

[Нажмите сюда для просмотра этой темы в оригинальном формате](#)

MotoFan.ru _ FAQ для конкурса _ Конвертирование Видео Для Просмотра На Телефоне

Автор: blackbox Среда, 28 Июня 2006, 13:44

Представляю способ качественного конвертирования видео в MPEG4 со звуком AAC с использованием open source программ.

Скачать мануал с картинками в формате CHM для просмотра в оффлайне:

[MPEG4 AAC coding.zip](#) (199.38кб) Кол-во скачиваний: 1087

Шаг 0.

Перед тем, как что-либо делать, убеждаемся в наличии следующего:

1. http://sourceforge.net/project/showfiles.php?group_id=9649 (версия 1.6.14 или выше)
2. <http://www.koepe.org/xvid.shtml> (версия 1.2 beta или выше)
3. <http://pessoal.onda.com.br/rjamorim/faac.zip> (версия 1.24.1 или выше)
4. <http://yamb.unite-video.com/download.html> (версия 1.5 или выше)
5. Понимание терминов: *кодек, контейнер, битрейт, частота дискретизации, ресемплирование, мультиплексирование, командная строка*
6. **Прямые руки**

При отсутствии чего-либо из пунктов 1-4 качаем недостающий софт по приведенным ссылкам. Если не хватает пункта 5 - в <http://www.justfuckinggoogleit.com/>. Если проблемы с последним пунктом - вы попали не по адресу. Если вы находитесь в танке, то рекомендую из него вылезти, прежде чем приступать к изучению данной инструкции.

Шаг 1.

- 1.1. Запускаем VirtualDub, открываем исходный файл.
- 1.2. В меню *Audio* выбираем *Full processing mode* - в этом режиме становятся доступны опции ресемплирования и изменения числа каналов (моно/стерео).
- 1.3. Открываем окно конвертера: *Audio > Conversion...*
- 1.4. Рекомендуемые настройки:

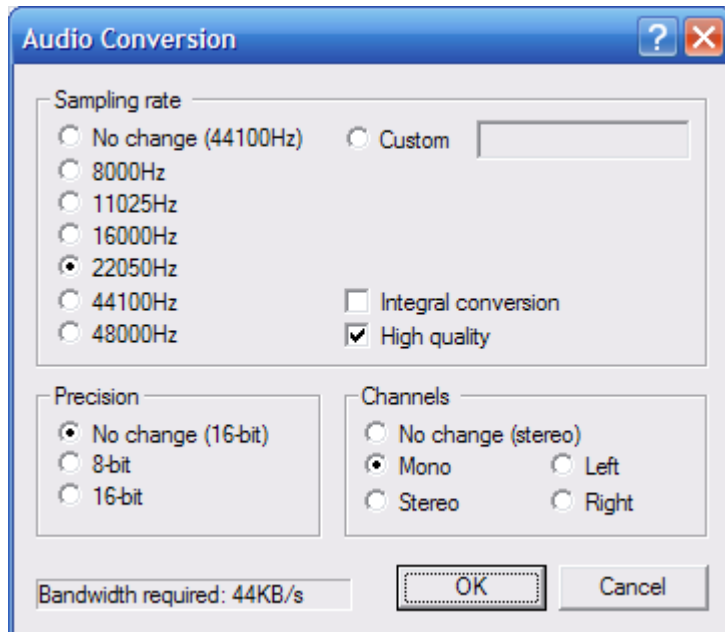
Sampling rate: 22050Hz, High quality

Precision: No change (16-bit)

Channels: Mono

- 1.5. Экспортируем звуковую дорожку: *File > Save WAV...*

- 1.6. Поскольку звук нам больше не нужен, выбираем *Audio > No audio*.

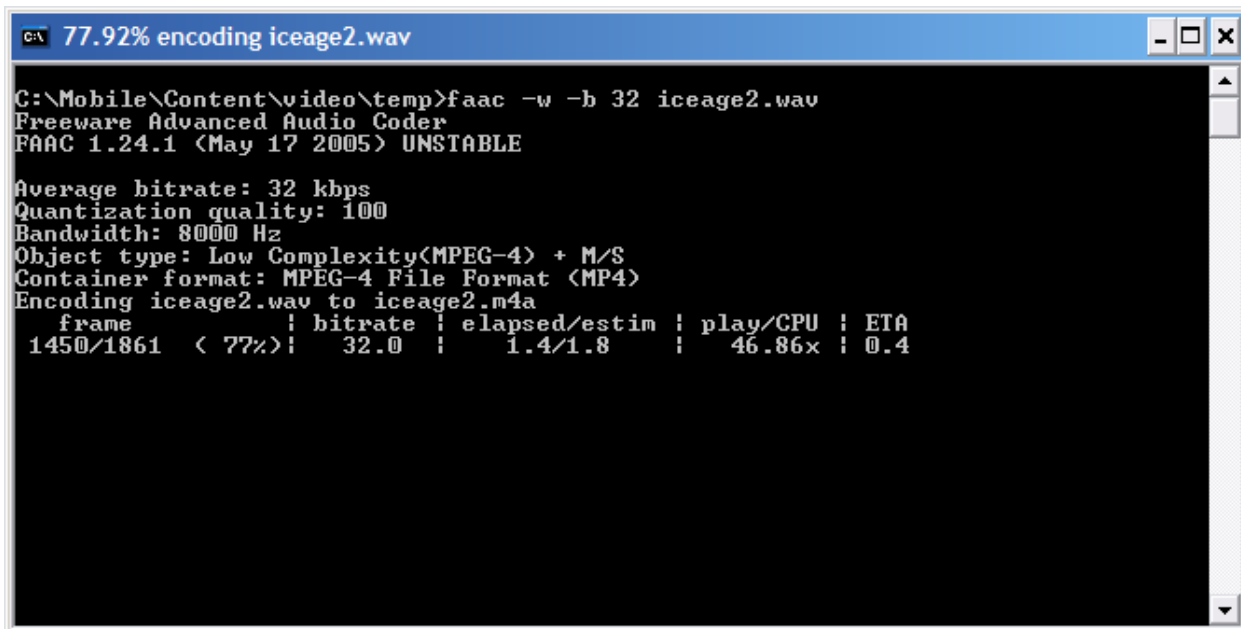


Шаг 2.

2.1. Сжимаем WAV файл, полученный на предыдущем шаге, с помощью энкодера FAAC. Эта программа похожа на LAME и работает в командной строке.

Запускаем процесс командой

faac -w [-b *bitrate*] *sound.wav*



```
C:\ 77.92% encoding iceage2.wav

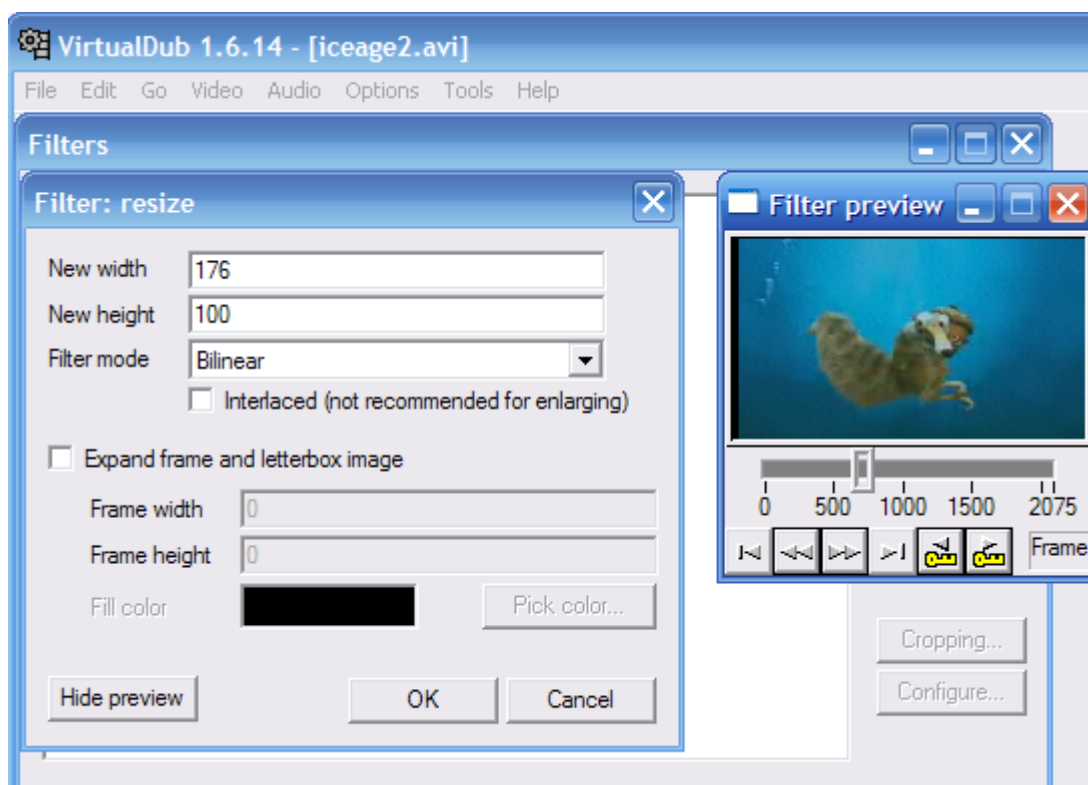
C:\Mobile\Content\video\temp>faac -w -b 32 iceage2.wav
Freeware Advanced Audio Coder
FAAC 1.24.1 (May 17 2005) UNSTABLE

Average bitrate: 32 kbps
Quantization quality: 100
Bandwidth: 8000 Hz
Object type: Low Complexity(MPEG-4) + M/S
Container format: MPEG-4 File Format (MP4)
Encoding iceage2.wav to iceage2.m4a
   frame      | bitrate | elapsed/estim | play/CPU | ETA
1450/1861    < 77%>|    32.0 |    1.4/1.8   |   46.86x | 0.4
```

Если не указывать битрейт, FAAC определит его автоматически. Для звука 22050Hz mono это будет примерно 48 кбит/с, однако если в исходном файле преобладает речь, а не музыка, битрейт можно спокойно понизить до 32-24 кбит/с. Имейте ввиду, что речь идет о среднем, а не постоянном битрейте, поэтому результат может немного отличаться от заданного значения.

2.2. На выходе получаем файл *sound.m4a* - это AAC в контейнере MP4 (тот же формат, который использует iTunes).

Не мешает прослушать полученный файл и убедиться в отсутствии явных звуковых артефактов - при наличии оных повторяем п.2.1 с большим битрейтом до достижения желаемого результата.



В поле *Filter mode* выбираем алгоритм ресемплинга.

Nearest neighbor - самый быстрый и тупой алгоритм. Не рекомендуется, поскольку приводит к появлению "ступенек" вместо диагональных линий.

Bilinear - относительно быстрый алгоритм. Дает побочный эффект в виде небольшого "размазывания" картинки, что, как ни парадоксально, может уменьшить количество артефактов сжатия в конечном файле (мягкие контуры требуют меньшего битрейта).

Lanczos3 - самый качественный (и медленный) алгоритм.

Кнопкой *Show preview* можно включить предпросмотр и оценить качество ресемплинга.

3.7. Жмем OK, OK и возвращаемся в главное окно VirtualDub.

Шаг 4.

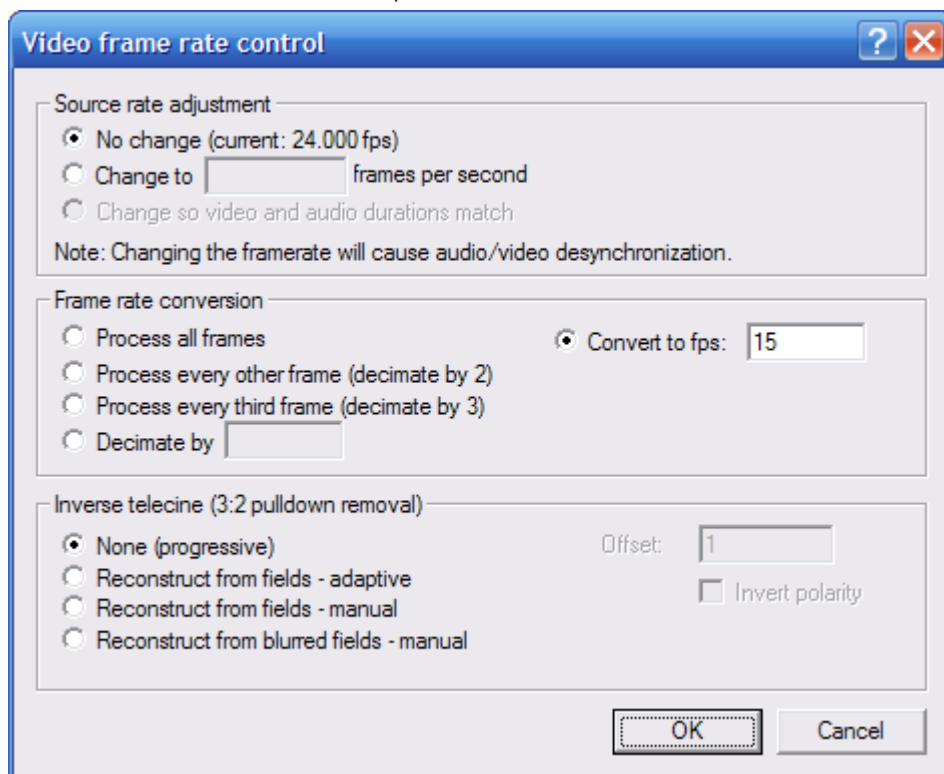
4.1. Настраиваем частоту кадров. Как известно, рекомендуемая частота кадров для E398 и E1 - 15 кадр/с.

Однако, если кодируется ролик без звука, то телефон способен воспроизводить и 30 кадров в секунду.

Если аудио в формате AMR, то в некоторых случаях максимальная частота может быть до 25 кадр/с.

Если же звук в формате AAC (как предполагается в этой инструкции), то желательно ограничиться 15 кадрами.

В VirtualDub жмем *Video > Frame Rate...*, выбираем опцию *Convert to fps* и в соседнем поле вводим значение желаемой частоты кадров.



Внимание, для тех кто по-прежнему сидит в танке: выбирать нужно именно опцию *Convert to fps*, находящуюся в группе *Frame rate conversion*, а не *Change to __ frames per second* в группе *Source rate adjustment*.

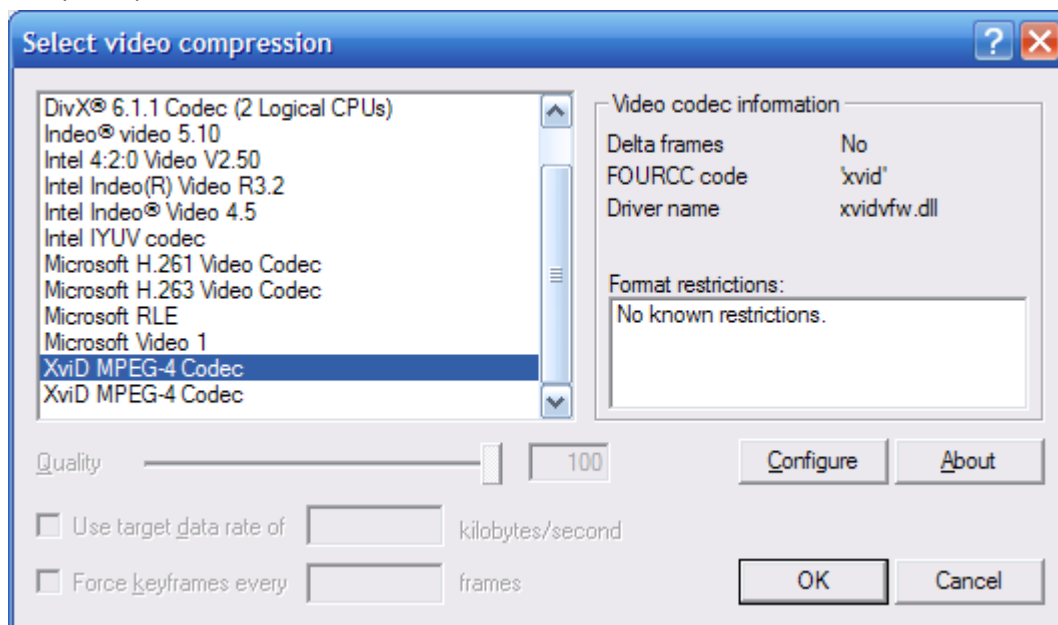
Если выбрать второе, то изменится частота кадров, но их количество останется прежним. Это приведет к эффекту замедленного движения и рассинхронизации аудио и видео ряда (звук будет опережать изображение).

4.2. Жмем OK.

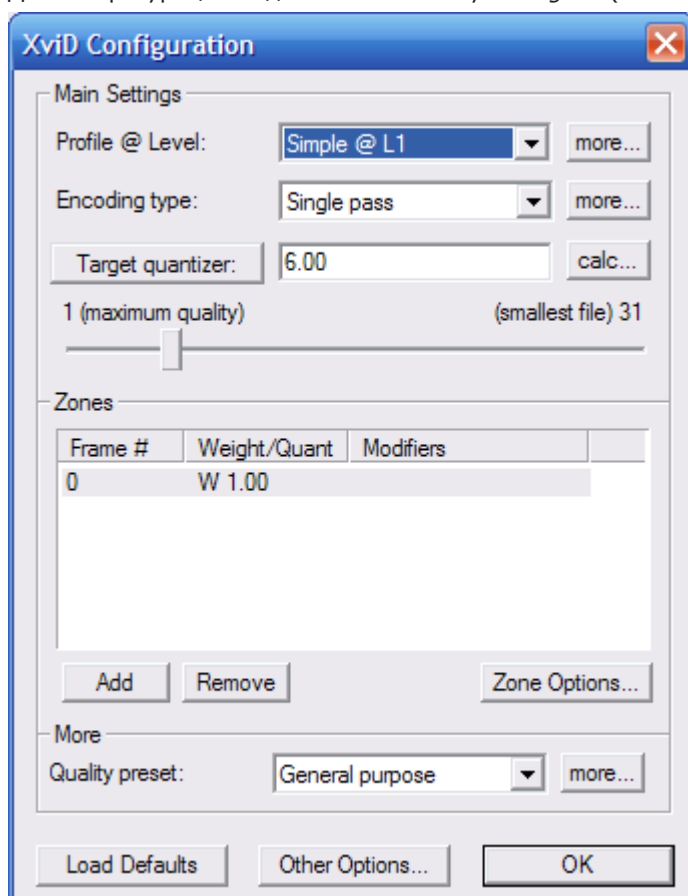
Шаг 5.

5.1. Осталось только настроить видео кодек.

Идем в *Video > Compression...* и выбираем "XviD MPEG-4 Codec". В списке их будет два - нужно выбрать тот, у которого FOURCC = xvid.



Для конфигурации кодека жмем кнопку *Configure* (логично, да?).



5.2. В первый раз не мешает нажать *Load Defaults*, чтобы вернуть все настройки по умолчанию.

После этого устанавливаем следующие параметры:

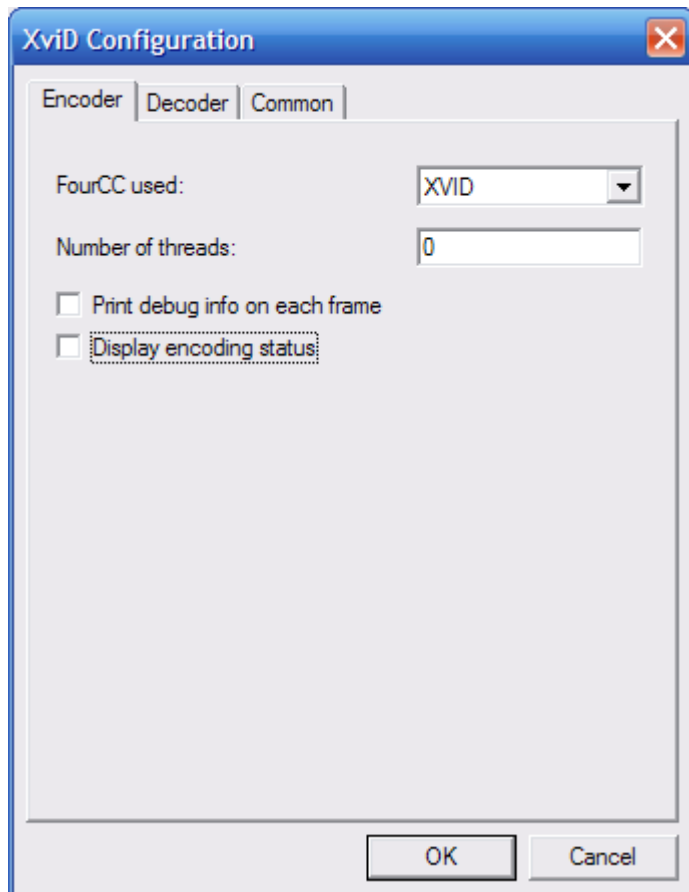
Profile @ Level: Simple @ L1

Encoding type: Single pass

5.3. Теперь разберемся с битрейтом. XviD при кодировании в один проход (single pass) имеет два

режима контроля битрейта: Target quantizer (кодирование с постоянным качеством) и Target bitrate (кодирование со средним битрейтом). Эти режимы переключаются соответствующей кнопкой. Выбор оптимального варианта зависит от исходного видео. Target quantizer предпочтительнее, так как дает постоянное качество на протяжении всего видео, однако на особо динамичных сценах битрейт может превышать возможности телефона. В этих случаях можно увеличить значение квантизатора, установить алгоритм ресемплинга bilinear (см. п.3.6.), или использовать режим Target bitrate. Для примера, трейлер Ледникового Периода 2 я кодировал с $q=6.0$.

5.4. Перед тем как закрыть настройки XviD, можно нажать *Other Options...* и отключить опцию *Display encoding status* - тогда во время кодирования не будет вылазить окно с дополнительной информацией.



5.5. OK, OK, OK.

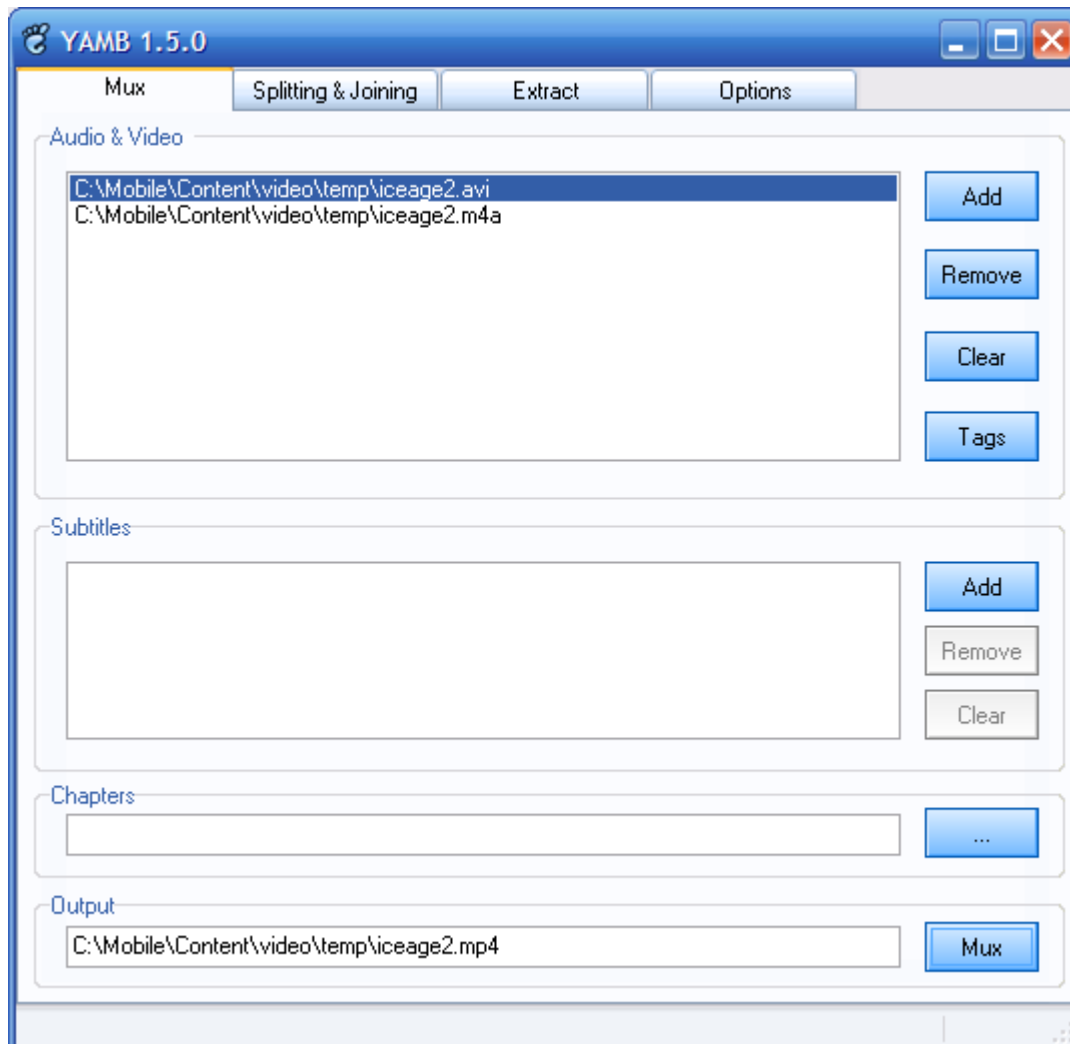
5.6. Начинаем собственно процесс кодирования.

В меню *File* жмем *Save as AVI...*, выбираем файл и ждем окончания процесса.

Шаг 6.

6.1. Если у вас хватило сил дочитать до этого места, можете радоваться - основная работа позади, осталось только мультиплексирование видео и аудио в контейнер MP4.

Эту операцию будем делать с помощью YAMB (Yet Another Mp4Box UI) - это графический интерфейс к консольной программе mp4box.



6.2. Кнопкой *Add* добавляем в список потоков видео файл (.avi), а затем аудио (.m4a).

6.3. Нажимаем **Mux** и получаем готовый .mp4 файл.

6.4. Заливаем его в телефон и смотрим!

P.S. Несмотря на кажущуюся объемность инструкции, на самом деле все ручные операции занимают всего несколько минут.

P.P.S. Этот способ пригоден как для коротких роликов, так и для полнометражных фильмов. Я успешно кодировал двухчасовые фильмы, которые проигрывались на телефоне без притормаживаний.

P.P.P.S. Загрузка процессора телефона зависит от трех величин: частоты кадров, битрейта видео и битрейта аудио. Если процессор при просмотре работает на пределе своих возможностей, то нажатия клавиш будут отрабатываться со значительным запаздыванием.

P.P.P.P.S. **Archy** на основе этой инструкции написал <http://motofan.ru/board/index.php?showtopic=55521>, который автоматизирует большинство перечисленных операций.