

# VirtualDub 메뉴얼 V1.0

만든이 : 쿨맨(하나넷 영화동호회 씨네)

## §1. 기초 사항

### 1. VirtualDub 은....

이 프로그램은 비디오 테이프나 오디오 테이프를 더빙하듯이 컴퓨터상에서 보다 편리하고 효율적으로 동영상 및 사운드를 더빙 할 수 있도록 해주는 공개 프로그램입니다.

이 프로그램은 기본적으로도 강력한 편집기능(자르기, 재압축, 필터링)을 제공하며, 화면중첩 등의 전문적인 기능은 제공하지 않지만, 완전 공개 프로그램이고 업데이트도 자주되는 편인데다가 여러가지의 비공식 버전들도 개발되고 있어 동영상 편집에 널리 사용되는 훌륭한 프로그램입니다.

초기버전에서는 asf 파일도 다룰수 있었으나, MS사의 요청에 의해 1.4 버전부터는 asf 파일을 다룰 수 없도록 기능이 다운그레이드 되었습니다. 하지만, 현재는 1.4d 버전을 기반으로 하고도 asf를 다룰 수 있는 개량된 버전(비공식 버전이지만...)이 나와있습니다.(이 프로그램은 DVD의 동영상 형태인 .VOB 파일을 직접적으로는 다루지 못합니다.)

이 프로그램은 여러가지 형태의 비디오 포맷에 대응할 수 있는 범용적인 프로그램이며, 가능한 작업도 여러 종류이지만, 본 메뉴얼에서는 최근 가장 유행하고있는 형태인 DivX 코덱의 동영상의 편집을 기준으로 설명하고자 합니다. 아울러, 본 메뉴얼에서 VirtualDub을 이용하는 모든 사용방법을 다 설명하지는 못합니다. 기본적인 사항 이외의 테크닉들은 각자의 상황에 따라 적용해 보시기 바랍니다.

프로그램을 사용하기 이전에 이 메뉴얼을 꼭 끝까지 읽고나서 사용하기 바랍니다.

### 2. 디지털 영상 기본 사항

★ 이 내용은 디지털 영상에 대한 기본적인 지식이 있으신 분들은 읽으실 필요가 없습니다만, 초보인 분들은 지루하더라도 반드시 읽어주시기 바랍니다.

#### 2.1. Frame rate

우리가 보는 동영상 파일은 결국 정지된 그림 화면들을 지속적으로 변경해가면서 보여주는 것입니다. 이때 1초에 몇 화면을 표시하는지가 frame rate가 되겠습니다.

- ① 영화필름 : 24~48 fps (frame per second)
- ② TV : 25~30 fps
- ③ 대부분의 컴퓨터 비디오 : 10~15 fps

## 2.2. Key frame 과 Delta frame

예를 들어, 같은 배경위를 사람이 움직이는 경우, 배경은 동일하고, 움직이는 것은 사람일 뿐이므로, 배경과 사람이 포함된 첫번째 프레임을 Key Frame 으로 삼고, 나머지 프레임(-Delta Frame)은 움직이는 사람만 저장 함으로써 파일의 용량을 줄일 수 있습니다.

대부분의 비디오 파일은 5~30 프레임마다 또는 1~10초 마다 하나씩의 키프레임을 가지고 있습니다만, 최근 등장한 인코딩 방식인 VKI(Variable Keyframe Interval)로 인코딩된 DivX 파일들의 경우에는 이 키프레임 간격이 일정하지 않습니다. 키 프레임이 많아지면 동영상 실행시 특정 부분으로 이동한다든지 하는 경우 훨씬 부드럽게 이동이 가능하지만, 상대적으로 파일의 크기는 대단히 커집니다.

따라서, 키 프레임 간격은 적절한 선에서 결정해 주어야 용량도 줄이면서 화질도 좋은 영상을 만들어 낼 수 있습니다.

## 2.3. Codec

동영상이나 오디오의 압축/해제 방식을 코덱이라 합니다. 압축되지 않은 데이터는 워낙 용량이 크기 때문에 이를 줄이기 위해 고안된 것이 코덱이며 이 코덱의 종류는 무수히 많습니다.

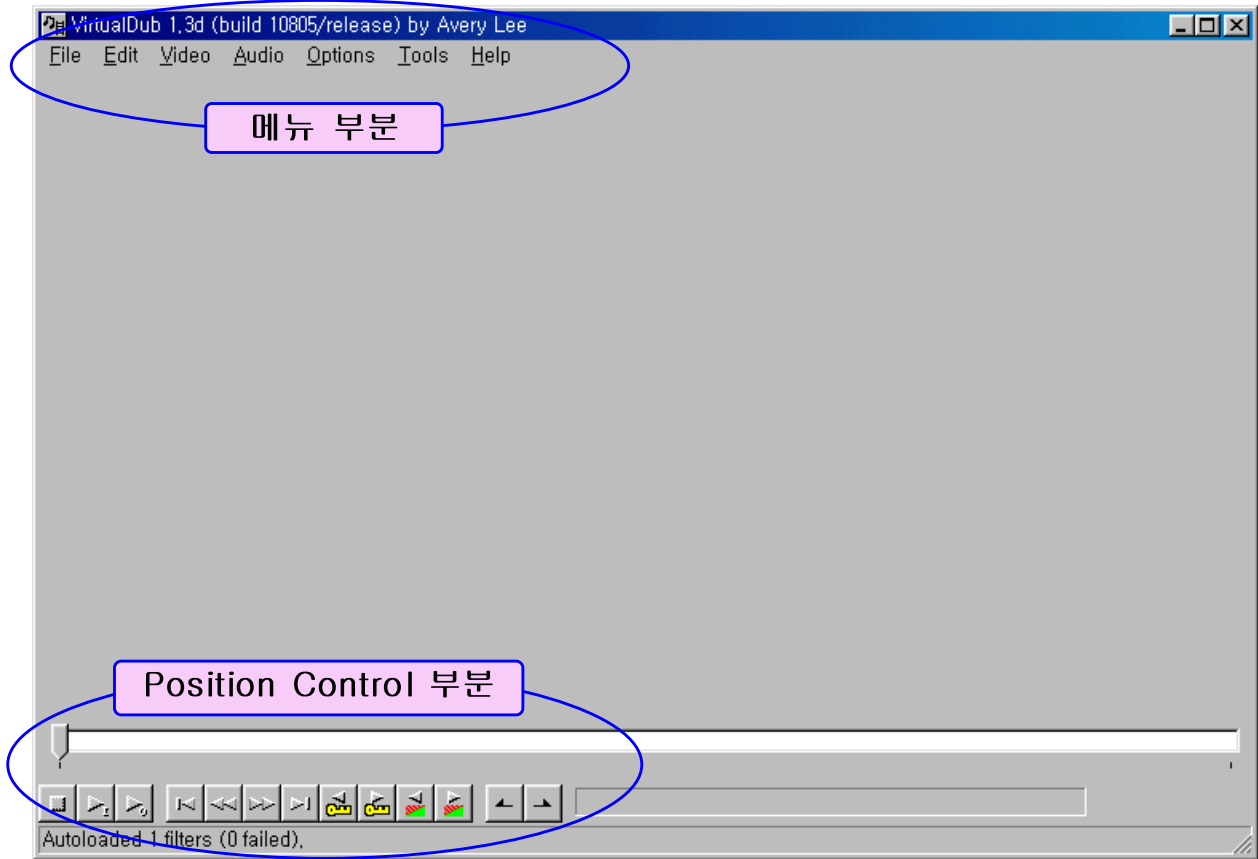
또한, 압축과정에서 사용된 코덱과 동일한 코덱이 설치되어 있지 않은 컴퓨터에서는 동영상 자체의 실행이 불가능합니다. 쉽게 예를 들자면, 흔히 알려져 있는 오디오 코덱중 하나가 Mpeg Layer 3 (MP3) 인데 일반 CD 플레이어에서 MP3 파일을 실행할 수 없는 것과 같습니다.

최근에 가장 인기를 끌고 있는 동영상 압축 코덱이 바로 DivX가 되겠습니다.





따라서, 향후 설명하게될 동영상 및 오디오 편집 과정에서 나오는 코덱들은 이 프로그램 실행 이전에 반드시 먼저 설치되어 있어야 합니다. (간혹, 왕초보 분들 중에는 코덱을 설치하고 나서 바로 프로그램을 실행하는 경우를 볼 수 있는데 기본적으로 코덱을 설치한 다음에는 리부팅을 해 주셔야 합니다.)










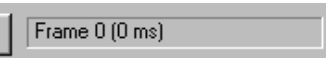

## §2. VirtualDub 의 기본 조작

VirtualDub을 실행했을때 나타나는 프로그램 창을 기준으로, 기본적인 조작 방법들을 설명합니다.  
아래의 화면이 VirtualDub을 실행한 모습입니다.



이 프로그램의 사용시 가장 빈번하게 만지게 되는 부분은 바로 Position Control 부분이기 때문에 이 부분의 조작방법은 확실히 익혀 두셔야 합니다.

1.  Stop
  - ① 현재 진행중인 미리보기 또는 더빙을 정지시킨다.
  - ② 이 버튼은 프로그램의 주화면에만 나타난다.
2.  Play Input
  - ① 현재 위치에서 입력 동영상을 실행시킨다.
  - ② 범위 설정을 비롯한 대부분의 프로그램의 기능이 불가능해진다.
  - ③ 이 버튼은 프로그램의 주화면에만 나타난다.
3.  Play Output
  - ① 현 위치에서 작업된 동영상을 실행시킨다.
  - ② 모든 프로그램의 기능이 활성화된다.
  - ③ 미리보기는 현위치에서 시작되고, 만약 엔드마크를 했다면, 엔드마크까지 진행된다.
  - ④ 이 버튼은 메인 스크린에서만 나타난다.
4.  Start (주로 사용됨)
  - ① 파일 맨 처음으로 이동한다.

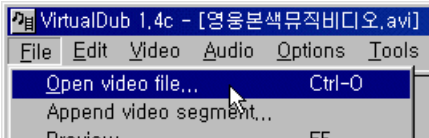
5.  Back (별로 쓸일 없음)
  - ① 한 프레임씩 뒤로 이동한다.
6.  Forward (별로 쓸일 없음)
  - ① 한 프레임씩 앞으로 이동한다.
7.  End (주로 사용됨)
  - ① 현재 위치에서 파일 맨 끝으로 이동
8.  Key Reverse (주로 사용됨)
  - ① 먼저부분의 키프레임으로 후퇴
  - ② Mpeg 파일에서는 마지막 GOP(Group Of Pictures) 시작위치로 후퇴
9.  Key Forward (주로 사용됨)
  - ① 다음 키프레임으로 전진
  - ② Mpeg 파일에서는 마지막 GOP(Group Of Pictures) 시작위치로 전진
10.  Scene Reverse (별로 쓸일 없음)
  - ① 먼저의 Scene 위치로 후퇴
  - ② 이 버튼의 클릭에 따라, 후퇴를 멈추거나, 계속 진행할 수 있다.
11.  Scene Forward (별로 쓸일 없음)
  - ① 앞의 Scene 위치로 전진
  - ② 이 버튼의 클릭에 따라, 전진을 멈추거나, 계속 진행할 수 있다.
12.  Mark In (주로 사용됨)
  - ① 삭제할 프레임의 시작위치를 지정
13.  Mark Out (주로 사용됨)
  - ① 삭제할 프레임의 마지막 위치를 지정
14.  Information Panel
  - ① 처음부터 현재 위치까지의 프레임수와 시간(ms)을 알려줌
15.  Position Slider
  - ① 어떤 프레임에 위치해 있는지 조절 (원본파일의 어떤 위치로도 이동가능)
  - ② 모든 프레임은 점표시로 나타나며, 프레임수가 많으면, 검은줄로 표시됨

### §3. VirtualDub 을 이용한 DivX 동영상의 인코딩

비디오 CD용 동영상인 .dat, .mpg, .mpeg나 .avi, .asf 파일들은 VirtualDub을 이용해 DivX 동영상을 인코딩 할 수 있습니다. 물론, FlaskMpeg에서도 mpg 파일을 다룰수는 있으나 FlaskMpeg 보다는 VirtualDub을 이용하시는게 더 좋습니다.

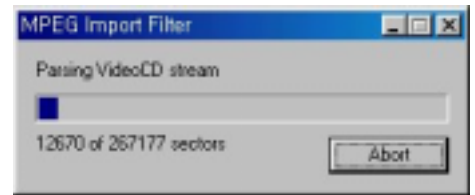
단, VirtualDub은 DVD의 .VOB 파일은 직접 다루지 못하므로, DVD를 인코딩하시는 경우에는 FlaskMpeg 프로그램을 이용하시길 권장합니다.(FlaskMpeg을 이용한 인코딩 방법은 씨네 프로그램 자료실에 이미 업로드해 놓았습니다.)

#### 1. 파일 불러오기



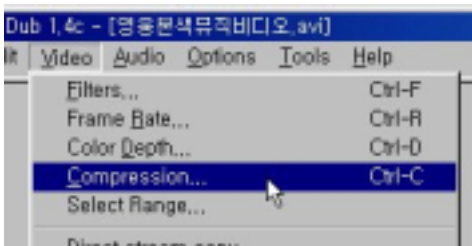
프로그램을 실행한 상태에서 "File"→"Open Video File" 을 선택해 변환하고자 하는 동영상을 불러옵니다.

이때 작업대상이 DAT 인 경우에는 오른쪽 그림과 같이 불러오는 과정에 시간이 좀 걸립니다.....^^;;



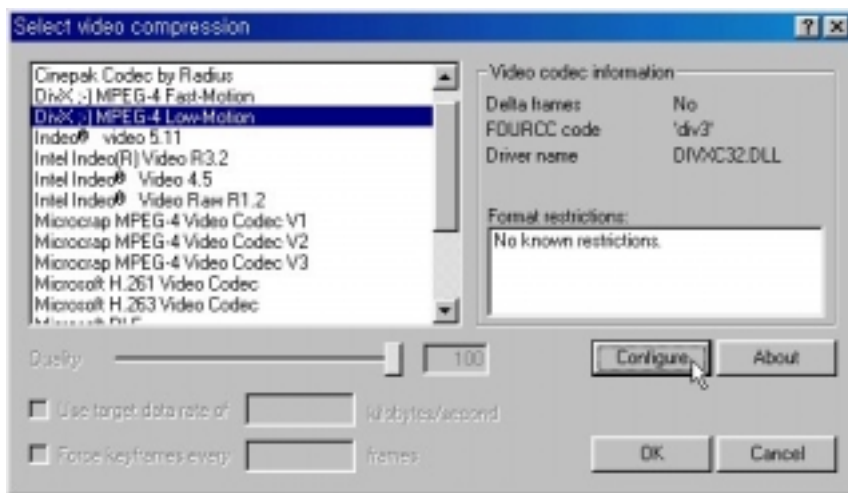
#### 2. 비디오코덱 설정하기

사용할 비디오 코덱을 선택하는 과정입니다.



먼저, "Video" → "Compression" 을 선택합니다.

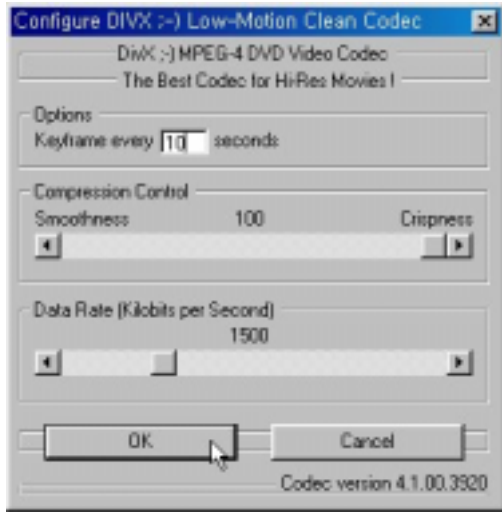
그러면, 아래쪽의 화면과 같은 코덱 선택 창이 뜹니다.



여기서 "DivX(-)MPEG-4 Low-Motion" 을 선택합니다. 물론, DivX(-)MPEG-4 Fast-Motion 을 선택해도 되지만, Fast Motion 코덱은 Bitrate 수를 높여도 별로 화질이 좋아지지 않는 등 문제가 있어, 주로 DivX(-)MPEG-4 Low-Motion 코덱을 사용하시길 권장합니다.

다음으로 "Configure" 버튼을 눌러 인코딩 설정을 합니다.

아래는 "Configure" 버튼을 눌렀을때 나타나는 설정 창입니다.



여기서, "Keyframe every [ ] seconds" 의 박스 안에는 몇 초 간격으로 키프레임을 삽입할 것인지 수치를 써 넣습니다.

본래 디폴트값은 대체로 3초 이나, 10초 정도면 적당할 것 같습니다.

Compression Control 에서는 Crispness 쪽으로 갈수록 화면이 더 바삭(?)해 집니다.(좀더 선명해지는 격인데, 대체로 100으로 그냥 둡니다.)

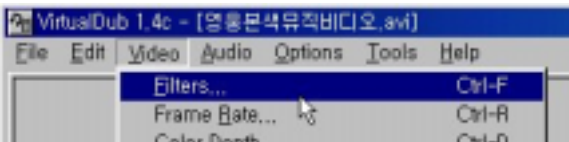
Data Rate 는 디폴트값이 1500 이지만, 적당히 조정해야 합니다. 대체로 1500 정도로 그냥 두면 CD 2장 정도의 분량이 되는 경우가 많고, 1000 정도면 1장에 들어갈 수 있을 것입니다.

이 수치가 커지면 그만큼 화질은 좋아지지만, 용량이 커지며 또 무턱대고 크게 잡는다고 화질이 눈에 띄게 좋아지는것도 아닌 경우가 많으므로 적당한 수치는 원본 데이터의 상태나 각자의 취향에 따라 선택해야 할 문제가 되겠습니다.

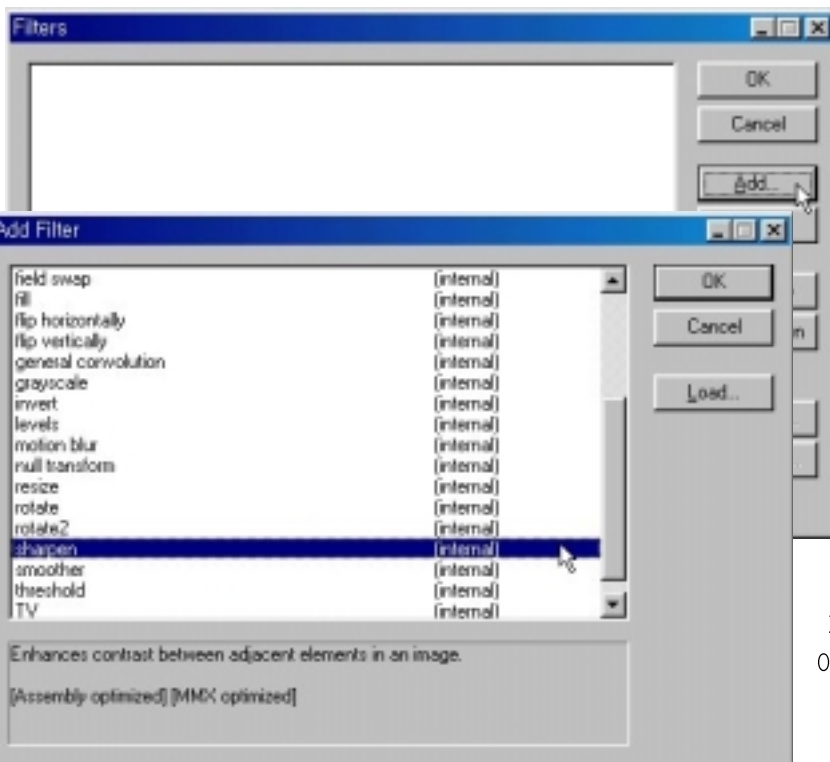
설정이 끝나면 당연히 "OK" 버튼을 눌러줘야 겠죠....^^;;;

### 3. 필터 등의 부가 기능 설정하기 (옵션 사항)

필터링 기능 등의 부가기능을 사용할 수 있습니다만, 이는 옵션 사항이므로 항상 필요한 것은 아니니 참고만 하시기 바랍니다.



"Video" → "Filters..." 를 선택합니다.



그러면, 왼쪽과 같은 빈 창이 하나 나타납니다.

여기서 "Add..." 버튼을 누르면 아래쪽에 보이는 "Add Filters" 창이 나타납니다.

선택가능한 필터들이 상당히 여러가지이며, 필터 이름을 클릭하면 아래쪽에 해당 필터의 기능들이 설명되므로 필요한 필터를 선택한 다음 OK 버튼을 누릅니다.

그러면 앞의 빈 화면에 필터가 추가되며, Filter 창에서 또 OK 버튼을 눌러 필터 선택을 마치면 되겠습니다.

기타로 "Color Depth" 등등의 기능은 그리 쓸일이 없을 듯 하기 때문에 설명하지 않습니다.

#### 4. 오디오 코덱의 설정

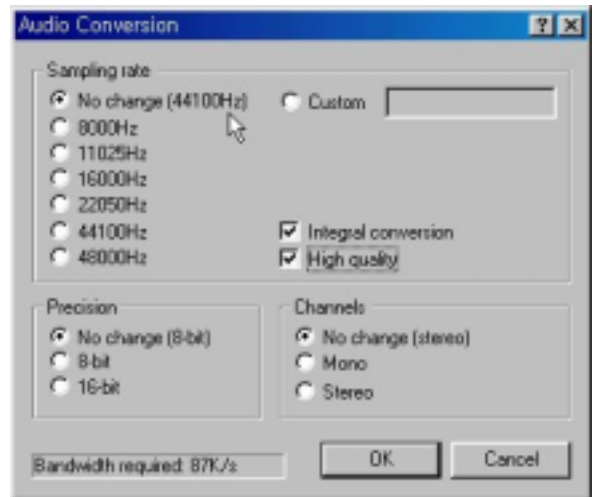
오디오 압축 코덱을 설정하는 과정입니다.



우선은 "Audio" → "Full processing mode" 로 설정 한다.

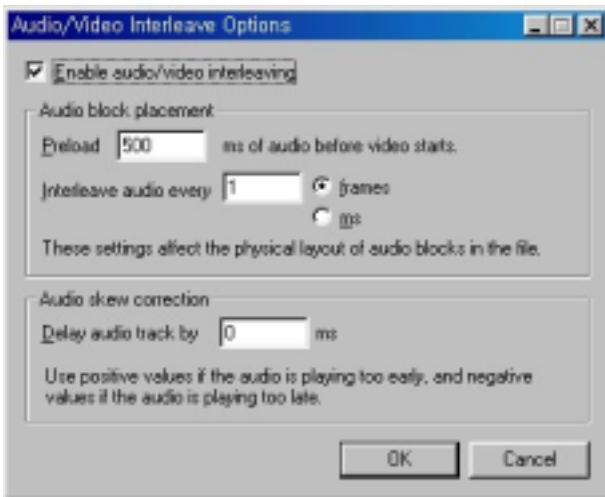
다음으로 "Audio" → "Conversion" 을 선택합니다.

그러면 오른쪽 그림과 같은 창이 뜹니다. ("No change" 옆에 써 있는 수치가 현재 불러온 동영상의 오디오 Sampling rate 입니다.) 오디오의 샘플링 레이트는 반드시 바꿀 필요는 없으며, 대체로 44100 이 사용되지만, 경우에 따라 48000Hz를 44100Hz로 바꾸는 등의 조작이 필요할 수도 있습니다.



이 창에서는 Integral conversion 과 High quality 앞의 박스에 체크표시를 해 주시는게 좋습니다.

다음 과정은 "Audio" → "Interleaving" 의 확인 입니다.



왼쪽 화면이 Interleaving 을 선택한 후의 화면입니다.

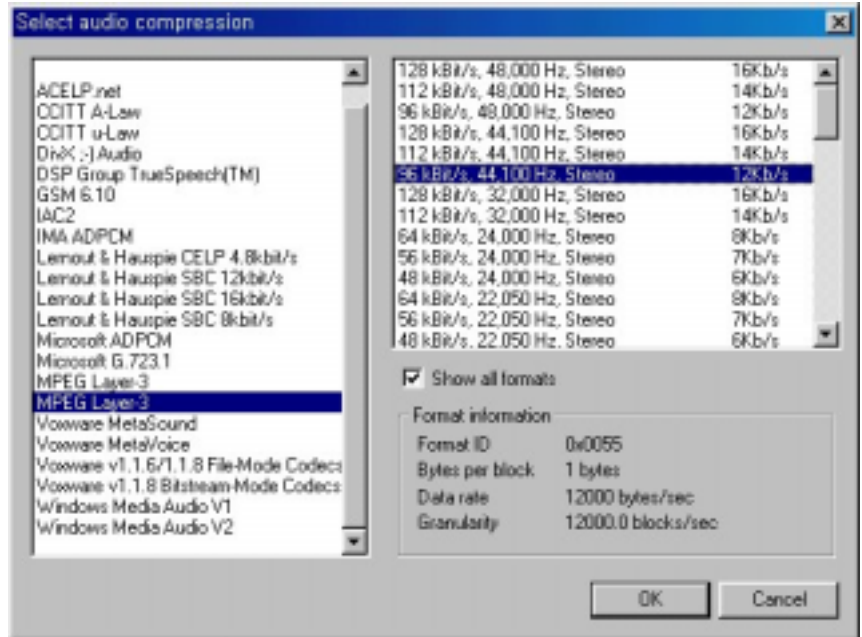
실제로 이 화면에 보이는 설정에서 바꿀 필요는 없습니다.

그 다음으로 "Audio" → "Compression" 을 선택합니다.

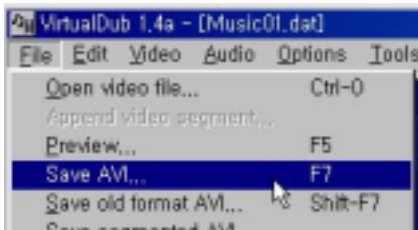
오른쪽 화면은 Compression 을 선택한 후 나오는 코덱 설정 화면입니다.

여기서, MPEG Layer-3 (MP3) 와 96kBit/s, 44100Hz stereo 를 선택합니다. 물론, 앞의 "Conversion" 메뉴에서 보았던 오디오 샘플링레이트가 48000 Hz 였다면 48000 Hz를 선택할 수도 있습니다.

만약, 코덱이 제대로 보이지 않는다면 "Show all formats" 앞의 박스에 체크하면 되며, 그래도 보이지 않는다면 아직 코덱이 설치되지 않는 경우이므로 코덱을 설치하셔야 합니다.(해당 오디오 코덱은 씨네 프로그램 자료실에 있습니다.)



## 5. DivX 코덱 동영상의 저장



자 이제 모든 설정이 끝났습니다. 이제 동영상을 저장하기만 하면 됩니다.

왼쪽 그림처럼 "File" → "Save AVI..."를 눌러 파일 이름을 지정해준 다음 저장하면 작업이 진행됩니다.

작업시간은 시스템 사양이나 파일 용량 및 사용 코덱, 설정된 조건 등등에 따라 워낙 변수가 많으므로, 얼마가 걸릴지 예측할 수 없습니다만 일반적으로 상당한 시간(하루 밤 정도)이 소요될 것입니다. (필자도 DAT나 Mpeg 파일의 변환은 해본적이 없어서.....^^;;)

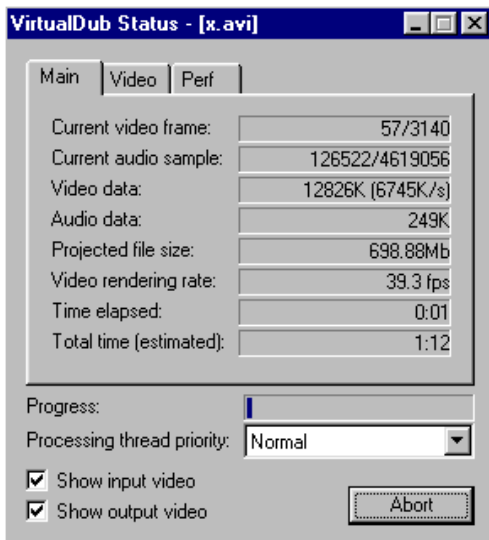
★ 주의 : .AVI 파일로 저장시 파일 이름을 지정할 때는 꼭 확장자(.avi)까지 써 넣어 주어야 합니다. avi 라는 확장자명을 안써주면 그냥 파일 이름만 덜렁 있는 이상한 모양의 파일이 만들어집니다.(물론, 나중에 탐색기 등에서 이름 바꾸기를 해줘도 되지만....) 이는 VirtualDub 자체의 버그인데 무슨 이유인지 해결이 아직 안되고 있습니다.



## [ 참고 ] VirtualDub 의 작업진행 표시창

아래는 VirtualDub의 작업 진행과정에서 보여지는 작업표시창에 대한 설명입니다. 실제로 main창 이외에는 별 쓸모가 없습니다.

### (1) Main 창



#### 1. Current video frame and audio sample

- ① 입력되는 영상 및 음성 중 작업이 진행되고 있는 부분을 표시
- ② 부분삭제나 편집등의 기능시 작업의 확인을 가능하게 해줌

#### 2. Video and audio sizes

- ① 현재 시점에서 출력파일의 크기를 추정해줌(어디까지나 추정값이므로 정확하지는 않습니다....^^:)

#### 3. Projected file size

- ① 출력파일의 최종크기를 추정해줌(어디까지나 추정값이므로 정확하지는 않습니다만, 작업이 거의 끝나갈 무렵에는 비교적 정확한 값을 보여주므로 용량의 추정이 가능합니다....^^:)

#### 4. Video rendering rate

- ① 프로그램이 초당 몇 프레임을 처리하고 있는지 표시

#### 5. Time elapsed / Estimated Time Remaining

- ① 작업시간(경과시간, 남은시간)이 표시됨

#### 6. Progress bar

- ① 작업 진행을 표시기

#### 7. Processing thread priority

- ① 프로그램이 동작하는데 관련되는 CPU time 을 조정하는 것으로, 건드리지 않는게 좋다.

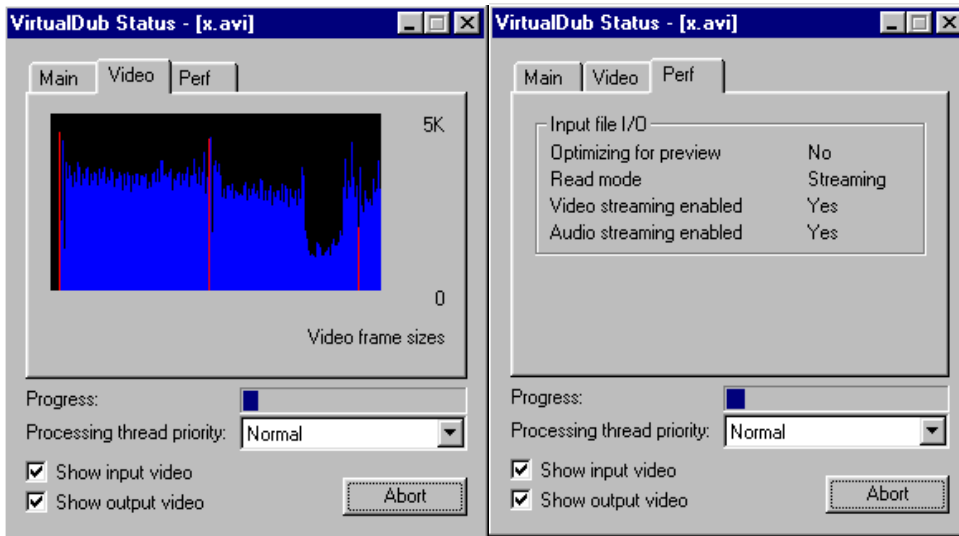
#### 8. Show input/output video

- ① 입출력 동영상을 표시함
- ② 미리보기시에는 입력쪽을 꺼두면 효과적이다(저 사양의 컴퓨터에서....)
- ③ 작업중일때는 이걸 끄나 켜나 작업속도에 별로 차이가 없다.

#### 9. Abort

- ① 작업을 중지하고, 지금까지의 출력파일을 마무리해서, 실행가능한 상태로 만든다.
- ② 한번 중지된 작업은 연결이 되지 않기 때문에 주의해야 한다. (잘못해서 마우스로 누르면, 끝이다....)

## (2) Video frame size monitor & Perf



### 1. Video frame size monitor

- ① 압축된 출력동영상의 프레임크기를 표시함
- ② 빨간선이 키프레임이고, 파란선이 델타프레임이다.
- ③ 그래프가 높게 그려지면, 프레임이 큰 것이다.
- ④ 이 그래프는 프레임의 크기에 따라 자동으로 스케일을 조절해가며 그려진다.
- ⑤ 미리보기 모드나, 압축되지 않은 출력의 경우에는 빨간색 블럭으로 그려진다.

### 2. Perf

- ① 작업상태를 보여주는것 뿐이다.(별 필요 없다.)

## §4. 기존 DivX 동영상의 용량을 줄이는 방법

### 0. 기존 동영상의 용량을 줄일 수 없나요...???

DivX 동영상을 구했는데, 용량이 좀 문제가 있는 경우가 간혹 있습니다. 650 MB를 넘는 것은 700MB 공CD에 구우면 되겠지만, 700MB를 넘는 경우에는 문제가 생깁니다.

간혹 700MB 넘는 동영상을 어떻게 구워야 하느냐에 대해 의문을 가지시는 분들이 많은데, 이런 경우에 대한 대책들부터 생각해 보고자 합니다.

#### 1) 800MB 공씨디가 있다는데....

아직까지 표준규격은 아니지만, 실제로 700 MB 이상의 공CD가 (800MB 짜리도...) 존재 합니다. 용산등에서 구할 수 있다고 하는데, 필자로서도 구경은 못해봤습니다.

이런 공 CD에 구우면 구울수 있습니다. 가장 간단한 방법이긴 합니다. 그러나, 이 방법도 아무나 적용되는 것이 아닙니다. 우선 CD 레코더 자체가 오버버닝을 지원해야 하는데 오버버닝을 지원하는 대표적인 기종이 티악, 플렉스터, 리코의 일부 기종인 것으로 압니다. 필자가 사용하는 소니나 HP 등의 기종은 오버버닝을 지원하지 않기 때문에 800MB 공씨디는 그림의 떡인 것입니다....ㅠ.ㅠ

또, 레코딩 프로그램으로는 네로 버닝 롬 의 최신 버전을 사용해야만 가능하다고 합니다.

#### 2) 700MB 에 네로로 구우면 가능하다는데....

이것 역시 오버버닝을 말하는 것이기 때문에, CD 라이터가 지원하지 않으면 소용 없습니다.

#### 3) VCD로 구우면 730MB도 들어간다는데....

VCD는 DATA의 용량이 아닌 시간으로 체크를 하기 때문에 가능한 얘기입니다만, 문제는 DivX 동영상을 mpg 형태로 바꾸면 그 크기가 엄청나게 커진다는데 문제가 있습니다. 물론 비슷한 용량으로도 만들 수는 있겠지만, 그때에는 화질의 저하가 불보듯 뻔하므로 진정한 의미의 DivX라고는 할 수 없습니다.

#### 4) 자르면 되지 않을까....

물론, DivX 동영상을 자르면 가능합니다. 그러나, 이렇게 되면 자막도 같이 잘라야 하는 문제점이 생깁니다.....^^;;

#### 5) 그럼 진짜 방법이 없나.....

여러 가지 방법을 생각해 볼 수 있습니다만, 대체로 아래의 몇 가지 방법을 생각해 볼 수 있습니다.

- ① 동영상은 그대로 두고 오디오 부분만 압축하는 방법
- ② 동영상을 압축하되 프레임레이트를 줄이는 방법
- ③ 동영상을 압축하되 비트레이트를 줄이는 방법
- ④ 동영상을 압축하되 픽셀수를 줄이는 방법
- ⑤ 위의 방법들을 적절히 혼용하는 방법

## 1. 동영상은 그대로 두고 오디오 부분만 압축하는 방법

이 방법은 우선 동영상의 화질을 전혀 손상시키지 않고, 시간도 비교적 짧게 걸린다는 점에서 좋은 방법입니다. 단지, 이런 방법으로 데이터 용량을 줄이는데는 한계가 있고 원본 데이터의 사운드 부분이 재압축의 여지가 있어야 한다는 점이 문제가 되겠습니다.

우선 이 방법을 적용하기 이전에 윈도우 탐색기에서 원본 동영상의 등록정보를 살펴봅니다.(오른쪽 마우스 버튼 누르면 나오는 팝업메뉴중 맨 아래에 "등록 정보"를 눌렀을 때 나오는 항목 중 "자세히"를 살펴 보면 오디오 등록정보가 있습니다.

일반적으로 DivX 동영상의 오디오는 MP3 코덱을 사용하는데, 대체로 44100Hz(또는 48000Hz) / 128kbps / stereo 가 많이 쓰입니다. 만약, 128 kbps 나 160 kbps를 사용한 경우라면 오디오 압축이 가능하지만 96 kbps를 사용한 경우에는 오디오 압축이 불가능합니다. MP3 코덱에서 96kbps 이하의 stereo는 없기 때문입니다.

물론, DivX;-) Audio 코덱 중의 96kbps를 적용하면 약간의 압축이 가능하긴 합니다만, 필자의 경험으로는 DivX 오디오 코덱은 싱크 부분에서 문제가 발생하는 경우가 많아 권하고 싶지 않습니다.

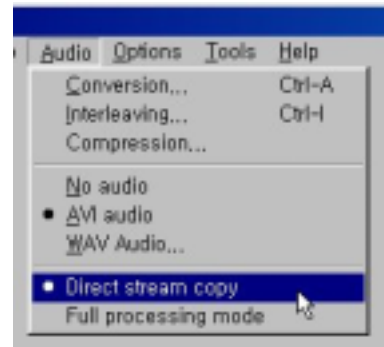
원본 동영상이 128kbps 라고 가정하고, 오디오 부분을 줄여 보도록 하겠습니다.

### (1) Audio Ddata 의 추출(.wav 파일의 추출)

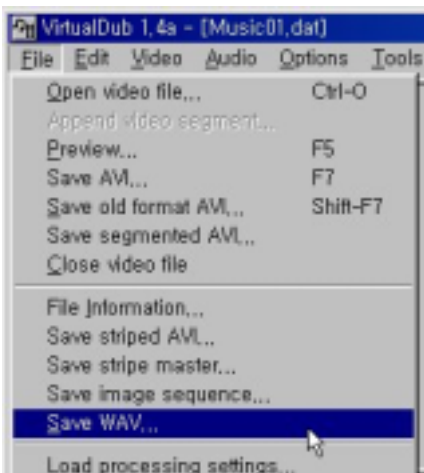
오디오 압축에서 가장 먼저 해야 할 일은 원본 동영상중의 사운드 부분을 wav 파일로 추출하는 과정입니다.



VirtualDub을 실행해 "File" → "Open Video File"에서 원본 동영상을 불러온 다음



왼쪽 그림에서처럼 "Audio" 의 설정을 "Direct stream copy" 로 설정합니다.  
(본래 이 설정은 디폴트 값으로 잡히지만, 다른 작업을 수행한 경우 혹시 변경되어 있을 수도 있기 때문에 재확인하는 것입니다.)



그 다음에 'File' → "Save Wav..."를 선택해 파일명을 써 주고, 사운드 파일만을 추출/저장합니다. (이때에도 파일명에는 반드시 확장자까지 붙여 주어야 합니다. 즉, "test.wav" 라고 확장자까지 써 주어야지 그냥 "test" 라고만 써 주면 확장자 없는 "test" 파일이 생깁니다. 이 역시 프로그램의 버그입니다....^^;;)

약간의 시간이 흐르면(몇분 정도..) 작업이 완료되며, 탐색기로 살펴보면 wav 파일이 만들어진 것을 보실 수 있을겁니다.

## (2) 추출해 놓은 wav 파일과 동영상의 합병

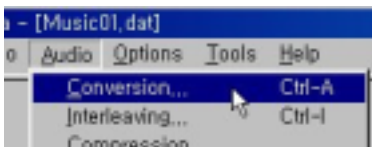
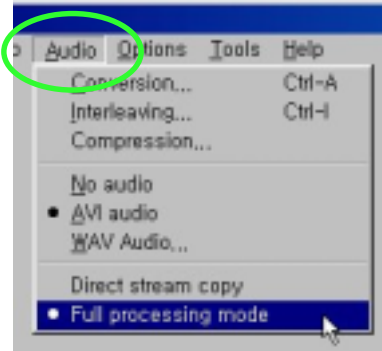
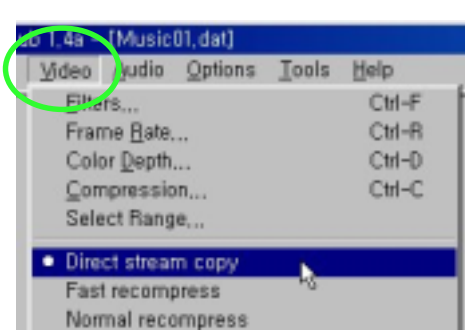
이제는 추출해 놓은 wav 파일을 재 압축하면서 동영상과 합병할 차례입니다.



동영상은 이미 불러져 있는 상태이겠지만, 만약 프로그램을 종료했다면 다시 원본 동영상을 불러온 다음

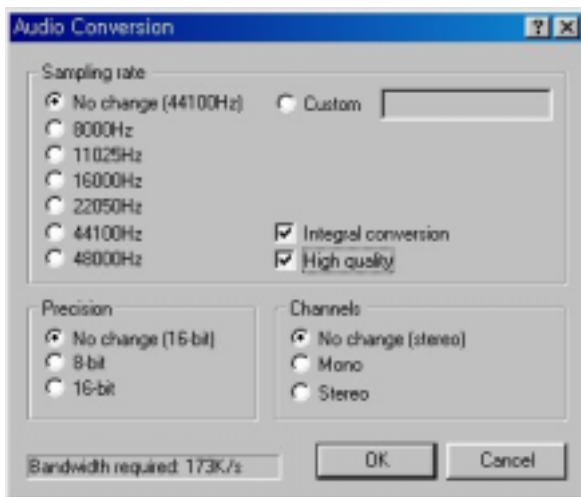
"Audio" → "WAV Audio..."를 선택해 앞에서 추출해둔 wav 파일(test.wav)를 불러옵니다.

다음으로, 아래의 그림처럼 "Video" 는 "Direct stream copy" 로 , "Audio" 는 "Full processing mode" 로 설정합니다.  
 (★ 이때 두 설정이 서로 바뀌지 않도록 조심하세요.....^^)



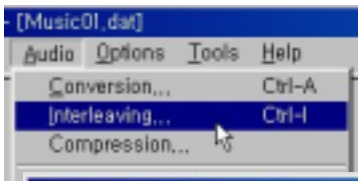
다음으로 "Audio" → "Conversion..."을 선택하면, 왼쪽 그림과 같은 창이 뜹니다.

여기에서는 원본의 오디오가 44100Hz 이기 때문에 화면에 보이는 것처럼 똑같이 설정하면 되지만, 만약 48000 Hz 인 경우에는 "No change" 가 아니라 아래쪽의 "44100 Hz"를 선택해 오디오의 샘플링 레이트를 줄일 수도 있습니다.



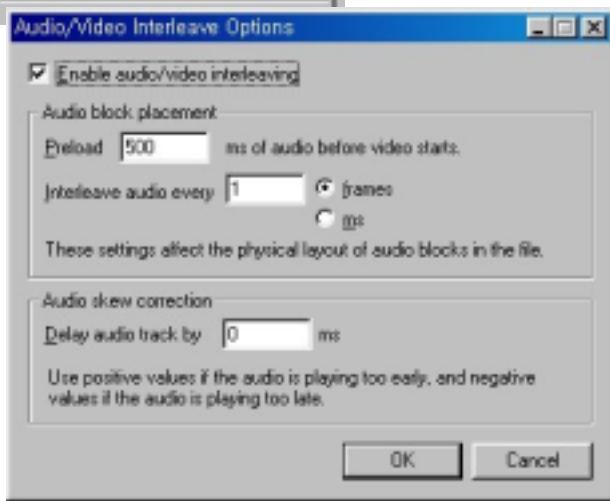
★ 대체로 48000 Hz의 사운드를 44100Hz로 줄여도 사운드 질에는 별반 차이를 느끼지 못합니다. (오디오 CD의 사운드가 44100Hz입니다.)

★ 어떤 경우에는 48000Hz를 44100Hz로 다운샘플링하면 오디오와 동영상 화면 싱크가 맞지 않는 경우가 있는데, 이걸 그 원본파일의 특성인 것 같습니다. 이럴 때는 꼭 44100Hz로 바꾸려고 하지 말고 그냥 48000Hz로 놔두는 편이 좋습니다. (물론, 억지로 44100Hz로 바꾸고 사운드 부분에 신경써서 몇가지 조작을 해보면 어느 정도 싱크를 수정할 수는 있습니다만 그것 보다는 차라리 48000Hz로 그냥 두는 것을 권하고 싶습니다.)

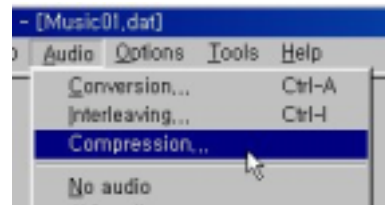


그 다음에는 "Audio" → "Interleaving" 부분을 확인합니다.

그림에 보이는 것처럼 설정되어 있으면 됩니다.

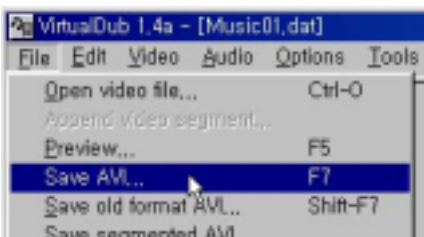
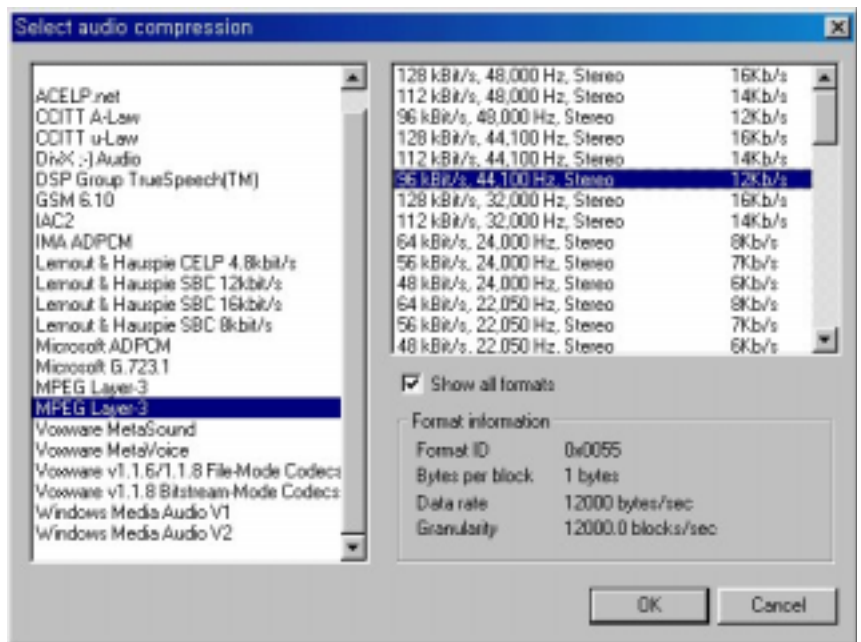


이제는 오디오의 재압축용 코덱을 설정해야 합니다. "Audio" → "Compression"을 선택하면, 아래와 같이 사운드 부분을 설정할 수 있는 코덱이 표시되는데,



여기서 "MPEG Layer-3"와 "96kbps / 44100Hz / stereo"를 선택합니다.

★ 만약, 앞에서 48000Hz를 선택했다면 여기서도 48000Hz를 선택해야 합니다.



자, 이제 설정이 다 끝났으면 "File" → "Save AVI..."로 동영상을 저장하면 되겠습니다.

★ 만족할 만큼 용량이 줄었나요? 필자의 경험으로는 DivX로 인코딩된 Sound of Music CD1의 경우 원본이 710MB였고, 이중 사운드 부분이 85MB 정도였는데 128kbps 데이터를 96kbps로 줄여 인코딩해서 최종 동영상을 693MB로 만들 수 있었습니다. 대략 10MB 정도가 줄어든 셈이죠.....(물론, 음질이 약간 떨어지긴 했지만, 별로 큰 차이를 느낄 정도는 아니었습니다.....^^)

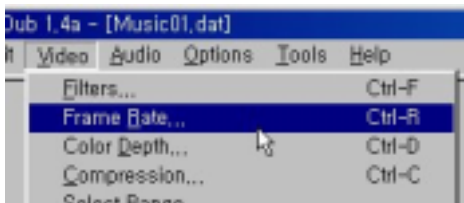
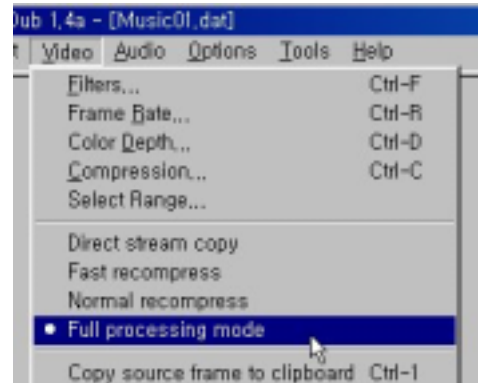
## 2. 동영상을 압축하는 방법

### 2.1. 프레임 레이트를 줄이는 방법

사운드 압축만으로 원하는 만큼 용량을 줄일 수 없는 경우에는 부득이 동영상을 줄여야 합니다. 동영상의 변환과정은 시간도 많이 걸리고 수준급 이상의 시스템 사양을 요구하는데, P3 500 / 128 MB RAM 이상이 권장 사양입니다.

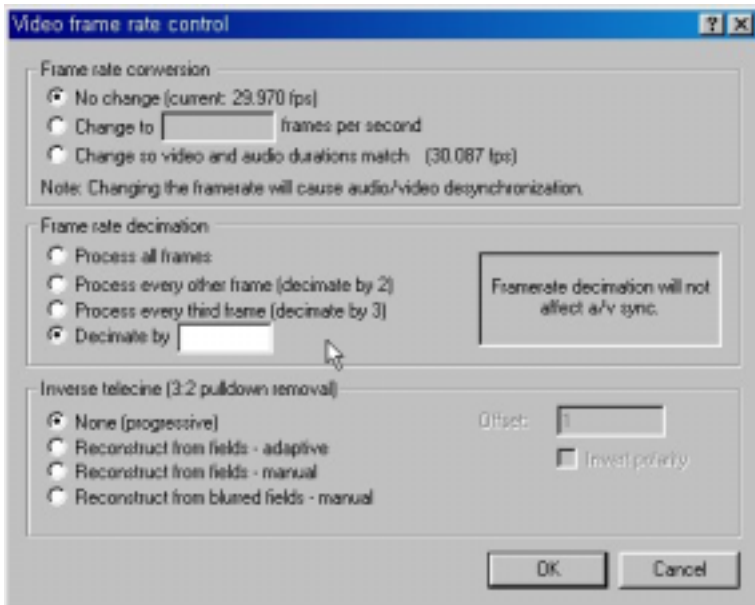
동영상의 압축과정은 사운드의 압축과는 별개이므로, 사운드 압축과 동영상 압축을 모두 적용할 수도 있습니다만, 사운드 압축은 앞에서 설명했기 때문에 동영상 압축에 대해서만 설명하도록 하겠습니다.

우선 원본 데이터를 불러온 다음 "Video" 부분이 "Full processing mode"로 설정되어 있는지 확인합니다.



그 다음으로, "Video" → "Frame Rate..."를 선택합니다.

그러면, 그림과 같은 설정창이 나오는데, 이 중에서 "Frame rate conversion"은 원본 동영상의 초(秒)당 프레임수를 변화시키는 것입니다. 이 부분에서 두 번째 줄에 있는 "Change to .....second"를 체크하고 박스안에 원하는 프레임레이트를 써 넣어서(예를 들어 "23.976") 원본의 프레임레이트를 바꿀수는 있습니다. 그러나, 이렇게 바꾸는 경우에는 오디오와 동영상의 싱크가 어긋나게 되기 때문에 이 부분은 "No change"로 그냥 두시길 권합니다.



두 번째 영역인 "Frame rate decimation"은 일정한 간격으로 한 프레임씩을 빼내는(없애는) 것입니다.

"Process every other frame"을 선택하면, 프레임수가 원본파일보다 절반으로 줄어들며

"Process every third frame"을 선택하면, 프레임수가 원본파일의 1/3 이 됩니다.

"Decimate by" 다음에 나오는 박스에 수치를 써 넣으면 그 수치분의 1로 프레임수가 줄어들게 됩니다. 이 방법은 오디오 싱크가 맞지는 문제가 없

고 하지만, 필자로서도 장담을 하지는 못하겠습니다.

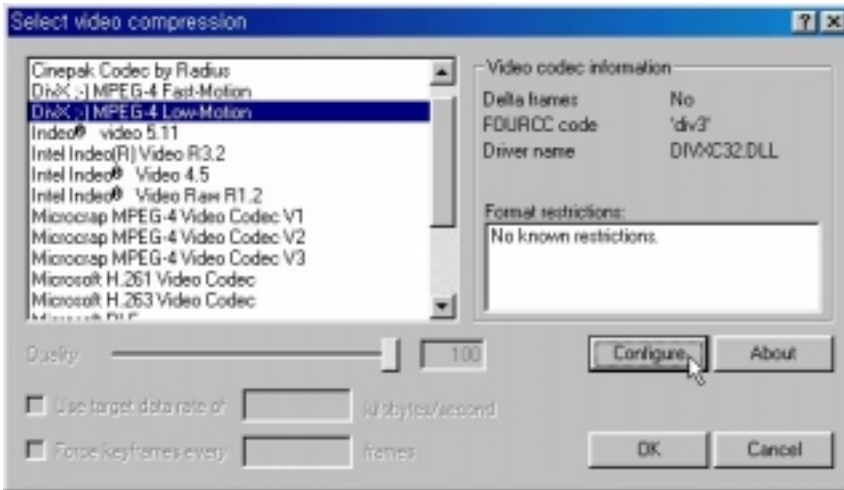
이제 설정이 끝났으면 "File" → "Save AVI..." 로 저장하면 됩니다.

★ 이 방법은 용량을 상당히 줄일 수는 있으나, 자막의 싱크나 사운드 부분에 대해서는 필자도 해본적이 없어 장담할 수 없습니다.

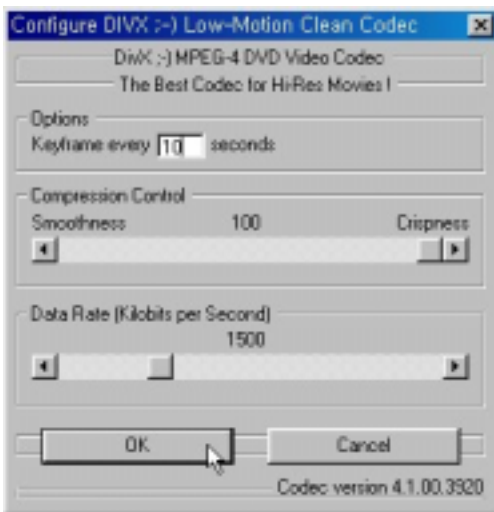
## 2.2. 비트 레이트를 줄이는 방법

이 방법은 원본 동영상을 좀더 낮은 비트 레이트로 재 인코딩하는 것입니다. 당연히 원본 화질보다 화질이 떨어질 수 있습니다.

우선은 원본 동영상 파일을 불러온 다음 "Video"를 "Full processing mode" 로 설정해 놓습니다.



다음으로 "Video" → "Compression"을 선택해 아래와 같은 비디오 코덱 설정창이 열리면, DivX(-)MPEG-4Low-Motion을 선택하고, "Configure" 버튼을 눌러 코덱의 옵션을 설정합니다.



왼쪽 그림과 같은 설정 창에서 중요한 설정은 Data Rate 가 되겠습니다.

Data Rate를 1000 이나 혹은 그 이하로 낮추어 설정합니다. (물론, 이 수치는 원본 동영상에 따라 달라지므로 참고적인 수치일 뿐이며 결정은 각자가 알아서 하셔야 할 것입니다. 누구도 정답은 알지 못합니다.)

설정이 끝나면 "File" → "Save AVI..." 해서 파일을 저장합니다.

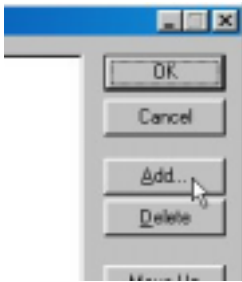
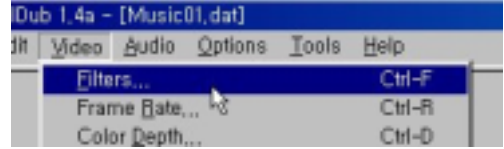
★ 이 설정 과정에서 앞에서 설명한 "Filter" 중의 "Sharpness" 등의 기능을 추가적으로 선택해 사용할 수 있으며, 이것 또한 각자의 선택에 따른 옵션 사항입니다.



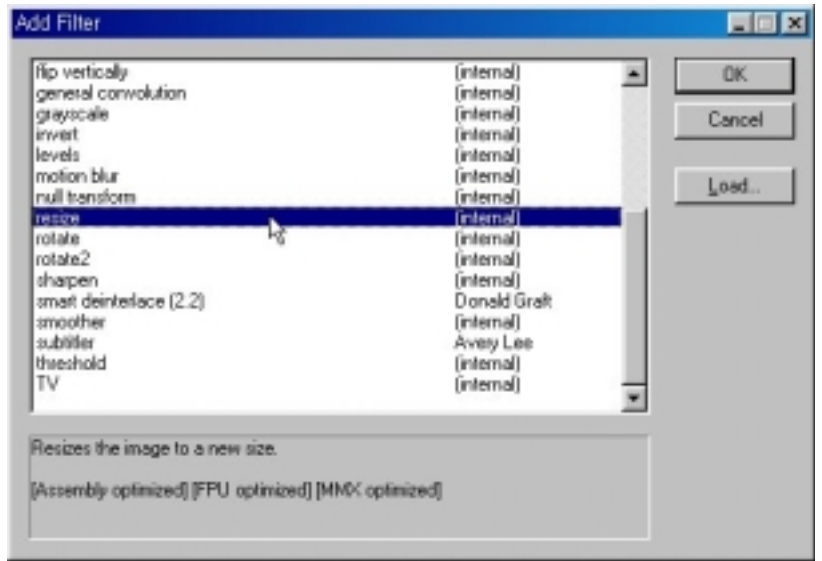
### 2.3. 픽셀 수를 줄이는 방법

이 방법은 동영상의 픽셀수를 줄이는 즉, 화면의 크기를 줄이는 방법입니다. 만약, 원본 동영상이 720×### 로 되어 있는 경우라면, 이를 적당한 크기로 줄여도 화질이 크게 떨어지지 않을 것입니다.

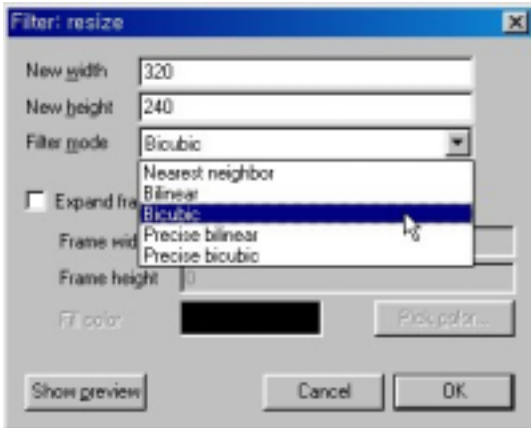
우선 원본 동영상을 불러온 다음 "Video"는 "Full processing mode" 로 설정해 놓고, "Video" → "Filter..."를 선택합니다.



그다음 나타나는 창의 오른쪽에 있는 메뉴중에서 "Add..." 버튼을 누르면 적용가능한 필터를 선택할 수 있는 아래와 같은 창의 뜹니다.



여기서, "resize"를 선택하면 또 다른 설정 창이 나타납니다.



왼쪽 그림과 같은 설정창이 뜨면, 우선 "Filter mode"를 "Bicubic" 으로 설정합니다.

그 다음으로 창 아래쪽에 있는 "Show preview" 버튼을 누르면, 원본 동영상을 보여주는 창의 뜹니다.



이제 이 창의 크기와 모양을 살펴가면서 "New width" 와 "New height" 에 적당한 수치를 써 넣어 봅니다. (이때 수치를 써 넣고 다음칸으로 커서를 옮기면 미리보기 화면 크기가 새로 지정한 크기로 바뀌니까 이를 보면서 수치를 결정하면 되겠습니다.)

★ 픽셀수를 결정할때는 반드시 16의 배수가 되도록 해 주시기 바랍니다. 동영상은 근본적으로 16×16 룰을 따르게 되어 있기 때문에 16의 배수가 되도록 설정을 해야 화질이 덜 손상됩니다.

적당한 크기가 설정이 되었으면 이제 동영상 코덱등을 설정하고 나서 파일을 저장하면 되겠습니다.

동영상 코덱이나 오디오 코덱의 설정은 이미 앞에서 설명한 내용이면 충분하리라고 봅니다.

앞에서 설명한 여러 방법들을 적절히 조합해서 각자의 취향에 알맞은 동영상을 만들어 보시기 바랍니다.

## [ 참고 ] VirtualDub 의 압축 모드 들

이 프로그램은 4가지의 기능모드를 가지고 있다.

### 1. 다이렉트 스트림 카피(Direct Stream copy)

① 기능 : 이 모드에서는 아무런 작업없이 입력파일을 그대로 출력파일로 복사합니다.

② 이점 :

- ㉠ 빠른 작업 : 많은 입력 파일로 부터 400fts 이상의 속도로 작업가능
- ㉡ 화질·음질의 저하 없음

③ 단점 :

- ㉠ 작업은 반드시 키프레임에서부터 시작되어야 함
- ㉡ 모든 프레임이 카피됨
- ㉢ 작업되는 동영상 볼수 없음

### 2. 빠른 재압축(Fast repack)

Fast repack 모드는 비디오 스트림의 재압축을 허용한다. 이 모드에서 이 프로그램은 하나의 비디오 포맷을 다른 포맷으로 압축하기 위한 가장 빠른 압축기를 자동으로 끌어온다.

① 장점 :

- ㉠ 자동으로, 포함된 비디오 코덱 중에서 가장 빠른 형태를 찾아 재압축이 진행된다.

② 단점 :

- ㉠ 코덱이 일반적인 이미지형태에 적용되지 않는 경우는 동작하지 않을수도 있다.
- ㉡ 작업진행중 동영상을 볼 수 없다.
- ㉢ 일부 비디오 코덱은 YUV 데이터를 올바르게 다루지 못하므로, 작업후의 동영상이 아래위가 뒤집어진 형태가 될 수 있다.

### 3. 느린 재압축(Slow repack)

이 모드에서는 입력 동영상을 RGB 모드로 풀은 후에 다시 선택된 비디오 압축기를 사용해 압축한다.

① 장점 :

- ㉠ 압축기와 해체기가 작동된다면, 어떤 형식의 입력동영상도 다른 형식의 출력동영상으로 재압축할 수 있다.
- ㉡ RGB 모드는 매우 안정한 형태이므로, 보다 많은 코덱과 호환될 수 있다.

② 단점 :

- ㉠ Fast repack 모드보다는 느리다.
- ㉡ 압축기와 해체기가 서로 호환되어야 한다. 만약, 해체기가 16bit RGB 모드로 동작하고, 압축기는 24bit RGB 모드로 동작한다면, 이 모드는 작동하지 않는다.

### 4. Full processing (디폴트로 잡힌다.)

필터를 포함한 모든 비디오 편집기능이 가능한 모드

① 장점 :

- ㉠ 모든 필터링 옵션이 가능하다
- ㉡ 어떠한 해체기-압축기의 조합도 가능하다

② 단점 :

- ㉠ 가장 느린 모드이다.
- ㉡ 모든 이미지가 32bit RGB 모드로 풀린 다음, 압축되므로 메모리 소요가 가장 많다.

## §5. 기존 동영상을 편집하는 방법

### 1. 동영상 파일 나누기

동영상 파일을 분할하는 프로그램들은 Avichop을 비롯해 여러 가지가 있습니다. 사용법도 간단하기 때문에 널리 쓰입니다. 그러나, 이런 프로그램들은 동영상 파일을 무조건 용량기준으로(그나마도 정확하지 않게.....IT.IT) 원하는 부분만을 잘라내거나 원하는 부분에서 정확히 나눌 수가 없습니다. 심지어 파일 분할과정에서 화면이 깨지는 경우도 발생합니다.

더구나, Avichop은 Win2000 에서는 제대로 동작하지 않는 버그가 있습니다.

따라서, 필자로서는 DivX 동영상 파일의 분할은 VirtualDub을 사용하시기를 강력히 권하고 싶습니다.

파일을 나누기 위해서는 우선 동영상의 시간이나 용량을 미리 생각해 두어야 합니다. 그런 다음 원본 동영상을 불러 옵니다.



동영상이 불러진 상태에서 프로그램 창 아래쪽의 위치선정 막대를 적당히 이동해 자르고자 하는 위치 근처로 옮겨 놓습니다. (이때 막대 아래쪽에는 이동하고있는 위치가 시간과 프레임수로 표시되며, 동영상도 화면에 나타납니다. 물론, 정지화면이지만...)



적당한 위치 근처까지 이동했으면, 왼쪽 그림에서처럼 노란 열쇠가 표시된 버튼 (키프레임 이동버튼)을 눌러가면서 적당한 위치를 잡습니다.

★ 파일의 분할은 반드시 키프레임에서 해야 동영상에 문제가 생기지 않습니다.



자르고자 하는 키프레임 위치가 결정되었으면, 왼쪽 그림에서처럼 구간지정 버튼을 누릅니다. (지금 그림에서 마우스가 위치해있는 버튼은 잘라낼 구간의 시작위치를 지정하는 버튼입니다.)



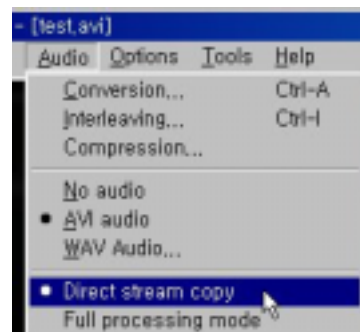
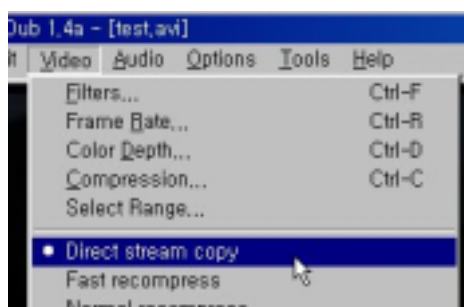
시작위치의 지정이 끝나면, 다시 위치이동막대를 움직이든, 아니면 키프레임 이동버튼을 누르든 해서 잘라낼 마지막 위치로 이동합니다. (대부분의 경우에는 파일 뒷 부분을 잘라내게 되므로 왼쪽그림에서처럼 맨 마지막으로 이동하는 버튼을 눌러 한번에 끝으로 가는 것이 일반적입니다.)



잘라낼 부분의 끝에 위치이동막대가 옮겨지고 나면 왼쪽 그림처럼 구간지정 버튼을 눌러 구간의 끝을 지정합니다. 그러면 오른쪽 그림에서처럼 지정된 구간이 파란색으로 표시됩니다.



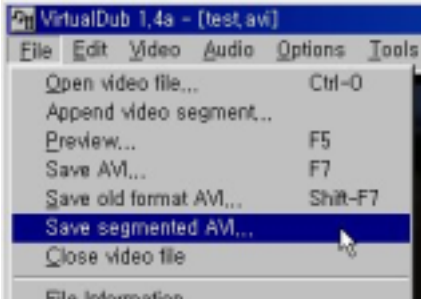
이제, 아래의 그림처럼 "Video" 와 "Audio" 모드를 전부 "Direct Stream Copy" 로 지정합니다.(프로그램을 처음 실행하면 Audio는 "Direct stream copy"가, Video는 "Full processing mode"가 디폴트로 잡히므로 특히 Video 부분에 주의해야 합니다.)



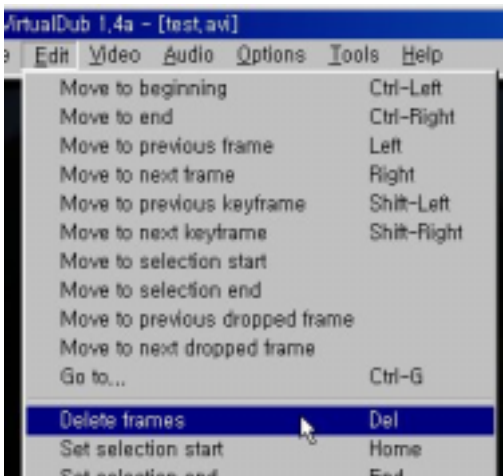
마지막으로 선택한 부분을 잘라내 저장하는 과정입니다.

이 과정에서 몇 가지 방법을 사용할 수 있습니다. 즉, 아예 원본 동영상을 잘라 짧은 동영상 2개로 분할 할 수도 있고, 아니면, 원본 동영상은 그대로 둔채로 두 개로 나뉘어진 동영상을 별도로 만들 수도 있습니다.

**(1) 원본 동영상을 잘라 짧은 2개의 동영상만 만들고자 하는 경우**



"File" → "Save segmented AVI..."를 선택해 앞에서 지정한 부분을 다른 이름으로 저장합니다.



그 다음으로 "Edit" → "Delete frames"를 선택해 앞에서 지정한 부분을 잘라낸 다음

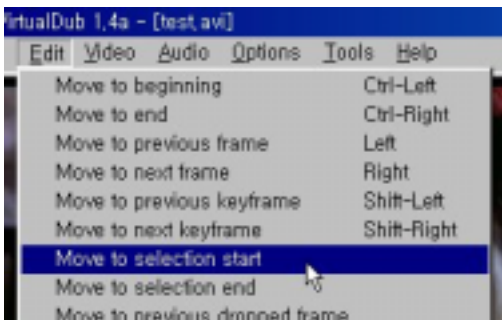
"File" → "Save AVI"를 선택해 원본파일을 원본파일과 같은 이름으로 저장합니다. 물론, "덮어쓸까요?"를 물어보면 당연히 덮어써야죠...

**(2) 원본 동영상과 짧은 2개의 동영상을 별도로 만들고자 하는 경우**

혹시나 작업과정에서 오류가 발생할 수도 있기 때문에, 필자의 생각으로는 (1)에서 설명한 것보다는 별도의 동영상을 만드는 쪽을 권하고 싶습니다. 단, 하드의 용량이 부족한 경우에는 어쩔수 없이 (1)번에서 설명한 방법을 사용해야겠죠...

이 과정도 나뉘어질 부분을 선택하는 과정은 앞에서 설명한 것과 같습니다. 한가지 다른 점은 영역 지정을 2차례 해야 한다는 것입니다.

즉, 앞에서 설명한 방법대로 영역 지정을 한 다음 "File" → "Save segmented AVI.."를 선택해 지정한 뒷 부분을 저장합니다. (이 과정은 (1)에서 설명한 것과 같습니다.)



다음으로, 다시 앞부분 동영상을 영역 지정하는 과정입니다.

왼쪽 그림처럼 "Edit" → "Move to selection start"를 선택합니다.



그러면 오른쪽 그림처럼 위치막대가 뒷부분 영역지정된 처음 위치로 옮겨지는데, 이때 그림에서 마우스가 위치한 것처럼 오른쪽 영역지정 버튼을 누릅니다.



다음으로 왼쪽 그림처럼 맨처음 프레임으로 이동 버튼을 누르면 위치막대가 처음 위치로 이동하게 됩니다.



이제 왼쪽 그림처럼 영역지정 버튼을 눌러 앞부분 영역을 지정합니다.

영역지정이 다 끝났으면, "File" → "Save segmented AVI.."를 선택해 앞부분 동영상을 저장합니다.

윈도우 탐색기를 열어 확인해 보면, \*.00.avi 와 \*.01.avi 의 두 개의 파일이 새로 저장되어 있는 것을 보실 수 있을겁니다.

## 2. 화면 정지 해결

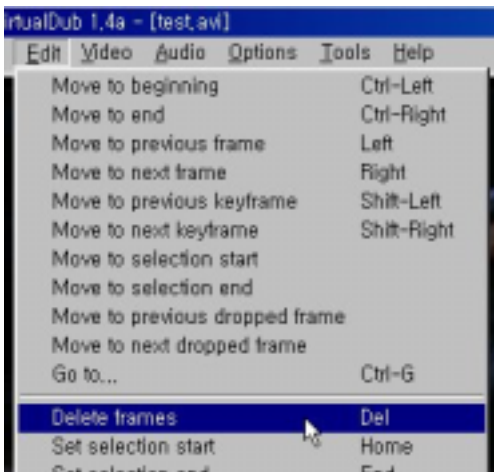
인터넷을 통한 전송과정에서 동영상 파일에 이상이 생기는 경우가 간혹 있습니다. 이럴 경우에 가장 좋은 방법은 이상이 없는 동영상 데이터를 구할 수 있으면 이를 이용해 패치 파일을 만들어 이상이 생긴 동영상을 패치하는 것입니다. 이런 패치의 제작은 국내에서는 거의 "하형진" 님 한분이 하고 계시며(하형진님 감사합니다...^^), 필자는 패치의 제작방법에 대해서는 알지 못합니다.

때로는 패치의 제작이 불가능한 경우도 있는데, 이런 경우에는 VirtualDub을 이용해 화면정지 오류를 없애 줄 수 있습니다.(★ 이 방법은 정상적인 패치 사용보다는 좋지 않고, 파일에 심각한 오류가 있는 경우에는 적용 불가능하므로 특정 부분에서 화면이 정지하는 오류이면서 패치의 제작이 불가능한 경우에 한해 적용하시기 바랍니다. )

우선 원본 동영상을 불러온 다음, 화면이 정지하는 위치까지 위치 이동 막대를 옮깁니다.



그 다음에 왼쪽 그림에서 보인 순서대로 한프레임 앞으로 이동해 위치지정하고 2~3 프레임정도 뒤로 이동해 위치지정을 합니다.



다음으로 "Edit" → "Delete frames"를 실행해 앞서 선택한 프레임을 잘라버립니다.

"Audio" 와 "Video"를 모두 "Direct Stream copy" 로 한 다음 파일을 저장하면 작업 끝입니다.

★ 결론적으로 이 방법은 이상이 있는 프레임 부분을 잘라내는 것입니다. 잘라낸 부분이 약간 표시날수는 있지만, 그리 큰 문제는 생기지 않을 것입니다. 또한 오디오와의 싱크도 약간 문제가 생길 우려가 있기는 합니다만, 이것 역시 큰 문제는 없을 것입니다.

### 3. 오디오 싱크의 조정

간혹 인코딩과정의 문제로 인해 동영상과 오디오가 맞지 않는 경우가 발생합니다. 이럴 경우 VirtualDub을 사용해서 어느 정도의 수정이 가능합니다.

#### (1) 동영상과 오디오의 싱크가 처음부터 약간씩 어긋나는 경우

대체로 이런 경우는 동영상과 오디오의 싱크가 약 0.5초 정도씩 차이가 나는 경우가 많습니다. 이렇게 처음부터 끝까지 같은 간격으로 차이가 나는 경우에는 간단히 수정이 가능합니다.

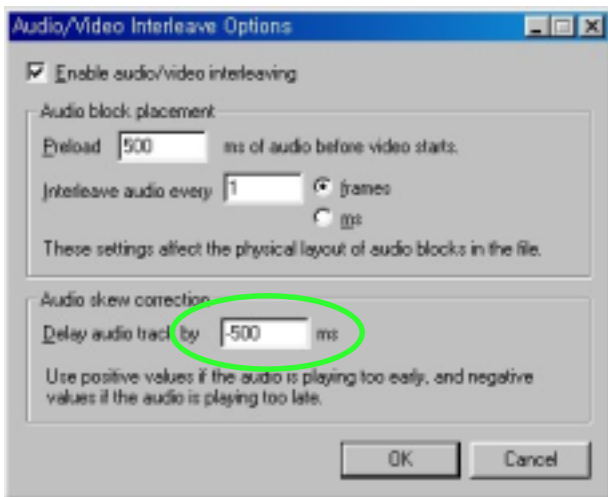
우선은 원본 동영상에서 사운드 부분을 .wav 파일로 추출해 놓습니다. (이 과정은 앞에서 이미 설명했기 때문에 생략합니다.)

다음으로, "Audio"는 "Full processing mode"로, "Video"는 "Direct stream copy"로 설정합니다.

이제 "Audio"에서 "WAV Audio..."를 선택해 미리 추출해 놓은 wav 파일을 불러옵니다.

"Audio"의 "Conversion" 설정항목은 모두 "No change"로 놓아두고, "Compression" 항목에서는 적당한 코덱을 설정합니다. (이 부분은 이미 앞에서 설명한 내용입니다.)

이 작업에서 가장 중요한 부분은 "Audio"의 "Interleaving..." 항목입니다.



왼쪽 그림은 Audio의 Interleaving을 선택했을 때 나오는 설정창입니다.

여기에서 타원형으로 표시된 부분에 적당한 숫자를 써넣음으로써 오디오를 앞/뒤로 조정할 수 있습니다.

지금 그림에서는 "-500"으로 설정했는데, 이 경우는 화면보다 오디오가 0.5초 정도 늦게 나오는 경우에 설정하는 값입니다.

만약, 오디오가 동영상보다 0.5초 정도 빠른 경우에는 "500"으로 넣어 줍니다.

결국 오디오를 당기려면 "-"값을, 미루려면 "+"값을 써넣으며, 시간단위는 ms(1/1000초)이므로, 1초를 1000으로 계산합니다.

설정이 다 끝났으면, "File" → "Save AVI..."로 저장하면 됩니다.

★ 만약, 작업이 다 끝난 동영상 파일을 재생했을 때 아직도 문제가 생긴다면, 위의 Interleaving 항목의 설정을 바꿔가면서 잘 될 때까지 반복해 보는 수 밖에는 없습니다....ㅠ.ㅠ

## (2) 동영상과 오디오의 싱크가 뒤로갈수록 점점 어긋나는 경우

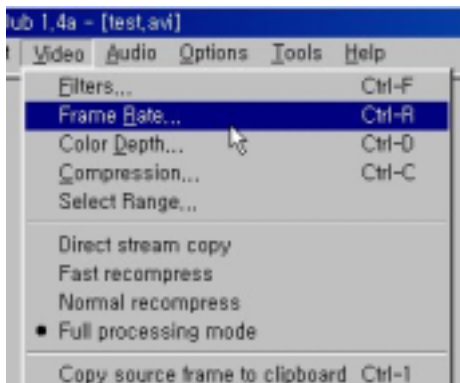
이 경우를 수정하는 것은 상당히 어렵습니다. 필자로서도 정확한 설정치를 제시할 수 없으며, 많은 시행착오를 거쳐야 할 것입니다.

우선, 이런 현상의 원인이 정확히 무엇인지부터 파악해 보셔야 할 것입니다. 다른곳에서는 멀쩡한 파일이 자신의 컴퓨터에서만 그럴다면 시스템상의 문제인 경우입니다. 실제로 동영상 자체의 문제이기보다는 시스템의 문제인 경우가 더 많습니다.

또, 일부 사운드카드의 경우에는 48000 Hz 의 샘플링 레이트의 사운드를 44100Hz로 재생해 문제가 되는 경우가 있습니다.

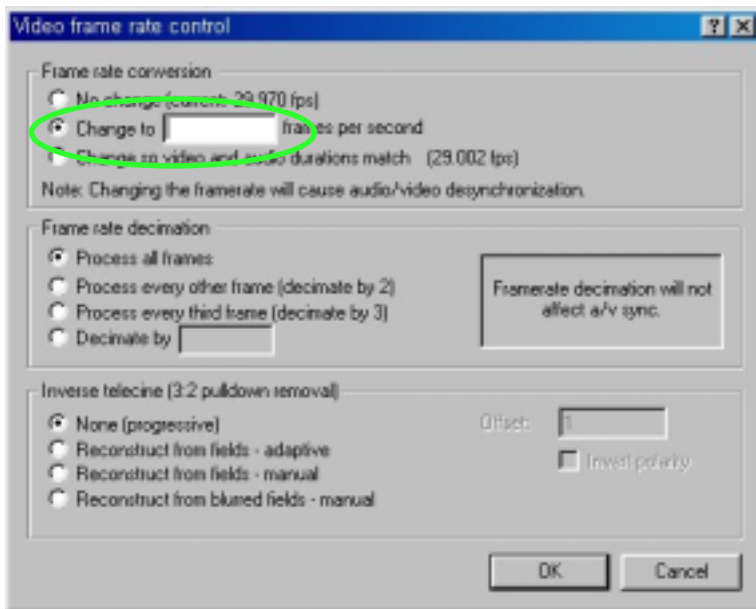
이런 시스템상의 문제가 아니라면, 이제 방법은 두 가지입니다. 하나는 동영상의 프레임 레이트를 조절해 재생속도를 바꾸는 것이고, 다른 하나는 사운드 데이터의 길이를 조절하는 것입니다.

VirtualDub 에서는 사운드 데이터의 길이 조절은 불가능하기 때문에 동영상의 프레임 레이트를 변화시켜야 합니다.



"Video"를 "Full processing mode"로 설정한 다음

"Frame Rate..."를 선택합니다.



왼쪽 그림과 같은 설정 창이 나타나면, 타원형으로 표시된 곳에 원하는 초당 프레임수를 써 넣습니다.

물론, 이는 대략 짐작으로 할 수 밖에 없으며 동영상 보다 사운드가 점차 늦어지는 경우에는 동영상을 늦게 플레이되도록 하기 위해 원본의 프레임 레이트보다 수치를 적게 써 넣어 주어야 하며(예: 원본이 29.970인 경우는 29.968처럼) 반대의 경우에는 수치를 많게 써 넣어 주어야 합니다.

다음 과정으로 동영상 압축코덱 설정 등을 해 준 다음 저장하면 됩니다. 이 과정에서, 사운드는 "Direct stream copy"로 그냥 둡니다. 또한, 저장명령을 실행하면 "오디오와 싱크가 어긋날 수 있다" 는 경고 메시지가 뜨는데, 그래도 무시하고 그냥 진행합니다.

★ 프레임 레이트를 변경한 경우, 자막이 맞지 않을 수도 있습니다.

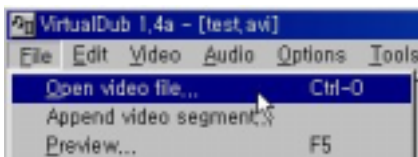
★ VirtualDub 이외에도 동영상의 프레임레이트만을 간단히 바꿀수 있는 공개 프로그램들이 있으므로 그런 것들을 사용하는 것도 좋겠습니다.



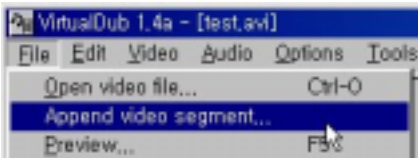
★ 사운드 데이터의 길이를 조절하는 것은 CoolEdit 같은 오디오편집 프로그램에서 가능합니다만, 상용프로그램이기 때문에 여기서는 다루지 않습니다.

#### 4. 두 동영상을 연결해 하나의 동영상으로 만들기

일부 스크리너버전의 경우 용량도 얼마 안되는게 1,2 편으로 나뉘어져 있어 이를 연결하고 싶어하는 경우가 있습니다. 이런 경우 VirtualDub을 이용해 아주 간단히 연결할 수 있습니다.



우선, "File" → "Opne vedio file..."을 이용해 1번 동영상을 불러 온 다음



다시 "File" → "Appedn video segment..."를 이용해 2번 동영상을 불러옵니다.

이제 "Video" 와 "Audio"를 모두 "Direct stream copy" 로 설정한 다음, "File" → "Save AVI.." 로 저장하면 됩니다.

★ 만약, 첫 번째 동영상의 뒷부분과 두 번째 동영상의 앞부분이 서로 얼마간 겹치는 부분이 있다면 이는 저장하기 전에 미리 잘라내 줘야 할 것이며, 잘라내는 방법은 이미 위에서 설명했으므로 생략합니다.

★ 동영상의 절단이나 합병 작업을 하면 가장 문제가 되는 것이 자막입니다. 이때 자막을 분리하거나 합병하는 방법은 필자가 씨네 프로그램 자료실에 별도로 올린 "자막 만들기 매뉴얼"을 참고하시기 바랍니다.

### ★★★ 만들고 나서 ★★★

벌써 오래전부터 VirtualDub 의 매뉴얼을 만들어보야겠다고 생각만 하고 필자가 게을러서 차일 피일 미루다가보니 아주 간단한 조작으로 변경이 가능한 것들도 프로그램 사용법을 몰라 답답해하시는 분들이 많고, 저한테 개인 메일로 문의하시는 분들도 계시기에, 더 이상 미룰수가 없어 결국은 만들게 되네요.....^^;;;

본래, VirtualDub 에서의 인코딩을 위주로 만들려고 한 것이 아니기 때문에 인코딩 부분이 좀 충실하지 못하지만, 동영상의 편집이 아닌 인코딩쪽에 관심이 있으신 분들은 제가 먼저 만들어 올렸던 FlaskMpeg을 이용한 인코딩 매뉴얼(씨네 프로그램 자료실에 있습니다.)을 참고하시기 바랍니다.

아울러, 본 매뉴얼상에 필자가 미처 알지 못한 오류가 있을 수도 있으며 또 본 매뉴얼은 아주 기본적인 사항들을 위주로 제작한 것인 만큼, 기본 기능들을 익히신 다음에 이를 조합/응용하거나 Filter의 적용 등의 기법들은 스스로 발전시켜 가시기 바랍니다.