

Setting Up an Audio-Video Workstation

Introduction

This document is a technical memo describing procedures involved in setting up an operating system and hardware for an audio-video workstation. The memo is based on almost 3 years of hands-on experience supporting our own voice over IP (VoIP) applications that are part of our [Web conferencing](#) and [distance learning system](#).

The vendors of audio and video capture hardware want us to believe that adding videoconferencing capabilities to a PC is a trivial task: unpack and insert hardware, install software, start up your Internet video phone, and—Voila!—talk to your buddies.

In reality, setting up a working, robust workstation for voice and video over IP applications is quite often an arduous task. If the initial hardware and software installation does not go flawlessly, the problem may become too difficult to solve for many users. This, as well as several other technical difficulties, such as inherent latency of audio conversations, caused current stagnation of the VoIP application market. Let's face it: VoIP still needs to catch up with the robustness and ease of use of the standard telephone system.

The experience we gained during our support work allows us to say that in many cases, the end users are not able to set up VoIP applications on their own, and almost always some sort of support or troubleshooting is necessary. The problems with basic setup were not related to our software at all. Mostly, the users either had technical problems with installation of the hardware and associated driver software, or were unable to proficiently use the audio-video conferencing software.

This is the situation most of the VoIP application vendors face. It is a no-win conundrum: no matter what, the manufacturer of the application software is held responsible for all problems. This note has been written to promote the only viable solution: educate the end users about technical aspects of the PC digital audio and video and provide them with a practical guide how to solve the most frequently encountered problems.

As indicated, the problems with implementing VoIP applications are twofold. The first, most basic problem is correct installation of audio & video hardware and drivers. The second problem is how to

use audio on a PC. This memo addresses these two problems in a sequence.

The memo provides a step-by-step system diagnostics procedure, which, if successfully completed, ensures that the user machine is properly set up, and that the user understands how to use the hardware and software involved in the process. We want to stress that if the basic problems with audio and video setup are not solved, our VoIP application, BuenaVista, may fail. We hope that this document will help users of our software to correct most of the problems. In the case of difficulties that the users cannot solve, we offer free [troubleshooting service](#).

Setting Up The Hardware

The first step in getting a PC to act as a videophone is to install necessary hardware. The most important component of such hardware are audio and video digitizers.

Every sound card has a built-in digitizer. Digitizer takes analog audio signal from a mike, TV, radio, VCR, etc., and, using the sampling process, converts it into digital form that a computer can handle. Computers cannot handle analog audio.

To serve as a source of digital audio, an analog audio device must be connected to a sound card. Sound cards typically have two types of audio inputs: a mike input, and a line input. These inputs differ in sensitivity: a mike input can handle a much lower electrical signal. If you connect your mike to the "line" input, your signal will be too weak to be useful. A radio receiver connected to the mike input will overload it and produce clipped, terribly distorted audio.

Make sure you connect devices to correct inputs.

Tip!

If you use wireless mikes, the output from the wireless receiver can fit either input. This varies from vendor to vendor, and the correct connection must be established by trail-and-error.

Video cards "capture" video signal from a camera. There are also "digital video cameras" on the market, connecting not to a dedicated card but to the USB port. Such cameras simply have a digitizer inside, and they don't need a card. They work, in general, in the same way as the cameras connecting to a PC card.

Video cameras can connect to the video capture cards using two popular connectors: an RCA jack, a.k.a. composite video, or an SVHS connector, a.k.a. hi-8 or "Y/C" video. SVHS video signal is always preferable, for two reasons:

1. The signal has higher resolution.

2. It is fed to the digitizer using independent signals for color and B/W component, so that the digitizer can do a better job in translating the analog signal into digital form.

The most difficult part in installation of an audio and a video capture card is drivers installation. Drivers are software modules enabling communication of the operating system and applications with hardware devices. Drivers are written according to a very detailed prescription provided by Microsoft, and, for each device, they must respond to a specific set of "calls" from the operating system. They also have to work with application software. The main idea is that any audio card must respond in some standard way to generic commands like "give me the audio stream from the microphone port." Wasn't it for such standard programmatic interface, applications would have to be programmed for each manufacturer's device separately.

Driver installation for new devices is traditionally one of the most fragile operations. Before "plug and play" was developed, the user had to carefully find out such parameters as interrupts and port numbers, to be used by devices in his/her computer, so that no two devices used the same system resource (which is industry techno-gibberish for the memory addresses and interrupt numbers). If there was a conflict, the machine could simply lock-up. The "plug and play" was supposed to solve this problem. The thing is, some people call this technology "plug-and-pray..."

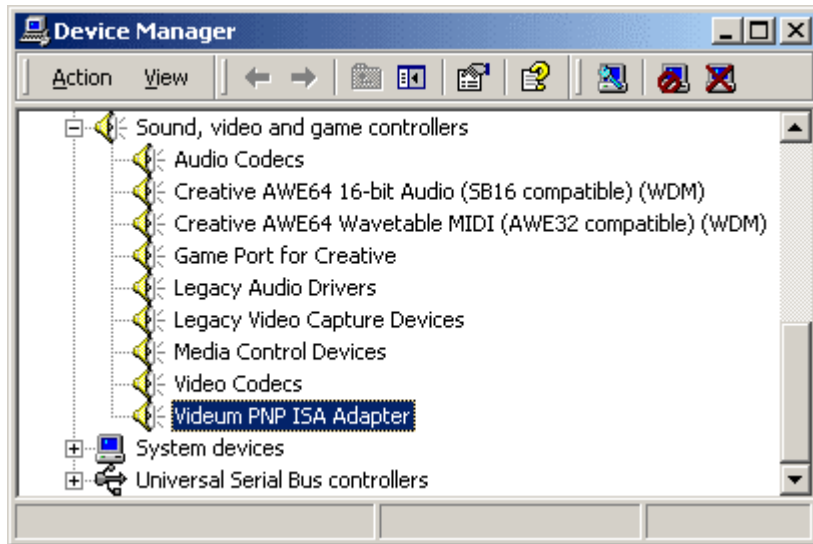
Device drivers for new devices can be installed in many different ways. Experienced users often assume they know how to install new drivers, and they ignore manufacturers guidelines. We strongly advice against it. Unless you are a real expert, and ready to debug your system by manually editing registry entries, you should always install drivers software exactly according the hardware manufacturer's instructions. Don't stray from the manual, or you may find yourself with a non-working system. If they ask you to reboot the system, please, do so. If they tell you not to use the *Add New Hardware* applet, don't. They know what they are doing.

Verifying Hardware and Software Installation for Audio

These instructions are provided for MS Windows 2000. The procedure is very similar for MS Windows NT 4.0.

Suppose you have installed new hardware and its device drivers. How do you check if it works? The very first thing to do is to check what the operating system thinks about it. For both audio and video capture card you must see if the devices are recognized by the system. Right-click on My Computer icon and select

"Properties," then select "Hardware" tab and click on "Device Manager" button. You should get a window similar to this one:



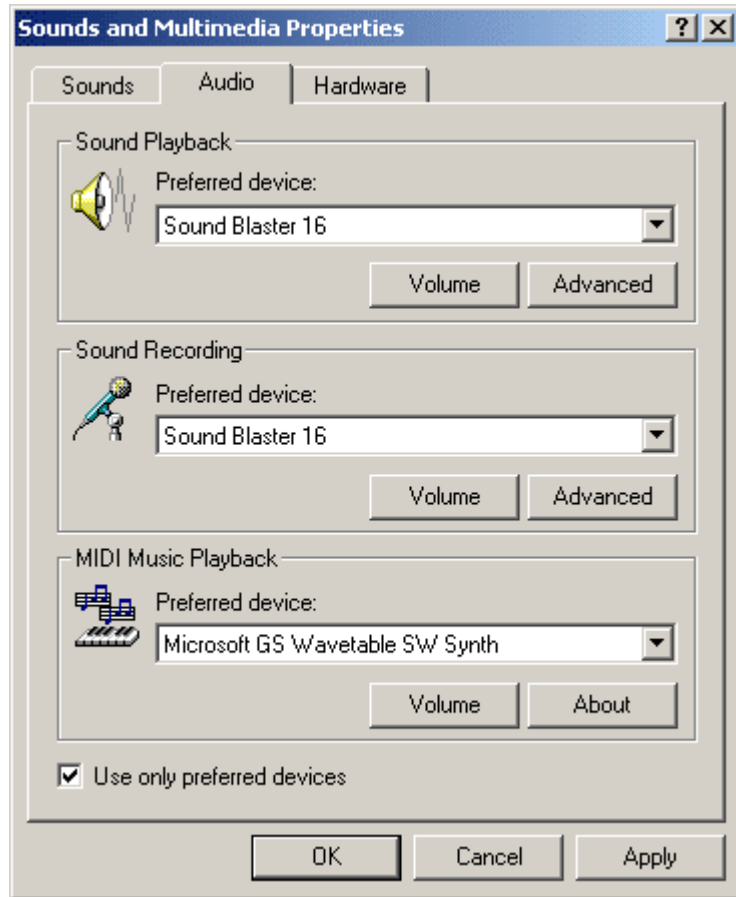
In MS Windows NT 4.0 or 98 you have to start "Multimedia" applet from Control Panel.

Expand "Sound, video and game controllers" node ("Audio Devices" and "Video Capture Devices" nodes in case of MS Windows NT or 98). If your drivers have been installed, you will see your audio and video capture devices listed, as in the example. If your devices are not there, or if they are listed twice, the driver installation procedure must have failed and you have to start it all over again.

Tip! *Drivers listed twice usually appear courtesy of "Plug and Play." If this happens, remove both instances and repeat installation procedure.*

If the devices are listed, it is a good sign. It doesn't mean that they are actually working, though!

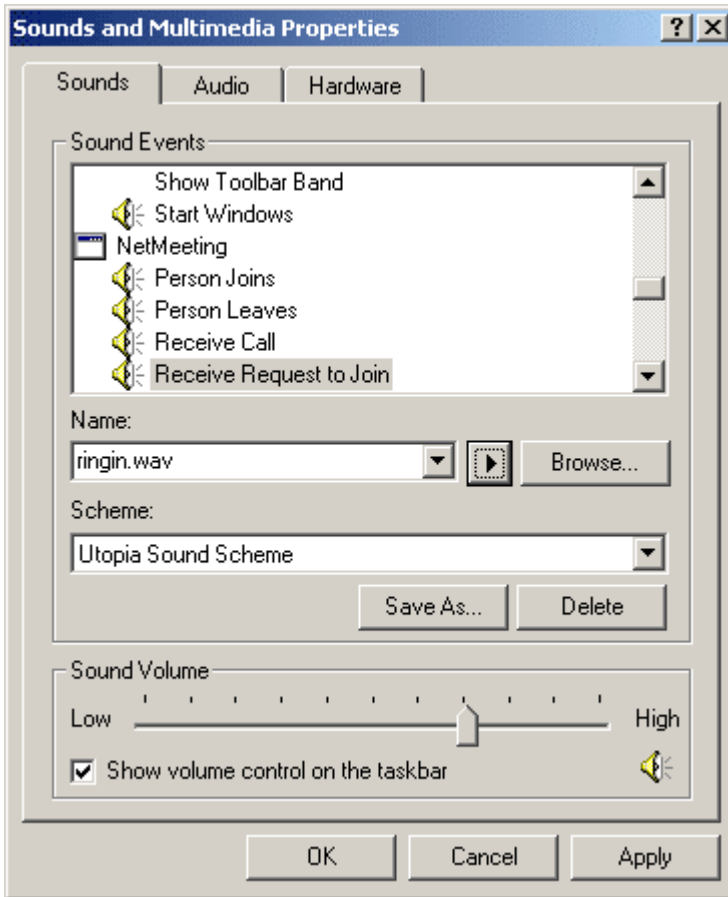
Go to Control Panel, and start "Sounds and Multimedia" applet (or switch to the Audio tab, in case of MS Windows NT or 98). You should see a window similar to the one on the right. The applet lists separately the playback and the recording devices.



If the dropdown lists have no entry for your device, your drivers did not install correctly. Remove the device as described above and repeat the installation procedure. If the audio devices are not listed properly in this tab, you can be certain that none of the VoIP applications will work.

You may have multiple audio devices in the system. If this is the case, you must select the device to which your mike and headphones are connected **before** you start the videoconferencing application. Switching audio devices when application is running already has no effect.

Proceed to the next step only if both devices are properly listed.



In the next step, you must test your audio device for proper operation. The first test is to play some audio snippets. The easiest way is to switch to the "Sounds" tab (or "Sounds" applet in MS Windows NT or 98). The applet is shown on the left.

Select any event from the list with the speaker icon next to it, and click the "play" button. You should hear a sound corresponding to the selected event. If you don't, check the playback volume (more below). Still no sound, check wiring. Wiring OK, your audio card does not work. Reinstall, call vendor's support, buy another card—all these things might help.

Again, if you cannot go through this check, there is no point in trying audioconferencing. Your system is not ready yet.

If audio is playing, it is time to test the mike.

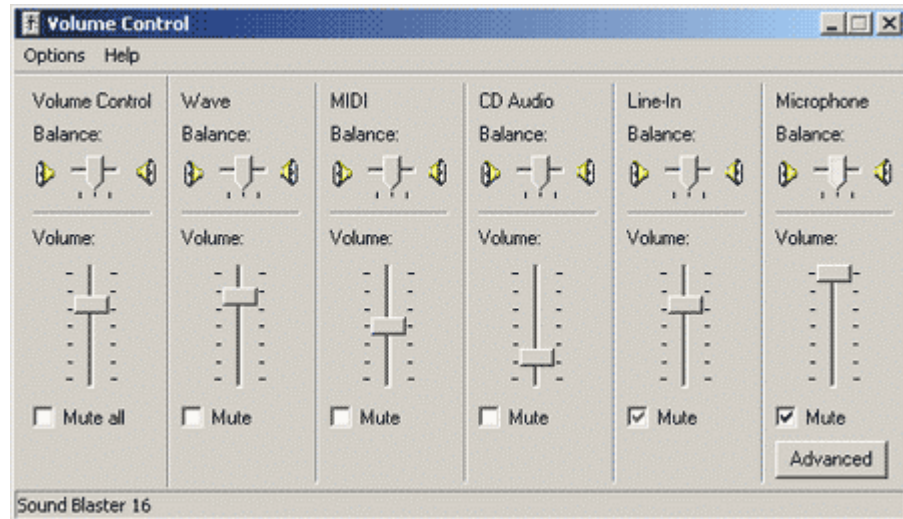
From the Start Menu, choose Programs, Accessories, Entertainment, and start Sound Recorder. This tiny applet is a front end to rather powerful sound recording and format conversion capabilities.



Press the Record button (rightmost one) and speak into the microphone. You should see the green line showing moving pattern similar to the one in picture.

Save the recorded audio in a file. Then, open the file and play it back using Sound Recorder. Adjust volume if necessary.

If the green line does not show anything, or if the recorded file just contains silence, it is time to take a look at one of the most confusing applications in the system: the Audio Mixer.

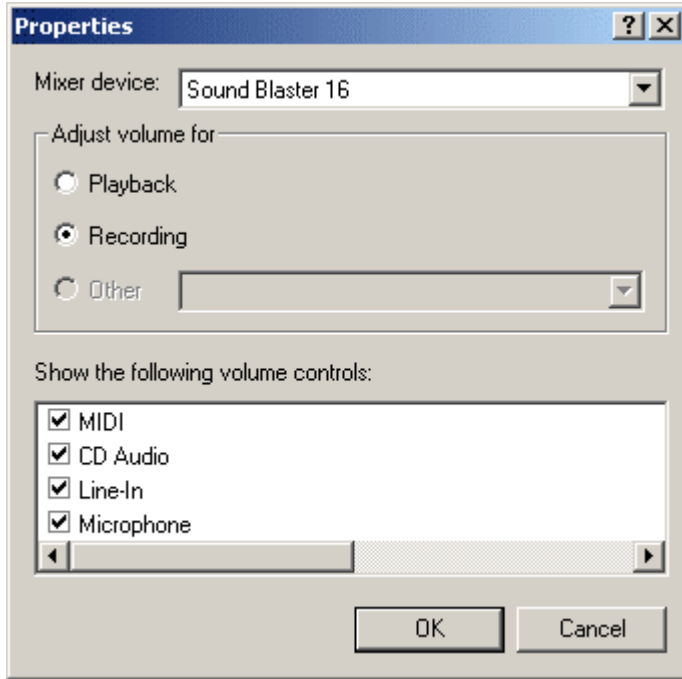


The Mixer is easily accessible from the system tray. If you installed the audio card correctly, you should have a small speaker icon in the tray. Double-click it, and you will see an interface like the one above. It is the playback volume control of the mixer.

This interface controls which audio sources will be played back by your card. In the picture above, the active sources are: Wave output (the audio is produced programmatically by an audio application), MIDI output (same thing except the file being played is synthesized), and CD Audio output (meaning, you can put an audio CD into your system CD-ROM drive, and listen to the music). Line-In and Microphone are muted. This means that there is no direct connection between the mike input of the card and it's speaker output. If you un-muted the mike, you would be able to hear yourself talking in the speakers or headphones. Please, note that the mike being muted here has nothing to do with being able to use the mike to talk to other people via a VoIP application! The mute checkbox **only** controls the internal routing of the signal in the audio card! Now, this would be fine, except that it does not apply to the volume control above the mute checkbox. Position of this slider **will** affect mike volume.

We have invariably found that the Audio Mixer confuses the users to the extreme. Why? Let's see. Click on the Options menu. You

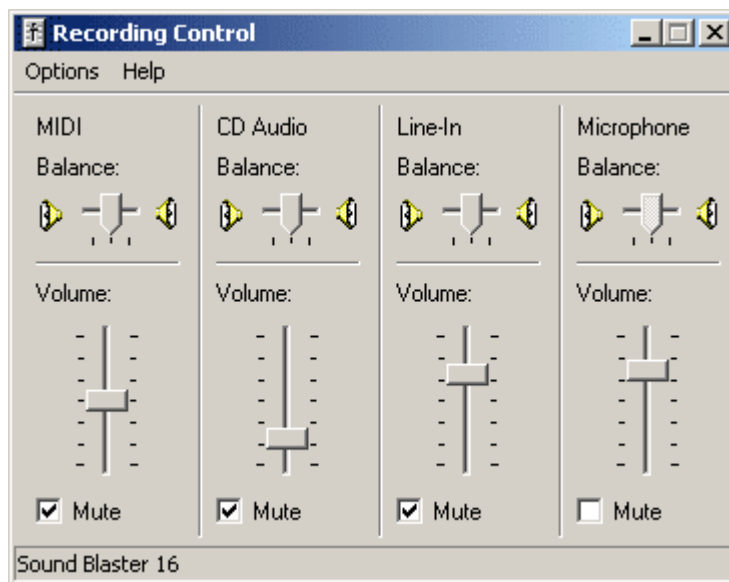
will get the window as below:



This window lets you open another "face" of the Audio Mixer: the Recording Control. It also lets you select controls for various inputs and outputs to be included in the interface. The list of check boxes is device dependent and can look differently for various audio device drivers.

The interface also lets you select the audio device being affected by the mixer. This is only important if you have more than one audio device in the system.

Clicking the "Recording" radio button opens the Recording Control of the Audio Mixer, as shown below:



From this interface you are able to control the audio sources which send audio streams to applications running on the system.

In case of MS Windows NT or 98 you won't see "Mute" checkboxes, but "Select" checkboxes instead. Only the audio sources for which the "Select" box is checked will be sending sound to the sound card.

Microphone slider affects mike sensitivity.

To explain function of Record Control better, suppose that the user wants to use a video conferencing software to play audio coming from an audio cassette player, not from the mike. The user would have to connect the cassette player output to the "Line-in" input of the audio card, and un-mute the Line-In control (or select it in MS Windows NT or 98). From this moment on, digitized audio stream from the cassette player will be available to all audio processing applications in the digital audio buffer of the sound card. Similarly, one could broadcast music by putting an audio CD in the CD player and selecting the CD Audio control.

If the audio playback and audio recording applications you have tested in two previous steps fail to produce expected audio, the first thing to check is the interface of the Audio Mixer. Make sure the correct devices are selected or UN-muted, that volume levels are in proper range, and repeat the tests.

For certain devices there may be more than one mike control. For instance, the Winnov audio-video capture card supports both camera mike and standalone mike. Always use the standalone mike in such situations!

Now you should be able to conclude the audio tests. If they work, you are all set for audio conferencing. If they don't, your system has an audio problem you will have to solve before you can use VoIP.

Verifying Hardware and Software Installation for Video

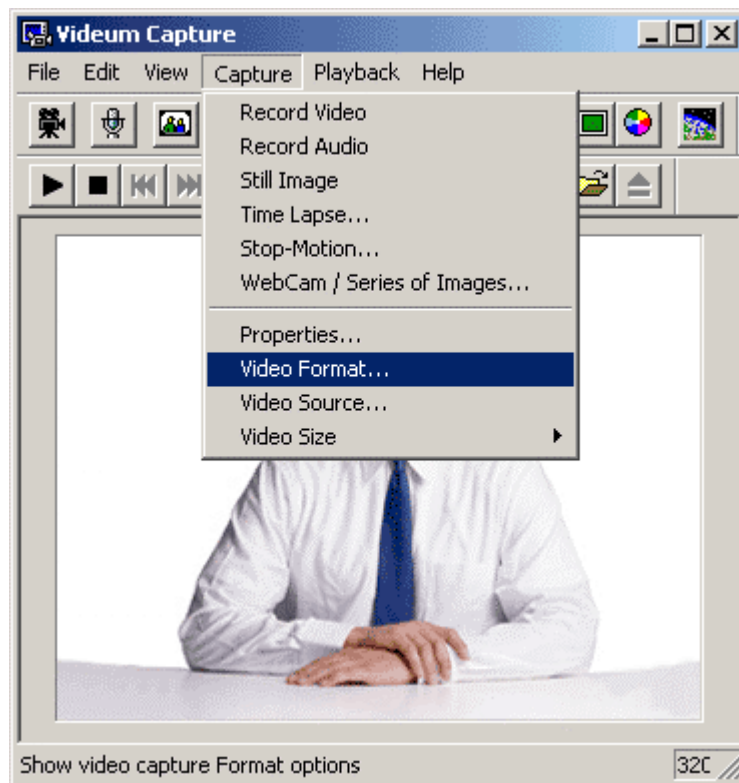
Video capture cards present a different set of problems. Conceptually, digital video is far simpler than digital audio. However, video capture driver problems tend to produce truly catastrophic system failures, including lock-ups and "BSOD" (Blue Screen O'Death) system crashes, even under relatively stable Windows NT system. This is because video capture drivers often work with video screen drivers, and by doing so, they "mess around" with most critical system components. Hence, using video

capture cards has a non-negligible negative impact on system stability, at least at the present "state of the art."

This situation is compounded by general deficiencies if the conceptual design of universal interface to both audio and video capture drivers. If the application using these drivers fails because of one subsystem's malfunction, the other system may be left in unpredictable state. Since the general programmatic interface does not support the "deep reset" functionality for either subsystem, quite often a malfunction of the application results in inability to restart it and resume operation. The only solution is system reboot, a dreaded operation, entirely unacceptable in situations such as lecture reception in TI Virtual Classroom. Hence, correct installation and robustness of video capture drivers is very, very important.

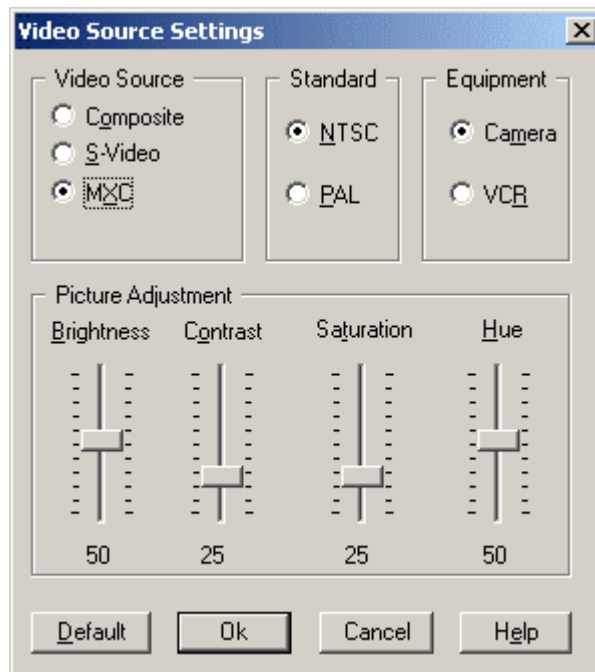
After you installed the drivers and verified that they appear in the Sounds and Multimedia applet (see above), it is time to test the video. Almost all video capture cards come with an application allowing for capture of video from the camera, digitization, conversion, and storage in a local file. This application must be used to verify correct driver installation before any attempt is made to use videoconferencing software.

We will show an example of such operation using a rather popular Video Capture utility from Winnov.



The picture above shows an output from a working MXC camera from Winnov, connected to the Winnov Videum. The menu show options that provide access to the settings of the video capture driver. It is important to understand that one can set up driver parameters from *any* application. The parameters are stored in the driver and won't change until they are modified by this or another application, or until system restart. Hence, it is possible (and, actually, recommended) that all driver parameters are set using test applications, before VoIP application is started for the first time.

The critical parameters of the video capture driver are usually accessible from two dialogs: Video Format and Video Source. The dialogs you will see are part of the device driver and will be different for each device.

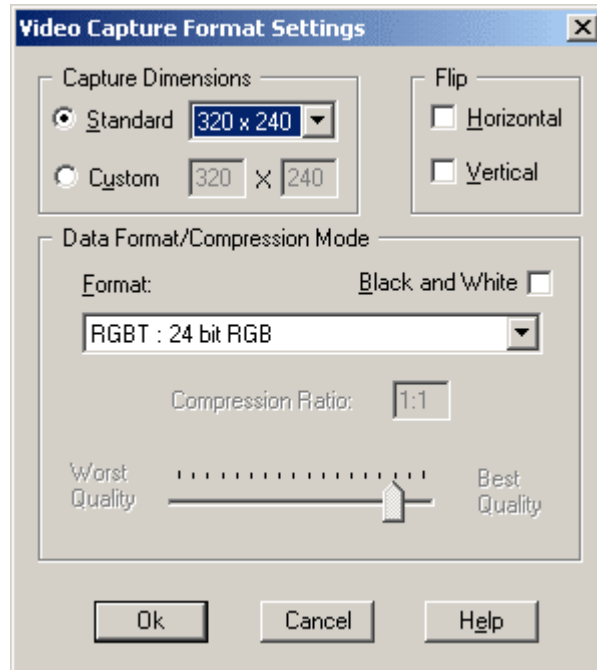


Video Source dialog allows the user to select which port is used to connect the camera to the video capture card, sometimes allowing the user to control . The picture to the right shows such a dialog for a popular Winnov Videum video capture board.

Many cards also allow selection of the TV color system (this is a property of the camera, and is usually NTSC for cameras sold in USA and PAL

for cameras sold in Europe), and adjustments for the image processing algorithm. In most cases, these controls should be left alone.

The Video Format dialog is used to select the format of the digitized video. Digital data will be available in the card's digital video output buffer. This format must be one of the formats understood by the application. BuenaVista understands and can properly handle the following formats : RGB24, RGB16, and YUV9. These are by far the most popular formats.



Some drivers display video format using euphemisms such as "millions of colors." Such descriptors are unclear and have little technical meaning. The settings can be tried, and they may or may not work.

Capture Dimensions define the size of the digitized video frame. BuenaVista always compresses QCIF-size video (176x144 pixels). The selected format can be equal to this value, or larger, but **never** smaller. Smaller format may cause BuenaVista to crash. Larger format will have a zoom effect: BuenaVista video window will show central part of the frame with object shown getting larger as the frame size increases (this is because the camera view is exactly the same no matter what the digitized video resolution is).

The video capture device setup must be absolutely correct before BuenaVista is tried. Non working or unstable video drivers can destabilize or crash the entire PC very easily. Make sure that your card is working flawlessly, or don't try to use video in BuenaVista.

Operator Errors

Even for correctly set drivers it is possible to crash a PC by abusive use of the interface.

Such a statement may raise reader's brows: computer abuse? Well, actually, yes...

Anybody who intensely uses a computer knows that in certain situations, under heavy load or when experiencing problems with disk or network access, a user that impatiently clicks on the screen just because s/he thinks that the system is too slow or hangs, can

actually cause system crash. This is because user's actions actually worsen the already critical condition in which the system tries to manage necessary resources using insufficient processing power. Such situation may cause various race conditions to surface and destabilize the system. This phenomenon is inherent to all current multi-process or multithreaded operating systems. An impatient user is often the straw that breaks camel's back. Clearly, impatient users are not the favorites of system administrators and software designers.

In the context of audio and video drivers, the "abusive use" consists of rapid requests to either switch from sending audio to receiving it in half-duplex cards, or to activate/deactivate video capture in video capture cards. The problem is that capture cards contain both digital and analog components, and that some cards have a very high latency (sometimes in seconds) for certain operations, including the two operations mentioned above. When subsequent on/off operations are requested while earlier requests have not been handled, the driver software is likely to show instabilities, otherwise not being easy to spot. Hence, a user wildly clicking at the video on/off toggle button can cause video driver to crash. As explained above, the crashed video capture driver can easily take the entire system down. It is recommended that the users treat the driver software with certain deference, since otherwise the driver may strike back...

Summary of the driver installation problems

Is BuenaVista going to fail when used with unsupported cards?

The answer is: we don't know. It isn't possible to test every single audio and video capture card currently on the market. We have seen the exotic cards to work perfectly with our software, and we have also seen how new release of the drivers can break a perfectly working system. The users are welcome to test and report either working or non-working hardware and software; we will compile these data and post them on the site. We are however reluctant to provide unpaid support for nonstandard system configurations, simply because of the cost of such efforts.

Advanced issues: mike and speakers volume thresholds

The graphics to the left shows the BuenaVista interface. The interface has three important elements: the speaker slider, the microphone slider, and the send/receive group of controls.



We have found out that the functionality of these controls is difficult to convey to the end users. Very often they assume that the sliders control the volume for mike and speakers, respectively. This is, unfortunately, not the case.

During a typical audio conference users aren't talking all the time. On average, 15 to 20% of the time there is silence—nobody is talking. This may sound surprising, but this number can actually be higher for multiparty conference calls, as users are waiting a moment to make sure nobody else wants to talk. Silence is a part of social protocol.

Internet conferences are not different. However, on packet networks, which we share with millions of other users, it is important not to clog the net by sending irrelevant information. Sending audio packets filled with silence, no matter how meaningful, wastes network resources.

For this reason BuenaVista implements very complex algorithm that

1. eliminates from the outgoing audio stream all packets in which volume level is lower than a certain threshold;
2. switches between sending and receiving based on the level of the outgoing and incoming signal levels.

These threshold levels are being controlled by the microphone and speaker sliders in the interface. The volume of different devices is controlled using standard Audio Mixer.

The position of the Microphone slider defines the threshold level for the audio signal, below which the signal is treated as silence and discarded before being sent to the network buffer. The signal level is integrated over a few milliseconds range to decrease switching frequency.

The Microphone slider can be used to adjust the threshold level to the mike sensitivity, and to the ambient noise level in the room. These parameters can change in a wide range and hence cannot be hardwired into the code.

Too low setting for the slider will cause ambient noise to be transmitted over the network. Too high setting will cause clipping effect: first phonemes of words or sentences may be lost. Setting up correct threshold is important for both audio quality and network performance. If the system experiences audio quality problems, a safe bet is to push the slider all the way left.

The Speaker slider defines a threshold for incoming audio to automatically switch from sending to receiving mode. By setting the slider all the way up, it is possible to mute all incoming audio. Setting it all the way down means the system will play back the entire incoming audio stream.

Speaker slider operation is only important for half-duplex audio cards and drivers. In the full duplex mode the system can concurrently send and receive audio.

The send/receive switch can work in either automatic or manual mode. Manual mode is essentially a "push to talk" mechanism, except that in the current implementation of BuenaVista the button does not "spring back"—it needs to be clicked again to toggle switch. In the automatic mode, the system switches between send and receive based on the slider threshold levels.

For full duplex drivers the toggle switch has no useful function. However, the sliders still control the threshold level for activation of transmission and reception of the audio signal.