP3 codec is available from many websites, enter the keywords "*Radium MP3 download*" into Google:

http://www.google.com

As a rough guide: When encoding to CBR MP3 128 kbps is generally considered *close to* CD quality, although bitrates from 112 kbps and upwards are normally acceptable for stereo sound. For mono sound halve these bitrates.

2. Audio contained in an AVI file must be correctly interleaved for stable playback performance—particularly when content is to be stored on optical media (such as CD-R or DVD-R), or other media with high seek times.

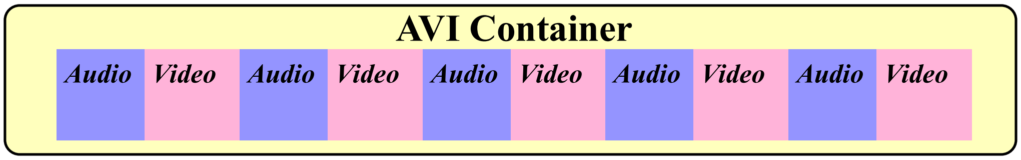
Consider this AVI container with one video and one audio stream:



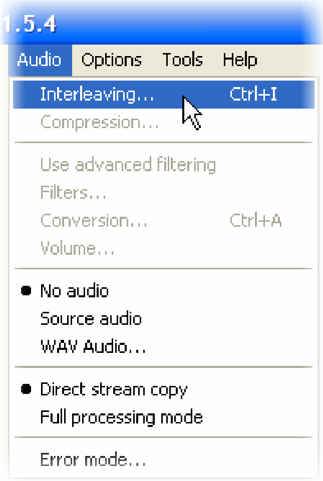
This represents the worst possible situation with regards to interleaving. During playback the system will need to seek back and forth within the file continuously in order to load the audio associated with the video as time progresses. If the viewer were to jump to a specific point in the video it might be hard for the media player to quickly locate the corresponding point in the audio stream, causing an undesirably high seek recovery time.

* The modifications performed by the Radium group are not legal and you should be aware that although popular the Radium MP3 codec is technically illegal.

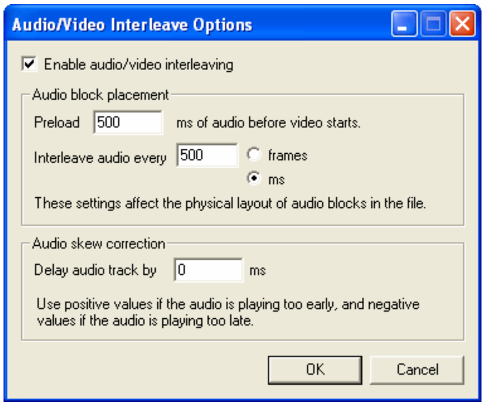
Now consider the corresponding correctly interleaved file:



Here audio chunks are interleaved within the file so as they fall closely alongside the related sequence of video. As playback progresses seeking within the file is minimized, and in fact with appropriate buffering as provided by most devices seeking will not even take place. This leads to smoother playback performance. Because audio chunks are stored alongside the corresponding video chunks if a viewer skips to a particular point in the video the media player will likely locate the associated audio and resume playback more promptly.



Using VirtualDub, you can access the *Interleaving* options from the *Audio* menu.



Audio/Video Interleave Options

☑ Enable audio/video interleaving

Audio block placement

Preload 500 ms of audio before video starts.

Interleave audio every 500 ○ frames ● ms

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

Audio skew correction

Delay audio track by 0 ms

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

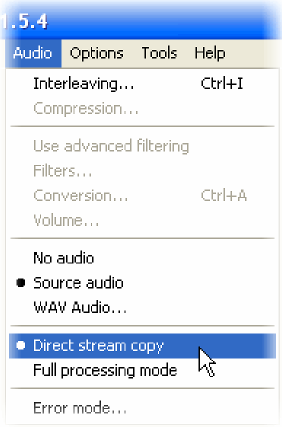
OK Cancel

Ensure "*Enable audio/video interleaving*" is enabled. The recommended audio preload is the default 500 ms. The recommended audio interleave is every 500 ms.

Note that the interleave options also present a method of adjusting audio/ video sync by adjusting the "*Audio skew correction*" value, either positively or negatively in milliseconds.

For example: If the audio was running two seconds behind the video a value of –2000 ms would correct for the delay. Conversely if the audio was running one and a half seconds ahead of the video a value of 1500 ms would be required.

3. VirtualDub contains two modes for processing audio, these are “*Direct stream copy*” and “*Full processing mode*”.



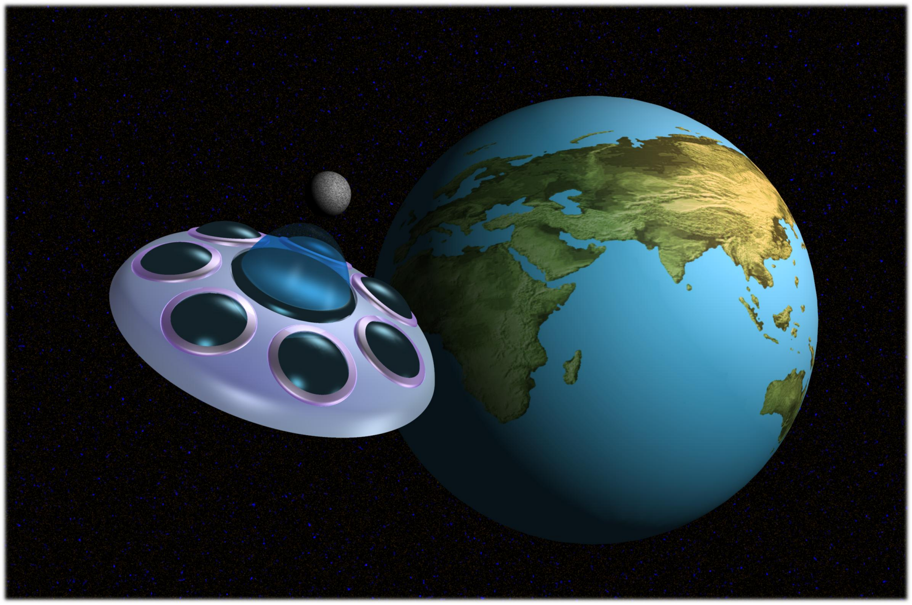
In *direct stream copy* mode no processing (other than interleaving) is performed on the audio stream. If it were desirable when editing an existing AVI file that only the video stream should be manipulated the audio mode would be configured as *Direct stream copy* in conjunction with “*Source audio*”. When the new AVI file was saved the original audio stream would be copied unmodified from the source.

In *Full processing mode* the given audio stream is decompressed and fully processed when saving, including the application of any filters or compressors you have selected.

Both *Direct stream copy* and *Full processing mode* have exactly the same semantics when applied to the video stream as opposed to the audio stream.

Tip: Using the *Audio skew correction* (see note 2) in combination with *Direct Stream Copy* mode for both audio and video streams allows you to correct the audio/video sync of any pre-existing AVI file without requiring recompression of either stream.

4. When performing Multipass encoding a lot of time can be saved if audio processing is enabled only during the last pass you intend to save. It is wasteful processing audio on every pass as each time the previous AVI file is overwritten, including any audio track it may have contained.



Forward

Forward

General concepts

In order to understand all of the encoder features that will be covered in the next section it is first necessary to cover the core concepts behind the MPEG 4 Video Standard upon which DivX 5.1 is based. Lets begin with the way videos are composed.

Frames

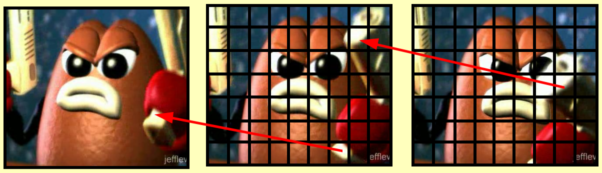
At its most basic a video is a series of pictures shown one after the other in quick succession. When the pictures are played fast enough the image appears to move. Each picture in a video is called a *frame* and the speed at which they are shown is the *frame rate,* given in *frames per second*, or *fps*.

Frame 1 Frame 2 Frame 3

If we were to slow a video down to such an extent that it were possible to see individual frames it would become evident there is generally little change in the picture between adjacent frames. Frames in sequence would appear very similar to each other, objects in each consecutive frame moving position only slightly. DivX takes advantage of this behavior as a means to compress the data required to store a video.

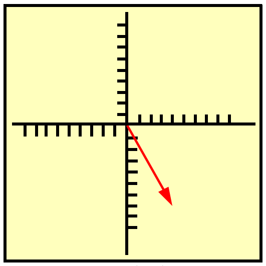
Macroblocks and motion

The encoder divides a frame into a grid of blocks, called *macroblocks*, and attempts to track the motion of each macroblock in the current frame back to a matching area of picture in the previous frame. This process is known as the *motion search*.



Frame 1 Frame 2 Frame 3

When the encoder finds a matching area for a block in the previous frame its position is recorded by use of a *vector*. A vector is simply a numerical representation of direction and magnitude. For example the vector *(4, -7)* might mean right 4 units, down 7 units.

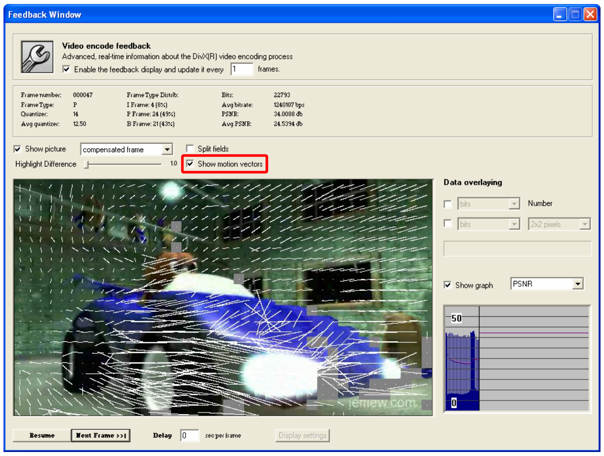


Vector (4,-7)

Because these vectors happen to represent the movement of a macroblock they are called *motion vectors*. In the illustrations the red arrows represent the motion vectors.

Storing the motion vector of each block from frame-to-frame allows DivX to recreate much of each new frame from the last using far less data than if each block were stored as a complete image. The process is similar to cutting up one frame into little pieces, then maneuvering them like a make-shift jigsaw puzzle using motion vectors in order to recreate the next frame. Of course the resulting picture is not perfect and DivX has to store some additional data to correct for the difference (known as the *residual*). Even so, recreating the picture using motion vectors and residual consumes far less storage than recording the entire image for every single frame.

The process of tracking blocks back to matching areas in the previous frame (known as the *reference* frame) is called the *motion search,* or *motion estimation*. The frame recreated from a reference frame by application of motion estimation is known as the *compensated* frame.



Feedback Window

Video encode feedback

Advanced, real-time information about the DivX(R) video encoding process

☑ Enable the feedback display and update it every 1 frames.

Frame number:	000047	Frame Type Distrib:		Bits:	22793
Frame Type:	P	I Frame: 4 (8%)		Avg bitrate:	1240107 bps
Quantizer:	14	P Frame: 24 (49%)		PSNR:	34.0088 db
Avg quantizer:	12.50	B Frame: 21 (43%)		Avg PSNR:	24.5394 db

☑ Show picture: compensated frame

☐ Split fields

Highlight Difference 1.0

☑ Show motion vectors

Data overlaying

☐ bits — Number

☐ bits — 2x2 pixels

☑ Show graph: PSNR

Pro

You can see the motion search in action using the encoder feedback window.

While encoding set *Show picture* to *Compensated frame* and turn on *Show Motion Vectors*.

The white lines overlaid on the compensated frame represent motion vectors returned by the motion search, tracking each block back to a matching area in the reference frame.

Where blocks are grey in the compensated frame the motion search failed to find a suitable match in the reference frame. These blocks will be encoded as image data as opposed to motion vectors.

Intra-frames and Predicted-frames

The DivX encoder makes use of three frame types during encoding. These are: *Intra-frames*, *Predicted-frames* and *Bi-directional-frames*. More commonly these are known as *I*, *P* and *B* frames. B-frames are discussed later in the guide as an advanced topic.

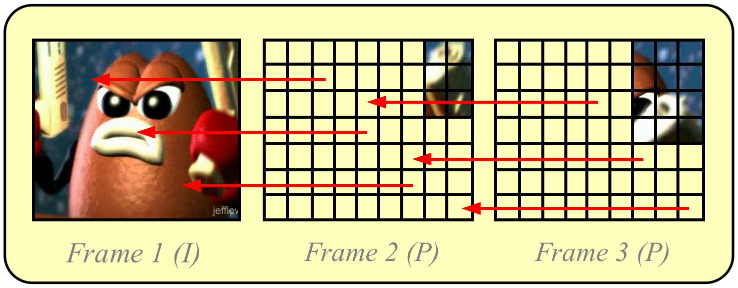
An intra-frame is one in which all of the macroblocks are stored as images rather than as motion vectors. As noted previously, recording the image of a block is the most expensive method in terms of storage and hence I-frames are the most expensive type of frame. Fewer I-frames in the video generally translates to better compression and I-frames are normally used by the encoder only when too few blocks can be tracked from the reference frame by the motion search algorithm.

An intra-frame

I-frames serve a very important purpose: All of the blocks in an I frame are stored as images, thus decoding an I frame reveals a complete picture without dependency on reference frames. For this reason I frames are also known as *key-frames*, and they are the only type of frame completely independent of all others.

A predicted frame is one that can contain forward-predicted blocks—in other words blocks that have been predicted via a motion vector from the previous frame by the motion search. It is normally unlikely that *all* of the blocks in a P-frame can be predicted, and where blocks can not be tracked from the previous frame an intra-block is used in its place, similar to those found in an I-frame. Because P-frames reconstruct much of the frame by applying motion vectors to the previous frame (*motion compensation*) they are far less expensive in terms of storage than I-frames.

One or more P-frames may follow an I-frame:



A greater ratio of P-frames to I-frames leads to a higher compression ratio.

Quantizers

DivX uses a technique called *quantization* to control the accuracy of the image data it stores. Quantizers are similar to the denominator in an expression such as:

$$\mathit{Result} = \frac{\mathit{Data}}{\mathit{Quantizer}}$$

… and the co-efficient in the inverse quantization:

$$\mathit{Data} = \mathrm{Quantizer} * \mathit{Result}$$

By maintaining some fixed Quantizer value (greater than one), it is possible to reduce the value of Data using the first expression, and then invert the process using the second expression.

However, after quantization the Result is normally rounded down to the nearest whole number. This has some interesting consequences for quantization with regards to image quality.

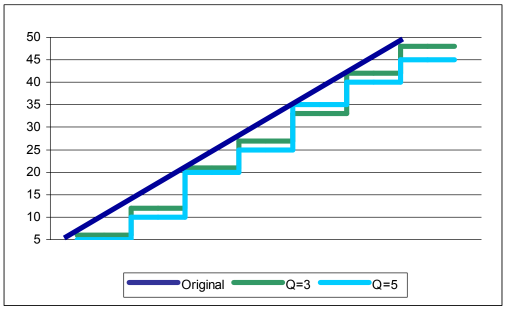
Examine the following two tables. The first quantizes a series of number with a fixed quantizer of 3. The second quantizes the same series of numbers but with a fixed quantizer of 5. Both tables round down all of the results to the nearest whole number then perform the inverse quantization and record the error from the original data series.

Quantizer = 3

Data series	7	14	21	28	35	42	49
Quantization data	2.33	4.66	7	9.33	11.66	14	16.33
Round-down result	2	4	7	9	11	14	16
Inverse quantization	6	12	21	27	33	42	48
Error	**14%**	**14%**	**0%**	**4%**	**6%**	**0%**	**2%**

Quantizer = 5

Data series	7	14	21	28	35	42	49
Quantization data	1.4	2.8	4.2	5.6	7	8.4	9.8
Round-down result	1	2	4	5	7	8	9
Inverse quantization	5	10	20	25	35	40	45
Error	**29%**	**29%**	**5%**	**11%**	**0%**	**5%**	**8%**



It soon becomes clear that higher quantizers introduce larger errors due to rounding before inverse quantization; thus when encoding lower frame quantizers lead to higher quality images.

Plotting results from the tables with respect to the original series illustrates lower quantizers recreating a more accurate signal.

Although lower quantizers give higher quality images they also produce larger file sizes. This is because the range of results encompassed by lower quantizers is larger than that of higher quantizers.

In the tables, results of data quantized at Q=3 could potentially include any value of interval 3 (*0, 3, 6, 9…)*. Data quantized at Q=5 has a larger interval (*0, 5, 10, 15…)*. Within a fixed range, I.e. 1 - 100, there are less possible values for Q=5 results than for Q=3 results. This leads to better compression for Q=5, and higher quantizers in general.

Under most modes the DivX encoder will fully manage control of the quantizer for you, although it is possible to specify a quantizer for encoding if you want to.

A key to the guide

From this point forward the guide will describe the features available in the DivX encoder. Throughout the rest of the guide an easy to follow iconic key will be used.

Features available only in the DivX Pro encoder will be marked **Pro**

The following color-coded sections will appear where appropriate to a feature:

This indicates a feature is not compatible with the DivX Certified Program. Use of the feature will mean your video will not be compatible with DivX Certified devices.

This indicates that use of the feature may have implications for DivX Certified devices.

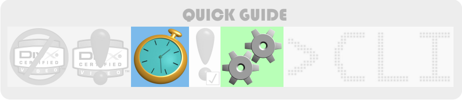
This indicates there are performance tips associated with the feature.

This indicates that use of the feature may have implications upon other options.

This indicates that there are usage notes associated with the feature.

>CLI This indicates there are CLI parameters associated with the feature.

Additionally, at the beginning of each feature description you will see the *Quick Guide* bar.



Here the Quick Guide bar indicates that there are performance tips and usage notes associated with the feature. All other icons are grayed out.

Bitrate mode

Bitrate mode

What is the bitrate?

In a computer the minimum unit of storage is one bit. A bit can represent two values, 0 or 1, and hence computers use the *binary* number system (base 2) as opposed to the *denary* system (base 10, 0-9) that humans use. When several bits are combined they can be used to store more complex data.

For practical purposes the smallest unit of storage software normally works with is actually one *byte*. A byte is composed of eight bits.

To avoid confusion when working with quantities of bits and bytes the following tables should be adhered to:

Bits

One bit	=	Smallest unit
One kilobit	=	1,000 bits
One megabit	=	1,000 kilobits
	or	1,000,000 bits

Bytes

One byte	=	8 bits
One kilobyte	=	1,024 bytes
One megabyte	=	1,024 kilobytes
	or	1,048,576 bytes
	or	8,388,608 bits

The bitrate describes how many thousands of bits per second *on average* the encoder should aim to spend when encoding the video. Given a video of any fixed duration encoding at a higher bitrate will lead to larger file sizes (and better quality video), while conversely encoding at a lower bitrate will lead to smaller file sizes (but lower quality video).

A key aim when encoding is to achieve a desired file size; after all the prime reason for video compression is to reduce storage requirements. Given the length of a video and a target file size it is possible to determine a suitable bitrate in just five simple calculations.

The following example demonstrates the bitrate calculation for encoding 60 minutes of video with 128 kbps MP3 audio for a 700MB CD-R. The algorithm is applicable to any video or target file size.

Bitrate calculation

60 minute video with 128 Kbps MP3 audio for 700MB CD-R

1. Calculate how many seconds long the video is

	60	*The number of minutes*
x	60	*The number of seconds in one minute*
=	**3,600 seconds duration**	

2. Calculate how many bits in total there are available

	700	*The number of megabytes*
x	1,048,576	*The number of bytes in one megabyte*
x	8	*The number of bits in one byte*
=	**5,872,025,600 bits available**	

3. Calculate how many bits will be consumed by audio

	128	*The number of kilobits per second for audio*
x	1,000	*The number of bits in a kilobit*
x	3,600	*The duration of the video in seconds from step 1*
=	**460,800,000 bits consumed by audio**	

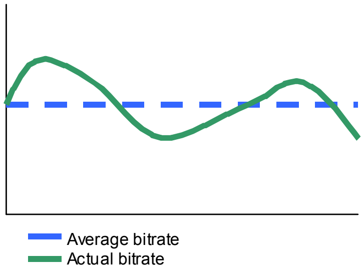
4. Calculate the bits remaining for video

	5,872,025,600	*The number of bits available from step 2*
-	460,800,000	*The number of bits consumed by audio from step 3*
=	**5,411,225,600 bits remaining for video**	

5. Calculate the video bitrate

	5,411,225,600	*The number of bits remaining for video*
/	1,000	*The number of bits in a kilobit*
/	3,600	*The duration of the video in seconds from step 1*
=	**1,503 kbps video bitrate**	

Note: Considering the AVI container itself requires some bits we might use a slightly lower bitrate to ensure hitting our target, e.g. 1,501 kbps



The bitrate calculated is the *average* bitrate for the video. DivX may choose to vary the actual bitrate throughout the video as it encodes, a technique known as *variable bitrate encoding*.

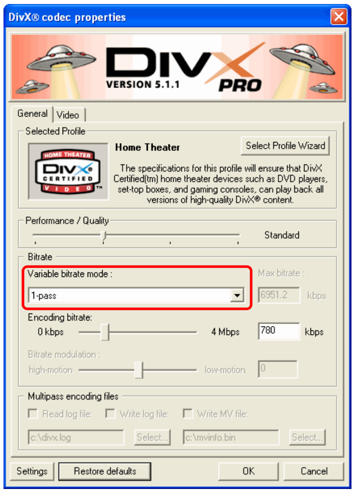
Since not all sequences in a video are equally complex in terms of image and motion, it is impossible to maintain a constant bitrate throughout the file without introducing substantial quality differences from frame-to-frame. Instead, DivX distributes the total bandwidth as appropriate with the goal of maintaining a consistent overall quality that permits it to meet the average bitrate. It does this via the *rate control*, which will be introduced shortly when multipass mode is covered in more detail.

1-Pass

In 1-Pass mode the encoder will output a working DivX video stream as it receives the source video.

This is both the easiest and fastest method of encoding DivX video while targeting an average bitrate, and is particularly suited to video capture applications where it is impossible to recreate the source video identically more than once. However, 1-Pass mode is the least consistent of all available modes in terms of video quality.

After running 1-Pass encoding your DivX file can be viewed immediately.



DivX® codec properties

DivX VERSION 5.1.1 PRO

General | Video

Selected Profile

Home Theater — Select Profile Wizard

The specifications for this profile will ensure that DivX Certified(tm) home theater devices such as DVD players, set-top boxes, and gaming consoles, can play back all versions of high-quality DivX® content.

Performance / Quality: Standard

Bitrate

Variable bitrate mode : 1-pass

Max bitrate : 6951.2 kbps

Encoding bitrate: 0 kbps — 4 Mbps — 780 kbps

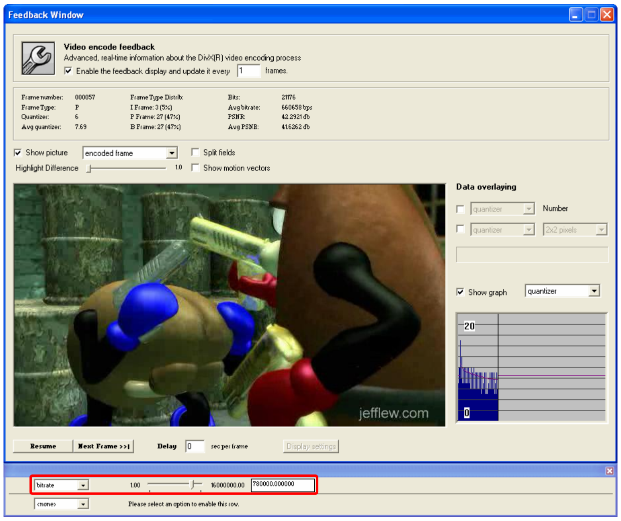
Bitrate modulation : high-motion — low-motion — 0

Multipass encoding files

□ Read log file: □ Write log file: □ Write MV file:

c:\divx.log Select... c:\mvinfo.bin Select...

Settings | Restore defaults | OK | Cancel



Feedback Window

Video encode feedback

Advanced, real-time information about the DivX(R) video encoding process

☑ Enable the feedback display and update it every 1 frames.

Frame number:	000057	Frame Type Distrib:		Bits:	21176
Frame Type:	P	I Frame: 3 (5%)		Avg bitrate:	660658 bps
Quantizer:	6	P Frame: 27 (47%)		PSNR:	42.2921 db
Avg quantizer:	7.69	B Frame: 27 (47%)		Avg PSNR:	41.6262 db

☑ Show picture: encoded frame □ Split fields

Highlight Difference 1.0 □ Show motion vectors

Data overlaying

□ quantizer Number

□ quantizer 2x2 pixels

☑ Show graph: quantizer

Resume | Next Frame >>| | Delay 0 sec per frame | Display settings

bitrate 1.00 16000000.00 780000.000000

<none> Please select an option to enable this row.

Pro

During 1-Pass encoding it is possible to modify the bitrate in real-time by selecting the *bitrate* option from the bottom panel of the encoder feedback window.

This feature is likely to be of particular interest when capturing from a live source. For example, if capturing from television you might want to reduce the bitrate during advertisements.

When you set the bitrate mode to *1-Pass* you will not be able to use the *Bitrate modulation* control or the EKG application.

1-Pass mode is particularly useful when you are capturing from a live source and you desire control over the bitrate/file size.

When capturing from a live source and attempting to maintain a consistent quality throughout the video you should use *1-Pass Quality Based* mode.

When working from static sources (for example re-processing existing stored video) you will achieve more consistent quality by using *Multipass* mode.

>CLI

Bitrate mode is one of:

-bv1 <*bitrate*>	1-Pass
-b1q <*quantizer*>	1-Pass, quality-based
-bvn1 <*bitrate*>	Multipass, 1st pass
-bvnn <*bitrate*>	Multipass, nth pass

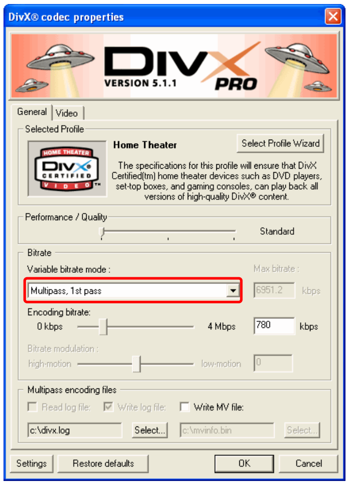
Bitrate is an integer value between 0 and 16,000 specifying the number of kilobits per second.

Quantizer is floating point value between 1.0 and 31.0 specifying the fixed quantizer to be used.

Multipass

In Multipass mode you are required to run the video through the encoder two or more times. On every successive pass the video must be identical to that which you used in your first pass.

As discussed in “*Bitrate mode—What is the bitrate?*”, DivX uses a *rate control* algorithm to vary the bitrate throughout the video in order to achieve a consistent quality while targeting the specified average bitrate. To do this effectively DivX has to be able to examine the entire video in order to determine where bandwidth should be allocated before it spends any bits at all.



Selected Profile: **Home Theater**

The specifications for this profile will ensure that DivX Certified(tm) home theater devices such as DVD players, set-top boxes, and gaming consoles, can play back all versions of high-quality DivX® content.

Performance / Quality: Standard

Bitrate
Variable bitrate mode : Multipass, 1st pass
Max bitrate : 6951.2 kbps
Encoding bitrate: 0 kbps – 4 Mbps 780 kbps
Bitrate modulation : high-motion – low-motion 0

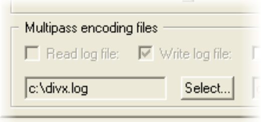
Multipass encoding files
□ Read log file: ☑ Write log file: □ Write MV file:
c:\divx.log Select... c:\mvinfo.bin Select...

Multipass, 1st pass

In 1-Pass mode the rate control strategy is not optimal because the encoder has no foresight into the content of the video. Therefore, in order to meet the specified bitrate the rate control must make extreme changes to the bitrate based upon the short term history as video is received.

Multipass mode is designed to avoid this extremism by allowing the encoder to first analyze the entire video before encoding it. This analysis is performed in *Multipass 1st Pass* mode, and during the first pass no actual video output is written (if you attempt to play your file after running a 1st pass you will see no video).

In Multipass 1st pass mode the encoder writes its analysis to the log file specified in the *Multipass encoding options* area of the configuration dialogue.



Multipass encoding files
□ Read log file: ☑ Write log file:
c:\divx.log Select...

Lets take a brief look at the log contents and what they mean:

```
##map version       8
nframes          5837
timescale       30000
passes              1
 seq  deltaT  type  total_bits  motion_complexity  texture_complexity  modulation
   0       0    I        7712           0.000000            0.251141    1.000000
   2    2400    P        2952           0.035268            0.011100    1.000000
   1   -1200    B        1144           0.015206            0.000000    1.000000
   4    3600    P        6984           0.067981            0.069732    1.000000
   3   -1200    B        2216           0.029374            0.000000    1.000000
   6    3600    P        8160           0.079010            0.082583    1.000000
   5   -1200    B        2264           0.029773            0.000745    1.000000
   8    3600    P       14176           0.083506            0.294013    1.000000
```

Multipass Log file

Log header

##map version <Value>

Value is the version number describing the format of the log file.

nframes <Value>

Value is the total number of frames listed in the log file.

Frame numbering begins from zero.

timescale <Value>

Value is some arbitrary number representing the number of intervals in one second of video.

The timescale is used in combination with *deltaT* (see below).

passes <Value>

Value is the number of passes performed when the log was written.

Frame list

seq <Value>

Value is the sequence number of each frame.

The sequence numbers may appear to be out of order but in fact this is because they are in *display* order. For example, in the section of log shown the sequence reads 0,2,1 for a set of I,P,B frames respectively. The order in which these frames must actually be displayed is I,B,P—or 0,1,2 respectively.

deltaT <Value>

Value is the change in time (based upon the *timescale* value in the header) between frames.

In the section of log shown the first three *deltaT* values are 0,2400,-1200. Recalling that frames are listed in decoding order this actually means the sequence is 0,1200,2400. With respect to the *timescale* value (30000 units) this means the interval between frames (1200 units) is (1200/30000) = 0.04 seconds, and we can derive from this frame-rate is 1/0.04=25 frames per second.

type <Frame_type>

Frame_type is the type of each frame and can be one of ‘I’,’P’ or ‘B’, representing intra-frame, predicted frame or bi-directional frame respectively.

total_bits <Value>

Value is the total number of bits used by the encoder to encode the frame.

motion_complexity <Value>

Value is a floating point number proportional to the number of bits required to encode all motion data in the frame.

This will be zero only for intra-frames.

texture_complexity <Value>

Value is a floating point number proportional to the number of bits required to encode the texture in the frame.

For intra-frames the value is associated with the bits required to encode the image. For predicted and bi-directional frames the value is associated with the bits required to encode the residual (difference between source and motion-compensated frames).

modulation <Value>

Value is a floating point number representing the modulation value set by the EKG application between passes.

Modulation acts as a co-efficient to the frame quantizer selected by the rate control algorithm. By raising the modulation value for a frame above one the quantizer will be increased resulting in lower quality for that frame. Conversely, lowering the value below one will cause the quantizer to be reduced thus raising the quality for the frame.

As an example, if the rate control were to determine quantizer 2 should be used but the modulation for the frame was 1.5 the result would be a quantizer of 2 x 1.5 = 3.

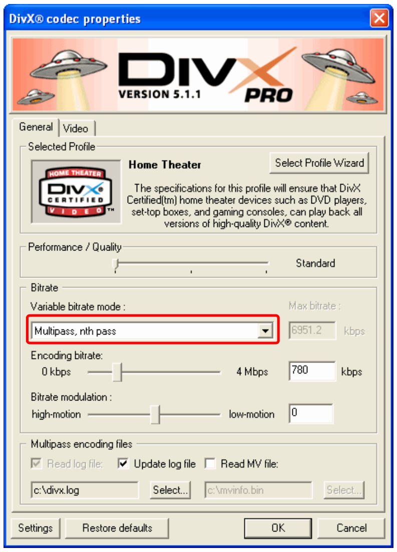
Modulation should not be confused with the *Bitrate modulation* option in the encoder configuration dialogue which is not directly related.

Because DivX Pro includes the EKG tool for graphically displaying and manipulating the contents of the log file it is rarely necessary to edit it manually.

Multipass, nth pass

Once first pass has been completed the same video must be encoded one or more times using *Multipass, nth pass* mode. DivX video is encoded and written to your DivX video file on every nth pass.

In nth pass the encoder will read the log file created by the previous pass and use the analysis to form a more optimal rate control strategy. This will lead to better bandwidth distribution throughout the file, obtaining a consistent quality while meeting the average specified bitrate.



DivX® codec properties

DIVX VERSION 5.1.1 PRO

Selected Profile: Home Theater

The specifications for this profile will ensure that DivX Certified(tm) home theater devices such as DVD players, set-top boxes, and gaming consoles, can play back all versions of high-quality DivX® content.

Performance / Quality: Standard

Variable bitrate mode : Multipass, nth pass

Max bitrate : 6951.2 kbps

Encoding bitrate: 780 kbps

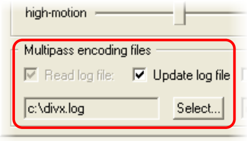
Bitrate modulation : 0

Multipass encoding files: ☑ Read log file: ☑ Update log file □ Read MV file:

c:\divx.log

c:\mvinfo.bin

The previous pass need not necessarily be a 1st pass. During each nth pass results of the new optimized rate control strategy can again be written back into the log file and used to further refine the rate control strategy for a successive nth pass.



Multipass encoding files: ☑ Read log file: ☑ Update log file

c:\divx.log

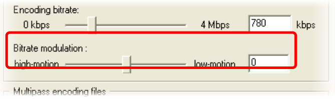
The log file will be updated during an nth pass if you have selected the option to *Update log file* in the *Multipass encoding files* area of the encoder configuration dialogue.

Although it is possible to run as many nth passes as you like, 98-99% of the optimal quality is typically obtained in three passes or less (one 1st pass followed by two nth passes).

Note that if you do not select to update the log file during nth pass then running successive passes will not improve the consistency of video quality because the rate control strategy will not be refined.

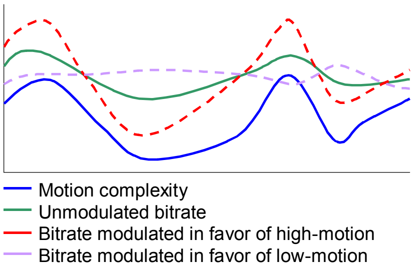
Bitrate modulation

By using the *Bitrate modulation* slider it is possible to bias the distribution of bandwidth towards frames with either low or high motion, allowing you to adapt the rate control to better suit the particular type of video being encoded.



Encoding bitrate: 780 kbps

Bitrate modulation : 0



Moving the slider left of center, or lowering the modulation value, biases bandwidth in favor of high-motion frames and consequently away from low-motion frames. Moving the slider right of center, or raising the modulation value, biases bandwidth in favor of low-motion frames and consequently away from high-motion frames.

When determining the most appropriate bitrate modulation it is important to remember that quality is subjective. The optimal bitrate modulation will vary from video to video and also upon the preference of the viewer and their personal sensitivity to different types of artifacting.

Let us consider some examples where it is beneficial to use the bitrate modulation control.

1. Consider the encoding of an action movie containing several explosion scenes or other fast-paced action and imagine the quality tending to degrade where there is significant motion in the scene. Moving the bitrate modulation slider left of center towards high motion will improve the quality consistency - the degree of adjustment relative to the level of correction required.

2. Once again taking an action movie as an example, imagine encoding at a very limited bitrate in order to create video suitable for Internet distribution. It might be desirable to intentionally bias bandwidth *away* from fast-motion scenes. Explosion scenes and other short fast-motion sequences may consume so much of the total available bandwidth that they actually starve the encoder of free bits as it encodes the majority of the video, leading to low-motion sequences that appear disproportionately degraded. Biasing bandwidth towards low-motion will cause the fast-motion scenes to appear at slightly lower quality, but the quality in the remainder of the video may appear much improved.

3. Consider a scenic video in which the camera is panning across a landscape and zooming in and out but pausing for periods to show features in detail. In this situation we might like to bias bandwidth so that low-motion scenes receive more bits than high-motion scenes. In this way we could enforce that the highest quality is afforded to scenes in the video where the camera focuses on detailed features in the landscape.

Fastest Performance/Quality mode is unavailable during Multipass encoding because it does not perform the motion estimation necessary for the rate-control to work effectively.

If you intend to use either *Slow* or *Slowest* Performance/Quality modes you can substantially decrease your encoding time by encoding all but the final pass using *Standard* Performance/Quality.

Running either a *Standard* pass followed by a *Slow* pass in place of two *Slow* passes or a *Standard* pass followed by a *Slowest* pass in place of two *Slowest* passes reduces encoding time by approximately 50%, while impacting so negligibly on quality as to be virtually indistinguishable.

MV re-use can reduce encoding time for *Standard* passes. See *MV Re-use* for further information.

Disable audio processing on all but the last nth pass to reduce encoding time.

Use Multipass mode to obtain the optimal quality consistency when it is possible to pass an identical source video through the encoder multiple times.

When performing more than 2 passes (1st, nth) in order to be effective in refining the rate control (and hence quality consistency) the option to *Update log file* must be enabled during each nth pass.

Generally 98-99% of the optimal quality is obtained in three passes or less (1st, nth, nth).

It is possible to use a *1-Pass, Quality-based* pass in place of *Multipass 1st pass* if you desire to perform the first pass based upon a fixed frame quantizer. Ensure that *Write log file* is enabled. The XVID encoder performs 1st pass at Q=2. Using this technique it is also possible to capture a live source via *1-Pass, Quality-based* mode and then recompress it, proceeding directly to nth pass.

>CLI

Bitrate mode is one of:

-bv1 <*bitrate*>	1-Pass
-b1q <*quantizer*>	1-Pass, quality-based
-bvn1 <*bitrate*>	Multipass, 1st pass
-bvnn <*bitrate*>	Multipass, nth pass

Bitrate is an integer value between 0 and 16,000 specifying the number of kilobits per second.

Quantizer is floating point value between 1.0 and 31.0 specifying the fixed quantizer to be used.

Log file:

-log <*filename*>

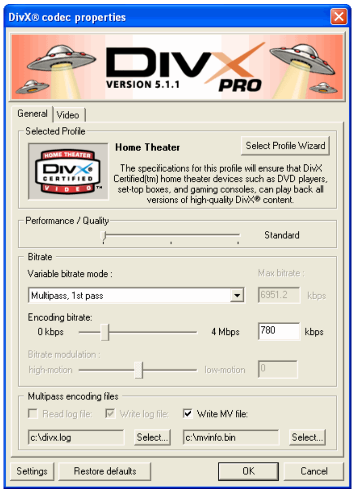
Filename is the log file name, fully qualified with a pre-existing path and enclosed in parentheses. The default is *"C:\DivX.log"*.

MV re-use

During multi-pass encoding it is necessary to pass the same source video through the encoder two or more times.

One of the more computationally intensive parts of encoding is motion estimation, where the encoder tries to find vectors predicting the movement of each block from frame-to-frame via the motion search. Since the source video is similar for every pass, the motion vectors should also be similar.

When enabled, the MV log records the motion vectors found by the motion search during the 1st pass and re-uses them during nth passes. Encoding time is thus reduced because motion estimation need take place only once.


DivX® codec properties

Selected Profile

Home Theater

The specifications for this profile will ensure that DivX Certified(tm) home theater devices such as DVD players, set-top boxes, and gaming consoles, can play back all versions of high-quality DivX® content.

Performance / Quality: Standard

Bitrate

Variable bitrate mode : Multipass, 1st pass

Max bitrate : 6951.2 kbps

Encoding bitrate: 0 kbps – 4 Mbps 780 kbps

Bitrate modulation : high-motion – low-motion 0

Multipass encoding files

□ Read log file: ☑ Write log file: ☑ Write MV file:

c:\divx.log Select... c:\mvinfo.bin Select...

For MV re-use, *Write MV file* must be enabled during the *Multipass, 1st pass*, and *Read MV file* during each *Multipass, nth pass*.

After the 1st pass it is not legal to change any encoder option while *Read MV file* remains enabled. If you wish to change options between passes you must disable the MV file at that point.

MV re-use can offer reduced encoding times under *Standard* Performance/ Quality mode, but marginally reduces video quality.

Where time is not a priority best results are seen when no MV re-use is enabled.

MV re-use has less significant effect on encoding duration when bi-directional encoding is enabled.

>CLI

MV re-use:

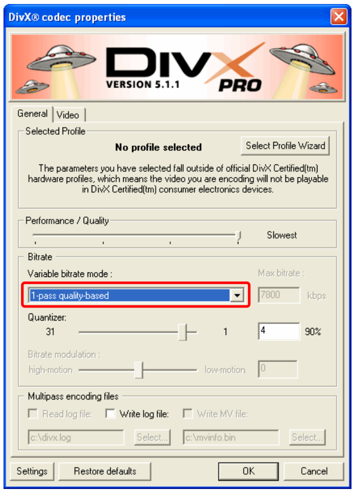
-mv <*filename*>

Filename is the MV log file name, fully qualified with a pre-existing path and enclosed in parentheses. The default is *"C:\mvinfo.bin"*.

1-Pass, Quality-based

QUICK GUIDE

In 1-Pass quality-based mode the encoder will output a working DivX video stream as it receives the source video, similar to in 1-Pass mode except that you specify the quality rather than the target bitrate.



DivX® codec properties

DivX VERSION 5.1.1 PRO

General | Video

Selected Profile

No profile selected [Select Profile Wizard]

The parameters you have selected fall outside of official DivX Certified(tm) hardware profiles, which means the video you are encoding will not be playable in DivX Certified(tm) consumer electronics devices.

Performance / Quality: Slowest

Bitrate

Variable bitrate mode : 1-pass quality-based

Max bitrate : 7800 kbps

Quantizer: 31 — 1 — 4 — 90%

Bitrate modulation : high-motion — low-motion — 0

Multipass encoding files

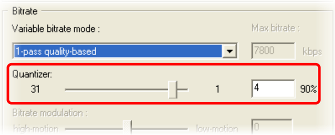
□ Read log file: □ Write log file: □ Write MV file:

c:\divx.log [Select...] c:\mvinfo.bin [Select...]

[Settings] [Restore defaults] [OK] [Cancel]

This mode is only accessible when profiles have been disabled from the *Select Profile Wizard*.

Quality is set by adjusting the *Quantizer* control. As described in *Forward—Quantizers*, the quantizer is the method DivX uses to control the accuracy of the image data it stores. Lower quantizers relate to higher quality, higher quantizers to lower quality.



Bitrate

Variable bitrate mode : 1-pass quality-based

Max bitrate : 7800 kbps

Quantizer: 31 — 1 — 4 — 90%

Bitrate modulation : high-motion — low-motion — 0

By fixing the quantizer you guarantee a consistent quality throughout your entire video file by ensuring the encoder stores the image data for each frame with a consistent accuracy. However, when you fix the quantizer it is not possible to set or predict the bitrate and hence file size because the encoder will always spend as many bits as are required to encode each frame at the given quantizer.

Quality-based mode is ideal when you need to be assured of a fixed quality regardless of file size, and particularly when capturing from live or analogue sources that you intend to later re-compress.

Note that even when the quantizer is 1 and the display reads 100% quality the encoded video will not be identical to the source. DivX is not a lossless codec, and there will always be some degradation from the source video regardless of the combination of settings that you choose.

After running a 1-Pass quality-based encoding your DivX file can be viewed immediately.

1-Pass, Quality-based mode is not explicitly unsupported by DivX Certified devices, however the *Video Buffer Verifier* is disabled in this mode and there is no guarantee that the video stream produced will not exceed the capabilities of the certified device.

When 1-Pass, Quality-based mode is enabled any *Video Buffer Verifier* CLI parameters are disregarded.

You cannot set a bitrate for 1-Pass, Quality-based mode.

MV re-use is not permitted under 1-Pass, Quality-based mode.

Use 1-Pass, Quality-based mode when capturing from a live source and attempting to maintain a consistent quality throughout the video without consideration of the resulting file size.

It is impossible to control or predict the bitrate or file size under 1-Pass, Quality-based mode, other than to know that encoding the same source at higher quantizer will produce smaller files and vice versa.

1-Pass, Quality-based mode is particularly useful for capture when you intend to later re-compress a video using Multipass mode. In fact, 1-Pass, Quality-based mode can be used in place of *Multipass, 1st pass* mode so long as *Write log file* is enabled. In this way it is possible to capture in 1-Pass, Quality-based mode and proceed directly to *Multipass, nth pass*. When making use of this technique take care to avoid overwriting your 1st pass file containing the captured video with the *Multi-pass, nth pass* output accidentally.

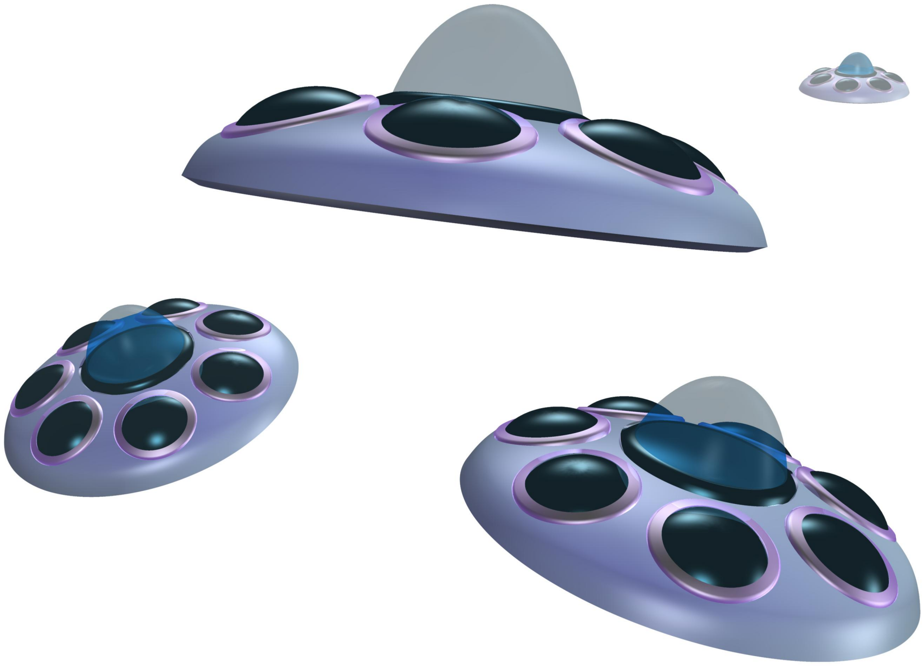
Under the same principle, 1-Pass, Quality-based mode allows you to perform XVID-like Multipass encoding. XVID performs its first pass with Q=2.

Bitrate mode is one of:

-bv1 <*bitrate*>	1-Pass
-b1q <*quantizer*>	1-Pass, quality-based
-bvn1 <*bitrate*>	Multipass, 1st pass
-bvnn <*bitrate*>	Multipass, nth pass

Bitrate is an integer value between 0 and 16,000 specifying the number of kilobits per second.

Quantizer is floating point value between 1.0 and 31.0 specifying the fixed quantizer to be used.



Performance/ Quality

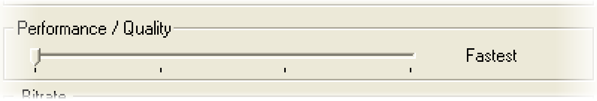
The OFFICIAL Guide

Performance/Quality

Fastest

QUICK GUIDE
>CLI

The fastest performance/quality mode causes the encoder to perform no motion search when encoding. Because no motion estimation is done all blocks will either be intra-blocks or predicted with a null motion vector. In this respect the encoded video will resemble an MJPEG sequence.


Performance / Quality
Fastest
Bitrate

When Fastest performance/quality mode is selected the encoder does not respect the following options: Quarter-pixel, Global motion compensation, Bi-directional encoding, Psychovisual enhancement, MV re-use, Scene change threshold.

The Maximum Keyframe Interval is limited to 60 frames when Fastest mode is enabled. Entering any interval larger than 60 frames will cause the encoder to default the Maximum Keyframe Interval to 10 frames.

Because fastest mode saves mainly intra-blocks it can consume an excessive number of bits. Generally this means 1-Pass mode is unsuitable when working with fastest mode and you should instead use only 1-Pass quality-based mode, fixing the quantizer much the same as you might set the compression factor when saving a JPEG image in a paint program.

If you choose to use Fastest mode in combination with 1-Pass mode the encoder may fail to meet the average bitrate unless a very high average bitrate is specified.

Fastest mode is unsuitable for use during multipass encoding because the rate control algorithm used by the multipass system relies on accurate motion complexity statistics to correctly distribute bandwidth throughout the video. Therefore, multipass does not support Fastest mode.

Fastest mode is ideal for capture situations where exceptionally high image quality is required - for example when you intend to later re-compress the video using another mode.

Performance/Quality:

>CLI

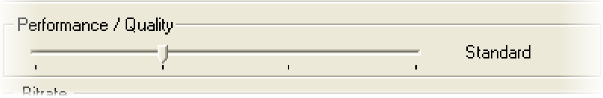
```
-pq <mode>
```

Mode is one of:

1	Fastest
5	Standard
64	Slow
192	Slowest

Standard

The standard performance/quality mode enables the motion search algorithm and is functionally equivalent to *Slowest* mode in previous versions of DivX 5, but contains algorithm enhancements that improve video quality.



Performance / Quality
Standard

Standard mode respects the *Maximum Keyframe Interval* and *Scene Change Threshold* settings (described later). Using these two settings it is possible to control the encoders decision process for frame-type selection with respect to intra-frames.

Standard mode is the only mode supporting motion estimation compatible with the QPel option.

Standard mode is the only mode supporting the *MV re-use* accelerant technique.

Performance/Quality:

-pq ***<mode>***

Mode is one of:

1	Fastest
5	Standard
64	Slow
192	Slowest

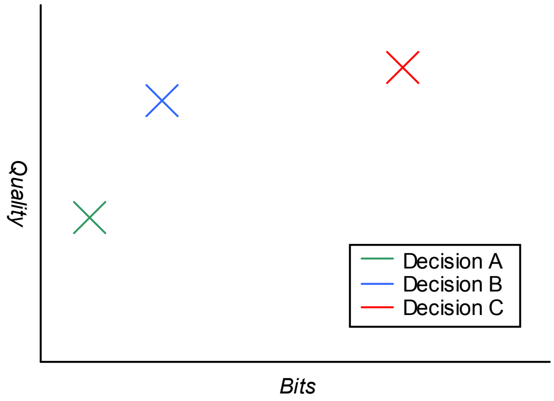
The Rate-Distortion algorithm

New to DivX 5.1 is the *Rate-Distortion* algorithm, enabled to varying degrees when the *Performance/Quality* setting is either *Slow* or *Slowest*. The rate-distortion algorithm allows the outcome of various decisions made by the encoder to be evaluated intelligently with respect to bit spend against quality gain where previously simple algorithms would govern the process.

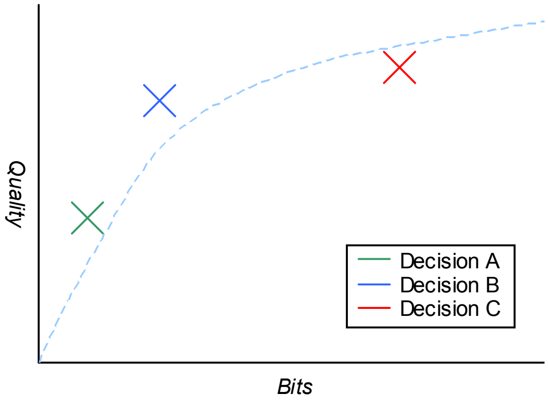
Every decision made as a video is encoded has an impact on both the *rate* and the *distortion*. The *rate* is a term used to describe the bits spent, *distortion* to describe the change in some measure of quality based upon any particular decision.

Decisions evaluated by the rate-distortion algorithm might include the coding of blocks (should they be intra-coded, predicted, or skipped), or which of several possible motion vectors for a block yields the best balance of bits spent against quality.

With respect to the illustration opposite, consider any decision the encoder might make and imagine three possible outcomes—A, B and C.



Under normal circumstances the encoder will make the best possible choice in terms of quality. In the illustration this is decision C. However, the rate-distortion algorithm might instead choose decision B since it offers comparable quality at a much lower bit spend.

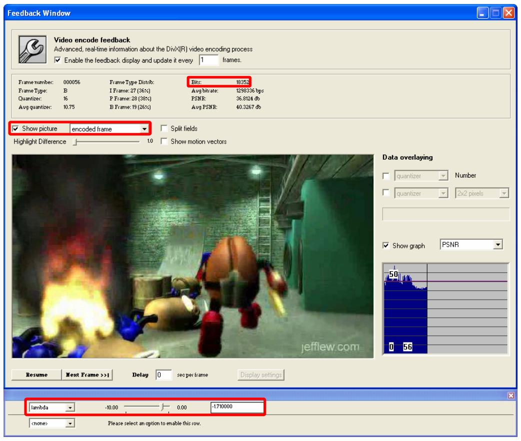


Notice that although decision B leads to marginally lower than optimum quality for this particular decision, overall quality will rise because the encoder has more free bits to spend.

The rate-distortion algorithm evaluates decisions based upon a rate-distortion curve, describing the change in distortion necessary for a rise in bit spend to be considered a good decision.

Here points above the rate-distortion curve evaluate as good decisions, those below the curve as bad decisions. The best decisions fall towards the upper-left of the plot (higher quality and lower bit spend).

It follows that the encoder must derive this rate-distortion curve from somewhere. The curve is approximated by some function inside the encoder accepting a value that changes the shape of the curve. This value is *Lambda*, and you can manipulate it via the Feedback window.



Feedback Window

Video encode feedback
Advanced, real-time information about the DivX(R) video encoding process
☑ Enable the feedback display and update it every 1 frames.

Frame number: 000056
Frame Type: B
Quantizer: 16
Avg quantizer: 10.75
Frame Type Distrib:
I Frame: 27 (36%)
P Frame: 28 (38%)
B Frame: 19 (26%)
Bits: 18352
Avg bitrate: 1298336 bps
PSNR: 36.8124 db
Avg PSNR: 40.3267 db

☑ Show picture: encoded frame
☐ Split fields
Highlight Difference 1.0
☐ Show motion vectors

Data overlaying
☐ quantizer Number
☐ quantizer 2x2 pixels
☑ Show graph PSNR

Delay 0 sec per frame

lambda -10.00 0.00 -1.710000
<none> Please select an option to enable this row.

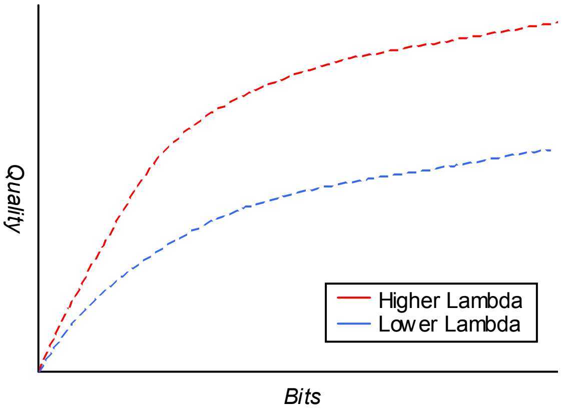
Pro

Begin encoding a video using either *Slow* or *Slowest* performance/quality modes. From the feedback window set *Show picture* to *Encoded frame* and enable the *Lambda* control in the bottom toolbar.

Pause encoding at any suitable frame and vary the *Lambda* control, observing the effect on image quality and the number of bits spent for the frame. By altering Lambda you are manipulating the shape of the rate-distortion curve.

Increasing Lambda (dragging the slider to the left) causes a sharper rise in the rate-distortion curve, meaning larger improvements are demanded for an increase in bit spend.

Decreasing Lambda (dragging the slider to the right) causes the rate-distortion curve to rise more slowly, meaning lesser improvements are acceptable for an increase in bit spend.



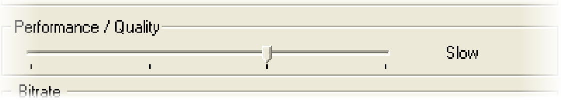
If the rate-distortion curve becomes too steep quality will be degraded because the encoder will not be able to achieve the quality demanded of it for a given rise in bit spend. If the rate-distortion curve becomes too flat quality will be degraded because the encoder will waste a lot of bits for very marginal improvements, reducing the number of free bits available overall.

Deriving the optimal value for Lambda is a very difficult process and under most circumstances you should not over-ride the encoder default. The encoder will normally vary Lambda automatically during encoding depending on circumstances, over-riding instead fixes Lambda for the duration of one pass. Lambda returns to its default mode of operation at the beginning of each pass.

Slow

Pro

Slow performance/quality mode enables the performance-optimized *Rate-Distortion* algorithm, designed to dramatically improve video quality at low bitrates while avoiding the performance hit incurred by the fully quality-orientated rate distortion algorithm.



The Rate-Distortion algorithm attempts to best balance bit spend against quality gain by intelligently evaluating all of the possible outcomes of various decisions made by the encoder.

In Slow mode the encoder uses the new Rate-Distortion algorithm to make frame type decisions based upon the best balance of bit spend and quality gain. Because of this the *Scene Change Threshold* settings become redundant and is ignored.

MV re-use is not permitted under Slow mode.

QPel is unsupported in Slow mode. If you have enabled both QPel and Slow mode the performance/quality mode will automatically default to Standard mode when encoding begins.

>CLI

Performance/Quality:

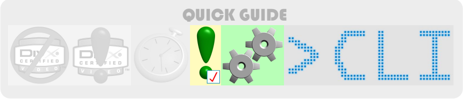
-pq *<mode>*

Mode is one of:

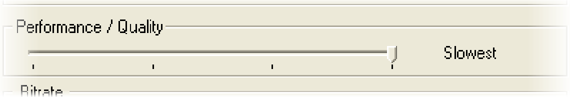
1	Fastest
5	Standard
64	Slow
192	Slowest

Slowest

Pro



Slowest performance/quality mode enables the quality-optimized *Rate-Distortion* algorithm, designed to dramatically improve video quality at low bitrates. Slowest mode will yield the highest possible quality available but at the expense of the longest encoding duration.


Performance / Quality
Slowest
Bitrate

The Rate-Distortion algorithm attempts to best balance bit spend against quality gain by intelligently evaluating all of the possible outcomes of various decisions made by the encoder.

In Slowest mode the encoder uses the new Rate-Distortion algorithm to make frame type decisions based upon the best balance of bit spend and quality gain. Because of this the *Scene Change Threshold* settings become redundant and is ignored.

MV re-use is not permitted under Slowest mode.

QPel is unsupported in Slowest mode. If you have enabled both QPel and Slowest mode the performance/quality mode will automatically default to Standard mode when encoding begins.

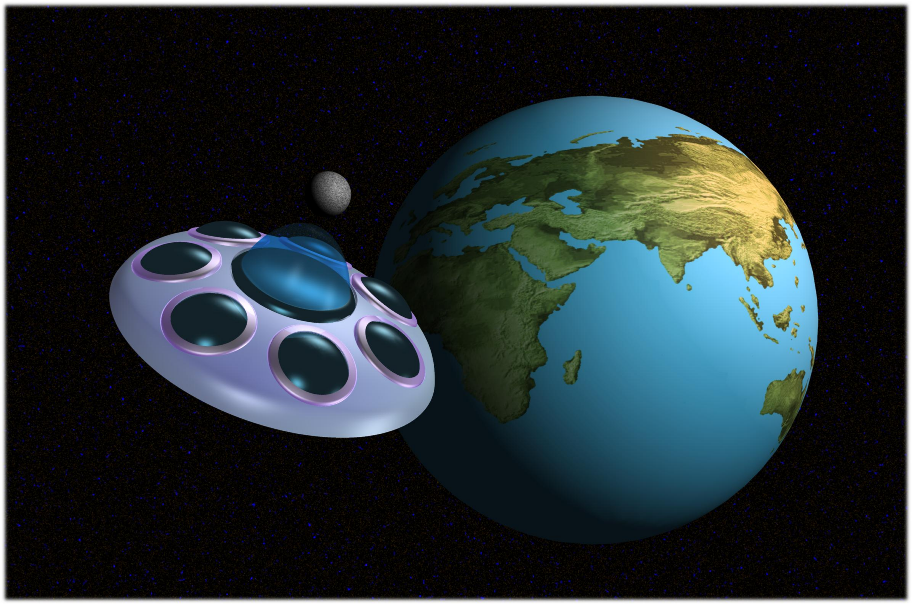
>CLI

Performance/Quality:

-pq ***<mode>***

Mode is one of:

1	Fastest
5	Standard
64	Slow
192	Slowest



Psychovisual Enhancement

Psychovisual Enhancement

What is psychovisual enhancement?

DivX is what is known as a *lossy* codec—that is after encoding and decoding a video the result will not be identical to the source, the video will have been degraded to some extent. In other words it will have lost detail.

DivX, like other lossy codecs you may be familiar with such as MP3 or Vorbis, uses encoding techniques based upon human perception to intelligently lose detail where it is least likely to be noticed before degrading the remainder of the picture. This perceptual technique forms the basis for *psychovisual enhancement*, allowing DivX to mask the artifacts inherent in MPEG4 video in areas of the picture where error is least perceivable.

To understand how this works an explanation of the method DivX uses to encoded the image is necessary. This is a complex subject that we will only touch on it briefly—do not be alarmed if you do not fully comprehend it.

The DCT, iDCT and Human Visual System

Inside a computers memory, images are normally represented as a two-dimensional array of *pixels* - the little colored squares that make up a complete picture.

Storing the color value of each and every pixel is very expensive in terms of bits and thus DivX uses a technique known as the Discrete Cosine Transformation to convert this series of values into frequency information. The actual process of the DCT is an advanced mathematical topic that will not be covered here but the main feature of the DCT result is a set of co-efficients representing the magnitude of the frequencies composing the input series in order of ascending frequency.

Y-axis
X-axis
jefflew.com

The human visual system is far less sensitive to high frequencies in an image than it is to low frequencies. One perceptual technique DivX uses during lossy compression is the reduction in accuracy of higher frequency co-efficients, saving bits while causing the least perceivable quality degradation.

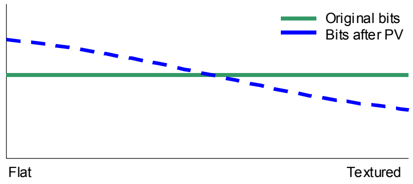
It is in fact these DCT co-efficients that are quantized by DivX when image data is encoded (see *Forward—Quantizers)*. Greater quantization of the DCT co-efficients means fewer bits are ultimately required to store them, but a less accurate image results when the *inverse discrete cosine transformation* (or iDCT) is performed during decoding—the process of restoring the original series (image data) from the DCT result. This error between source and encoded image is known as *quantization noise*.

The psychovisual enhancement system

As described, image data is stored by DivX as a set of co-efficients linked to different frequencies in the image. The psychovisual enhancement process actually manipulates these co-efficients to produce two distinct effects:

1. In flat areas of the image co-efficients are manipulated so that fine details are enhanced. If we were to encode in 1-Pass Quality Based mode (i.e. at a fixed quantizer) this enhancement would naturally *increase* the bits spent on flat areas of the image. However, at a *fixed* average bitrate the effect is actually that textured areas of the image will receive fewer bits and hence when psycho-visual enhancements are enabled artifacts will be masked in textured areas of the image where they are least visible.

   This works because of a balancing effect that is created - the encoder has a finite number of bits to spend and if flat areas are to consume more bits then textured areas must consume fewer bits.



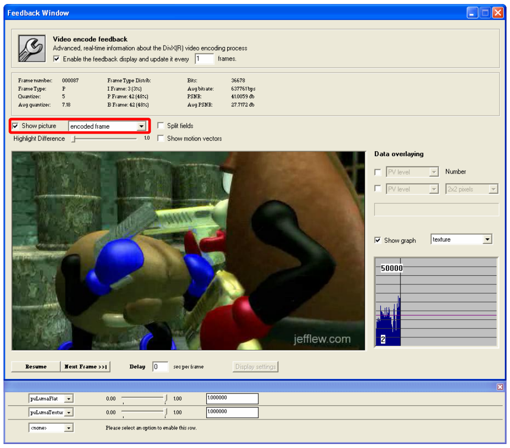
2. In strongly textured areas of the image co-efficients are manipulated so that fewer bits are spent and thus quality is slightly degraded. If we were to encode in 1-Pass Quality Based mode (i.e. at a fixed quantizer) this enhancement would naturally decrease the bits spent on heavily textured areas of the image. However, at a *fixed* average bitrate the effect is actually that flatter areas of the image will receive more bits (once again due to balancing) and hence when psycho-visual enhancements are enabled artifacts will be masked in textured areas of the image where they are least visible.

Because both psychovisual enhancement methods mask artifacts in textured areas where they are least visible the video quality appears to improve when psychovisual enhancements are enabled. You can visualize the concept by imagining two characters having a conversation while standing in front of a tree—the psychovisual enhancement process might enhance details in the characters faces at the expense of causing some artifacting amongst the leaves of the tree where it would be least perceivable.

Experiencing PVE

Pro

The *best* way to understand the psychovisual enhancements is to watch them working on an image. The feedback window provides an excellent method of doing this, and also allows you to configure the degree to which flat and texture psychovisual enhancements are performed.



Feedback Window

Video encode feedback
Advanced, real-time information about the DivX(R) video encoding process
☑ Enable the feedback display and update it every 1 frames.

Frame number:	000087	Frame Type Distrib:		Bits:	36678
Frame Type:	P	I Frame: 3 (3%)		Avg bitrate:	637761 bps
Quantizer:	5	P Frame: 42 (48%)		PSNR:	41.0059 db
Avg quantizer:	7.18	B Frame: 42 (48%)		Avg PSNR:	27.7172 db

☑ Show picture: encoded frame
□ Split fields
Highlight Difference 1.0
□ Show motion vectors

Data overlaying
□ PV level — Number
□ PV level — 2x2 pixels
☑ Show graph: texture

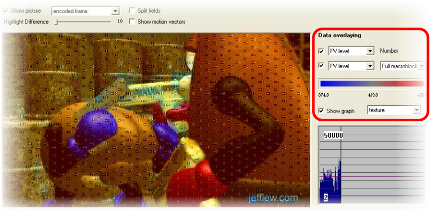
Delay 0 sec per frame

pvLumaFlat	0.00	1.00	1.000000
pvLumaTextur	0.00	1.00	1.000000
<none>	Please select an option to enable this row.		

Set up a 1-Pass encoding using VirtualDub and any source video of your choosing with *Fast* psychovisual enhancement mode enabled. Begin encoding and click the *Pause* button on the encoder feedback window when a suitable image to work with appears.

Set *Show picture* to *encoded frame* and enable the *pvLumaFlat* and *pvLumaTexture* controls in the bottom panel.

With the video paused slowly drag each slider back and forth from zero (no psychovisual enhancement) to one (full psychovisual enhancement), watching the image carefully as you do so. Since the aim of psychovisual enhancement is to mask artifacts where you are least likely to perceive them you should see that both effects appear to improve quality when enabled to fuller extents.



☑ Show picture: encoded frame
□ Split fields
Highlight Difference 1.0
□ Show motion vectors

Data overlaying
☑ PV level — Number
☑ PV level — Full macroblock
874.0 419.0
☑ Show graph: texture

By enabling the *Data overlaying* controls and configuring them to show the *PV level* and color the *Full macroblock* based also on the *PV level* it is possible to view graphically the changes made in a more statistically-orientated way.

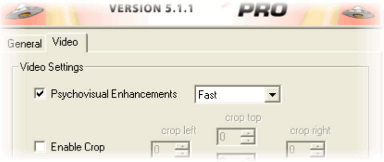
Where information is removed from the image the number overlaid will be positive and the macroblock tinted red. Where information is added to the image the number overlaid will be negative and the macroblock tinted blue.

Fast psychovisual enhancement

Pro

Fast psychovisual enhancement mode attempts to manipulate the DCT co-efficients selectively so that the resulting noise in the decoded image is positioned where it will be least visible, for example in areas of strong texture.

QUICK GUIDE



Video Settings

☑ Psychovisual Enhancements: Fast

□ Enable Crop

crop left: 0

crop top: 0

crop right: 0

When encoding a multipass video without *MV re-use* enabled you can save time by enabling psychovisual enhancements from the second-last pass onwards only. This will have a minimal impact on quality but can potentially reduce encoding time substantially.

Fast psychovisual enhancement mode generally produces a more intense effect than Slow psychovisual enhancement mode.

When choosing which psychovisual enhancement mode to use do not equate speed with quality. Each psychovisual enhancement mode produces slightly different results.

Care should be taken when applying psychovisual enhancement to noisy sources as undesired results can occur. If your source is a particularly old video or has been acquired via analogue capture you may need to apply *Source pre-processing* (described later) to reduce the level of noise present when applying psychovisual enhancement.

Psychovisual enhancement should not be applied when encoding as interlaced. The psychovisual enhancement system is not yet interlace-aware and may enhance combing artifacts undesirably.

>CLI

Psychovisual enhancements:

-psy <*mode*>

Mode is one of:

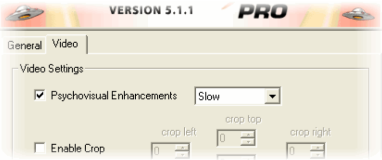
0 Disable psychovisual enhancement
1 *Fast* psychovisual enhancement
2 *Slow* psychovisual enhancement

Slow psychovisual enhancement

Pro

Slow psychovisual enhancement mode analyses each block and the blocks surrounding it in turn to ensure any enhancement does not introduce significant blocking and ringing artifacts, thus slow mode is less likely to introduce artifacts than fast mode.

QUICK GUIDE >CLI



VERSION 5.1.1 PRO

General | Video

Video Settings

☑ Psychovisual Enhancements: Slow

crop top: 0
crop left: 0
crop right: 0
□ Enable Crop

When encoding a multipass video without *MV re-use* enabled you can save time by enabling psychovisual enhancements from the second-last pass onwards only. This will have a minimal impact on quality but can potentially reduce encoding time substantially.

Slow psychovisual enhancement mode generally produces a less intense effect than Fast psychovisual enhancement mode and lowers the probability that visible artifacts will be introduced as a consequence of the process.

When choosing which psychovisual enhancement mode to use do not equate speed with quality. Each psychovisual enhancement mode produces slightly different results.

Care should be taken when applying psychovisual enhancement to noisy sources as undesired results can occur. If your source is a particularly old video or has been acquired via analogue capture you may need to apply *Source pre-processing* (described later) to reduce the level of noise present when applying psychovisual enhancement.

Psychovisual enhancement should not be applied when encoding as interlaced. The psychovisual enhancement system is not yet interlace-aware and may enhance combing artifacts undesirably.

>CLI

Psychovisual enhancements:

-psy <*mode*>

Mode is one of:

0	Disable psychovisual enhancement
1	*Fast* psychovisual enhancement
2	*Slow* psychovisual enhancement

Source pre-processing

Source pre-processing

What is pre-processing?

In the modern computer age digital video can be acquired from a wide and varied array of sources. DV Camcorders, DVD video discs, TV capture cards and even web-cameras have all become so accessible at the consumer level that for DivX to consistently produce high quality output it is necessary to provide a means of correcting video issues attributable to particular sources.

Pre-processing allows the encoder to clean phenomena such as noise, grain, flickering and color inconsistency from the source video prior to encoding it. An immediate benefit of pre-processing is an increase in the viewers perception of quality. With respect to encoding, a reduction in noise and general inconsistency in the texture from frame-to-frame yields a lower bit-spend when encoding the image because the DCT is less complex and the motion search is more successful.

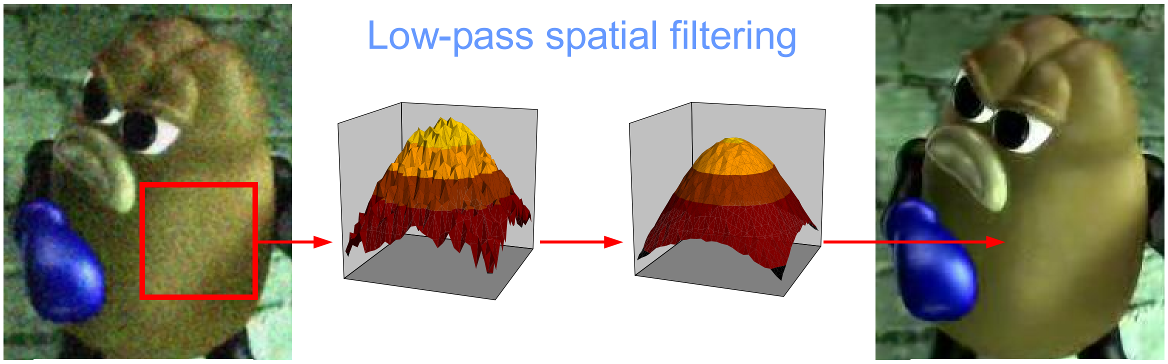
When to use pre-processing

Pre-processing is most effective when applied to video taken from an analogue source - typical examples being VHS tape, TV capture, and camcorder footage. All of these sources are likely to suffer from noise—high-frequency variations around the true color of each pixel. Noise causes the image to appear fuzzy or speckled.

Digital sources are also susceptible to noise. DV camcorders and web-cameras often amplify the signal arriving at their image sensor in low-light conditions, introducing a lot of noise to the picture. Web-cameras are also notable for an increase in noise proportional to the resolution, frame-rate and lack of strong lighting. Even DVD video re-mastered from film can show the fine grain of the original film media. This particular effect is a special case as it is sometimes desirable in the encoded video, adding warmth to the picture. DivX includes a special decoder feature allowing the artificial application of grain to any video, even when the original grain was removed during encoding.

Spatial filtering

Pre-processing consists of two filter types, the first is a spatial filter. The spatial filter is concerned with each pixel in the frame and the pixels surrounding it. Noise manifests itself as high frequency changes between the colors of adjacent pixels. By applying a low-pass filter (one that rejects high frequencies) to an image it is possible to stabilize the colors.



Low-pass spatial filtering

Noisy image

Sampled data

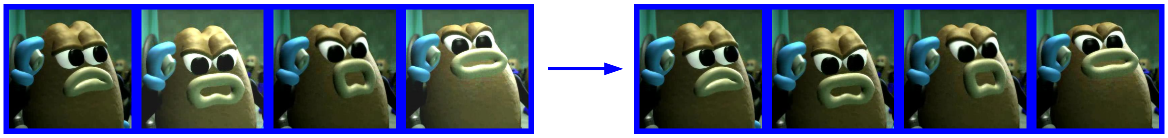
After low-pass filter

Filtered image

Temporal filtering

The second type of filter employed by the pre-processing algorithm is a temporal filter. While the spatial filter can reduce noise within a single frame it does not consider the changes in each pixel over time, I.e. from frame-to-frame. A good example of temporal artifacting is flickering video, where even though there may be little noise within each individual frame there is *temporal noise* affecting the intensity of pixels over time. Temporal noise is also responsible for producing grain or discoloration patterns that change throughout a video as it is played.

Just as it is possible to perform low-pass filtering within a frame by examining adjacent pixels it is also possible to consider the color of each individual pixel with respect to its color in several previous frames. By applying an adaptive low-pass filter to co-located pixels throughout time any temporal noise related to lighting or color tonality between frames can be reduced.



Temporal filtering

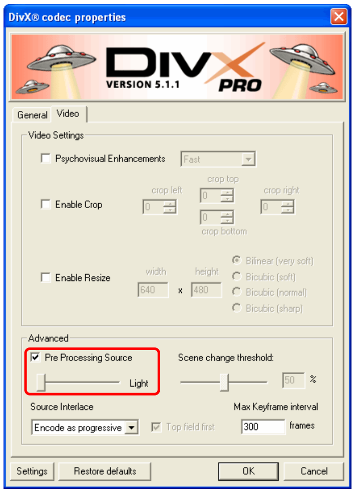
Source pre-processing

Source pre-processing applies spatial and temporal filtering prior to encoding in order to reduce noise in the source video.

Four presets are defined allowing control of the filter intensity. These are *light*, *normal*, *strong* and *extreme*.

Both *light* and *normal* presets are very carefully balanced and neither should cause visible degradation to the video.

Sources with heavy noise present may require pre-processing using the *strong* or *extreme* presets. These presets may introduce a visible smoothing effect in the video, therefore when using these two presets it may be advisable to process short sample clips under each to find the best balance between noise reduction and smoothing.



DivX® codec properties

DIVX VERSION 5.1.1 PRO

General | Video

Video Settings

□ Psychovisual Enhancements: Fast

□ Enable Crop — crop left: 0, crop top: 0, crop right: 0, crop bottom: 0

□ Enable Resize — width: 640 x height: 480
● Bilinear (very soft)
○ Bicubic (soft)
○ Bicubic (normal)
○ Bicubic (sharp)

Advanced

☑ Pre Processing Source — Light

Scene change threshold: 50 %

Source Interlace: Encode as progressive — ☑ Top field first

Max Keyframe interval: 300 frames

Source pre-processing is recommended when the source video contains either visible noise, unstable color tonality or flickering.

Source pre-processing is particularly important when encoding from analogue sources, old films, or consumer-level digital video equipment—especially when video was filmed under poor lighting conditions.

>CLI

Source pre-processing:

-pre <*Strength*> or **-pre <*Temporal*>,<*Spatial*>,<*SpatialPasses*>**

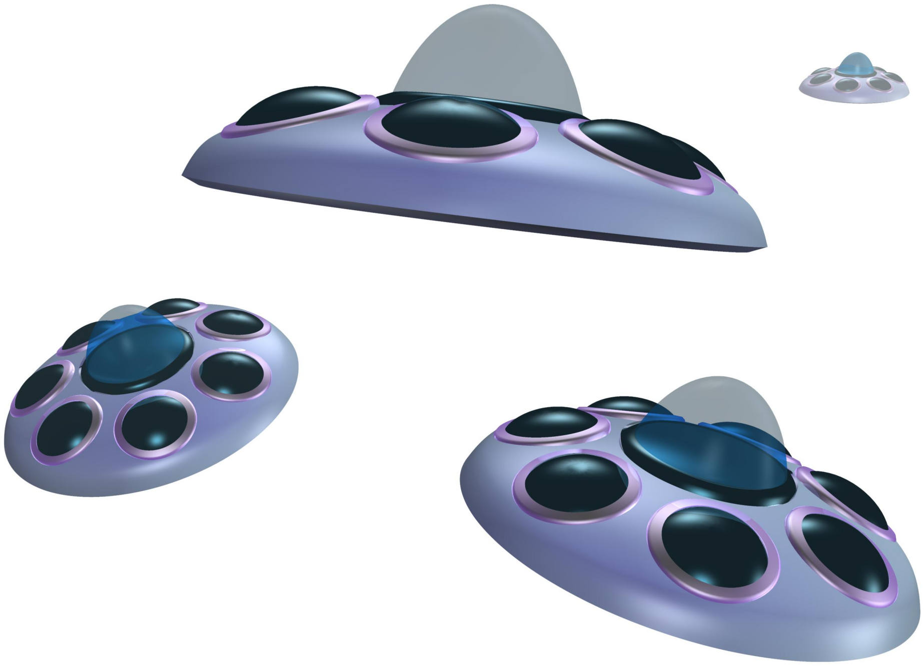
Strength is one of:

1	Light
2	Normal
3	Strong
4	Extreme

Temporal is the temporal level from 0.0 to 1.0

Spatial is the spatial level from 0.0 to 1.0

SpatialPasses is the number of passes using the Spatial filter from 1 to 3

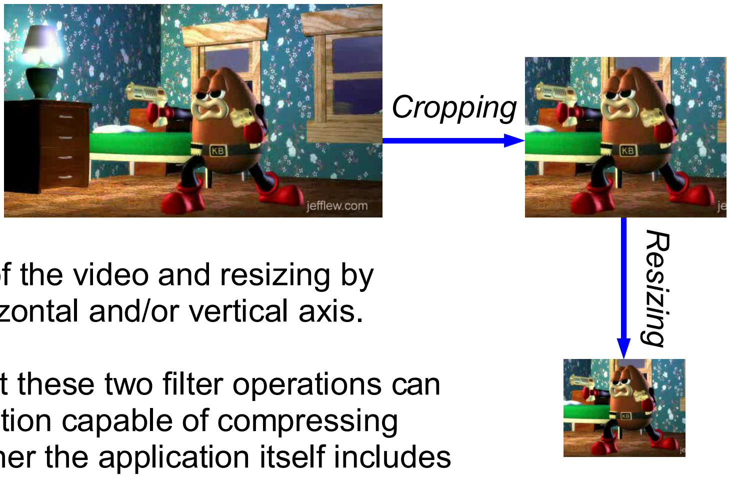


Crop and Resize

Crop and Resize

Cropping and resizing via the encoder

Cropping and resizing are two of the most basic filters than can be applied to video, and yet not all video applications provide this functionality. Both techniques alter the dimensions of the video area, cropping by discarding areas of the video and resizing by scaling the video along its horizontal and/or vertical axis.



The DivX encoder ensures that these two filter operations can be performed from any application capable of compressing DivX video regardless of whether the application itself includes this functionality by providing its own crop and resize controls. The encoders internal filters are optimized with regards to quality and performance and may offer better performance than those in your video application.

Why crop or resize?

By appropriately cropping and resizing a source it is possible to improve the quality of the encoded video as well as decrease encoding time.

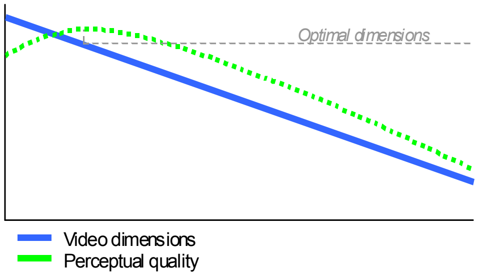
Cropping is most useful when processing video sources that feature borders around the exterior of the picture area, a common attribute of both widescreen DVD video and analogue capture content. Although the borders themselves are encoded with relative ease, where the border meets the picture a high contrast edge is created that consumes a large number of bits when encoded.

jefflew.com

This extra bit spend means throughout the video less bits are available to encode the actual picture and thus video quality suffers. By cropping borders no bits are wasted outside the actual picture area and thus overall quality is improved.

Resizing is used in a variety of situations, the simplest case being where there is a desire to reduce the video dimensions. At lower bitrates reducing video dimensions may assist in preserving quality by reducing the number of blocks required to make up the picture - with fewer blocks each block should receive a greater number of bits on average.

It should be noted that there is a moving balance to be drawn between dimensions and quality at any particular bitrate. A small reduction in dimensions will generally mean fewer blocks and thus a greater bit spend per block giving higher overall quality, but if the video dimensions are reduced too far then within each block the texture and motion may become so complex that intra-blocks require more bits to store and the motion search becomes less effective leading to a greater proportion of intra-blocks, intra-frames and lower overall quality. At lower resolutions fine details and sharp edges may also become degraded.



Resizing is frequently used to scale video on its vertical axis to correct the *aspect ratio* when encoding from DVD. The aspect ratio describes the relationship between the horizontal and vertical dimensions of the video with respect to their proportionality. If this ratio is not correct video will appear to be either compressed or stretched vertically when viewed, leading to characters that appear either short and fat or tall and thin.

jefflew.com

Original aspect ratio

jefflew.com

Higher aspect ratio

jefflew.com

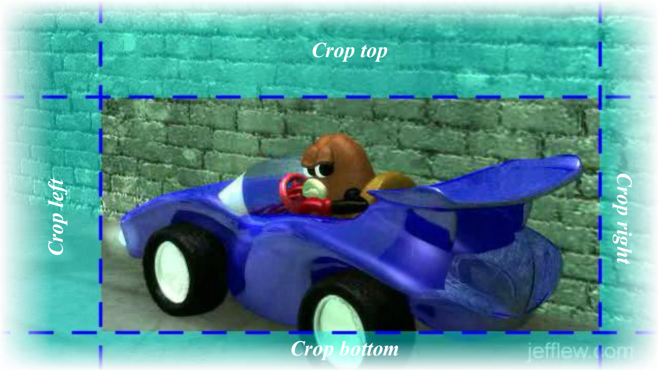
Lower aspect ratio

Crop

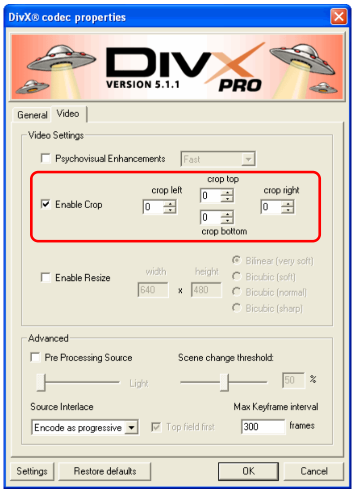
Pro

Crop allows you to specify the number of pixels from each edge of the video that will be discarded before encoding.

The cropping area is fixed throughout the video.



QUICK GUIDE


DivX® codec properties

Video Settings

☐ Psychovisual Enhancements Fast

☑ Enable Crop

crop left 0

crop top 0

crop bottom 0

crop right 0

☐ Enable Resize

width 640 x height 480

● Bilinear (very soft)
○ Bicubic (soft)
○ Bicubic (normal)
○ Bicubic (sharp)

Advanced

☐ Pre Processing Source

Light

Scene change threshold: 50 %

Source Interlace: Encode as progressive

☑ Top field first

Max Keyframe interval: 300 frames

Cropping in some applications, such as VirtualDub, can require colorspace conversions that reduce encoding performance and minimally degrade the video. Where crop and resize are the only filtering operations required greater encoding performance will be seen using the encoders crop filter.

You should crop any borders surrounding a video before encoding it to avoid wasting bits. This will improve the overall quality of your video.

Cropping is performed before resizing by the encoder. The dimensions of the video after both cropping and resizing have been performed must be divisible by four. For optimal performance the dimensions should be divisible by sixteen.

>CLI

Cropping:

-c *<left>,<right>,<top>,<bottom>*

Left, right, top and *bottom* are the number of pixels from each edge of the image that will be discarded before encoding.

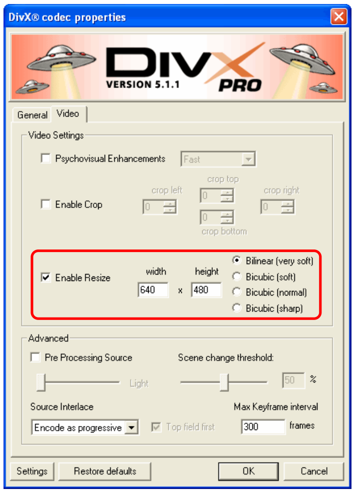
Resize

Pro

Resizing scales the video dimensions on one or both axis.

QUICK GUIDE >CLI

You can select the type of sampling that is performed in the source image to calculate the color of each pixel in the resized image. Different sampling options create slightly different effects in the resized image.


DivX® codec properties

DivX VERSION 5.1.1 PRO

General | Video

Video Settings

□ Psychovisual Enhancements Fast

□ Enable Crop crop left 0 crop top 0 crop bottom 0 crop right 0

☑ Enable Resize width 640 x height 480

● Bilinear (very soft)
○ Bicubic (soft)
○ Bicubic (normal)
○ Bicubic (sharp)

Advanced

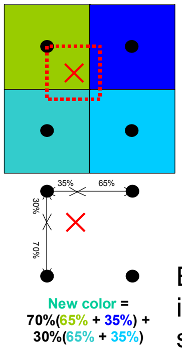
□ Pre Processing Source Light

Scene change threshold: 50 %

Source Interlace: Encode as progressive ☑ Top field first

Max Keyframe interval 300 frames

Settings | Restore defaults | OK | Cancel



Bilinear sampling is best used when reducing an image. The color of each pixel in the resized image depends on where the pixel would fall in the source image.

The color is interpolated from the four surrounding pixels based upon the theoretical position of the new pixel.

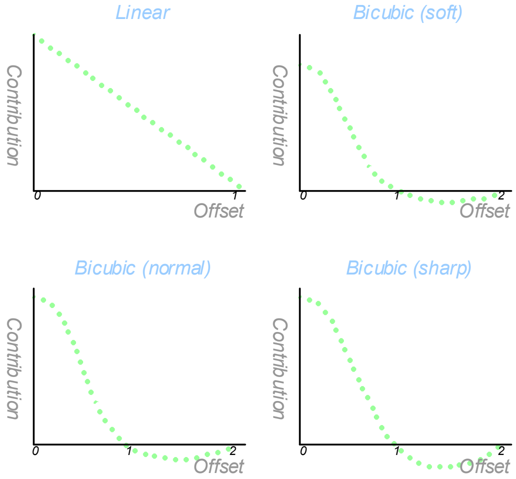
Because bilinear sampling considers only 2x2 pixels in the source and interpolates linearly it produces a softened image on resizing and can cause pixelation in enlarged images.

Bicubic sampling is more complex than bilinear sampling, calculating the color of each pixel in the resized image from 4x4 pixels in the source

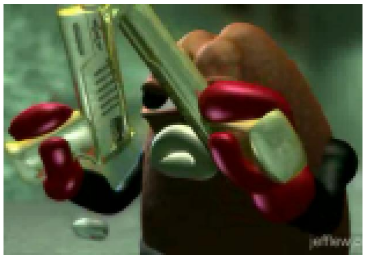
Instead of simply interpolating linearly between pixels bicubic sampling weights the contribution of each pixel in the source using a spline function based upon the offset of each pixel from the pixel being calculated.

Changing the spline parameters allows the sharpness of the resized image to be controlled. Three presets - *soft*, *normal* and *sharp*—are available to choose from.

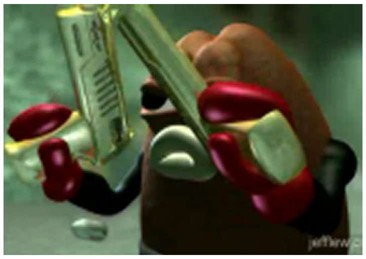
Bicubic sampling produces better quality over bilinear sampling when enlarging an image but because more samples are taken and calculations are more complex for the bicubic method it also takes longer to compute.



Source



Bilinear resample



Bicubic resample

The encoder resize filters are very highly optimized. Consider using them in place of external resize filters provided with video editing applications. Resizing in certain applications, such as VirtualDub, can require colorspace conversions that reduce encoding performance and minimally degrade the video. Where crop and resize are the only filtering operations required greater encoding performance will be seen using the encoders resize filter.

Cropping is performed before resizing by the encoder. The dimensions of the video after both cropping and resizing have been performed must be divisible by four. For optimal performance the dimensions should be divisible by sixteen.

Bilinear sampling is faster than bicubic sampling and is suitable for reducing video. Although it is valid to use bilinear sampling when enlarging better quality is obtained by use of bicubic sampling.

A common use for resizing is aspect correction when encoding from DVD. You can easily calculate the dimensions for the correct aspect ratio given the original video dimensions and the aspect details from the DVD.

> *Example:*
> *The video dimensions are 704x360 and the DVD box reports 2.35:1 aspect.*
>
> To calculate the correct vertical resolution:
>
> *(1 / Aspect)* x *Horizontal dimension*
>
> *Or:*
>
> *(1 / (2.35 / 1))* x *704 = 299.57*
>
> Recall that dimensions must be divisible by four, or more optimally by sixteen. 704 is already divisible by sixteen, but it is necessary to round 299.57 up to 304.

>CLI

Resize:

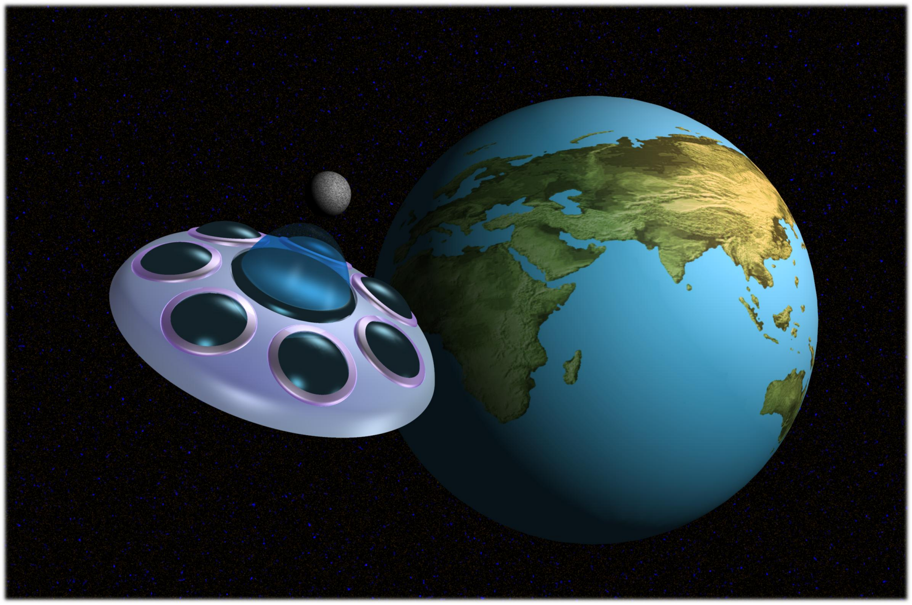
-r <*horizontal*>,<*vertical*>,<*mode*>

Horizontal is the horizontal dimension of the resized video in pixels.

Vertical is the vertical dimension of the resized video in pixels.

Mode is one of:

1	Bilinear (very soft)
2	Bicubic (soft)
3	Bicubic (normal)
4	Bicubic (sharp)



MPEG4 Tools

The OFFICIAL Guide

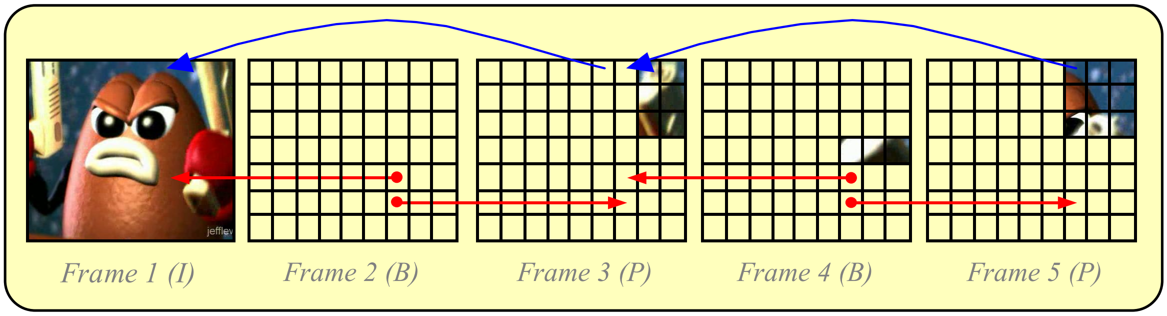
Bi-directional encoding

What is bi-directional encoding?

As discussed in *Forward—General concepts,* there are three different frame types available for the DivX encoder to select from. These are intra-frames, predicted-frames and bi-directional-frames, or *I*, *P* and *B* frames respectively.

Recall that in an I-frame all blocks are intra-blocks and are encoded as the image filling the block. In a P-frame blocks can be either intra-blocks or forward predicted blocks—those described by a vector referencing a matching area of image in the previous frame (known as forward prediction).

Bi-directional frames can contain blocks that are intra, forwards predicted, backwards predicted, or both forwards *and* backwards predicted. This means that B-frames reference not only the previous frame but the next frame also.



In the diagram frames 2 and 4 are bi-directional frames and any macroblock may be predicted from either the previous (forward) or next (backward) frame. If a block is both forwards and backwards predicted then when motion compensation takes place the block image will be the result of blending the appropriate areas of the forwards and backwards frames together.

Notice that frame 4 contains some intra-blocks because in the example these blocks could not be predicted from either the forwards or backwards frame.

As described earlier in the guide, predicting blocks consumes far less bits than encoding them intra-blocks. Because B-frames can be both forwards and backwards predicted in general B-frames will have a lower proportion of intra-blocks than any other type of frame.

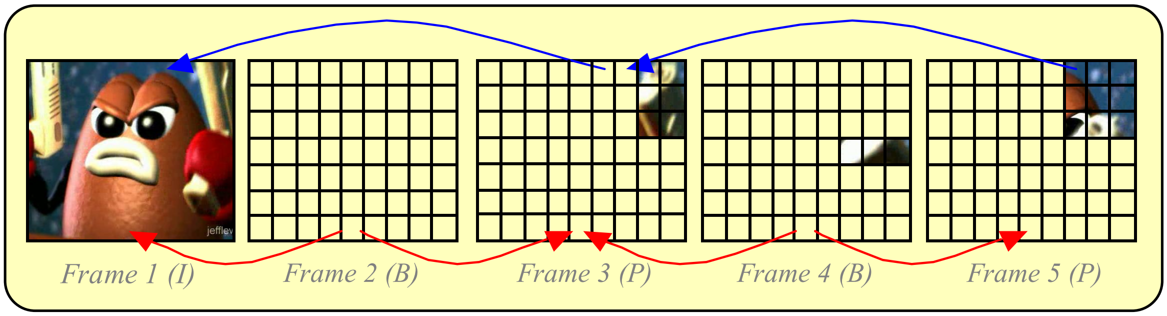
Since B-frames themselves are never referenced by any other frame it is possible for the encoder to use a higher quantizer than is normal for their coding since it need not be concerned with impacting upon the quality of other frames due to compound errors.

These two key features of B-frames lead them to offer the greatest compression ratio of any of the available frame types.

The ordering of frames

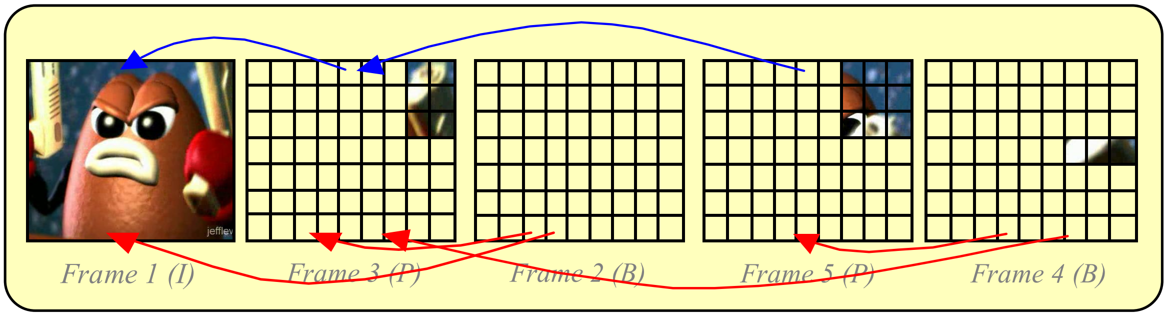
Backward prediction introduces issues relating to the sequence in which frames must be encoded and decoded. Whereas forward prediction, as used by P-frames, simply references the last frame that was decoded, a backwards predicted block in a B-frame makes reference to some future frame that has not yet been decoded. Therefore, in order to decode B-frames it is necessary to decode future frames first. This is why frames may appear out-of-order in the multipass log file (see *Bitrate mode—Multipass)*.

Consider once again the above illustration. Here the blue arrows represent forwards prediction in the P-frames and the red arrows represent either forward (left) or backward (right) prediction in the B-frames.



It is clear from the diagram that the B-frames can make reference to future frames because there are red arrows pointing to the right. The encoder must re-order these frames so that they are in the correct order for decoding.

Now consider the re-ordered sequence of frames:



Although Frame 2 still makes reference to both frame 1 and frame 3, frame 3 is decoded first.

It is now clear that in the re-ordered sequence all frames can be decoded, as every frame makes reference only to others that have been previously decoded (all arrows point left). The decoder is responsible for displaying the decoded frames in the correct order.

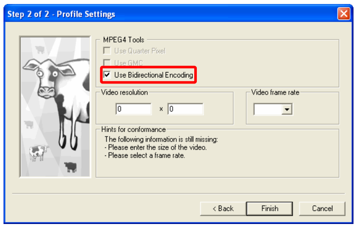
Bi-directional encoding

Pro

Bi-directional encoding allows the encoder to use the bi-directional frame type, supporting both forwards and backwards motion prediction.

B-frames offer the best compression ratio of all frame types and are invaluable to producing high quality video at low bitrates.

Bi-directional encoding can be toggled from the second step of the *Select Profile Wizard* dialogue under *MPEG4 Tools*.

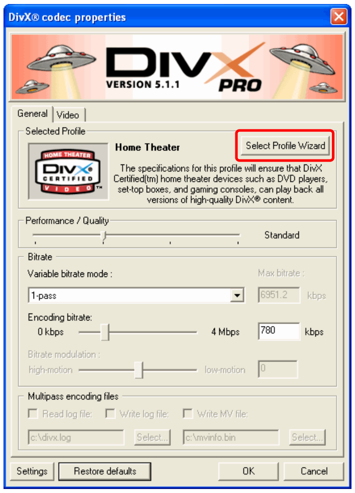

Step 2 of 2 - Profile Settings

MPEG4 Tools
□ Use Quarter Pixel
□ Use GMC
☑ Use Bidirectional Encoding

Video resolution: 0 × 0

Video frame rate:

Hints for conformance
The following information is still missing:
- Please enter the size of the video.
- Please select a frame rate.


DivX® codec properties

Selected Profile
Home Theater
The specifications for this profile will ensure that DivX Certified(tm) home theater devices such as DVD players, set-top boxes, and gaming consoles, can play back all versions of high-quality DivX® content.

Performance / Quality: Standard

Bitrate
Variable bitrate mode: 1-pass
Max bitrate: 6951.2 kbps
Encoding bitrate: 0 kbps – 4 Mbps: 780 kbps
Bitrate modulation: high-motion – low-motion: 0

Multipass encoding files
□ Read log file: c:\divx.log
□ Write log file:
□ Write MV file: c:\mvinfo.bin

The *Handheld* profile does not support Bi-directional encoding.

Bi-directional encoding does not make use of the *MV file* during a multipass encoding. Enabling bi-directional encoding will always increase encoding time.

Bi-directional encoding is not performed when *Performance/Quality* mode is set to *Fastest*.

Bi-directional encoding can greatly improve video quality at low-medium bitrates. At extremely high bitrates it may be beneficial to disable it.

>CLI

Bi-directional encoding:

-b

Enabled only when CLI parameter is present.

Quarter Pixel

What is quarter-pixel?

Y-axis
X-axis
jefflew.com

In a digitized video each frame is composed of a 2-dimensional array of *pixels*, the small colored squares that when viewed collectively from a distance make up the picture.

Recall from *Forward-Macroblocks and motion* that during the motion search DivX attempts to locate a matching image for every block in the current frame from a similarly sized area in the reference frame. One way to think of the motion search is to imagine the encoder taking each square of video in the current frame and overlaying it in various positions in the reference frame until it finds a good match, continually maneuvering it and evaluating the difference.

The simplest way to perform this operation would be to compare each block in the current frame against the surrounding area in the reference frame, moving the block one pixel in any direction at a time. The relative vector offsets from the origin of the current block during this search might be *(0, 0)*, *(0, 1)*, *(0, 2)*, *(1, 0)* and so on.

This scheme would be called whole-pixel because the search was performed against the reference frame at 1-pixel resolution.

In fact, DivX actually searches at half-pixel resolution by default, achieved by use of filtering. Applying a filter to the block allows the encoder to create a virtual block that represents how the block would appear maneuvered half a pixel in any direction against reference frame. The motion search can then return a vector accurate to half-pixel resolution. The relative vector offsets from the origin of the current block during a half-pixel resolution search might be *(0, 0)*, *(0, 0.5)*, *(0, 1)*, *(0, 1.5)*, *(0, 2)*, *(0.5, 0)*, *(0.5, 0.5)*, *(0.5, 1)* and so on.

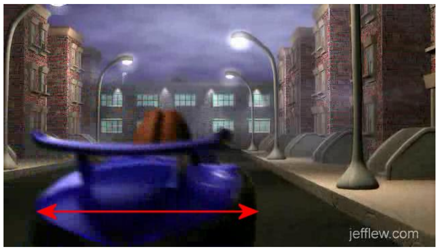
Quarter-pixel, as its name suggests, performs the motion search accurate to quarter-pixel resolution. Relative vector offsets from the origin of the current block during a quarter-pixel resolution search might be *(0, 0)*, *(0, 0.25)*, *(0, 0.5)*, *(0, 0.75)*, *(0, 1)* and so on.

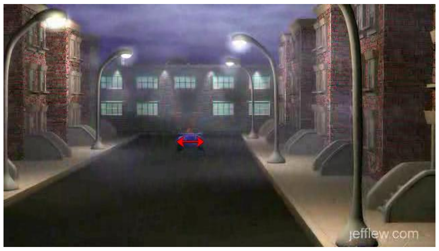
Why use quarter-pixel?

Motion vectors accurate to quarter-pixel resolution allow each motion-compensated frame to be reconstructed more accurately than those accurate to half-pixel resolution. The result tends to be that video appears sharper and of slightly higher quality.

Consider that the eye is most sensitive to motion where objects in a scene are moving very slowly. It would be difficult, for example, to accurately see how many pixels a rocket had traveled across the screen after firing, but if instead we were watching a snail crawling in front of the camera then accuracy would be far more important.

The effect of perspective (where distant objects appear smaller than near objects) biases the improvements made by quarter-pixel motion search with respect to an objects depth in the scene. With respect to the image at the camera objects will appear to move slowly in the distance where perspective causes space to appear compressed, and fast near to the camera where perspective has the least significant effect.





In this scene the effect of perspective is apparent—although the car does not physically shrink as it becomes more distant it occupies less area in the image. We could say that distant objects have lower spatial resolution in the image. Quarter pixel counter-acts this.

Imagine a character walking across the video behind the car while it is positioned near to the camera. The motion accuracy would not be critical because the resolution of near objects is already high. In contrast, imagine the same character walking across the video behind the while it is distant from the camera. The character would occupy far fewer pixels and motion accuracy would be more important for fluid and accurate motion.

Therefore, although quarter-pixel reduces the compensation error throughout the entire video its most significant impact is on the fluidity in motion of objects distant from the camera.

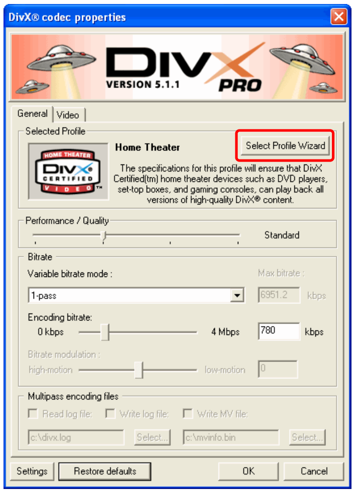
Quarter-pixel

Pro

Quarter-pixel increases the resolution of the motion search to one quarter of one pixel, double that of the default half-pixel resolution.

Quarter-pixel resolution motion vectors reduce the error in the motion-compensated frame but increase the search time, leading to longer encoding durations.

Quarter-pixel can be enabled from the second step of the *Select Profile Wizard* dialogue under *MPEG4 Tools* only when *Disable profiles* has been selected during the first step.



DivX® codec properties

DivX VERSION 5.1.1 PRO

General | Video

Selected Profile

Home Theater

[Select Profile Wizard]

The specifications for this profile will ensure that DivX Certified(tm) home theater devices such as DVD players, set-top boxes, and gaming consoles, can play back all versions of high-quality DivX® content.

Performance / Quality: Standard

Bitrate

Variable bitrate mode : 1-pass

Max bitrate : 6951.2 kbps

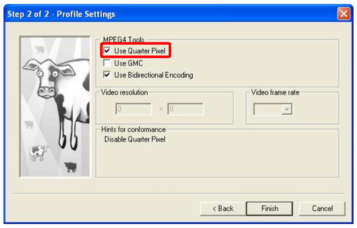
Encoding bitrate: 0 kbps – 4 Mbps 780 kbps

Bitrate modulation : high-motion – low-motion 0

Multipass encoding files

□ Read log file: □ Write log file: □ Write MV file:

c:\divx.log Select... c:\mvinfo.bin Select...



Step 2 of 2 - Profile Settings

MPEG4 Tools

☑ Use Quarter Pixel

□ Use GMC

☑ Use Bidirectional Encoding

Video resolution: 0 x 0

Video frame rate

Hints for conformance

Disable Quarter Pixel

Support for Quarter-pixel decoding is not required by any DivX Certified device. If you encode your video using quarter-pixel you may be unable to play it using DivX Certified devices.

Quarter-pixel motion search resolution can substantially increase encoding duration.

Quarter-pixel motion search resolution is not supported by the Rate-Distortion algorithm as used by the *Slow* and *Slowest Performance/Quality* modes. If you enable Quarter-pixel in combination with these modes the encoder will default to *Standard Performance/Quality* mode automatically.

If you are not creating content strictly for your personal use be aware of the decoding implications of quarter-pixel motion accuracy. Slower computers may perform very poorly when attempting to decode quarter-pixel video—some may fail to decode in real-time even when the quality slider is at its minimum setting. Others may require substantially reduced post-processing, leading to visible artifacting that can counter-act the quality improvements offered by quarter-pixel.

By increasing the accuracy of the motion vectors quarter-pixel also increases the bit spend on predicted frames. The result can actually be detrimental to quality at low bitrates. Although the motion compensation will be more accurate and the picture will appear sharper, fewer free bits will cause the rate control to enforce higher quantizers throughout the video.

>CLI

Quarter-pixel:

-q

Enabled only when CLI parameter is present.

Global motion compensation

What is global motion compensation?

Global motion compensation reduces the bit spend for motion estimation by deriving the motion of some blocks from a global motion present in the frame, as opposed to explicitly specifying a unique vector for each.

jefflew.com

Video sequences frequently contain features such as panning and zooming where many of the blocks will share a common motion attributable to the behavior of the camera.

Imagine the resulting effect on the motion vectors if the camera were to pan right across a scene. It would be reasonable to assume that many of the blocks would share a motion vector opposite but relative to the panning motion of the camera, with the exception of any blocks representing objects moving independently in the scene.

Likewise, if the camera were rotated clockwise all of the blocks representing a static scene would appear to move anti-clockwise around the center of the picture. If the camera were to zoom in, all blocks would appear to move outwards from the center of the picture towards the edges. Its entirely possible that the camera might zoom, rotate and pan at the same time.

During motion estimation, global motion compensation finds and records the global motion attributable to translation, rotation and zooming of the camera. Motion vectors for blocks whose motion is consistent with the global motion can be calculated during decoding as opposed to being explicitly specified, and hence less bits are spent recording motion vectors when global motion compensation is enabled.

When global motion compensation is not used the encoder remains efficient in recording translation. Each motion vector is recorded as the difference from the vector of its neighboring block, thus when vectors are largely consistent throughout a frame very few bits can be spent recording them.

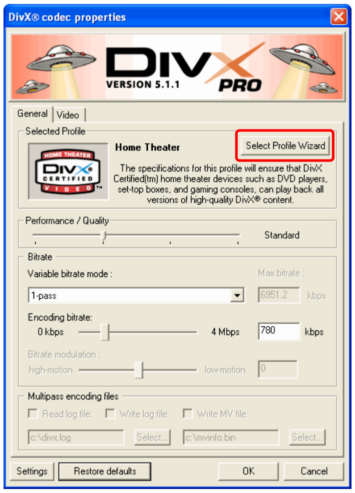
Global motion compensation

Pro

QUICK GUIDE

>CLI

Global motion compensation reduces the bit spend for motion estimation by deriving the motion vectors of some blocks during decoding from a common global motion present in the scene, as opposed to specifying a unique vector for each.



DivX® codec properties

General | Video

Selected Profile

Home Theater

Select Profile Wizard

The specifications for this profile will ensure that DivX Certified(tm) home theater devices such as DVD players, set-top boxes, and gaming consoles, can play back all versions of high-quality DivX® content.

Performance / Quality: Standard

Bitrate

Variable bitrate mode : 1-pass

Max bitrate : 6951.2 kbps

Encoding bitrate: 0 kbps — 4 Mbps 780 kbps

Bitrate modulation : high-motion — low-motion 0

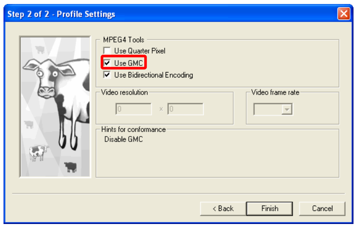
Multipass encoding files

□ Read log file: □ Write log file: □ Write MV file:

c:\divx.log Select... c:\mvinfo.bin Select...

Settings | Restore defaults | OK | Cancel

Global motion compensation can be enabled from the second step of the *Select Profile Wizard* dialogue under *MPEG4 Tools* only when *Disable profiles* has been selected during the first step.



Step 2 of 2 - Profile Settings

MPEG4 Tools

□ Use Quarter Pixel

☑ Use GMC

☑ Use Bidirectional Encoding

Video resolution: 0 x 0

Video frame rate

Hints for conformance

Disable GMC

< Back | Finish | Cancel

Support for Global motion compensation is not required by DivX Certified devices. Although some hardware devices claim to support GMC, this functionality is not required by the DivX Certified Program or tested by DivXNetworks.

If you are not creating content strictly for your personal use be aware of the decoding implications of global motion compensation. Slower computers may perform poorly when decoding video with global motion compensation.

Global motion compensation can improve video quality by reducing the bits spent on motion estimation, leading to the rate control reducing quantizers and thus improving quality.

>CLI

Global motion compensation

-g

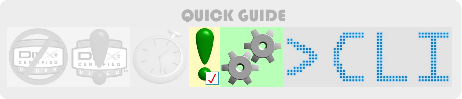
Enabled only when CLI parameter is present

Advanced

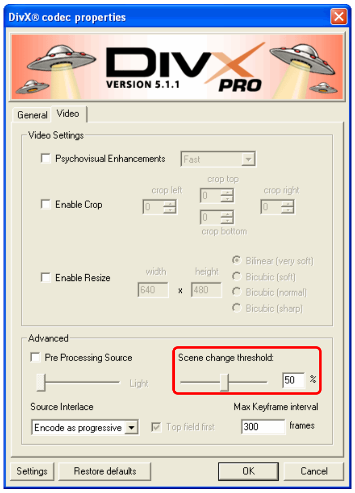
The OFFICIAL Guide

Advanced

Scene-change threshold



As described in *Forward—Predicted-frames and Intra-frames*, each frame in a video can be one of three types: intra, predicted or bi-directional. It follows that there must be some logic controlling frame type selection—the encoder is not human and can't actually recognize where one scene changes into the next.


DivX® codec properties

General | Video

Video Settings

□ Psychovisual Enhancements: Fast

□ Enable Crop: crop left 0, crop top 0, crop right 0, crop bottom 0

□ Enable Resize: width 640 x height 480; ● Bilinear (very soft) ○ Bicubic (soft) ○ Bicubic (normal) ○ Bicubic (sharp)

Advanced

□ Pre Processing Source: Light

Scene change threshold: 50 %

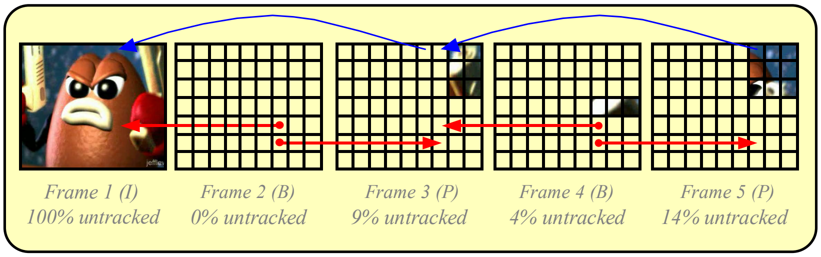
Source Interlace: Encode as progressive; ☑ Top field first

Max Keyframe interval: 300 frames

Recall that when encoding predicted or bi-directional frames the encoder performs the motion search for each block composing the frame into one or more reference frames. For P-frames the reference frame is the previous frame in the video, for B-frames the reference frames are both the previous and next frames.

Where a block can't be matched in a reference frame it will be encoded as an intra-block and instead of prediction via a motion vector its image will be stored. One way to select where to place intra-frames is therefore to set some threshold level specifying the percentage of blocks that must be intra-coded in order for the frame to be encoded as an intra-frame. This is the *scene-change threshold*.

The scene-change threshold defines the percentage of blocks not tracked from the reference frames by the motion search required to trigger a scene-change detection. When a scene-change is detected a key-frame (intra-frame) will be used in place of a predicted or bi-directional frame.



Here if the threshold were 13% frame 5 would become a key-frame (intra-frame).

Note that what the encoder views to be a scene-change does not necessarily correspond with what humans actually perceive to be a scene-change—no matter what value you configure the scene-change threshold to be there is no guarantee that the encoder will key-frame on an actual scene change, just as there is no guarantee that the encoder will not key-frame elsewhere.

The scene-change threshold is respected by the encoder only when *Fastest* or *Standard* Performance/Quality modes are selected.

Key-frames (intra-frames) are useful throughout a video for a number of reasons. Where too few blocks can be predicted from one frame to another it can be more economical to key-frame rather than use one of the predicted frame types. More frequent key-frames can reduce the seek recovery time in media players when the viewer attempts to skip to a specific point in the video. Due to the nature of predicted frames a media player must decode every frame from the nearest past key-frame until the seek target in order to resume playback at the desired frame.

Reducing the scene-change threshold will cause an increase in the proportion of key-frames to predicted frames throughout the video. Key-frames are the most expensive in terms of bit-spend of all frame types and using too many key-frames will starve the encoder of free bits causing the rate control to return higher quantizers, in turn reducing the overall quality of the video.

Raising the scene-change threshold will cause a decrease in the proportion of key-frames to predicted frames throughout the video. Fewer key-frames translate to more free bits causing the rate control to return lower quantizers, in turn increasing the overall quality of the video. However, too few key-frames will lead to long seek times when the viewer attempts to skip to a specific point in the video and increase the recovery time should the video stream become damaged (e.g. when in transit over a broadcast network).

The default scene-change threshold is 50%. It is recommended the scene-change threshold not be lower than 40% or higher than 60% for most content.

>CLI

Scene-change threshold:

-sc <*Threshold*>

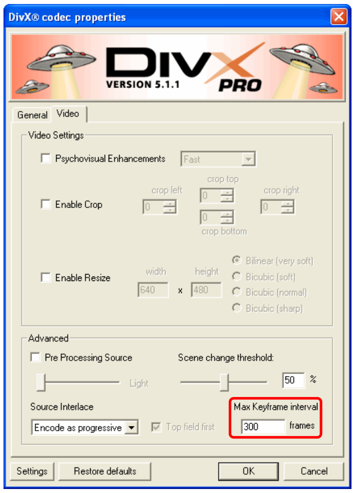
Threshold is the percentage of intra-blocks required to trigger scene-change detection.

Maximum key-frame interval

The maximum key-frame interval defines the maximum number of consecutive frames that may be a predicted type.

If *maximum key-frame interval* frames have been encoded since the last intra-frame, including of the current frame, the current frame will be forced intra.

A fixed limit on the maximum interval between key-frames is essential in ensuring that seek and error recovery is prompt throughout the encoded video – particularly for low-motion content that might otherwise have a very low proportion of key-frames.



DivX® codec properties

DivX VERSION 5.1.1 PRO

General | Video

Video Settings

☐ Psychovisual Enhancements [Fast]

☐ Enable Crop — crop left 0, crop top 0, crop bottom 0, crop right 0

☐ Enable Resize — width 640 x height 480; ● Bilinear (very soft) ○ Bicubic (soft) ○ Bicubic (normal) ○ Bicubic (sharp)

Advanced

☐ Pre Processing Source — Light

Scene change threshold: 50 %

Source Interlace: Encode as progressive ☑ Top field first

Max Keyframe interval: 300 frames

More frequent key-frames can reduce the seek recovery time in media players when the viewer attempts to skip to a specific point in the video. Due to the nature of predicted frames a media player must decode every frame from the nearest past key-frame until the seek target in order to resume playback at the desired frame.

Reducing the maximum key-frame interval will cause an increase in the proportion of key-frames to predicted frames throughout the video. Key-frames are the most expensive in terms of bit-spend of all frame types and using too many key-frames will starve the encoder of free bits causing the rate control to return higher quantizers, in turn reducing the overall quality of the video. Conversely, raising the maximum key-frame interval will cause a decrease in the proportion of key-frames to predicted frames throughout the video. Fewer key-frames translate to more free bits causing the rate control to return lower quantizers, in turn increasing the overall quality of the video.

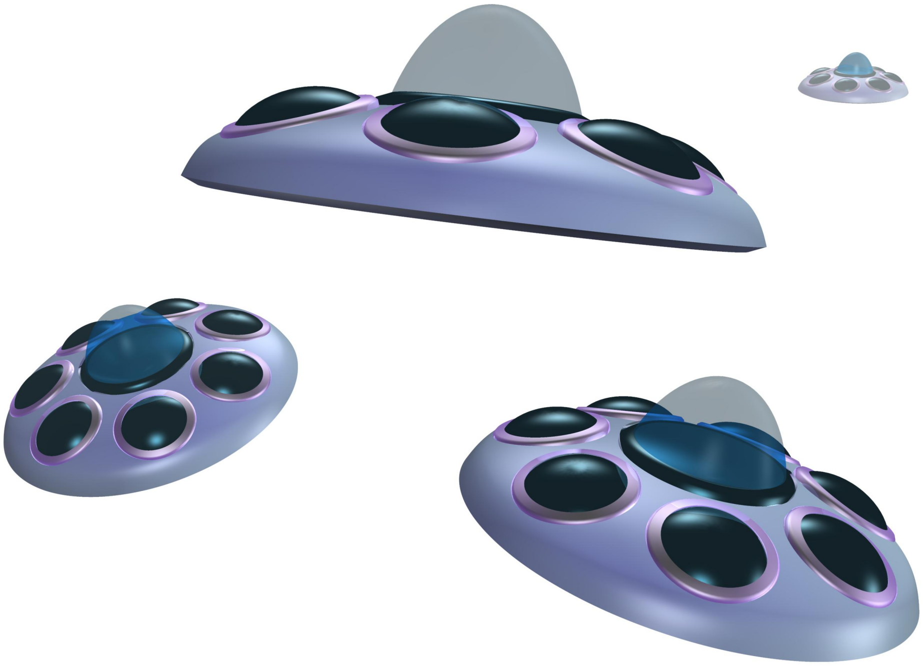
The maximum key-frame interval is limited to 60 frames when *Fastest* performance/quality mode is selected.

>CLI

Maximum key-frame interval:

-key <*frames*>

Frames is the maximum interval between key-frames (intra-frames), evaluated inclusive of the current frame.

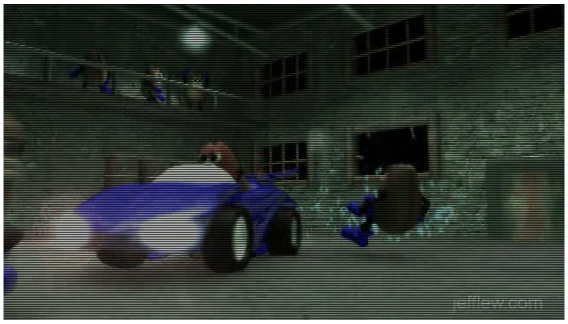


Interlacing

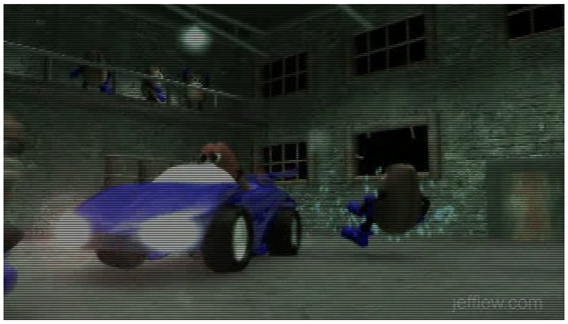
Interlacing

What is interlacing?

Interlacing is a method adopted from the old days of analogue television that allowed the frame rate of a video to be doubled by broadcasting only half a frame at a time. This was achieved by dividing each frame into two fields, each occupying alternating horizontal lines. We call these the top and bottom fields.



Top field



Bottom field

Here you see two fields composing one frame of video. First the top field is broadcast and scanned out on odd lines by the television, then the bottom field is broadcast and scanned out on the even lines. In this way a complete picture is recreated.

jefflew.com

The complete frame

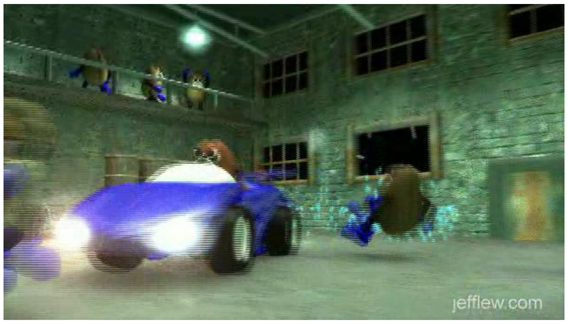
An interlaced camera may capture all the even lines in the picture first and then the odd lines a fraction of a second later, leading to jagged edges that appear where there is motion in the reconstructed frame.

Consideration of interlacing is important to DivX encoding because some sources (notably DV video and some DVDs) are interlaced and will suffer severe quality degradation and artifacting if encoded without respect to interlacing.

Why de-interlace?

During playback of an interlaced video half of one frame is continually being merged with the previous picture. If the encoder were to treat each frame as *progressive* (a complete picture that is not interlaced) then inevitably artifacting would occur when half of a new frame was merged with the remaining half of the previous frame, particularly where the new frame differs significantly from the last.

Interlacing artifacts therefore appear as two slightly different pictures being merged on alternate horizontal lines and are most visible where there is fast motion in a scene.



Interlacing artefacts

De-interlacing is the process of taking an interlaced video and recombining the fields so as to create a progressive, non-interlaced output.

Consider that if the encoder were to assume interlaced content was in fact progressive then when motion occurred in the image interlacing artifacts would be present. Greater motion causes greater artifacting and inconsistency in the texture between frames. The motion search must be able to match blocks in the current frame back to similar areas in the reference frame and an inconsistent picture will cause this process to fail. Consequently, a higher proportion of blocks will be intra-coded leading to a higher bit spend per frame, forcing the rate control to use higher quantizers and lowering the overall quality of the encoded video.

Maintaining interlacing

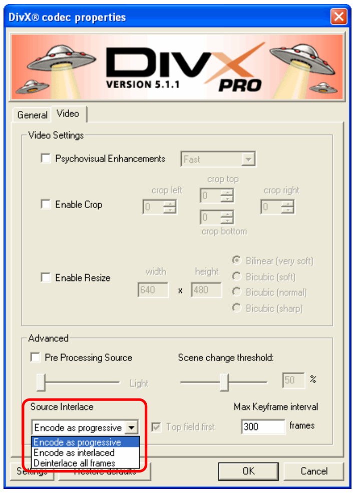
There are some circumstances where it is desirable to maintain the original interlaced fields discreetly so as they can be recreated without artifacting for display on an interlaced device. One example of such a situation is playback via a DivX Certified device to an interlaced television.

To achieve this the encoder can be set to *Encode as interlaced*. Fields will be preserved, however encoding each field individually requires a higher bitrate than is normally required when encoding a progressive source.

Source interlace

Source interlace specifies the interlacing format of the source video and how interlacing should be handled by the encoder.

QUICK GUIDE
>CLI


DivX® codec properties

Video

Video Settings

☐ Psychovisual Enhancements: Fast

☐ Enable Crop: crop left 0, crop top 0, crop right 0, crop bottom 0

☐ Enable Resize: width 640 x height 480; ● Bilinear (very soft) ○ Bicubic (soft) ○ Bicubic (normal) ○ Bicubic (sharp)

Advanced

☐ Pre Processing Source: Light

Scene change threshold: 50 %

Source Interlace: Encode as progressive (Encode as progressive, Encode as interlaced, Deinterlace all frames)

☑ Top field first

Max Keyframe interval: 300 frames

1. **Encode as progressive**
The encoder will assume that the source video is progressive (not interlaced) and no de-interlacing will be performed prior to encoding.

2. **Encode as interlaced** *Pro*
The encoder will assume that the source video is interlaced and will encode each field in every frame separately.

3. **De-interlace all frames** *Pro*
The encoder will assume that the source video is interlaced and will de-interlace it, encoding as progressive.

The *Handheld* and *Portable* profiles do not support interlaced video.

If your source video is interlaced you should select to *de-interlace all frames* to progressive unless you have a specific reason for preserving interlacing. Encoding as interlaced requires substantially higher bitrates in order to achieve equal perceptual quality to progressive content.

All video from DV video cameras is interlaced.

In DivX 5.1 the DivX decoder does not support real-time de-interlacing during playback. If you encode as interlaced the interlaced fields will be visible when played via a progressive display, such as a PC monitor.

If you wish to resize and de-interlace, you *must* de-interlace before resizing. If you use the encoders resize filter then de-interlacing is always performed before resizing. If you use an external resize filter you must also use an external de-interlace filter so that de-interlacing takes place before resizing.

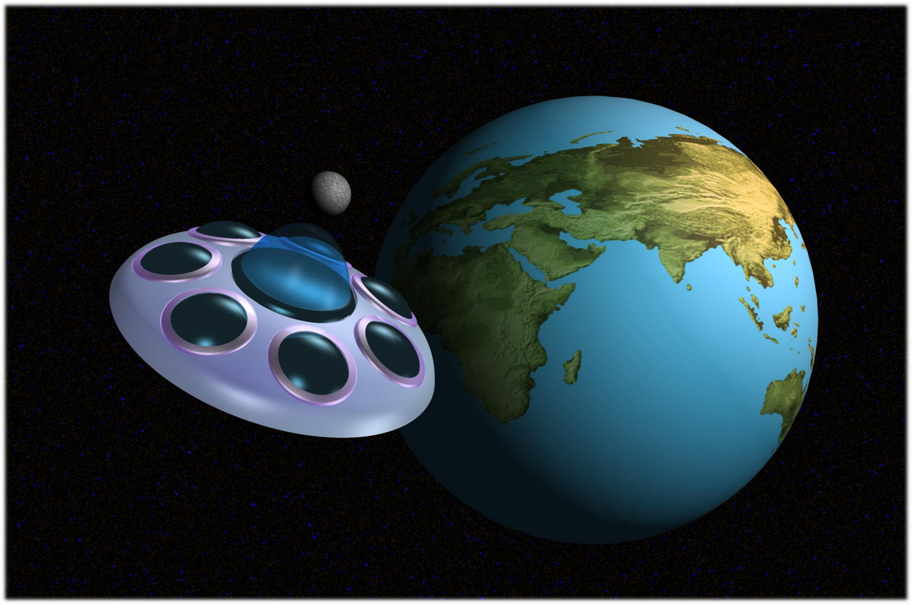
>CLI

Source interlace:

-d <*mode*>

Mode is one of:

1	Encode as progressive *(parameter may be omited)*
2	Encode as interlaced
3	De-interlace all frames



Video Buffer Verifier

The OFFICIAL Guide

Video buffer verifier

What is the video buffer verifier?

The encoding profiles available in DivX 5.1 create video that is guaranteed to play back on DivX Certified devices. To facilitate this it is necessary to control aspects of the video stream produced by the encoder in such a way that the capabilities of the target device are not exceeded. The video buffer verifier is part of this process.

The video buffer verifier forms part of a *virtual decoder* that is attached to the encoder, limiting its output in terms of certain device capabilities. These are the sustainable rate at which the device and receive the video stream, the buffer associated with the video stream within the device, and how full that buffer should be before playback begins.

When playing DivX videos using some types of devices, such as a fast desktop computer, these considerations can often be overlooked without any significant consequences. However, when encoding for broadcast over an IP network (e.g. the Internet), or for a hardware device with fixed sustainable throughput, it is critical for uninterrupted playback that the abilities of the device are not exceeded.

The video buffer verifier explained

Consider a hardware DVD player featuring DivX support. The DVD-ROM drive in the unit may have a limited sustainable transfer rate and a fixed-size cache associated with it.

Suppose a sustainable transfer rate of 2000 kbps from the drive and a 4096 kilobit read cache. If the encoder were to encode at 2500 kbps for a long period this would be 500 kbps in excess of the drives ability to read from the disc. Assuming the 4096 kilobit cache was full to begin with a 500 kbps under-run would exhaust the buffer and cause playback to pause after around 8 seconds.

As this example illustrates, it is important that the encoder considers the sustainable bitrate, buffer size and initial buffer occupancy to avoid pausing.

Video buffer verifier

The video buffer verifier prevents the encoder exceeding the ability of a target system to read and buffer the video stream as to prevent pauses in playback.

Each DivX Certified Profile has specific VBV parameters associated with it that ensure playback performance on DivX Certified devices. Although it is possible to alter the VBV parameters in a way that would not negatively impact playback performance, changing the VBV parameters is not recommended except by experienced users with specific reasons for doing so.

The default VBV parameters are:

Profile	Bitrate	Buffer	Initial occupancy
Handheld	128000	262144	196608
Portable	768000	1048576	786432
Home theatre	4000000	3145728	2359296
High-definition	8000000	6291456	4718592

The default initial occupancy is 75% of the buffer.

Using the video buffer verifier it is possible to create MPEG4 video streams suitable for streaming, even though DivX Pro does not include a streaming server itself.

By careful configuration of the VBV parameters you may be able to improve playback on older MPEG4 compatible DVD players that have not been certified by DivX Networks.

Although the initial occupancy is always obeyed by the video buffer verifier and is useful particularly at the beginning of an encoding session, it may not always be respected by a decoder.

>CLI

Video buffer verifier:

-vbv *<bitrate>, <buffer size>, <initial occupancy>*

Bitrate is the maximum sustainable read bitrate in bits per second of the target system.

Buffer size is the size in bits of the buffer associated with the video stream in the target system.

Initial occupancy is the number of bits from the video stream assumed by the encoder to have been pre-buffered by the target system before playback begins.

Profiles

Profiles

What is a profile?

As DivX grew over the years into a professional-grade standard for low-bitrate video, hardware manufactures realized its potential in the consumer market and developed devices claiming to be *DivX compatible*.

In truth these players would typically support some, but not all, versions of DivX, and then only limited features from these versions. The capabilities of hardware devices claiming DivX compatibility varied so widely that it became a hit-and-miss affair attempting to encode DivX video that could be played consistently well on all of them—a task that often seemed impossible.

DivXNetworks have therefore designed a system of *profiles* for DivX Certified™ devices—those that have been rigorously test by DivXNetworks to ensure they meet the high standards the certification program demands. Associated with each profile are a set of minimum capabilities a device must posses in order to attain certification.

Why use profiles?

By selecting one of the four available profile modes from the *Select Profile Wizard* you instruct the encoder to consider the minimum capabilities of devices certified for that profile. Encoder features unsupported by any given profile will be automatically disabled and the encoder will produce only video streams suitable for the certified device.

Creating video streams for DivX Certified devices is as easy as selecting the profile badge as displayed on the certified device from the *Select Profile Wizard* prior to encoding.

HANDHELD
DIVX®
CERTIFIED
VIDEO™
PORTABLE
DIVX®
CERTIFIED
VIDEO™
HOME THEATER
DIVX®
CERTIFIED
VIDEO™
HIGH DEF
DIVX®
CERTIFIED
VIDEO™

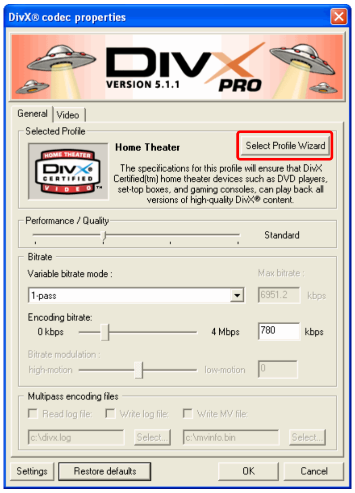
The DivX Certified™ Profiles

Profiles

QUICK GUIDE >CLI

Profiles enforce that the encoder creates a video stream compatible with DivX Certified devices.

By selecting the profile of the DivX Certified device from the *Select Profile Wizard* prior to encoding the video stream will always play correctly on the device.



DivX® codec properties

General | Video

Selected Profile

Home Theater

Select Profile Wizard

The specifications for this profile will ensure that DivX Certified(tm) home theater devices such as DVD players, set-top boxes, and gaming consoles, can play back all versions of high-quality DivX® content.

Performance / Quality: Standard

Bitrate

Variable bitrate mode : 1-pass

Max bitrate : 6951.2 kbps

Encoding bitrate: 0 kbps — 4 Mbps 780 kbps

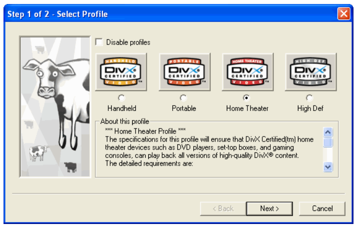
Bitrate modulation : high-motion — low-motion 0

Multipass encoding files

□ Read log file: □ Write log file: □ Write MV file:

c:\divx.log Select... c:\mvinfo.bin Select...

Any given profile may limit the available encoder options based upon the minimum requirements defined by the certification program for that profile. CLI parameters, such as the *video buffer verifier*, will also be configured automatically when a profile is selected to reflect the abilities of the target device.



Step 1 of 2 - Select Profile

□ Disable profiles

○ Handheld ○ Portable ● Home Theater ○ High Def

About this profile

*** Home Theater Profile ***
The specifications for this profile will ensure that DivX Certified(tm) home theater devices such as DVD players, set-top boxes, and gaming consoles, can play back all versions of high-quality DivX® content.
The detailed requirements are:

It is possible to encode video suitable for certified devices even when profiles are disabled by carefully choosing encoder options.

Certified devices are not required to support *Global Motion Compensation* or *Quarter-pixel* under any profile, and when profiles are disabled you should take care to set the *video buffer verifier* CLI parameters as appropriate.

Handheld profile does not support *Interlacing* or *Bi-directional encoding*.

Portable profile does not support *Interlacing*.

Quarter-pixel and *Global Motion Compensation* will always be unavailable when profiles are enabled. *Interlacing* will be unavailable when either *handheld* or *portable* profiles are selected. *Bi-directional encoding* will be unavailable when *Handheld* profile is selected.

1-Pass Quality-based mode is unavailable unless profiles are disabled.

When encoding for non-certified players that claim DivX compatibility for greatest compatibility you should use *Home Theatre* profile.

For detailed profile information see *DivX Certified™ Program—What's in a profile?*

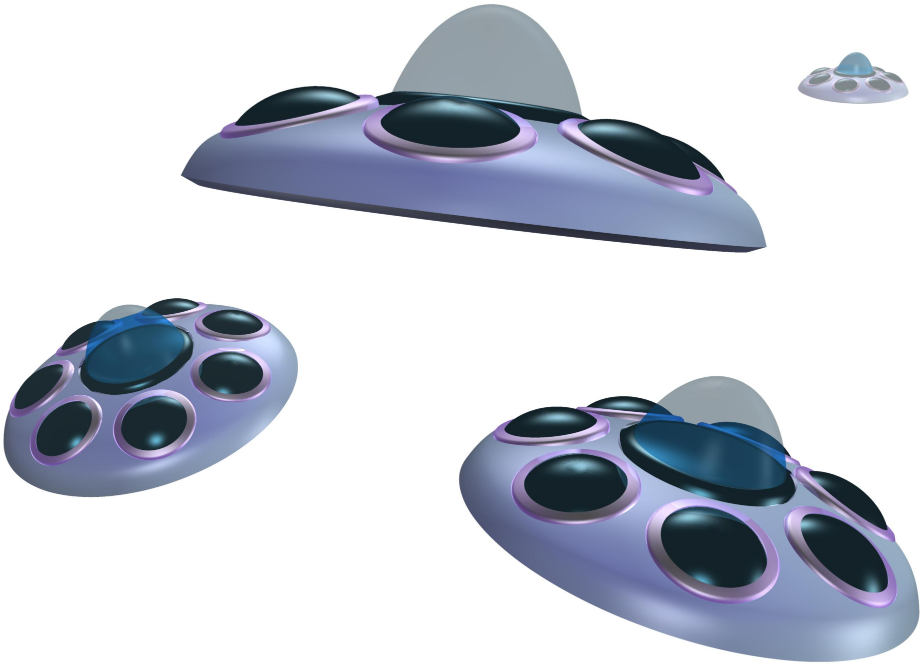
>CLI

Profiles:

-profile <*number*>

Number is one of:

0	Disable profiles
1	*Handheld* profile
2	*Portable* profile
3	*Home Theatre* profile
4	*High-definition* profile



DivX Certified™ Program

The OFFICIAL Guide

DivX Certified™ Program

What is the DivX Certified Program?

The certification program was developed by DivX Networks to ensure the ability of certified devices to play all versions of DivX video without problems.

DivX Certified devices come in various forms, each has a suitable profile associated with it that defines its abilities.

►Example Certified Devices

HANDHELD
DivX®
CERTIFIED
VIDEO™

Cell phones, low-end camera phones, PDAs

PORTABLE
DivX®
CERTIFIED
VIDEO™

Multimedia jukeboxes, video cameras, still cameras

HOME THEATER
DivX®
CERTIFIED
VIDEO™

DVD players, PVRs (DVD players and recorders), some types of vide camera

HIGH DEF
DivX®
CERTIFIED
VIDEO™

High-definition DVD players, HDTV PVRs

Every certified device is extensively tested by DivXNetworks before being awarded certification. Only devices that say "DivX Certified" and carry one of the four official certification logos have been tested by DivXNetworks as capable of playing all versions of DivX video content.

If a device does not display the logo then do not assume that it is capable of playing all versions of DivX video: *"If it doesn't say it, it doesn't play it".*

What's in a profile?

Each profile is associated with a set of minimum features a device must support in order to achieve certification.

The following table lists profile-specific features.

	HANDHELD DivX® CERTIFIED VIDEO™	PORTABLE DivX® CERTIFIED VIDEO™	HOME THEATER DivX® CERTIFIED VIDEO™	HIGH DEF DivX® CERTIFIED VIDEO™
Maximum resolution	176 x 144 15 fps	352 x 240 30 fps 352 x 288 25 fps	720 x 480 30 fps 720 x 576 25 fps	1280 x 720 30 fps
Macroblocks per second	1485	9900	40500	108000
Maximum average bitrate	128 kbps	768 kbps	4000 kbps	8000 kbps
Maximum peak bitrate during any 3 seconds of video	800 kbps	4000 kbps	8000 kbps	32000 kbps
Minimum VBV buffer size	32 KB	138 KB	384 KB	768 KB
Bi-directional encoding support	No	Yes	Yes	Yes
Interlaced video support	No	No	Yes	Yes

Devices can exceed these minimum requirements but you should never rely upon this when performing profile encoding.

By selecting the correct profile of the target DivX Certified device from the *Select Profile Wizard* prior to encoding you ensure the video stream will always play correctly on the device.

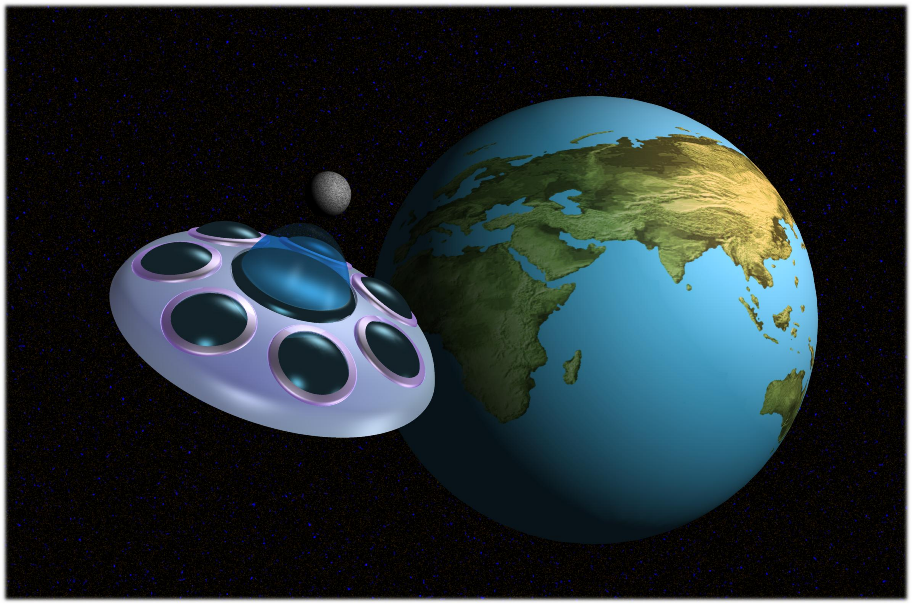
DivX Certified Program requirements

Aside from profile-specific features, there are requirements that *all* DivX Certified devices must meet.

The following table lists the generic requirements of certified devices.

Feature	Required
All DivX 3.11 movies on 1 CD, anything under 1 mbps average bitrate	Yes
All DivX 4 content	Yes
DivX 5 content with no GMC and no QPel	Yes
DivX video created for Video on Demand	Yes
DivX video created on a DivX Certified encoding device	Yes
MP3 audio in DivX video both CBR and VBR	Yes
DivX 3.11 movies on 2 CDs (high bitrates)	No
DivX 5 content with GMC or QPel	No
XVID content	No
ADPCM audio, PCM audio, Ogg Vorbis audio	No
AVI files with bad audio/video interleaving	No

Content that is not supported by certified devices can be effortlessly converted using Dr DivX (http://www.DivX.com/divx/drdivx).



E.K.G.

Electrokompressiongraph

What is the EKG?

We saw in *Bitrate mode—Multipass* that after each pass of a Multipass encoding a log file is written containing an analysis of the video encoded during the pass, used in turn to optimize the rate-control strategy for any successive pass.

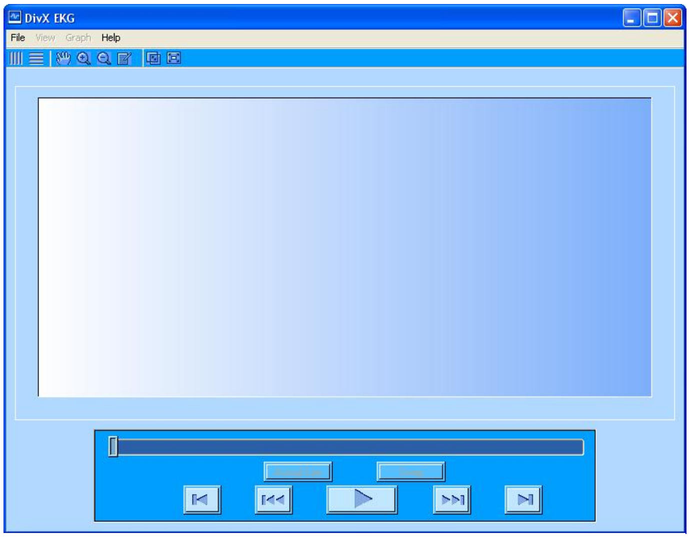
Normally the rate-control will use the information from the log file in an unbiased fashion, lending equal weight to all frames and basing frame quantizers upon the bitrate and frame complexity. However, the encoder itself can not conceive of the actual content of a video as humans can, or accurately determine how the human eye will perceive the quality of any particular sequence of video to be.

At low-bitrates it is often desirable to intentionally bias the rate-control as to raise or lower the quality of a particular section of video. For example, when encoding a movie in its entirety it is likely the viewer will be less concerned with the quality of the titles than that of the movie itself. It would thus be desirable that the titles are encoded with a higher quantizer (lower quality), and the bits freed by doing so used to enhance the quality of the actual movie. Alternatively, where a particular video sequence demands exceptional quality it is desirable to bias the rate-control in such a way that a lower quantizer is selected and bits are drawn from the remainder of the video in order to enhance the sequence.

The EKG makes this possible by allowing the data in the log file to be displayed and manipulated graphically between passes, displaying the video associated with the data for ease of use.

The EKG offers fine tuning unavailable in other encoding software.

A complete description of the log file data was covered in *Bitrate mode—Multipass*.



The Electrokompressiongraph

How the EKG works

As discussed in *Forward-General concepts-Quantizers*, DivX uses quantizers to control the picture quality in each frame. Lower quantizers equate to higher quality and bit-spend, while higher quantizers equate to lower quality and bit-spend.

The aim of the EKG application is to control the *modulation* value associated with each frame in the multipass log file. This modulation value actually manipulates the frame quantizer chosen by the rate-control by acting as a co-efficient.

Frame quantizer = Modulation * *Quantizer*

Thus if the rate-control would normally encode a particular frame at Q=6 but a modulation of 2.0 was specified the frame would instead be encoded at Q=12. A modulation of 1.0 would leave the original quantizer unchanged (1.0 x Q = Q), and a modulation of 0.5 would halve the quantizer so that the frame would be encoded at Q=3.

Notice that the EKG can not show the actual quantizer for any particular frame as this is a decision made on-the-fly as video is encoded by the rate control. Modulation allows you to manipulate quality in terms of proportionality, not in terms of fixed quantizers.

Also notice that the relationship between quantizer value and quality is inverse, a lower quantizer gives rise to higher quality. Therefore, a modulation of 2.0 is the equivalent of 50% quality, while a modulation of 0.5 is the equivalent of 200% quality.

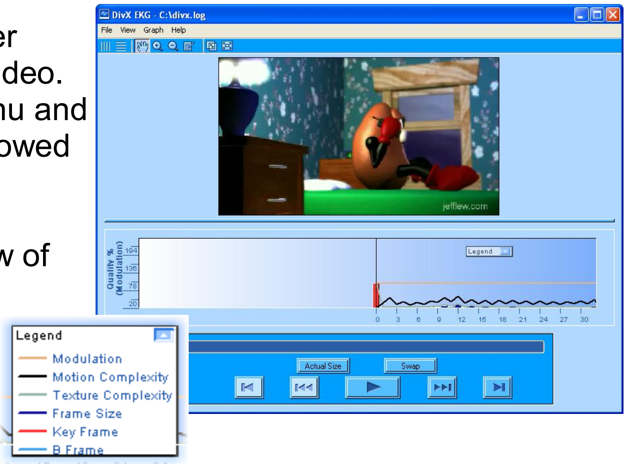
Using the EKG

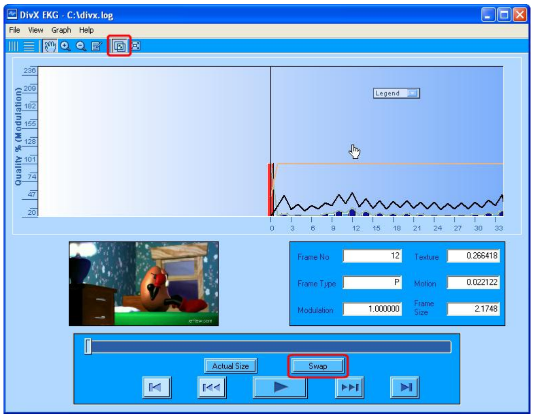
Pro

Using the EKG it is possible to display graphically the contents of the multipass log file after only the 1st pass, however in order to preview the encoded video as you manipulate the modulation control at least one nth pass must have been performed. Making best use of the EKG will therefore require at least three passes—the EKG application being used between passes two and three.

1. Open the EKG application after encoding two passes of any video. Select *Open* from the *File* menu and open the multipass log file followed by the encoded AVI file.

The EKG will display a preview of the video and graph the data from the log file. Hover the mouse over the *Legend* bar to see the graph legend.



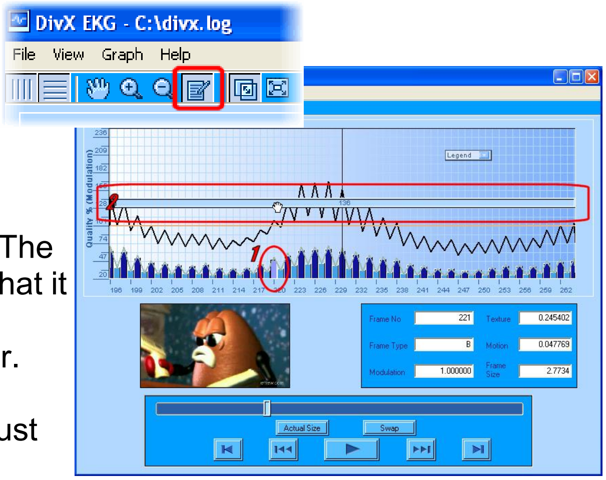


Frame No 12
Frame Type P
Modulation 1.000000
Texture 0.266418
Motion 0.022122
Frame Size 2.1748

2. Click the *Swap* button to display the contents of the log file numerically. Hovering the mouse over a frame in the graph will display its statistical information. While the drag tool (hand icon) is selected from the toolbar at the top of the EKG window you can safely drag the graph area using the left mouse button without making any changes. You can also use the zoom-in and zoom-out toolbar buttons to zoom the graph.

The two left-most toolbar buttons add grid-lines to the graph area. You can also select which statistics are graphed and how they are displayed from the *Graph* menu.

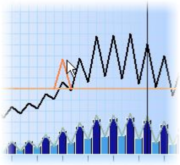
3. Click the *Edit* button on the toolbar to enter editing mode. Clicking on the graph will now operate the modulation control system.



Frame No 221
Texture 0.245402
Frame Type B
Motion 0.047769
Modulation 1.000000
Frame Size 2.7734

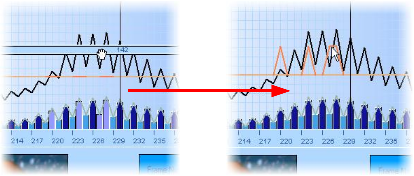
4. Click on any frame to select it. The bar will turn purple to indicate that it has been selected and the modulation grab-bar will appear.

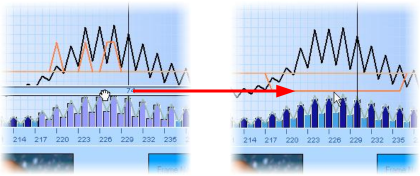
Drag the bar up or down to adjust the modulation for the selected frame. As you do so the modulated quality will appear as a percentage in the center of the grab-bar.



After you release the grab-bar the new modulation for the frame will be set and reflected in the orange modulation line that runs throughout the graph.

You can select multiple frames to modulate simply by clicking on each before adjusting the drag-bar.





You can also select an entire sequence of frames by clicking the first frame, then holding shift while you click on the last frame.

5. After making your changes, choose *Save Changes* from the *File* menu. You must now run one more nth pass to incorporate these changes into your DivX video.

Modulation settings are preserved between passes. You can revise your modulation settings at a later time if you wish.

DIVX®

DivX Decoder

The OFFICIAL Guide

DivX Decoder

About the decoder

The DivX decoder is capable of playing all DivX 3, 4 and 5 series video, applying special *post-processing* effects that enhance video quality, and making best use of the advanced features provided by your graphics hardware to improve playback performance.

The DivX decoder will be used by all third-party media players to play DivX video. You need only configure it once to impose changes on all of your media players.

It is worth noting before discussing the decoder features that some third-party MPEG4 decoder filters can commandeer DivX decoding from the genuine DivX decoder in media players that are based upon Microsoft's DirectShow platform.

You can check that the genuine DivX decoder is playing your DivX videos simply by playing any DivX video in a DirectShow based media player (such as Microsoft's *Windows Media Player*) and watching for the DivX logo in the lower right hand corner of the movie. The logo should appear for several seconds when playback begins if the genuine DivX decoder filter is in use.

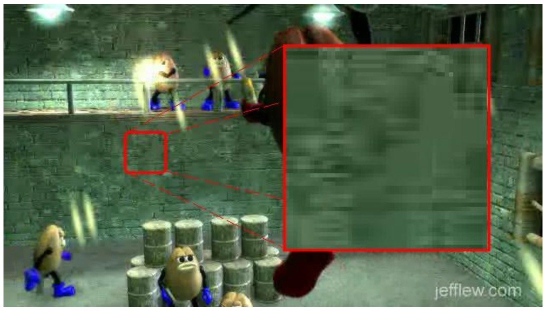
To access the decoder configuration dialogue referred to in the following pages choose the *Config.exe* program icon in the *DivX* programs group on your *Start* menu.

Post-processing

Post-processing refers to effects applied to the video after decoding to enhance the picture in some way.

The DivX decoder contains three post-processing techniques. These are de-blocking, de-ringing, and the film effect.

De-blocking



The most visible artifact occurring in low-bitrate DivX video is called *blocking*, and appears as small 8x8 squares in the video. The artifact is caused by the DCT algorithm used by DivX to encode the image, and becomes strongest where video is encoded at very low bitrates due to high quantization.

De-blocking blends the transition along the block edges so that individual blocks become less apparent.

De-ringing

Ringing is another artifact caused by the DCT algorithm. Ringing appears as an outline or shadowing around contrasting edges in the picture.

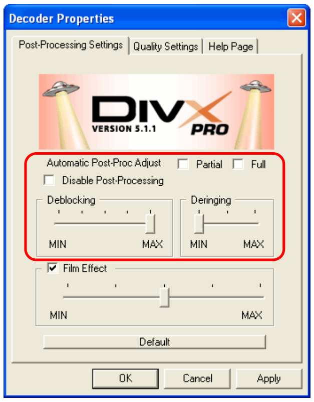
jefflew.com

Ringing is harder to see than blocking and requires more computation to correct. By default the encoder will apply full de-blocking before applying any de-ringing, depending on real-time decoding performance.

Automatic post-processing

The DivX decoder can automatically adjust the level of de-blocking and de-ringing based upon your computers real-time playback performance. This prevents the decoder dropping frames if the CPU becomes overloaded.

Automatic de-blocking is enabled when *Partial* auto post-processing is selected. Automatic de-blocking and de-ringing are enabled when *Full* automatic post-processing is selected.



Automatic Post-Proc Adjust □ Partial □ Full

□ Disable Post-Processing

☑ Film Effect

Post-processing levels

The manual de-blocking and de-ringing controls do not control the strength of the post-processing, but rather where and how it is applied.

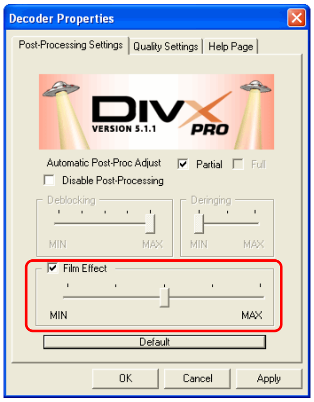
Each pixel in a DivX video is composed of three color channels—one luminance channel and two chrominance channels. The eye is most sensitive to changes in the luminance channel and so de-blocking gives preference to this.

De-blocking level	Horizontal luminance de-blocking	Vertical luminance de-blocking	Horizontal chrominance de-blocking	Vertical chrominance de-blocking
0 *(MIN)*	No	No	No	No
1	Yes	No	No	No
2	Yes	Yes	No	No
3	Yes	Yes	Yes	No
4 *(MAX)*	Yes	Yes	Yes	Yes

De-ringing level	Horizontal de-ringing	Vertical de-ringing
0 *(MIN)*	No	No
1	Yes	No
2 *(MAX)*	Yes	No

Film effect

The film effect restores warmth to the image and masks the artificial smoothing effect that can result from low-bitrate encoding by applying a film grain effect to the picture.



Decoder Properties

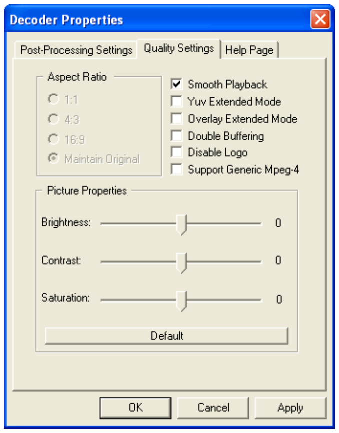
Automatic Post-Proc Adjust ☑ Partial ☐ Full
☐ Disable Post-Processing

☑ Film Effect

The film effect is particularly useful in counteracting the effects of strong source pre-processing applied during encoding.

Quality settings

The quality settings page allows configuration of decoder options relating to the rendering of DivX video.



Aspect Ratio
○ 1:1
○ 4:3
○ 16:9
● Maintain Original

☑ Smooth Playback
☐ Yuv Extended Mode
☐ Overlay Extended Mode
☐ Double Buffering
☐ Disable Logo
☐ Support Generic Mpeg-4

Picture Properties
Brightness: 0
Contrast: 0
Saturation: 0

►Smooth playback

When a DivX video contains bi-directional frames smooth playback makes use of a buffer so the decoder need only decode one frame for every frame it displays.

If the frame sequence were 1I, 2P, 3B, 4P, 5B the decoder would decode and buffer 1I, decode 2P and display 1I, decode 3B and display 3B, decode 4P and display 2P, decode 5B and display 5B, ...

The last frame of the video may be dropped due to the buffer but because only one frame is decoded at a time smooth playback enhances performance for slower CPUs.

With smooth playback disabled b-frames are decoded at the same time as the frames they reference and then displayed at the appropriate time.

If the frame sequence were 1I, 2P, 3B, 4P, 5B the decoder would decode and display 1I, decode 2P and 3B and display 3B, wait and display 2P, decode 4P and 5B and display 5B, wait and display 4P.

No frames are dropped when smooth playback is disabled but greater demands are placed upon the CPU.

For an explanation of frame orders refer to *Bi-directional encoding—The ordering of frames*.

►YUV Extended

When enabled the decoder will output video in YV12 color mode. YV12 is the internal color format used by DivX and thus is the fastest method of decoding as data can be sent directly to the video card.

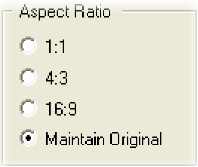
When YUV Extended is enabled the *Brightness*, *Contrast* and *Saturation* controls are unavailable.

►Overlay Extended

When enabled the decoder will attempt to use the hardware overlay instead of the software overlay. The hardware overlay is a feature of some graphics cards that increases both rendering performance and quality when video is scaled.

When overlay extended is enabled only one video file may be played concurrently.

Overlay extended mode allows you modify the pixel aspect ratio of the display.



Aspect Ratio

○ 1:1
○ 4:3
○ 16:9
● Maintain Original

►Double Buffering

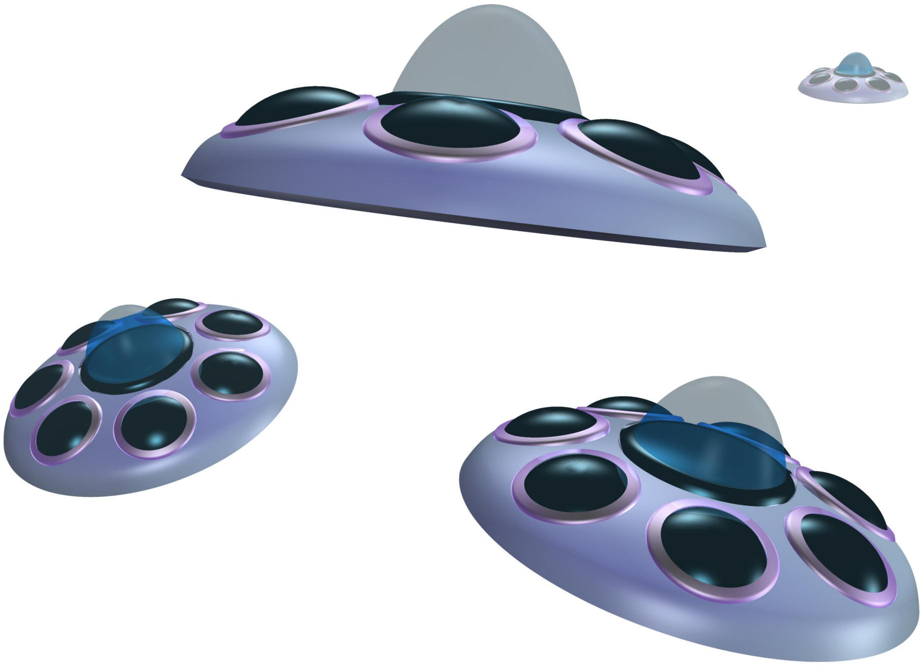
Double buffering causes the video card to reserve additional memory for receiving video data. Double buffering improves the smoothness of playback but may not work with older graphics card that have very little onboard memory.

►Disable Logo

When enabled this option will prevent the decoder overlaying the *DivX Video* logo in the lower right-hand corner of the video during the first few seconds of playback.

►Support generic MPEG4

When enabled the decoder will support playback of various MPEG4-based video formats, including XVID.



Acknowledgements

Acknowledgements

Thanks and credits

Permission to illustrate this guide with images taken from *Killer Bean 2* was very kindly given by Jeff Lew. See his work online and read about his upcoming DVD, *Learning 3D Character Animation*:

http://www.jefflew.com

Thanks to the entire DivX Advanced Research Centre (DARC) team for their assistance in writing this guide:

- Eugene Kuznetsov
- Andrea Graziani
- John Funnell
- Adam Li
- Adrian Bourke
- Cheng Huang
- Jérôme Rota
- Darrius Thompson

Thanks also to Tom Huntington (Corporate Communications Manager) for his editorial review and Scott Green (Art Director) for his cover artwork and other illustrations.

- Alastair Mayo (DigitAl56K)

Legal