

Wissenswertes rund um 3ivx

(by Selur)(Version 0.0.7)



(3ivx is an international trademark by 3ivx Technologies)

0. Allgemeines

1. Der Videocodec

1.1 Basic options

1.1.1. Encoder Mode

1.2 Advance Options

1.2.1. Keyframes

1.2.1. Encoder Features

1.2.3. Pixel Aspect Ratio

2. 3ivx Config

2.1 3ivx D4 Audio Decoder

2.2 3ivx D4 Audio Encoder

2.3 3ivx D4 Media Muxer

2.4 3ivx D4 Media Splitter

2.5 3ivx D4 Video Decoder

2.5.1. 3ivx D4 Video Decoder

2.5.2. Direct Show Priority

3. 3ivx in Graphedit

3.1 wav/mp3/ogg=> aac

3.2 avi (mp4Video&mp3) => mp4 (mp4Video&aac)

3.3 Anmerkungen zu anderen Umwandlungen

4. Anhang

4.1 Danksagungen

Anmerkung:

Da es sich hierbei um die ersten Versionen dieses Dokumentes handelt, nicht wundern wenn was nicht so 100%ig passt. Korrekturen kann man mir gerne über das offizielle Flaskmpeg&DVDtoOgm – Board (<http://www.flaskmpeg.info/board>) oder das deutsche Doom9&Gleitz – Board (<http://doom9.gleitz.de>) zukommen lassen. ;)

0. Allgemeines:

@3ivx Technologies:

Wollte nur anmerken, dass ich mich sehr freuen würde wenn ihr mich nicht aus einem wie auch immer gearteten Grunde wegen diesem Dokument oder irgendetwas was ich in ihm schreibe / schreiben werde verklagen würdet. Da ihr meine Email habt und man bis dato gut miteinander ausgekommen ist würde ich mich freuen etwaige Beschwerden, ohne Verklagt zu werden aus der Welt zu schaffen.

Wieso ist dieses Dokument entstanden?

Nachdem "Wissenswertes rund um Xvid" durchaus Anklang gefunden hat und ich durch Zufall einer der Beta-/Seeding-Tester des Codecs war, habe ich mich entschieden zu versuchen, etwas Vergleichbares auch für 3ivx zu schreiben. In der Hoffnung, dass es möglichst vielen Leuten ein bisschen hilft.

Was ist 3ivx?

Vor allem an dieser Stelle erscheint mir wichtig zu erwähnen, dass 3ivx nicht nur ein Videocodec sondern ein komplettes Filterpaket ist, mit dessen Hilfe man nicht nur mp4 Video, sondern auch mp4 Audio und auch echte ISO-konforme mp4-Files erstellen kann.

3ivx ist ein Produkt von 3ivx Technologies, einer australischen Firma, welche sich vor allem aus Mpeg-4-, Hardware- und Software- Spezialisten, Designern und Entwicklern zusammensetzt. Auf genauere Informationen zur Firma und ihrer Entstehungsgeschichte will ich jetzt nicht eingehen, da sie eine sehr informative Homepage haben. (<http://www.3ivx.com>)

Wie spricht man 3ivx aus?

3ivx wird auf Englisch "Thrivex" ausgesprochen. Wer mir dies nicht glauben will kann aber gerne mal (auf eigene Kosten) bei den 3ivx-Leuten anrufen und sich anhören wie sie ihren Firmennamen aussprechen.

1. Der Videocodec:

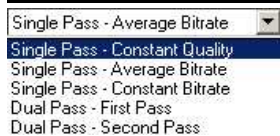
Wie Xvid und Divx ist der 3ivX Videocodec auch ein Mpeg4 Videocodec. Sicher wäre es interessant und auch von einigen Lesern dieses Schriftstückes gewünscht, dass ich hier einen kleinen Vergleich der Fähigkeiten der Codecs vollführe. Dies werde ich jedoch nicht tun, da ich einerseits der Ansicht bin, dass man aus Codecvergleichen nur Hinweise aber keine Weisheiten ziehen kann und andererseits auch einfach nicht die Lust und Zeit habe einen ausführlichen Codectest, der den einzelnen Codecs gerecht werden würde, durchzuführen.

Vielmehr möchte ich hier kurz die Optionen erläutern die einem User zur Verfügung stehen, wenn er z.B. in Anwendungen wie VirtualDub, DVDtoOgm, GordianKnot, Flaskmpeg, Xmpeg, usw. bei den Videoeinstellungen des Codecs findet.

1.1. Basic Options:

In den Basic Options werden die Grundeinstellungen des Codec vorgenommen. Hier stellt man die Datenrate, die minimale und maximale 'Qualität' und den etwaigen Encoding Mode ein.

1.1.1. Encoder Mode:



Genauso wie auch die bereits schon erwähnten Codecs (Xvid&Divx) unterstützt auch 3ivx, einige grundlegenden Encoding Arten zu denen ich nun kleine Erläuterungen gebe.

Single Pass – Constant Quality:

In dieser Einstellung wird der gesamte Videostream mit einer konstanten Qualität, mit einem konstanten Quantizer encoded. Ein kleiner Quantizer steht hierbei für kleine Rundungsfehler und damit für eine hohe Genauigkeit und eine größere Dateigröße. Prinzipiell würde ich diese Einstellung vor allem benutzen wenn man schnell etwas mit einer bestimmten Qualität encoden will. Vor allem wenn man sich entscheidet ein Videosignal anfangs mit maximaler Qualität zu encoden um es später noch weiter zu bearbeiten ist dieser Mode empfehlenswert.

Single Pass – Average Bitrate:

In diesem Modus wird der Codec versuchen zwischen einem maximalen und einem minimalen Quantizer-/Qualitätswert zu bleiben und gleichzeitig durchschnittlich eine gewisse Datenrate zu halten. Dies ist vor allem dann nützlich wenn es schnell gehen muss und man zwar in etwa bei einer gewissen Datenrate (=> Dateigröße) bleiben will, jedoch nicht die Zeit hat einen Dual pass Encodingvorgang durchzuführen. Zwar wird man im Allgemeinen nicht so gute Ergebnisse erreichen wie mit einem Dual pass Verfahren, jedoch erhält man normalerweise ein sehr gutes Ergebnis bei einer abschätzbaren Zielgröße mit nur einem Durchgang. Wichtig in dieser Einstellung sind vor allem *Maximum-/Minimum Quality* und die *Average Bitrate*. (gehe auf beides etwas später ein)

Single Pass – Constant Bitrate:

Wählt man diese Einstellung so wird der Codec versuchen etwaiges Bildmaterial immer mit der gleichen Bitrate zu encoden, was vor allem für Material wichtig ist, welches über Datenkanäle mit einer beschränkten Kapazität übertragen werden soll. Notfalls werden auch einzelne Bilder verworfen falls es anders nicht möglich ist die Datenrate zu halten bzw. einen konstanten Datenstrom zu gewährleisten. (z.B. weil der PC auf dem encoded wird zu langsam ist.)

Allgemeines zu Dual pass Encodingverfahren:

Allgemein versucht der Codec, wenn man ein Dual pass-Verfahren wählt, die subjektive Qualität konstant zu halten und gleichzeitig nicht über die gewünschte Dateigröße zu geraten. Im *First pass* - Pass wird die Quelle analysiert und etwaige Erkenntnisse gespeichert. Diese werden dann im *Second pass* benutzt um die vorhandene Datenrate möglichst ideal auf die einzelnen zu komprimierenden Szenen zu verteilen. Wichtig ist das immer erst der *First pass* durchgeführt werden muss vor dem *Second pass*, da sonst keine gesammelten Daten/Erkenntnisse zur Verfügung stehen. Auch sollten zwischen dem First und Second pass keine der *Advanced Options* geändert werden, da dann die gesammelten Daten durchaus nicht mehr akkurat sein können.

Dual Pass – First Pass:

In diesem Pass wird das zu komprimierende Material mit maximaler Qualität gespeichert. Standardmäßig wird hierbei keine Datei erstellt, aktiviert man aber die Option *Output Video During First Pass*, so wird eine, je nach Input sehr große Datei mit maximaler Qualität erstellt. Während dieses Encodingvorganges wird die Datei und ihre Komprimierbarkeit analysiert und so gesammelte Daten in eine Datei gespeichert, deren Pfad man unter *First Pass Log* angeben kann.

Dual Pass – Second Pass:

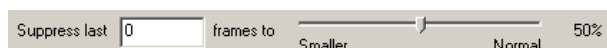
Im zweiten Durchlauf werden nun die Daten aus dem ersten Durchlauf verwendet um eine möglichst ideale Verteilung der Datenrate zu erlangen.



Hier hat man nun die Möglichkeit zu wählen ob diese Verteilung darauf abzielen soll möglichst ideale Qualität oder eine möglichst konstante Datenrate zu erreichen. Dies geschieht, indem man bei *Dual Pass Goal* entweder *Best Quality* oder *Constant Bitrate* auswählt. Unter *Dual Pass Bitrate* kann man die angestrebte durchschnittliche Datenrate in kByte pro Sekunde angeben.

Wichtig ist hierbei, dass 3ivx mit $k = 1024$ rechnet und nicht wie z.B. Xvid mit $k=1000$. Da ein Byte acht Bit entspricht kann man jedoch einen DivX3.11;) Calculator zur Hand nehmen und das Ergebnis einfach durch 8 teilen.

Wenn man auf *Best Quality* abzielt hat man auch noch die Möglichkeit anzugeben, dass eine gewisse Anzahl an Bildern am Ende des Inputs mit einer niedrigeren Datenrate encoded werden.



Persönlich bevorzuge ich die Datenrate auf etwa 15 – 20% für den 'Abspann' zu senken.

*So, jetzt noch ne kleine Anmerkung und dann war's das
auch von meiner Seite zu den Basic Options.*

Nicht nur der Codec ist ausschlaggebend wie gut das Ergebnis ist

Die Anwendung mit der man Encoded bzw. deren Filter (Deinterlacer, Resizer&Co) können auch einiges an der Qualität des Resultates verändern. Wenn man z.B. eine niedrige Datenrate anstrebt ist es oft sinnig das Material etwas weich zu zeichnen (smoothen), da man so dem Codec die Arbeit erleichtert, dadurch dass man weniger Details im Bild hat, die man wahrscheinlich eh nicht sehen würde.

Wichtig ist auch das Ausgangsmaterial auf dem der Codec arbeiten soll. Dunkle, detailarme und langsame Szenen lassen sich besser komprimieren als helle detailreiche Szenen. Wundert euch also nicht wenn ihr mit euren Standardeinstellungen nicht immer gleich gute Ergebnisse habt.

1.2. Advanced options:

In den Advanced Options geht es wie der Name schon sagt um die Erweiterten Einstellungen von 3ivX. Zu beachten ist hier, dass *Mpeg-Quantizer* zwar die Bildqualität verbessern, jedoch vom verwendeten Decoder die Unterstützung des *Advanced Simple Profiles* verlangt, was bei der Wiedergabe am PC nicht zu Problemen führen sollte jedoch bei Hardwareplayern zu beachten ist.

1.2.1. Keyframes:

Das hier festzulegende Intervall gibt an, nach wie vielen Frames spätestens ein I-/Key-Frame gesetzt werden soll. Einerseits ist es aus kompressionstechnischen Gründen sinnvoll nur an Szenenwechseln bzw. in Szenen wo starke Änderungen zu den Vorgängern auftreten ein I-Frame zu setzen, d.h. man sollte nicht zu viele I-Frames setzen. Andererseits werden I-Frames auch als Sprungpunkte benötigt, d.h. man will nicht zu wenige, da man sonst nur in groben Sprüngen im Film umherspulen bzw. springen kann. Als guter Richtwert hat sich ein Wert eingebürgert der etwa dem 10fachen der Framerate entspricht. Für PAL Filme wäre dies ein Wert von 250. (NTSC =240 bzw. 300) Wichtig ist zu beachten, dass ein Wert von 0 zur Folge hat, dass jedes Frame ein Keyframe ist.

1.2.2. Encoder Features:

Hierbei handelt es sich um zusätzliche Features welche beim Encoden maßgeblichen Einfluss auf die Qualität und die Geschwindigkeit beim Encoden haben können.

Half Pixel Motion:

Dieses Feature sollte meiner Erfahrung nach immer aktiviert sein, da der Codec dann eine genauere Bewegungssuche durchführt. Genauer gesagt wird nicht mehr nur jedes Pixel sondern jedes halbe Pixel betrachtet, was zu besserer Kompression und Qualität führt.

Four Vector Motion:

Wie *Half Pixel Motion* sollte auch dieses Feature immer aktiviert sein da dazu führt das 16x16 Macroblöcke in vier 4x4 Macroblöcke eingeteilt werden und für jeden einzelnen der 4x4 Blöcke wird ein einzelner Motionvector gespeichert. Dies führt zu besserer Kompression und dadurch auch besserer Bildqualität bei gleicher Datenrate.

MPEG Quantizer:

Hierbei handelt es sich um ein Advanced Simple Profile Feature von 3ivx, welches den Codec dazu veranlasst einen alternativen Quantizertyp zu verwenden. *Im Allgemeinen würde ich empfehlen dieses Feature zu aktivieren*, da sie meist ein schärferes Bild erzeugen als die Standardquantizer (h.263). Zu beachten ist jedoch, dass im Gegensatz zu den normalen (h.263) Quantizern, MPEG Quantizer auch eher dazu tendieren bei niedrigeren Datenraten zur Macroblockbildung zu tendieren.

Adaptive Quantisation aka. Lumi Masking:

Adaptive Quantisation betrachtet die absolute Helligkeit einer Szene.

Aktiviert wird es wenn eine nicht (fast) strahlend weiße oder (pech-) schwarze, Szene betrachtet wird.

In solchen Szenen im mittleren Helligkeitsbereich weist Adaptive Quantisation Texturen die nicht leicht mit den Augen zu erkennen sind einen höheren Quantizer zu und versucht so Datenrate zu sparen. *Wenn man relativ niedrige Datenraten anstrebt würde ich empfehlen dieses Feature zu aktivieren, persönlich aktiviere ich es immer. Verwendet man das DualPassverfahren wird dieses Feature automatisch nicht benutzt, auch wenn man es aktiviert hat.*

1.2.3. Pixel Aspect Ratio (PAR):



Das Pixel Aspect Ratio (Bildpunkteverhältnis) beschreibt das Verhältnis zwischen der Breite und Höhe eines zu encodenden Bildpunktes. Anfangs mag sich dies relativ unnütz anhören, tat es zumindest für mich, wenn man aber ein bisschen darüber nachdenkt ist es recht praktisch insofern der Player den man verwendet die Information auch wieder aus dem Videostream lesen kann. Ist ein Player hierzu in der Lage kann über diese Einstellung festgelegt werden wie das Seitenverhältnis eines Films sein soll, was einem wiederum ermöglicht anamorph zu Encoden.

(=> man kann sich das Resizen und damit verbunden Qualitätsverlust beim Encoden sparen)
Zur kurzen Erklärung sei noch angemerkt, dass 1:1 das Standard-Computer-Aspect-Ratio ist. Alle anderen Optionen sind denke ich selbsterklärend. Wichtig ist eventuell noch für Leute die auf Custom Pixel Aspect Ratio (PAR), dass man nur Werte zwischen 1 und 255 gewählt werden können.

Dies waren eigentlich alles was mir so an erwähnenswerten Dingen zu den Advanced Options und zum 3ivX Videocodec an sich eingefallen sind.

Alles was jetzt noch kommt hat nichts mehr direkt mit dem Videocodec zu tun sondern mit dem Filterpack und dessen Verwendungen.

2. 3ivx Config:

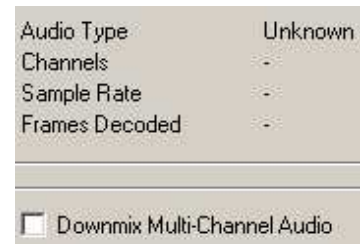
(bei Windows Systemen über Start => Programm => 3ivx D4 4.5 => 3ivx Config zu erreichen)

In der 3ivx Config hat man die Möglichkeit einige grundlegende Einstellungen der neben dem Videocodec im Filterpack enthaltenen Filter vorzunehmen. Meist wird man hier zwar nichts groß ändern, jedoch denke ich das ich lieber mal was dazuschreibe ehe ich noch dauernd diesbezüglich gelöchert werde.

2.1. 3ivx D4 Audio Decoder

Der Audio-Decoderfilter erlaubt den verschiedenen Media Player (MPC, BsPlayer, WMP,...) AAC (advanced audio codec) und HE-AAC abzuspielen. Bei AAC und der Erweiterung HE-AAC handelt es sich um den sehr effizienten Standardaudiocodec für Mpeg4. Im Gegensatz zu einigen anderen Filtern die den Playback von AAC beherrschen werden alle gängigen Profile (LC, HE, LTP, MP) und MultiChannel Audio unterstützt. (anderer guter Filter ist CoreAAC)

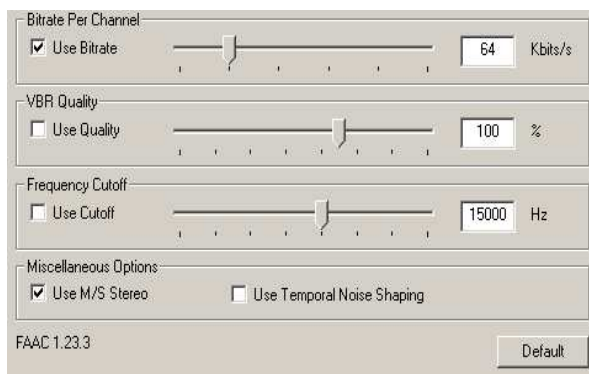
Spielt man gerade einen (HE-)AAC Stream ab kann man hier ablesen ob es sich um AAC oder HE-AAC handelt, wie viele Kanäle, welche Samplerate (32/44.1/48Hz) der Audiostream hat und wieviele Audioframes schon decoded wurden.



Aktiviert man die *Downmix Multi-Channel Audio* – Option so wird ein Multi-Channel Eingangssignal auf ein Stereosignal umgewandelt, was dann weiter an die Boxen/Anlage geleitet wird.

2.2. 3ivx D4 Audio Encoder:

Hier kann man einige generelle Einstellungen des AudioEncoderFilters (FAAC) vornehmen, den man z.B. mit Graphedit verwenden kann um verschiedene Audioformate nach AAC umzuwandeln.



2.2.1. Bit Per Channel

Mit dieser Option legt man fest wieviel Datenrate **pro Kanal** verwendet wird, ein Wert von 64kBit/s würde also bei einem Stereosignal zu einer effektiven Datenrate von 128kBit/s führen. Man bedenke hierbei auch, dass AAC im Vergleich zu z.B. mp3 bei gleicher Datenrate eine bessere Qualität liefert. Persönlich würde ich nicht mehr als 64kBit/s pro Kanal wählen, da zumindest ich dann den Unterschied zum Original nicht mehr höre.

Momentan funktioniert es nicht wenn man in das Kästchen direkt einträgt, aber mit dem Slider geht's ohne Probleme.

2.2.2. VBR Quality:

Use Quality aktiviert den VBR Modus (= variable Bitrate) von FAAC. Dieser liefert eine bessere Qualität als der ABR Modus, welcher benutzt wird falls man stattdessen "Bitrate per Channel" aktiviert, um z.B. die endgültige Dateigröße für den Audioteil genau zu definieren oder wenn man MP4-Dateien live streamen will, was ebenfalls besser klappt, wenn die Bitrate möglichst konstant bleibt. Bei 2 Kanälen entspricht 100% etwa 128kBit/s. Bei einem 5.1 Stream mit 150% Quality wäre man etwa bei 450kBit/s. *Persönlich bevorzuge ich 110%*. Für Movie-Soundtracks reicht sehr wahrscheinlich auch eine Qualität von 80% mit einem Cutoff bei 13 kHz, was dann auch zu einer ziemlich guten Verringerung der Bitrate/Dateigröße führt (etwa 100 kbps/stereo = 250 kbps/6-channel).

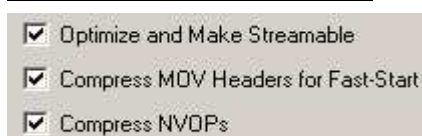
2.2.3. Frequency Cutoff:

Diese Option muss man nur anfassen, wenn man etwas anderes als 100% Quality nimmt. In solchen Fällen sollte der Cutoff pro +/- 10% um +/- 1000Hz geändert werden. (150% => 20 000Hz, 50% => 10Hz) Der Sinn der Option ist bei AAC, die Bitrate weiter zu senken (hängt sowohl von -q als auch von -c ab, nicht nur von -q) und dabei noch einigermaßen gut zu klingen. Eine krasse Fehleinstellung wäre z.B. 50% Qualität bei standardmäßigen 15 kHz Cutoff, weil bei einer so geringen Bitrate so hohe Frequenzen gar nicht mehr vernünftig kodiert werden können.

2.2.3. Miscellaneous Options:

Aktiviert man *Use M/S Stereo* so werden nicht wie normalerweise der linke und der rechte Kanal separat verarbeitet, sondern in ein Monosignal (Rechts + Links) und in ein Stereozusatzsignal (Links - Rechts) überführt. Dadurch sind mehr Bits für den "Mittenkanal" zur Verfügung. Wodurch man normalerweise eine besser Qualität bei gleicher Datenrate erhält. Durch die Option *Use Temporal Noise Shaping* (TNS) wird besonders die Sprachqualität bei niedrigen Bitraten verbessert. Leider wird sie aber manchmal auch verschlechtert, deshalb ist standardmäßig die auch *TNS* deaktiviert. Eigentlich soll TNS die Auflösung für impulsive Signale verbessern helfen, also z.B. bei Zischlauten oder Schlagzeugbecken, hat aber bei früheren AAC-Versionen eher zu neuen Artefakten geführt. Wie es sich im Moment verhält und was damit zukünftig passiert, ist noch nicht klar, da ungetestet. Prinzipiell also die Finger davon lassen, wenn man nicht erstmal testen will.

2.3. 3ivx D4 Media Muxer:



In Media Muxer Einstellungen hat man die Möglichkeit einige Optimierungen zu aktivieren die immer dann zum Tragen kommen, wenn der Muxer (z.B. im Graphedit) verwendet wird um eine .mov oder .mp4 Hülle mit Inhalt zu füllen.

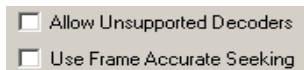
Die Option *Optimize and Make Streamable* passt den Stream so an, dass man einerseits beim Download eines solchen .mp4 Files und früher oder teilweise heruntergeladene Teilstücke angucken kann und andererseits das Springen in so einer Datei etwas flotter ist.

Compress MOV Header for Fast-Start, ist eine Option die man nur benötigt wenn man .mov (Quicktime) Dateien erstellen will. Aktiviert man diese Option in einem solchen Fall starten so erstellte Files etwas schneller als standard (unoptimierte) Quicktimedateien, da die Header komprimiert (gzip) werden.

Zur Option *Compress NVOPs* will ich folgendes sagen: NVOPs sind nicht kodierte Frames, welche benutzt werden um effizient dropped Frames oder Frames, die mit ihrem Vorgänger identisch sind zu speichern. Leider kann Quicktime momentan nicht richtig mit NVOPs umgehen (bekannter bug von Quicktime 6.4). *Prinzipiell sollte man aber wenn man nicht gerade .mov files erstellen will diese Option aktivieren.*

2.4. 3ivx D4 Media Splitter:

Der Media Splitter ermöglicht es einem z.B. in Graphedit Streams aus .mov und .mp4 files zu demuxen. Damit man auch an nicht 3ivx eigene Filter in Graphedit 'andocken' kann sollte man nicht vergessen die Option *Allow*



Unsupported Decoders aktivieren. Die Funktion *Frame Accurate Seeking* erlaubt bzw. ermöglicht es genauer in einem Stream zu arbeiten und sollte deshalb auch aktiviert werden, falls der Filter zu dem man connecten will kein Problem damit hat.

2.5 3ivx D4 Video Decoder:

Bei den Video Decoder Optionen kann man *PostProcessing* aktivieren.



Es handelt es sich dabei um relativ CPU intensive automatische Filter, welche einerseits die Schärfe etwas verbessern und andererseits das Rauschen und Makroblockbildungen verschleiern sollen. Allgemein wird empfohlen hier die Option *Use Auto-Select for Video Filter Level* zu wählen.

Da ich persönlich aber die Datenrate bei meinen Encodes hoch genug wähle, so dass sie an sich eine gute Qualität haben und etwaige Filter meist nur schaden würde ich empfehlen die Option zu deaktivieren. Natürlich kann man auch selbständig das Filterlevel festlegen, hierbei sollte man aber immer ein Auge darauf haben wie hoch die CPU Auslastung beim Playback steigt.

Ein manchmal sehr nützliche Option ist *Force Overlay*, mit ihr wird dafür gesorgt, dass ein 3ivx - Videostream auch auf jeden Fall den Videooverlay, was einem in den meisten Media Playern die Möglichkeit eröffnet, dass man die Helligkeit, Kontraste und teilweise noch mehr während der Wiedergabe ändern kann..

Unter *Direct Show Priority*, kann man festlegen wie aggressiv der Decoderfilter dafür sorgt das Material was es abspielen kann auch mit ihm wiedergegeben wird. Unter *Enable compatibility...* kann man festlegen für welche Videostreams außer 3ivx der decoder zuständig seien soll.



Persönlich installiere ich eigentlich für jeden Videostream einen einzelnen Decoder, wenn man dies aber nicht will kann man ohne bedenken hier auch das Decoden von DivX3/4/5 und Xvid aktivieren, da der 3ivx Decoder sehr flott und qualitativ hochwertig ist.

So, das war es ersteinmal zu den Audiosettings, wenn der Encoder genauer getestet wurde werden hier sicher noch einige Erweiterungen kommen. Würde mich also über Feedback freuen.

3. 3ivx in Graphedit:

Da es sich bei 3ivx ja um ein Filterpaket handelt und Graphedit der mir einzig bekannte Weg ist mit diesen Filtern einfach zu arbeiten will ich hier nun auf einige Anwendungsmöglichkeiten von Graphedit mit dem Filtern von 3ivX eingehen.

Ich gehe hierbei bewusst nicht auf die Möglichkeit ein den Videocodec von 3ivx zu benutzen, da ich persönlich denke, dass es wesentlich komfortablere Möglichkeiten gibt einen Videostream zu erzeugen.

3.1 wav/mp3/ogg=> aac:

Hier will ich nun kurz und knapp erklären wie man eine gegebene .wav Datei in ein mp4 file mit aac audio encoded.



(in etwa so sollte der Graph am Ende aussehen)

0. Graphedit starten

1. *File=>Render Media File=>* hier das zu encodende wav/mp3/ogg file aussuchen

2. das *Default DirectSoundDevice* auswählen und mit *Entf* (auf der Tastatur) löschen

3. *Graph=>Insert Filters=>3ivx Filters=>3ivx D4 Audio Encoder* auswählen und *Insert filter* drücken

4. *output* des *Wave Parsers/Vorbis Decoders/mp3 Decoders* mit dem *XForm In* des *3ivX D4 Audio Encoders* verbinden

5. *Graph=>Insert Filters=>3ivx Filters=>3ivx D4 Media Muxer* auswählen und *Insert filter* drücken

6. *XForm Out* des *3ivX D4 Audio Encoders* mit dem *Input 1* des *3ivx D4 Media Muxer* verbinden

7. *Graph=>Insert Filters=>DirectShow Filters=>File writer* auswählen und *Insert filter* drücken (hier das Verzeichnis auswählen und den Namen eingeben für das zu erstellende aac file)

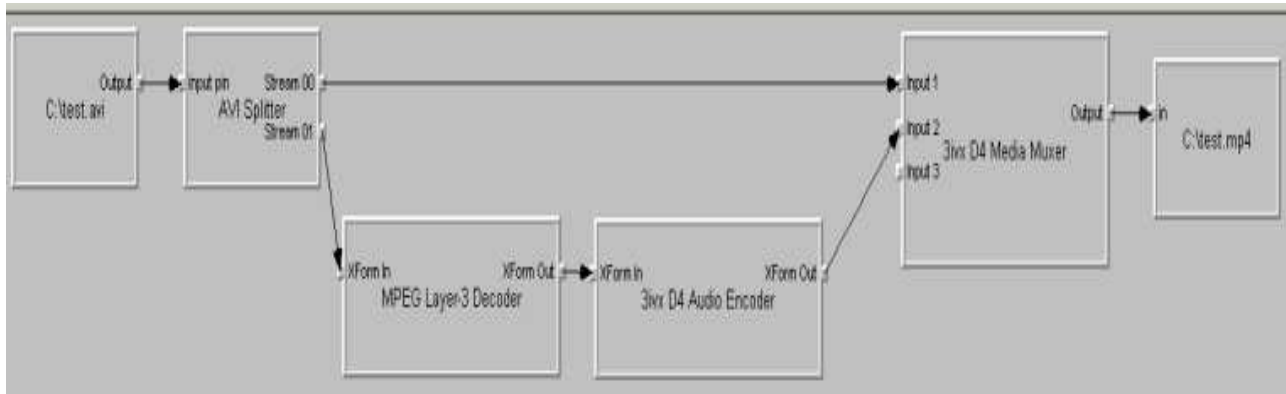
8. den *Output* des *3ivx D4 Media Muxer* mit dem *in* des *File writers* verbinden

9. *Graph=>Play*

(nun wird ein aac File an der angegebenen Stelle erstellt)

Anzumerken ist, dass man die Audio Encoder settings entweder dadurch ändern kann, dass man einen rechten Mausklick auf den 3ivx D4 Encoder macht und dann über Filter properties zu den Audio Encoder Settings kommt, oder wie im Abschnitt 2.2 macht.

3.3 avi (mp4Video&mp3) => mp4 (mp4Video&aac)



(so in etwa sollte der Graph am Ende aussehen)

0. Graphedit starten

1. *File=>Render Media File=>* hier das zu encodende wav/mp3/ogg file aussuchen
2. das *Default DirectSoundDevice* auswählen und mit *Entf* (auf der Tastatur) löschen
3. den *Video Renderer* auswählen und mit *Entf* (auf der Tastatur) löschen
4. *output* des *Wave Parsers/Vorbis Decoders/mp3 Decoders* mit dem *XForm In* des *3ivX D4 Audio Encoders* verbinden
5. *Graph=>Insert Filters=>3ivx Filters=>3ivx D4 Media Muxer* auswählen und *Insert filter* drücken
6. den Video Decoder (*Xvid/DivX/3ivx/ffdshow/..*) auswählen und mit *Entf* (auf der Tastatur) löschen
7. den *Stream 00* des *Avi Splitters* mit dem *Input 1* des *3ivx D4 Media Muxer* verbinden
8. den *XForm Out* des *3ivX D4 Audio Encoders* mit dem *Input 2* des *3ivx D4 Media Muxer* verbinden
9. *Graph=>Insert Filters=>DirectShow Filters=>File writer* auswählen und *Insert filter* drücken (hier das Verzeichnis auswählen und den Namen eingeben für das zu erstellende mp4 file)
10. den *Output* des *3ivx D4 Media Muxer* mit dem *in* des *File writers* verbinden
11. *Graph=>Play* (nun wird ein mp4 file an der angegebenen Stelle erstellt)

3.4 Anmerkungen zu Graphedit:

Ich denke die Funktionsweise von Graphedit sollte durch die zwei oberen Beispiele soweit erklärt sein, dass man mit ein bisschen Grübeln auch selbst hinbekommen sollte ogm, mkv oder sonstige Files nach mp4 zu muxen. Etwaigen Problemen wird man sich sicher gerne im offiziellen *Flaskmpeg&DVDtoOgm* und im *Doom9&Gleitz Board* annehmen. Falls hier öfter Probleme auftreten sollten, werde ich dieses Dokument dementsprechend erweitern.

4. Anhang

4.1. Danksagungen:

Erstmal möchte ich dem 3ivx Team für dieses coole Filter Suite, für die Hilfe betreffend FAAC, dem Flaskmpeg&DVDtoOgm Team für die Fehlerkorrekturen und GMB für die Screenshots danken.

4.2. Changelog:

25.11.2003: Kommenatre zu FAAC korrigiert dank Hans-Jürgen :)