

VirtualDub documentation: Capture

9 min read • [original](#)

How fast of a CPU do I need?

This is a hard question to answer, but I can tell you what I've been able to do:

- Celeron 300 @ 450, Hauppauge WinTV, 44KHz 16-bit stereo PCM, 640x480 YUY2 at 29.97 fps compressed with PicVideo at quality 18, on-screen display disabled.
- Pentium III 733, Hauppauge WinTV, 44KHz 16-bit stereo PCM, 640x480 YUY2 at 29.97 fps compressed with Huffvuv in Predict Median, Preview on.

Note that these are with a 16-bit desktop. You will take a sizeable hit with a 24-bit or 32-bit desktop if you have Preview on.

VirtualDub says I don't have a capture device, but I know I have one.

VirtualDub needs a Video for Windows capture driver to capture. Most Firewire (DV) devices do not provide a VFW driver, and thus cannot be used by VirtualDub at all. Also, ATI appears to be shipping their current devices with a WDM (Windows Driver Model) driver only; this can be used indirectly by VirtualDub through a Microsoft wrapper, but it is crippled in functionality and it also appears that the wrapper is buggy.

The wrapper will show up as "Microsoft WDM Image Capture (Win32)." If it works for you, great.

I can capture DV through VirtualDub, but I get the audio out of my sound card instead.

Video for Windows only captures audio through a sound device; with DV, the audio is interleaved with the video itself. I don't have any DV specs, so I can't make VirtualDub extract the audio.

I get no sound.

First, check that you aren't missing the pass-through. Most video capture cards don't have on-board sound capture and instead use your sound card, so if you don't connect the cable from the capture board to the sound input, you won't get any sound.

Next, bring up Volume Control (`sndvol32.exe`), go up to the menu and choose Properties, then switch to Recording controls. Make sure Line-in, Aux, or whatever input you are using is selected. Voice chat programs are notorious for automatically switching inputs for you, and some video capture drivers are known to reset volume levels as well.

Finally, if you have a Hauppauge WinTV, check for an `AUDIOSEL.EXE` application on your hard disk or driver CD. It can be used to force the audio pass-through on for those cards. (Note: If you are using an external source with

I get sound, but it has cracks and pops in it.

My video has greenish lines in it.

Occasionally, I see thin horizontal strips in the captured video that looks like they came from the last frame???

No one has a rock-solid answer for why these problems occur, but it appears to be caused by contention on the PCI bus, which then prevents the sound card and video capture devices from emptying their buffers in time. This problem is reported more frequently on motherboards that have a VIA chipset, or

in systems that have a SoundBlaster Live! sound card. In the former case, try upgrading your VIA 4-in-1 drivers first, and if that is not sufficient, check for a motherboard BIOS update that specifically addresses your problem. As for the Live!, the Creative driver is known to cause problems by lowering the latency timer of the PCI bus. (The result is similar that of making the red lights appear every ten seconds at a four-way intersection.) In that case, try installing the Microsoft drivers instead.

Also, check the websites for the hardware manufacturers to see if they have utilities which may help. For instance, Pinnacle has a PCI adjustment utility for some of their cards.

If you are using RGB24 for your raw capture format, switch to YUY2 to drop your raw datarate by 33% (see below), which may be enough to lower PCI bus load to workable levels.

I see a greyish, wavering line at the bottom of my capture. It looks sort of like the picture, but distorted.

If it's only a few lines at the bottom, it's likely VHS head switching noise. Ignore it or crop it out afterward.

I see a lot of thin lines in my image, like a comb.

That's interlacing -- what you're seeing is two fields, taken at slightly different times, and combined together to form a single frame. You'll see it whenever you capture with a height more than 240 lines (NTSC) or 288 lines (PAL). Interlacing is not a defect in the video and cannot be "fixed" -- it's a fundamental aspect of the way video is encoded.

Which capture format should I choose if I'm going to use video compression?

Ultimately, you will need to choose a format which is compatible with the codec, but if you have a choice, always choose YUY2, UYVY, or YVYU over RGB24. Here's why:

TV cards generally use one of two chips, either the Zoran ZR36120 (FlyVideo-derived devices), or the Brooktree/Conexant BT8x8 series (everyone else). If you look at the datasheets for these chips you will find that they either accept YUV 4:2:2 data from the decoder or produce it internally. Guess how they produce RGB24 data when you ask for it? You guessed it, the YUV 4:2:2 data is *upsampled* to RGB24. This means that the extra color resolution that RGB24 gives is entirely fabricated.

Choosing YUY2/UYVY/YVYU over RGB24 typically gives you a sizable speed increase, and gives you better compression with HuffYuv.

RGB16 introduces horrible banding and signal loss, while RGB32 just adds an extra 8-bits of junk to RGB24. There is no point in using either.

Why do I get errors when I try to capture with a height above 240 (288 for PAL/SECAM)?

This following only applies to devices based on the BT848(A) or BT878(A).

NTSC video is composed of alternating fields of 240 scanlines, and PAL/SECAM of 288. If you specify a height equal to or lower than that, the capture driver only snags one field. If you specify a taller image, both fields are grabbed. In either case when you specify a size smaller than the height of the field(s) captured, the result is then scaled down.

The BT8x8(A) can only DMA to one destination at a time, and the output pixel format can only be set on a per-field basis. If you specify Overlay mode, then the capture driver will instruct the chip to DMA one field to memory for capture, and the other field to the video card. The catch is that if you are capturing both fields, both DMA lists have to point to main memory. This means that overlay mode is impossible when both fields are captured, and depending on your exact driver, you will either have overlay disabled when the capture starts, or, as is more often the case, get a cryptic error about being out of memory. Note that this is not a bandwidth issue -- you can capture 640x240 in overlay mode, but not 80x480.

To "fix" the problem, specify Preview mode; this unfortunately is slower since the CPU is now doing a

blit from the captured frame to the video card. It may be that your system is not fast enough to handle this, in which case you will need to disable the video preview entirely and capture blind. A secondary consequence of all of this is that you see exactly what is being captured in Preview mode, while you are actually seeing the other field in Overlay. If you have a strange source where one field is much cleaner than the other, it is possible with an Overlay display that you will see a clean display on screen and end up capturing crap. (I have had this happen.)

This may not apply to you if you have such a chip built onto your video card, since in that case the chip can DMA into video memory, and then other video hardware can transfer from there into main memory for capture.

Other people can capture 768x576, but my card doesn't do more than 640x480.

It's likely that the other people are dealing with a PAL (Phase Alternating Line) standard, while you are dealing with the NTSC (National Television System Committee) standard. PAL has fields of 625 scanlines at 50Hz while NTSC is 525 scanlines at 59.94 Hz. In addition to the resolution difference, PAL people should capture at 25 fps while NTSC people at 29.97.

I have lots of codecs in the normal video compression dialog, but some disappear when I go to capture!

You can only use codecs which are compatible with the current capture format, so try changing the capture format in the Video Settings or Video Format dialogs. RGB formats will give you the widest selection; YCrCb (YUV) formats will whittle the list down, and exotic formats such as VCR2 or MJPG will usually clear the list. This restriction is necessary because the format may require hardware assist to decode, and it may be impossible for VirtualDub to decompress the video during capture in order to transcode to another format.

What's the difference between compatibility mode and normal (internal) capture?

Compatibility capture mode (F5 key) is a no-frills, straight Video for Windows capture -- it lets Video for Windows do as much work as possible and is the failsafe mode you should use to diagnose problems when regular capture mode doesn't work. It has a number of disadvantages:

- Timing correction is disabled, so audio on long captures may be out of sync.
- The capture file will be invalid if it exceeds 2GB.
- Only one capture file is allowed -- no multisegment captures.
- VirtualDub won't be able to display the compressed video frame size.
- Video filtering options that change the size of the frame aren't allowed.
- Disk I/O is buffered, so you may encounter problems when pushing the limits of your hard disk bandwidth.

It's basically like Vidcap32.

Use normal capture mode (F6) when possible. If you find that you can't capture even in compatibility mode, chances are something is seriously wrong with your system configuration.

The audio isn't in sync in the capture file.

It's what we get for putting audio and video capture on separate clocks.

If you are running VirtualDub 1.4c or earlier, check the Knowledge Base for a possible workaround if your sound card isn't very accurate. (This problem will be fixed in V1.4d.) Also, if you're using compatibility mode, to capture, don't -- for VirtualDub to resync the audio during capture, you must not be using compatibility mode, and timing correction must also be active (it's in the capture menu).

The "lock video stream to audio stream" option in settings only works for compatibility mode; it sets the frame rate of the video stream so that it's the same length in time as the audio one. This is a nasty way to get audio in sync, because it will make editing harder -- to coerce the clips to a single frame rate for

rendering, frames will have to be dropped or duplicated. It also doesn't work well if the timing relationship between audio and video drifts over the course of the capture.

Why am I dropping frames?

Perhaps one of:

- Insufficient CPU power. If you're running near the limits (>90% CPU), then you can sporadically drop frames when busy scenes pass by.
- Slow hard disk. Note that most hard disks cannot handle capturing full frame, uncompressed video; that requires 18MB/sec for 16-bit RGB/YUY2 and 27MB/sec for 24-bit RGB, not counting filesystem and seek overhead. Also, if you don't have DMA set on an IDE disk, you'll get slower performance than usual and additionally will load the CPU significantly during disk access, often by 40% extra or more.
- Bus bottleneck.
- Bad source, such as VHS video tape, or a signal interruption such as changing the channel.
- Timing correction necessary to keep the audio synchronized. Most sound cards will deviate a small fraction from the video capture device, requiring that a few frames be dropped to keep the audio in sync. Another possibility is that the source is outputting a slightly slow or fast frame rate; old videotape can do this, as well as game consoles.

The key to remember is that one dropped frame per thousand is nearly indiscernable, but ten dropped frames at a single point is.

Why does my video capture stop at 1 hour, 11 minutes, but I still get audio?

This is a bug in many video capture drivers. The exact limit is actually 2^{32} microseconds, or 1 hour, 11 minutes, 34 seconds. Many TV tuner devices are susceptible to this bug, as are the miro/Pinnacle DCxx devices; you may be able to fix the problem simply by upgrading to the latest capture drivers. There are two incarnations of this problem. In the non-fatal version, video frames are still sent to the application, but the timestamp on the video stream starts over from zero. VirtualDub will correct for this problem automatically, allowing you to capture beyond 71 minutes. The other possibility is that the driver stops sending data altogether. VirtualDub will notify you if this occurs, but will not be able to capture past 71 minutes in this case.

Vanilla capture devices based on the Conexant/Brooktree BT848/878 reference drivers are usually either immune to the problem or have the non-fatal form of it. I'm told that the bug in Pinnacle's DC30(+) drivers still exists but is nonfatal as of the 1.41a drivers. As of this writing, the bug in the DC10/10+/20/20+ drivers is fatal and cannot be worked around.

There are apparently a few drivers that are even susceptible to half this limit (35 min.) due to using *signed* 32-bit arithmetic. VirtualDub does not yet detect or attempt to bypass the 35 minute limit that results.

I'm getting an error when I try to capture an AVI file bigger than 4GB.

You'll get this problem if you either (a) are running Windows 95/98/98SE/ME, or are saving to a FAT32-formatted partition. Either limits you to 4GB. If you want to capture a *single* file bigger than 4GB, you must have your capture partition formatted as NTFS and you must be running Windows NT/2000. The error message itself is caused by Windows, and not by the limitations of the AVI file format.

So how do I capture more than 4GB?

You must do all of the following:

- Use normal capture (F6 key) and not compatibility mode capture (F5 key).
- Check Capture/Enable multisegmented capture.
- Add one or more drives to Capture/Spill drives. Set the thresholds to 50MB, and make sure you use a full path (i.e. E:\ and not E:). Also, set priority of all drives to 0 to avoid a bug in V1.4c.

VirtualDub will then capture in segments to a bunch of 2GB files.

How do I change the channel or capture in MPEG-1/2?

You can't -- this functionality isn't available through the Video for Windows capture interface. The application that comes with your capture device can do it because it uses a proprietary interface that other programs can't use.

Original URL:

http://virtualdub.com/docs_capture.html