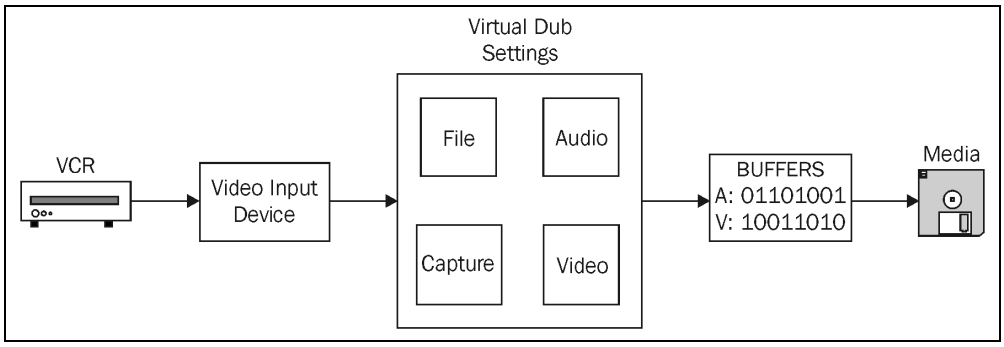
PACKT PUBLISHING VirtualDub Video: Sample Chapter

3
Capture Preprocessing

It is time to make some software settings for the actual capturing process, assuming we have set up all the equipment in the right order from the previous chapter. These settings include file, video, audio, and capture settings inside VirtualDub, and are necessary to get an appropriate video file. The following diagram is a graphical representation of the capture preprocessing workflow:



We will study all these configurations step by step, in this chapter. Moreover, we will consider a practical example in which the entire capturing process, from receiving input analog signals to preparing them for writing in a popular DVD format, will be illustrated.

By the end of this chapter, we'll be able to:

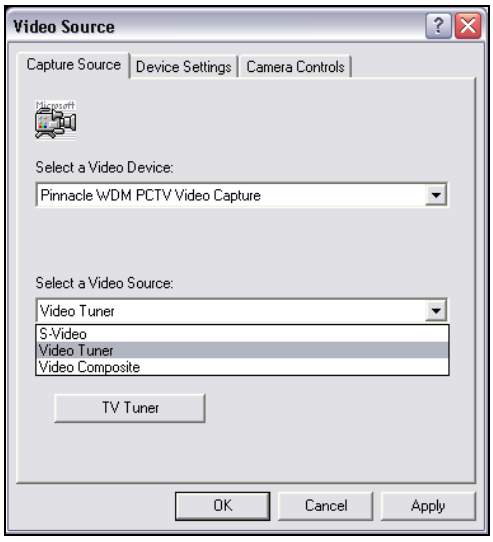
- Select a suitable approach (VFW or WDM) for the capturing process, based on your Windows version
- Specify file settings including: Name, Path, Allocated space

- Define suitable timing and disk I/O for a capturing process
- Achieve multisegment capturing
- Set the audio compression and volume
- Set the video characteristics: frame size, frame rate, color depth
- Set the video standard for capturing
- Set the desirable format and compression for captured data
- Define suitable buffer size and number
- Save capture settings for future use

Defining an Input Source for VirtualDub

We want to connect VirtualDub to the source so that it will receive the captured data. To do this, we can use the companion application that came with the capturing card, enable the input channel that we want to capture, and close the application.

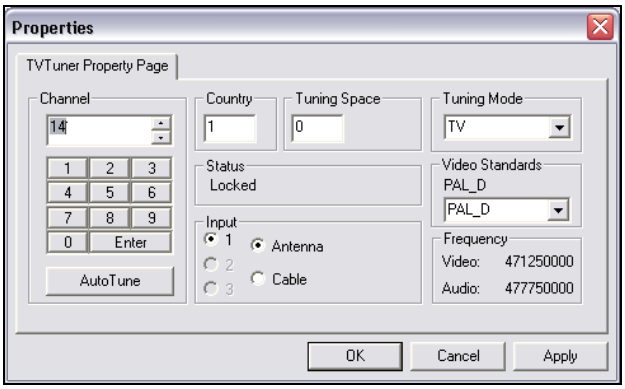
Run VirtualDub and select File | Capture avi. Another alternative is to open the Video Source window by pressing *S* in the Capture mode window (or selecting Video | Source from this window) and, in the Select a Video Source drop-down list, choose the desired input source. There are a number of input sources for video, which may vary based on the capturing card:



Choose the TV input source (i.e. Video Tuner), and click on TV Tuner in this dialog box. In the next dialog (assuming you've set the correct options while installing the new

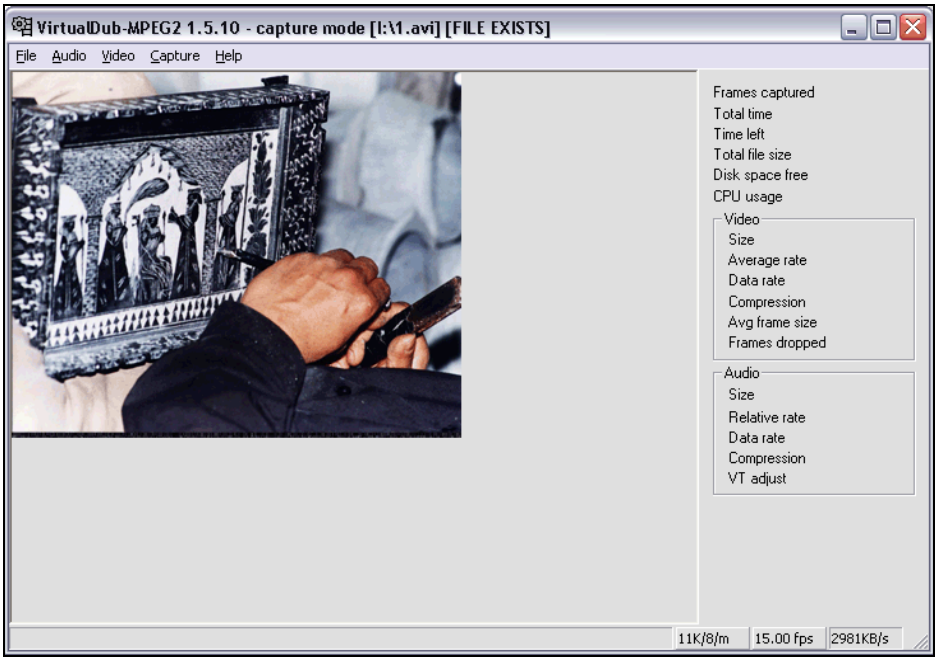
capture card), leave the settings unchanged, and just enter the number of the desired channel in the Channel group. Now, press *Enter*.

If no channels have been detected yet, it's possible to find them automatically by pressing the AutoTune button. After pressing AutoTune, the channel number, and Video and Audio frequencies change rapidly to find a channel:



Press OK to close the above dialog box. If all cable connections and channel settings are correct, the desired image will be shown in the preview area.

In the right area of the following screenshot, there is some data about the current job. These fields will show some statistical information during the capturing process:



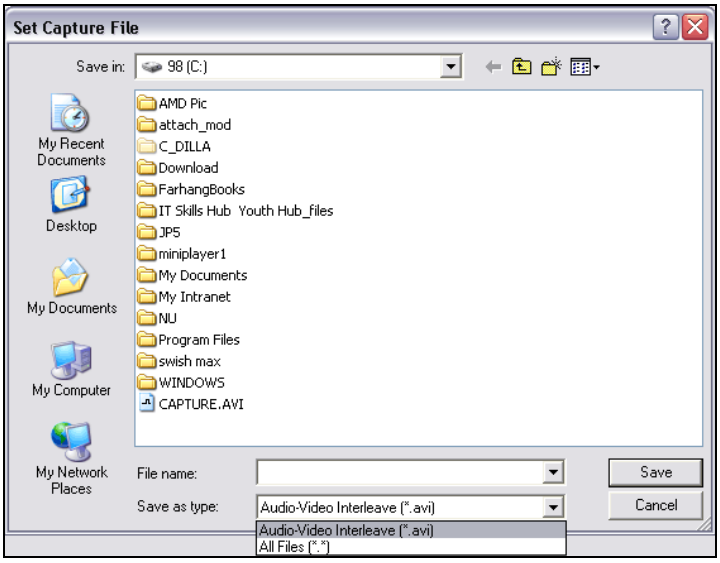
At the bottom of the above screenshot is the status bar, which shows capturing events and three buttons for video and audio settings. VirtualDub computes the required space for storing captured data per second, depending on the quality we set for video and audio, and shows the results in the next field.

Now that we have input signals from an analog resource, it's time to make other settings inside VirtualDub required to achieve digital data. These settings can be divided into four categories: **File settings**, **Audio settings**, **Video Settings**, and **Capture settings**.

File Settings

File settings in VirtualDub include defining the name and path, and allocating space for storing capturing data. All these settings can be done from the File menu:

By default, the first captured data is stored with the name of `capture.avi` in the `My Documents\Captured Files\Videos\` folder:



However, we can set our favorite name and path by using the File | Set capture file option (or pressing *F2*). We can define the desired settings in the window that appears, as illustrated above.

After defining the base file name, every time we capture analogue signals, a counter increments the number of the file name and prepares it for the next capture.

For example, suppose we set the file name `sample.avi` at the root directory of `c:\`. After the first capture if we try to capture data one more time, the number 1 will be added to the file name (i.e. `c:\capture1.avi`); for the next capture 2 will be added (i.e. `c:\capture2.avi`), and so on.

Regardless of what number we use for file names, in the next capture, that number will be automatically incremented.

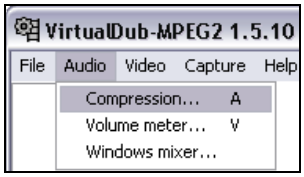
For example, if you set `44.avi` as the first file name, the next file will be captured with the name `45.avi`.

There are two options in the File menu to go back and forth in the existing and unused file IDs. These are File | Previous file ID and File | Next file ID.

If the file already exists, an alerting tag will be shown in the title bar of the capturing window:

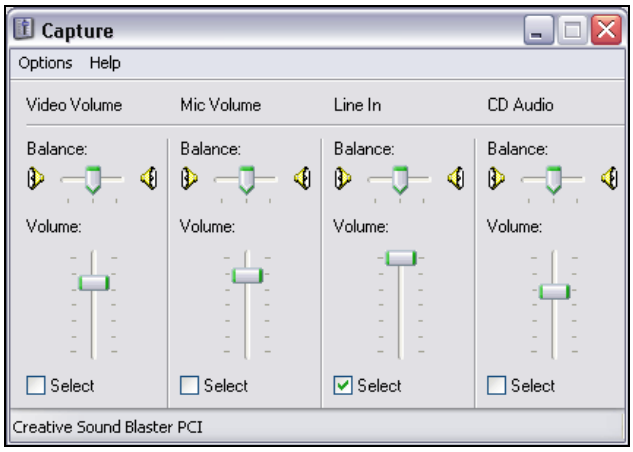
Audio Settings

There are three options in the Audio menu for receiving audio signals from the capture card, setting sound volume and channel balance, and the sound compression scheme:

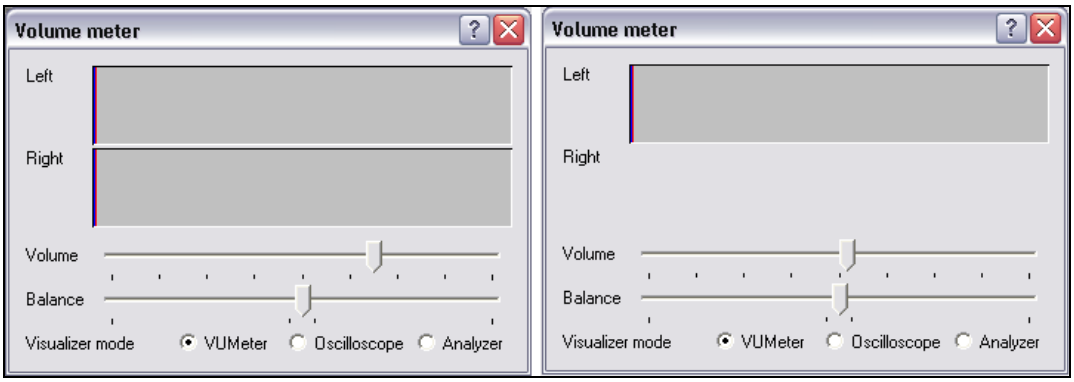


If the captured signals are without sound, then we must check out the internal Windows mixer to make sure Line in is selected as the audio source.

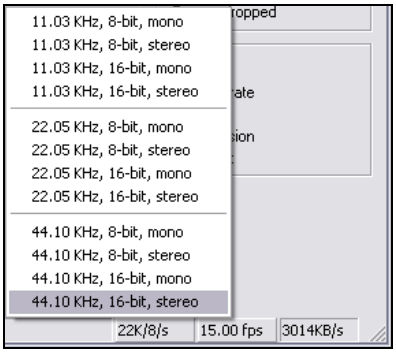
Selecting the Audio | Windows mixer option will show the mixer, in which the audio input source can be set:



In the Volume meter dialog box, we should see one of following images, depending upon whether we have stereo or mono settings. We can use the VUMeter, Oscilloscope, and Analyzer radio buttons for changing sound balance, volumes, and the graphical representation of audio signals:



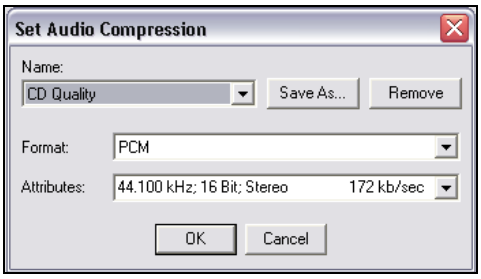
We can set stereo or mono sound at the bottom of the VirtualDub window. Just click on this button and hold the mouse button for a while. From the following list, we can choose our desired sound settings for capturing. For the example given above (capturing for DVD), we need to select 44.10 KHz, 16-bit, stereo, as shown in the following window:



In the context of multimedia, a huge amount of space is required to save uncompressed data. Therefore, we must use some compression algorithms to reduce the file size. In VirtualDub, all audio characteristics, including format and capturing quality, can be found under the Audio | Compression option. Please note that in this dialog box the format is different from the codec that will be used later for saving the captured data.

If you do not want to be concerned with audio details, and just want a good audio scheme, select CD Quality from the Name drop-down list.

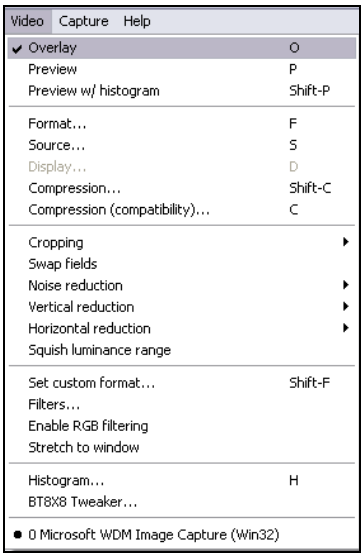
The Format section defines how to arrange audio signals in a file and the Attributes section sets the basic sound quality:



If we define custom settings in this dialog box, we can use the Save As button to save it with our desired name. For future usages, we can just select that setting from the Name drop-down list. We can delete that scheme from the list by pressing the Remove button.

Video Settings

The Video menu provides many more options than the Audio menu. Some of these options are used for visualization in VirtualDub capture mode, and some of them are used for video capture settings:

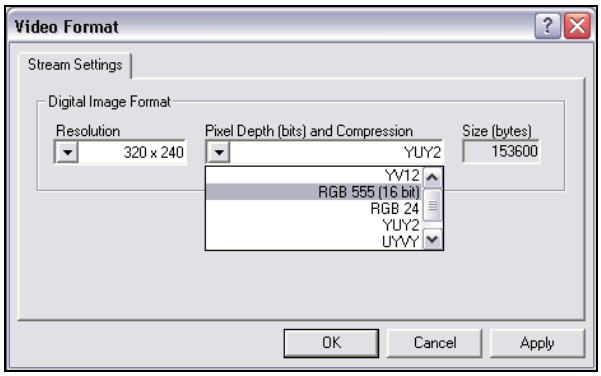


The first step in the video settings is defining the video format. Press *F* (or choose Video | Format) to activate the Video Format dialog box. There are two drop-down lists that we can use for setting the frame size and pixel format.

A box next to these two fields shows the size of each uncompressed frame, based on the selected options. For example, if we select 320x240 for Resolution and RGB 555 (16 bit) for Pixel Depth, then the size of each frame can be computed as:

320 (pixels) x 240 (pixels) x 2 (bytes or 16bit) = 153600

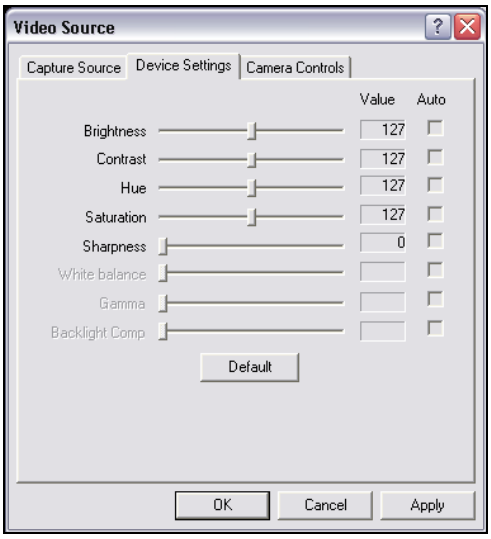
This field is designed to give an overall idea about the size of the captured video:



Select YUY2, since we have decided to capture data for a DVD. We can then choose the MPEG 4 codec later. After setting frame size and pixel format, press Apply, and then OK to close this window.

Look at the current captured image in the preview area. Some color and light corrections maybe necessary, depending upon what channel or input source signals we receive:

Press *S* one more time (Video | Source), and in the Video Source dialog box, go to the Device Settings tab. All sliders in this area are self-explanatory and the changes can be seen while tuning them in the preview area. Pressing the Default button will return all sliders to their original position:

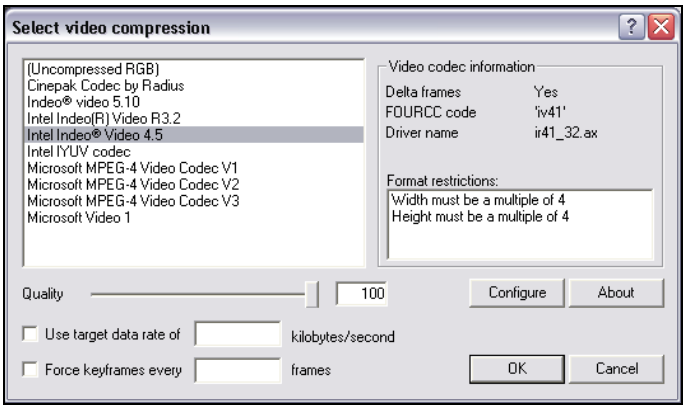


Press Apply, and OK after you achieve the desired settings:

Compressing Input Signals while Capturing

We can compress input signals while capturing. There are various options in the Video | Compression (*Shift*+*C*) window, based on what format we choose for video.

For example, if we select RGB for pixels format, the following compression codecs will be available:



Remember that it's possible to add some additional codecs to Windows. So the new codecs will be shown in the compression windows if the pixel format (color space) can support them.

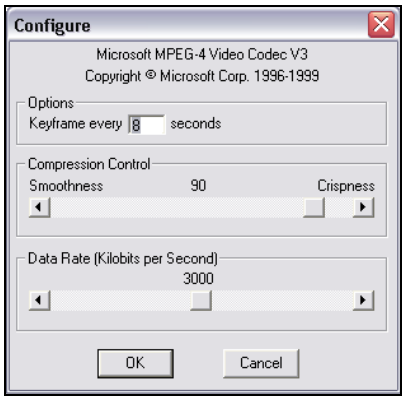
> Additional codecs like Huffyuv and MJPEG can be downloaded from the following sites:
>
> **Huffyuv**:
>
> `http://neuron2.net/www.math.berkeley.edu/benrg/huffyuv.html`
>
> `http://www.free-codecs.com/download/HuffYUV.htm`
>
> **MJPEG**:
>
> `http://www.morgan-multimedia.com/`
>
> `http://www.free-codecs.com/download/Motion_JPEG_Codec.htm`

Every codec in the window has some options that can be used for size and compression settings. There is one global rule for almost every codec setting: increasing the quality results in increasing the file size.

You may wonder why there are two Video | Compression options in VirtualDub: one in the main VirtualDub window (*Ctrl*+*P*), and one in the VirtualDub capture mode (*Shift*+*C*).

This is because the end codecs tend to be too slow to be used on slow computers in real-time. So we use fast codecs to capture data signals, and slow codecs (that can compress better than fast codecs) to do the final encoding.

For our DVD example, choose Microsoft MPEG-4 Video Codec V3 from the list. Then press the Configure button and set the sliders, as shown in the following screenshot:



Cropping Videos

We can crop the video during the capture process. Click on Video | Cropping | Enable, and then select Video | Cropping | Set bounds. In the following screenshot, X1 and X2 set the left and right margins and Y1 and Y2 set the up and down margins respectively:

Some frames may drop because cropping during capture needs heavy processing. Therefore, it is strongly recommended that you use cropping after capturing video, or use small values for cropping.

Removing Unwanted Noises from Videos

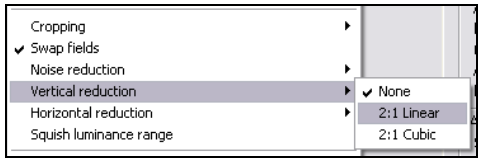
Another technique for video optimization is reducing the noise. This can be enabled by selecting Video | Noise reduction | Enabled:

We can set the noise threshold with the slider that appears after selecting Video | Noise reduction | Threshold.

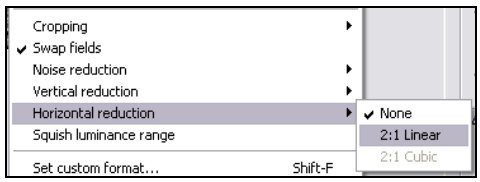
Reducing File Size

There are two options for horizontal and vertical reduction. This feature is useful for powerful capture cards capable of getting signals at high resolution. A good capture card is not enough, and we need a powerful CPU too, or we may lose some frames during the reduction process.

Vertical and horizontal reduction options can be found in the Video menu. For better quality use the 2:1 Cubic option:



For more compression, use the 2:1 Linear option:

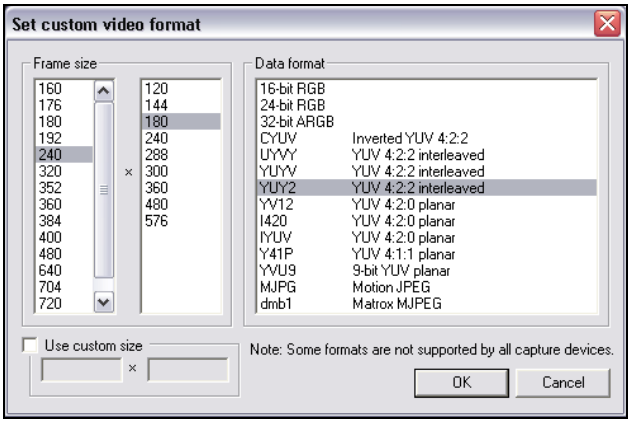


Unusual Resolutions and Formats

For authoring standard VCDs, SVCDs, and DVDs it's better to use standard PAL, or NTSC settings. However, in some situations, if you prefer to use different resolutions and pixel formats you can refer to Video | Set custom format and select the resolution and data format that you wish.

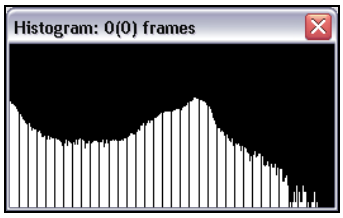
If your preferred resolution does not exist in the frame size group, check the Use custom size box, and insert the desired resolution manually. Not every resolution is allowed. For example, if you enter a resolution in which the video width is less than its height, the following error message box will appear:

When you decide to use custom format, it's preferable to use a low resolution and YUV family data format as illustrated below. The number of dropped frames is consequently reduced and the video will have reasonable quality:



Histogram

The **histogram** option provides a graphical representation of the colors in the current frame. With this ability, we can study the color structure of input signals during capture and make any necessary changes. Activate this feature through Video | Histogram:

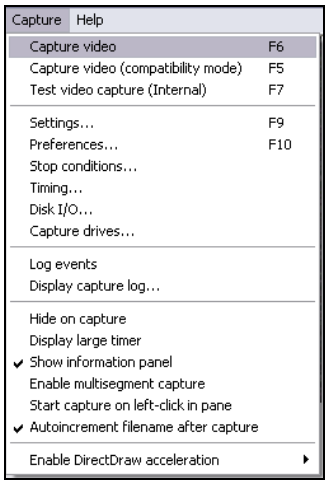


Use histogram only if your CPU is powerful enough. Otherwise, you may lose some frames during capture.

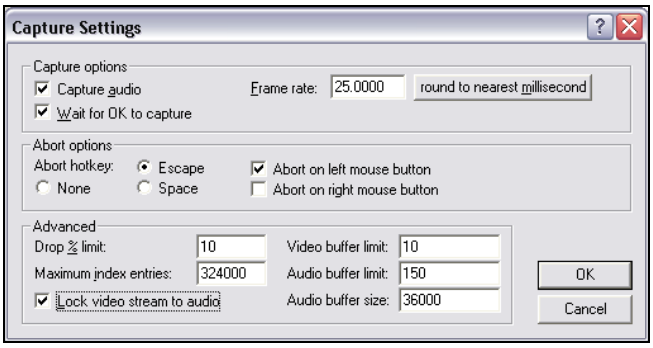
Capture Settings

All options we need for capturing (from configuration to starting the real job) lie in the Capture menu. Pressing *F6* (Capture | Capture video) or *F5* (compatibility mode) starts the actual video capturing.

Before doing this, however, we need to do some configuration in the Capture menu:

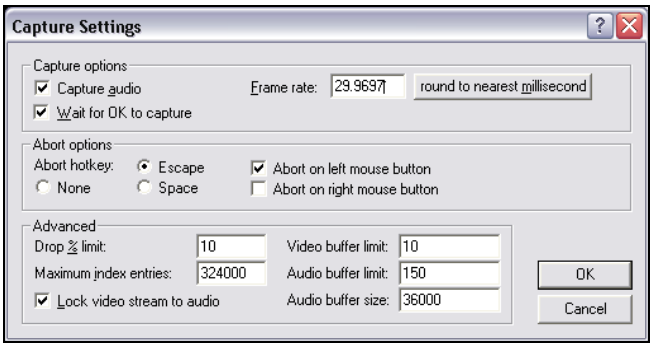


Press *F9* (or choose Capture | Settings) to activate the Capture Settings window. For standard videos (PAL and NTSC), we should set different values in this window. For example, in order to capture for the PAL system we should use the following settings:



There are four controls in the Capture options group. Selecting Capture audio saves audio signals too, and activating Wait for OK to capture will show a message box like the one above, after pressing *F5* or *F6*. For continuing capture, we press OK.

For NTSC, the suitable settings are:



The most important control in this group is Frame rate, which is used to set the video capturing frame rate. It is possible to define frame rates with up to four digits after decimal point. We used this feature for setting NTSC standard. NTSC has 29.97 frame rates, but because of an internal issue in VirtualDub, we must use 29.9697, for NTSC.

The last control in this group is a button that can be used for computing the frame rate, nearest to the current number.

In the Abort options group we can choose the keys and mouse buttons that can abort the capturing process. Selecting Escape and Space will abort capturing if one of these keys is pressed. We can also choose the left and right mouse buttons for this purpose.

The Advanced group includes some serious controls for capturing. The percentage of frames that are allowed to be dropped during capture is set in the Drop % limit field.

The Buffer

A **buffer** is a piece of memory that is used to temporarily hold data while waiting for the hard disk. The number of buffers is limited by available free disk space. Although it's possible to use up to 10 audio buffers, it's better to use a 10 video buffers and 4 audio buffers combination.

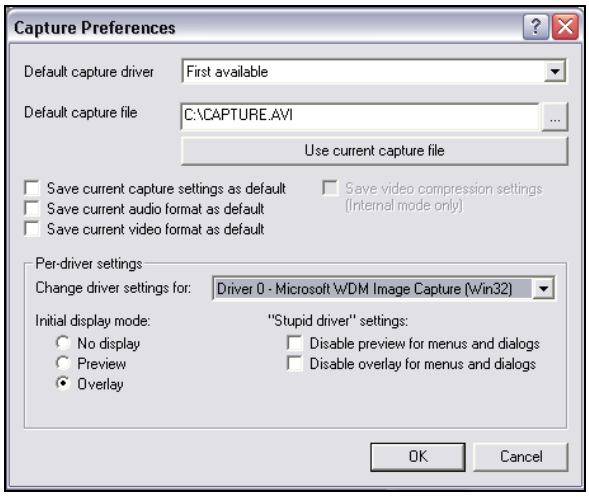
In the Audio buffer size field, we can set the required space for gathering audio signals in bytes. If you specify a small size for this field, CPU load will increase noticeably—and you will lose some frames, as a result. Setting a big value for this field will show lack of video synchronization.

Saving Current Configurations for Future Reference

Pressing *F10* (Capture | Preferences) will show the following window, in which we can set the capturing preferences. If you plan to use a special path and file name for capturing, then type it in the Default capture file field.

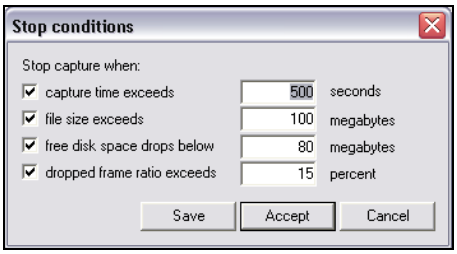
For saving current capture, audio, and video settings check the Save current capture, Save current audio, and Save current video boxes respectively.

Leave the other controls unchanged because we've set them in previous menus:



How can we tell VirtualDub to stop capturing, if unwanted situations occur?

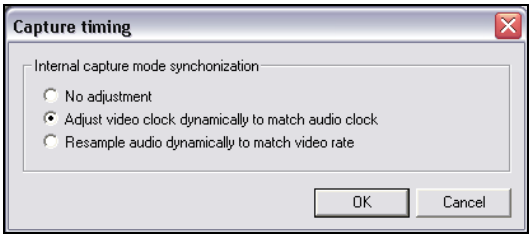
Capture | Stop conditions shows us a window for defining capturing stop conditions:



The Synchronization Problem

Sometimes in the capturing process, video, and audio don't match correctly. There are many reasons for this, including: unusual video frame rates, weak processors, and background activities during running VirtualDub.

With the Capture | Timing option we can solve this problem. We can resample audio during capture to match the video frame rate by selecting the last option of this window. By selecting this option, we can ignore dropped frames:



Chunks

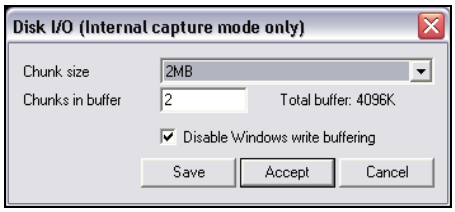
Chunks are synchronized video and audio packages that sit in the buffer before they are written on the disk. Through Capture | Disk I/O we can set the size and number of chunks in each buffer.

The size of each resultant buffer (based on chunk size and number settings) is a very important factor for suitable capturing. If we select buffers that are too small, we will get too much file I/O and data will be synchronized correctly, but the CPU load and HDD activities will increase, and VirtualDub may drop some frames. If we select buffers that are too large, the synchronization will be affected.

After testing various settings, try to balance these factors based on the hardware that you have, to get the optimized result.

In fact, we must use this section for matching VirtualDub with the speed of the computer. It's recommended that you use 1MB chunk size, and 14 chunks for disks with at least 2MB cache, and a 1GHz CPU.

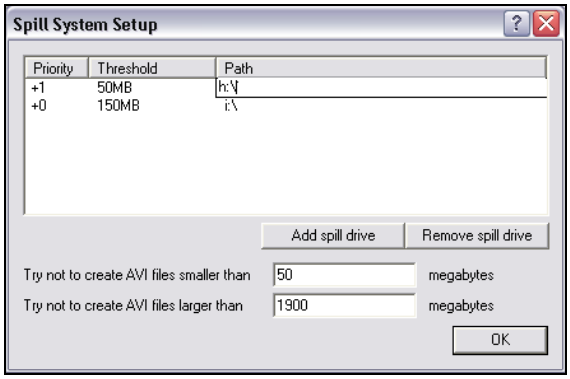
At the bottom of this window, we can see that the Disable Windows write buffering option has been checked. This prevents any disk management application (and Windows, itself) from interfering:



Capture Duration Restrictions

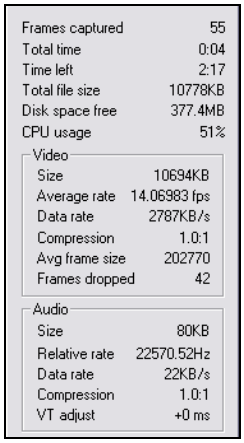
Is there any restriction for capturing duration? Yes, there is. In fact, the only restriction for file size is the disk space.In VirtualDub, we can capture AVI files with more than 2 GB! There is an ability called Spill System (Capture | Capture Drives), which can be used for dividing large files into several parts. VirtualDub will see these parts as one logical file in the next call.

In Spill System Setup we can specify the partitions of captured data:



Defining more than one spill drive is allowed. However, this is not the whole story. For activating spill drives that we've defined, we should enable the multi-segment capture feature. Just check the corresponding option: Video | Enable Multisegment capture.

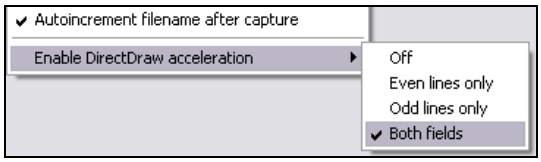
If you think every thing is alright, press *F6* to start the capturing process. The right pane of the window shows some statistical information. This area indicates how many frames have been captured, how much of CPU capacity used, how many frames dropped, etc:



If you wish to see a larger counter during capture, select Capture | Display Larger Timer.

Keep watching, and if you find that the CPU usage is more than 90%, you may notice that the value of dropped frames increases. Just review all your settings one more time and make any necessary changes (e.g. reduce capture resolution).

If you are using Win XP or 2K, activate the Capture | Enable DirectDraw Acceleration | Both fields option for better performance. Make sure that Video | Overlay is selected:



For Win9x and ME, deselecting this option will be useful for better performance. If your hardware is not powerful enough, deactivating Video | preview maybe helpful too.

Leave the computer alone during capture or at least don't ask it to do another heavy task!

To stop, just click the VirtualDub window, press the *ESCAPE* key, or use any other settings that you've made for stopping capture. The File | Exit capture mode takes us back to the VirtualDub main window.

Summary

We have been introduced to some capture preprocessing that is very important for creating optimized videos. We learned that VirtualDub configuration for capturing falls into four categories:

- File settings
- Audio Settings
- Video Settings
- Capture Settings

Moreover, we learned that before writing captured signals on disk they should be buffered for better disk I/O management and optimized data capturing. Also, with the help of some VirtualDub internal mechanisms called Spill System Setup and Multi Segment Capturing we can capture and save files with more than 2GB.