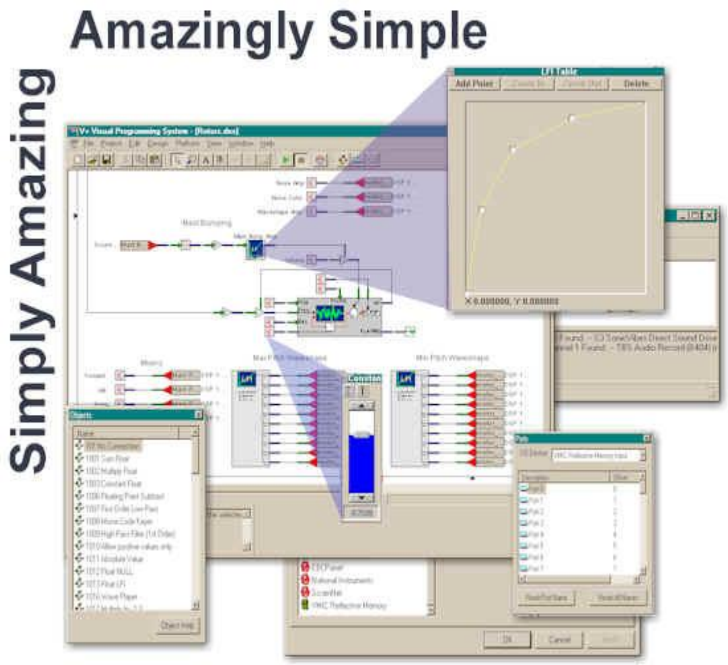
FXDirect Audio Systems User Manual



SimPhonics, Inc.

Printed: 10/25/2010
Filename: FXDUM1 - FXDirect Audio System User Manual, V1.2

REVISION HISTORY

Ensure you have the latest release of this document before relying on this information.

Version	Revision	Date
1.0	Initial Release	January 13, 2000
1.1	Updated manual references to VCOMM+.	May 19, 2003
1.2	Replaced Cool Edit with Adobe Audition	May 12, 2004

This document was authored using Microsoft Word 2000 and is maintained at the SimPhonics web site in .DOC format and HTML format. See the reference Section. This document may be copied freely for any purpose.

For your convenience this document may contain references to web addresses. If you are reading this document on a computer using Microsoft Word or compatible software, clicking these references will point your browser to a location on the web. If you are reading a hard copy, these references will appear as underlined blue text. See SimPhonics home page at www.simphonics.com for more information.

INTRODUCTION

PURPOSE

This document is intended to familiarize users with the FXDirect system, it's use within the V+ environment, methods and techniques for the generation of audio, and it's role within the SimPhonics product line.

BEFORE READING THIS MATERIAL

The reader must first be familiar with the V+ visual programming language. FXDirect is an extension to V+. A trial version of the V+ visual programming language as well as the V+ User's Manual is available for download from SimPhonics' website. See the references at the end of this document.

An understanding of the principles of digital audio is also a requirement. Persons using FXDirect obviously intend to use digital audio for some purpose; therefore, most users will have a working knowledge of digital audio principles.

Also, the reader should have a basic understanding of real-time systems and the simulation of synthetic audio environments.

ACRONYMS

API	Application Programming Interface
ATIS	Automated Terminal Information Service
CPU	Central Processing Unit
DIS	Distributed Interactive Simulation
DMA	Direct Memory Access
DSP	Digital Signal Processing
FAA	Federal Aviation Administration
IRQ	Interrupt Request
ISA	Industry Standard Architecture (bus format)
LFI	Linear Function Interpolation
MMX	Multi-Media Extensions
NSP	Native Signal Processing
PAR	Precision Approach Radar
PCI	Peripheral Component Interconnect (bus format)
SNR	Signal to Noise Ratio
SRC	Sample Rate Conversion

ABOUT SIMPHONICS

In addition to other products, SimPhonics specializes in products that generate artificial audio environments.   These environments are typically flight simulators, or other devices that require a simulation of a sound environment based on a model that is generated in real-time in response to user input.

The SimPhonics product line spans hardware, software and services that combine to form advanced systems. These highly integrated systems have been in use for over twelve years in the simulation and training industry. SimPhonics systems consist of commercial-off-the-shelf software, computer equipment, and peripherals, thereby minimizing the cost of initial acquisition and life cycle costs, while enhancing sound fidelity, reliability and maintainability. SimPhonics was first to offer simulation audio systems with the introduction of the FX series digital audio products, and the product line continues to grow with new and exciting additions. From DIS networked audio to sound modeling, our low-cost, off-the-shelf technology has been proven in the field in a variety of applications.

As of 1 January 2000, over 160 SimPhonics systems have been delivered to over 25 Foreign and US military and commercial customers. The rapid increase in popularity of SimPhonics products resulted in a current backlog of over 200 systems; mostly repeat orders from satisfied customers. These systems range from V+ Development Systems to full sound simulation and radio/interphone control, mixing and simulation, including networked audio for use in Distributed Interactive Simulation (DIS) applications.

In September of 1999, the SimPhonics sound system for Northwest Airlines Training Corporation's (NATCO) DC-9 received Federal Avionics Administration (FAA) Level D certification.

Our revolutionary new Visual Programming System, V+, dramatically simplifies the creation and integration of our systems and ease of use by our customers, thereby removing one of the largest cost factors in developing sound simulation. The highly successful V+ programming environment has also expanded the utility of SimPhonics software beyond our traditional bread-and-butter sound systems. V+ is capable of providing complete intelligent, reconfigurable Input/output processing, including sound, for simulation, training, or other industrial processes. Utilizing the latest 32-bit WindowsNT technology means wider support and lower life-cycle cost.

WHY USE FXDIRECT?

The need for modeled or dynamic audio is not always obvious. This is reinforced by the Hollywood motion picture industry, sometimes considered to be the forefront of audio special effects development. Motion picture audio development, or special effects, follows a predetermined path, and therefore the audio can be developed off-line and painstakingly recorded and synchronized with the video. This type of audio is statically generated from a recording. The only dynamic input to the generation of the audio is time.

However, there are systems that have many dynamic inputs whose values are not known until the instant the generation of the audio is required. The most notable of these are flight simulation and training systems, as well as interactive games. A flight simulator for example, must reproduce the complete audio environment of the aircraft in order to be realistic, including engine, aerodynamic, and other environmental sounds. This requires dynamically generating the audio in real-time.

That is what FXDirect is all about, generation of dynamic audio in real-time in response to dynamic input variables, using standard Windows WAV type hardware and software.

The most important an enabling feature is the V+ visual programming environment. Using V+ greatly simplifies the developmental effort, and requires no text-based software. If you can draw a picture and understand simple engineering technique and control algorithms, you can program FXDirect to do almost anything with audio.

DIGITAL AUDIO PRIMER

For those of you familiar with digital audio principles, you may skip this section, and go on the to FXDirect section.

BASIC DIGITAL SAMPLING THEORY

Before jumping into using FXDirect, let's take a quick look at the principles of digital audio. Remember that this section is only a very brief overview of the terms and basic digital audio principles and is not intended as a technical reference.

There are an infinite number of points along a genuine analog waveform, each point's amplitude having a value from zero to n, with n being the maximum amplitude of the waveform. When such a waveform (sound) is digitized via some digitalization or recording process, there are a finite amount of "points" along this wave, each point having a duration and amplitude. These points are called samples, since they represent a sample of the waveform at that point in time. The duration is determined by the sampling rate.

A typical sampling rate is 22,050 times per second. This means each sample represents 1/22050th of a second of sound. The amplitude at each sample (illustrated in green) is played for the entire 1/22050th of a second. As illustrated in Figure 1, the higher the sampling frequency the closer the sampled waveform approximates the original analog waveform.

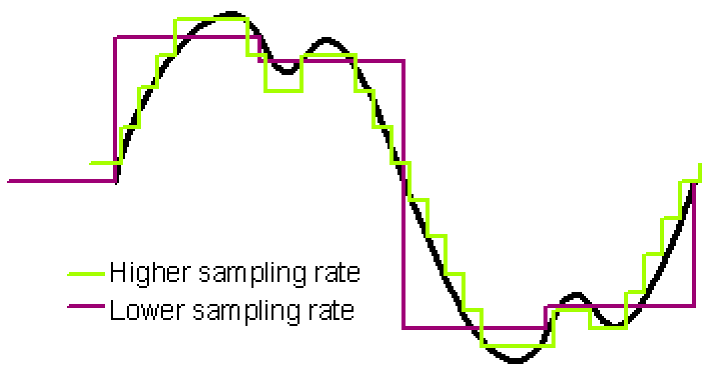


Figure 1

These samples, once collected into "chunks" or buffers, or as an entire waveform in a sound file, may be manipulated mathematically by V+ via FXDirect.

PROCESSING SAMPLES

In the past, SimPhonics produced audio using dedicated Digital Signal Processing (DSP) hardware and software, such as the SimPhonics FX-30 system. This system generated discrete audio samples from algorithms running on an embedded DSP ISA card.

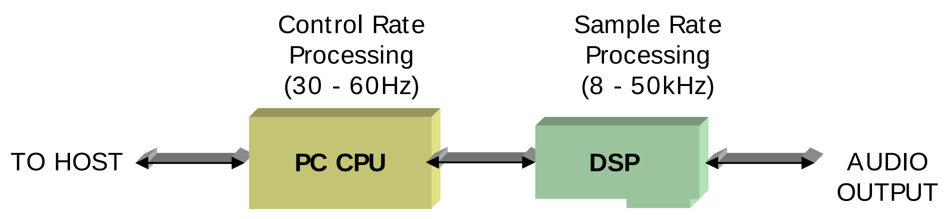


Figure 2

As illustrated in Figure 2, a typical FX-30 system consists of a number of ISA based cards connected together in a computer chassis.

The PC is used to generate control information from the host model that controls the DSP algorithms. The DSP in turn generates the actual audio samples. The external DSP was necessary since the PC did not have enough power to handle the large processing loads. Also, the ISA bus did not have the throughput capabilities to stream more than a few channels of audio. The FX-30 system manipulated individual samples. All the processing was performed once for each sample. The processor was interrupted at the sample rate, a sample was calculated, or manipulated, and the sample was sent on it's way for eventual output. The processor would then wait, or perform some background tasks while waiting for the next sample interrupt to arrive. This type of processing is called sample rate processing. This means that typical processor iteration rates are equal to the sample rate, or anywhere from 8000 to 50,000hz. While the FX-30 system is still in use today for mixing many inputs to many outputs, the current system of choice for the simulation of sound and moderate sound mixing is the FXDirect system. For more information about the FX-30 system, see http://www.simphonics.com/fx30/fx30.htm

Recently introduced by SimPhonics, the FXDirect system performs most of the signal processing functions using the processing power of the PC, rather than using a dedicated DSP card. This is now possible due to the increased speed and MMX instruction set of Pentium based PCs as well as a number of other technological advancements in PC sound handling capabilities. The following section discusses the basic operation of the FXDirect system.

FXDirect deals with arrays, or buffers of samples. Instead of the processor manipulating individual samples, a large number of samples are processed at one time. We can get away with this as long as the buffer length does not exceed about 1/30th of the sample rate. This is known as control rate processing. If the control rate drops below 30Hz, there can be noticeable abnormalities in the audio output.

STREAMING VERSUS STATIC BUFFERS

Using FXDirect, audio can be sent to the output device as a stream of audio samples, or as a static buffer. FXDirect uses streaming for audio that is not known in advance, such as text-to-speech audio and networked audio inputs such as Distributed Interactive Simulation (DIS) voice audio. Static buffers are used for FXDirect objects. The entire file is loaded into the secondary buffer during initialization before execution begins.

Note: When using static sounds, care must be taken to ensure that enough memory is available for all of the static buffers. The system may continue to load the files, even though there is not enough memory.

INSIDE FXDIRECT

FXDirect is a product name for SimPhonics V+ objects that use a PC to generate and process audio samples. These objects work directly with Microsoft's DirectSound, and are visual objects within the V+ visual programming environment.

Before DirectX®, sound on a PC was generally limited to playing discrete sounds one at a time. Also, multiple applications could not share a sound channel and play their audio while another application was using the sound card.

Thanks to Intel's NSP (Native Signal Processing) initiative, MMX instructions became available on the PC's Pentium processor. Microsoft then developed an object oriented API that uses these MMX instructions for the generation of audio and their software and is now called DirectX®, of which DirectSound® is a subset.

DirectSound®, allows V+ to mix multiple streams of audio, called Secondary Buffers, coming from WAV files, Text-to-speech processes, DIS voice connections, or any other audio source stream into an output channel called a Primary Buffer. Each sound card channel is directly associated with a Primary Buffer. V+ refers to Primary Buffers as "***channels***" internally.

Figure 3 illustrates this process. Source streams of audio from multiple WAV files, a TEXT-TO-SPEECH audio stream and a DIS voice stream are shown being mixed by DirectSound into the output channel on a sound card.

Note: The TEXT-TO-SPEECH capability is offered by SimPhonics as a separate extension to the V+ Programming environment. It allows synthesis of text strings passed by the host computer. This capability is useful in many simulation applications such as the Automated Terminal Information Service (ATIS) broadcasting local airfield and weather conditions, or the generation of a controller's Precision Approach Radar (PAR instructions to the pilot. A separate TEXT-TO-SPEECH User's Manual is available at SimPhonics' website.

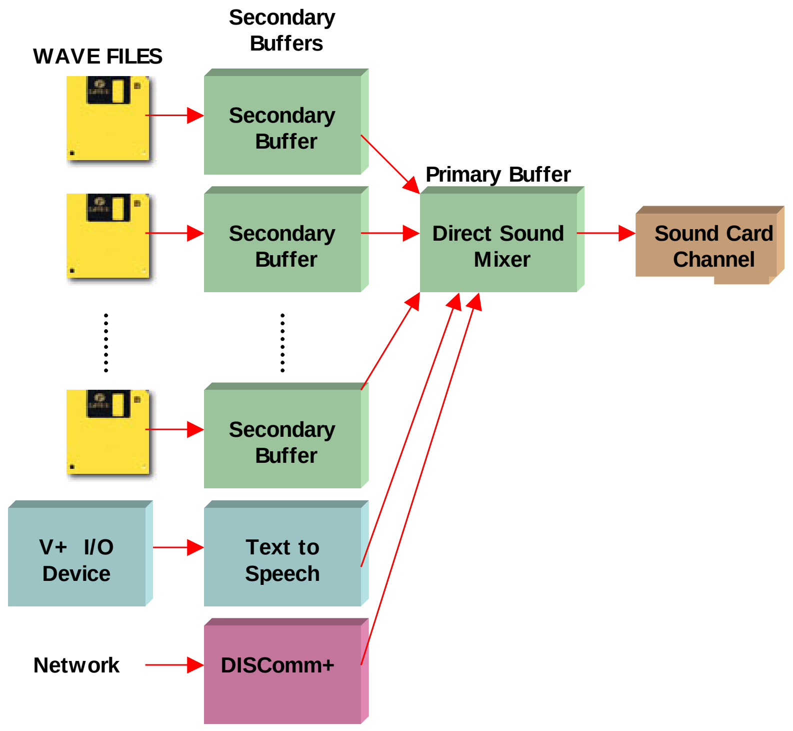


Figure 3: Typical V+ FXDirect Design

SAMPLE RATE CONVERSION

One of the most important functions DirectSound® provides in addition to mixing is sample rate conversion (SRC). SRC is necessary since audio input streams may be different sample rates, and they must be converted to the same sample rate before they can be presented to the primary buffer. In the example shown above, a DIS network stream could be coming in at 8kHz, while a sound is playing to that same channel at 44.1kHz. DirectSound® performs all of the sample rate conversion for us.

USING FXDIRECT WITH V+

Within the FXDirect visual environment of V+, secondary buffers are associated with objects on the worksheet, known as FXDirect objects as shown in Figure 4. There are many different variations of the FXDirect object within V+, but all are associated with a secondary buffer, and include a filename for the user to select a WAV file for use with the object.

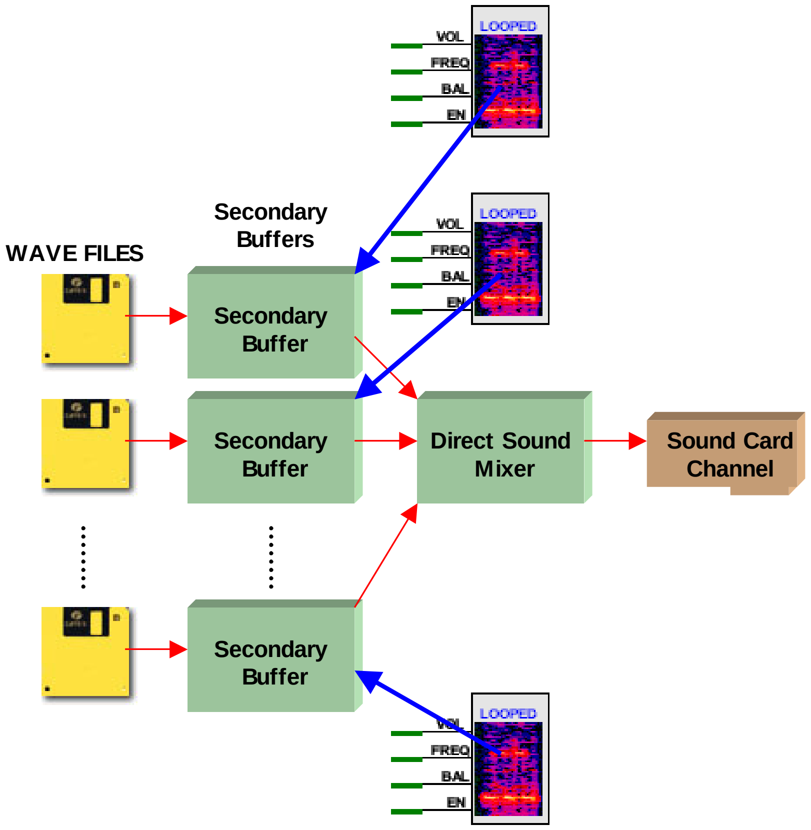


Figure 4: Mixing multiple Sounds in Primary Buffer

A PRACTICAL EXAMPLE

This is probably a good time to put all that theory into practice to reinforce understanding of FXDirect. Using the basic setup shown in Figure 4, let's see how that would apply to the aural simulation of a simple turboprop engine.

Note: The V+ turboprop engine worksheet and its two associated sample wave files are available at the SimPhonics Website http://www.simphonics.com/FXDirect/designs/turboprop/ for download. Once you have visited the site, be sure you have the following files required for this exercise:

Apu.wav	The WAV file for the turbine
Prop.wav	The WAV file for the propeller
Turboprop.des	The V+ design worksheet for the Turboprop Engine

In the real world, we would first locate or accomplish a recording of the turboprop engine to be modeled at various power settings. We should also determine at some point what parameters will determine the characteristics, frequency, and amplitude of the composite sound, and this information would normally be documented in an Interface Control Document (ICD). We should then do a spectrum-analysis of the composite sounds, which will determine the primary components and allow us to isolate them. Once those components have been identified, software tools can filter out components selectively so that we can make individual Wave recordings of the various component sounds. In this example, we will assume the composite sound made of a turboprop engine consists of the turbine and the propeller sounds.

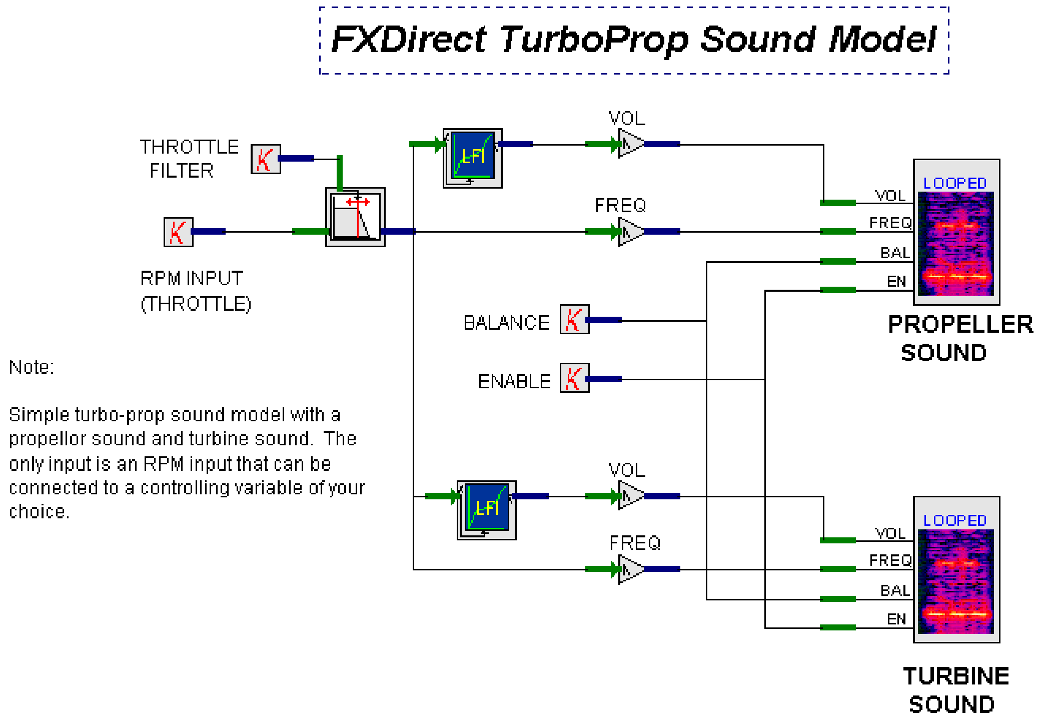
If it is not possible to do an actual recording, then we could try to synthesize the sound components using the various software tools (such as Adobe Audition) available. For the sake of simplicity, we will also assume that the sound of a turboprop engine is composed only of the turbine and propeller sounds. That is quite an oversimplification, since there are undoubtedly other components of the sound such as intake and exhaust roar and propeller blade loading which are distinguishable by the cockpit crew. But the purpose here is to provide an illustration of the FXDirect concept, not to produce a high fidelity turboprop sound.

Armed with the latest version of V+ and the FXDirect extension to V+, we next need to develop a design worksheet, which describes the turboprop engine sound. As a minimum we will define:

- **Input ports** - these are the simulation variables to which the sound reacts
- **A process** - a combination of V+ and FXDirect objects wired into a system, which represents the generation and control of the sounds to be generated.
- **Output Ports -** usually the available channels (speakers, headsets etc) into which the various sounds are mixed and output.

The V+ Development system allows us to drag all the above components from an object list box onto the design worksheet. The trick is to know which objects you will need, but that skill will come rapidly with the on-line context-sensitive help feature and some experience. As the primary workhorses of this turboprop sound simulation, we will use two looped wave players (Object # 1067), one for each sound component. We also need some filters and constants to control the sounds during testing.

The resultant design would look somewhat, but not necessary exactly like Figure 5, which is an actual Turboprop.des worksheet you downloaded. To avoid complicating the worksheet, an assumption was made that the only input parameter determining the sound is the throttle setting, which is an obvious oversimplification. A filter is added downstream of the throttle to account for the gradual (not instantaneous) reaction of the engine to throttle setting.


FXDirect TurboProp Sound Model
THROTTLE FILTER
RPM INPUT (THROTTLE)
LFI
VOL
FREQ
BALANCE
ENABLE
LOOPED
VOL
FREQ
BAL
EN
PROPELLER SOUND
TURBINE SOUND
Note:
Simple turbo-prop sound model with a propellor sound and turbine sound. The only input is an RPM input that can be connected to a controlling variable of your choice.

V+ Design of Simple Turboprop Engine

Note: All objects on the worksheet have "pins", which represent the inputs to, and outputs from that object. The looped wave player objects have a built-in output, that of one channel of audio. The inputs and outputs of these objects are then wired together into a visual block diagram, which incidentally, also forms the executable design. Simply amazing!

Since the relative amplitude (volume) of the two sounds elements is not necessarily linear over the entire throttle range from 0 to 100%, we will add a Linear Function Interpolation (LFI) table for each sound which will look up the correct amplitude of the sound element (Y axis) as a function of the input (Throttle, X-Axis). The two LFI tables can be edited in real time by dragging the various points in the table to a new location (see Table example the follows). New points can be added if necessary.

The two looped wave players have four "pins" each, consisting of Frequency, Amplitude,

Balance and Enable. The amplitude and frequency are the prime operators of FXDirect on the sample wave file, which result in generation of the characteristic sound component at correct pitch and volume, which is then streamed into the primary buffer. The enable pin simply enables or disables the wave player, which in this case would typically be related to a simulator freeze parameter (which would be an input port in a functional simulation turboprop model). If the simulator would enter the simulation freeze state, the sounds would be disabled. For the sake of simplicity in our demonstration, we set enable to a value of 1 (on).

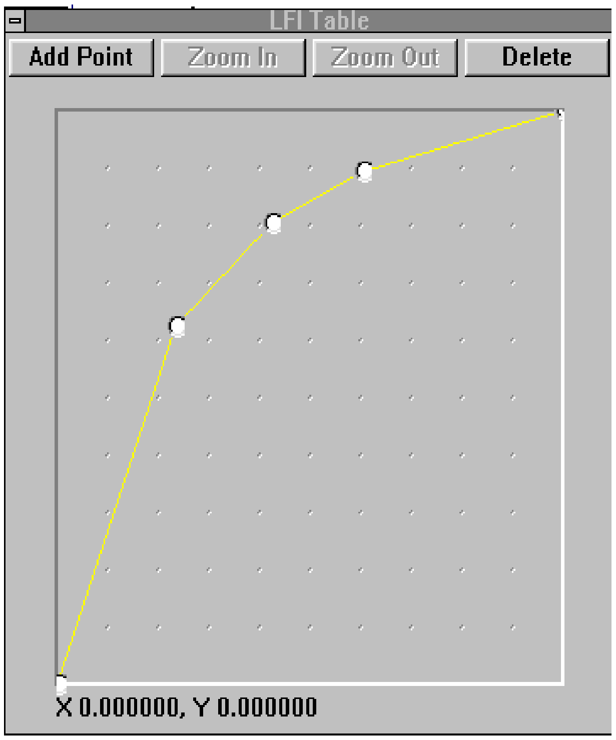


Figure 5 - LFI Table

Finally, the Balance pin relates to the characteristic of a PC sound card to produce a stereo output, and FXDirect in turn allows us to assign the output to one of two channels or speakers. For demonstration purposes, we have set the limits on the balance parameter from +1 to –1, allowing us to distribute the sound to the right speaker only (+1), equally balanced over left and right speakers (0), to left speaker only (-1). There is more on this later in this manual.

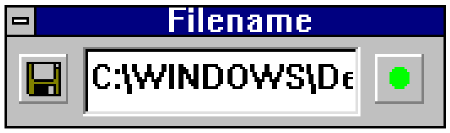
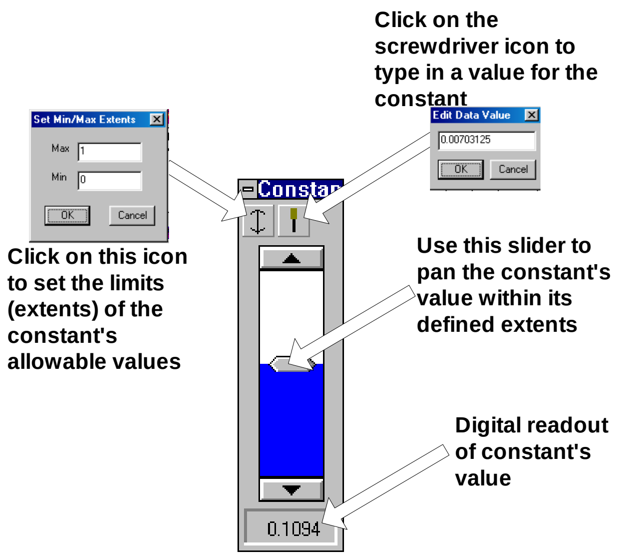


Figure 6 - Assigning a WAV file to FXDirect Looped WAV Player

Note: If you have downloaded the Turboprop design (*.des) file and the two associated wave (*.wav) files from the SimPhonics site, be sure that the V+ design worksheet can find their path and file names on your computer. To do this, double-click each of the two looped wavefile player objects on the design worksheet. Click on Filename, and make certain that the complete path to the appropriate wave file is shown in the text box (Figure 7). Edit the path as necessary.

If you do not remember the exact location of the wave file, click on the diskette symbol on the left of the wave player object, which will allow you to browse for the location of the file and select it. Once the right path and filename have been determined either through typing it in or browsing, click on the round green button to the right to associate that Wave sound file with the looped wave player object.


Click on the screwdriver icon to type in a value for the constant
Set Min/Max Extents
Max
1
Min
0
OK
Cancel
Edit Data Value
0.00703125
OK
Cancel
Constan
Click on this icon to set the limits (extents) of the constant's allowable values
Use this slider to pan the constant's value within its defined extents
Digital readout of constant's value
0.1094

Now that you have a simple but fully functional FXDirect design, it is time to experiment. Click on the green Run button on the toolbar to start the platform, and a turboprop engine sound should be emerging from one or both of your computer speakers. Click on any or all the "constant" objects and use the pan control to change the value of the parameter to see what the effect is. Click on the LFI Table objects and drag the LFI plot points up and down to determine the effect of the changes made. All this can be done in real time, when the platform is running, providing immediate audible feedback on the results of the changes.

Figure 7: Using the PAN Control

Note: If you hear an occasional and faintly familiar musical sound while running the demonstration, chances are you have not assigned one or both of the looped wave players to the correct wave file. In this case, Windows assigns the Windows98 startup sound *(c:\windows\meadia\The Microsoft Sound.wav)* to the wave player as a default file.

MULTIPLE CHANNELS

With DirectSound®, multiple audio signals can be mixed and played at the same time to one or more channels. This is an advantage to using FXDirect, since the number of channels is limited only by the processor performance and the power and speed will always be improving, not to mention the reduction in cost.

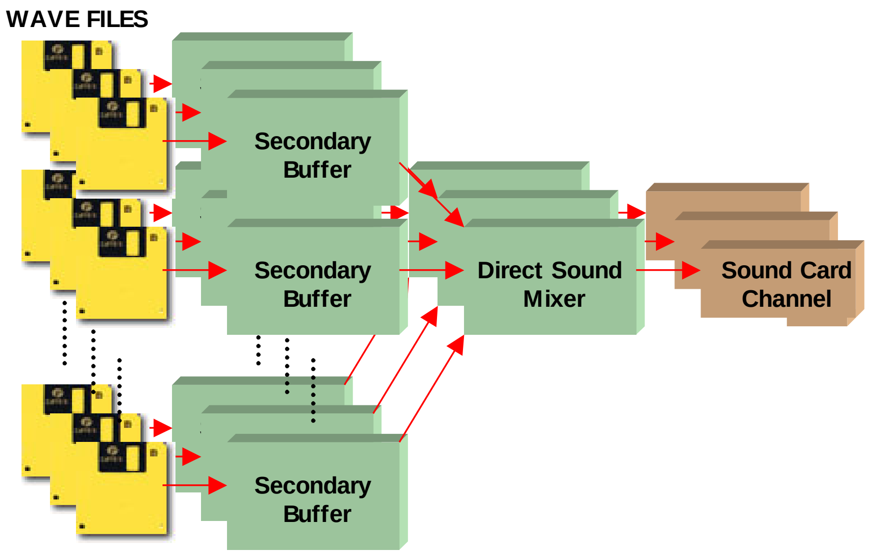


Figure 8: Use of Multiple Channels

V+ FXDIRECT OPERATION WITH DIRECTX®

FXDirect is an extension of the V+ visual programming system. Since FXDirect is really a visual "wrapper" around DirectSound, the same limitations and requirements for DirectX® apply to FXDirect. For example, Windows NT 4.0, with service pack 4, only features DirectX 3.0. That version has some performance limitations, and does not include input support.

FXDIRECT COMPATIBLE SOUND CARDS

DirectSound is rather new, and as such, only a few sound cards are fully compatible with the new standard. While most of the standard PCI based sound cards will work, they do so via a DirectSound® emulation driver, which is not fully DirectSound compliant. The real issue is the number of independent channels that will work with DirectSound, since most simulation applications require more than two channels of sound output. Although theoretically it is possible to install multiple sound cards in a PC chassis, in reality this often results in resource conflicts.

As of this printing the only sound card system that can produce more than four channels using DirectSound®, is the GadgetLabs Wave/824, using the DirectSound® driver from Stage Research. This system consists of an internal PCI card that drives eight channels of audio from an external rack-mounted chassis. See www.gadgetlabs.com for more information.

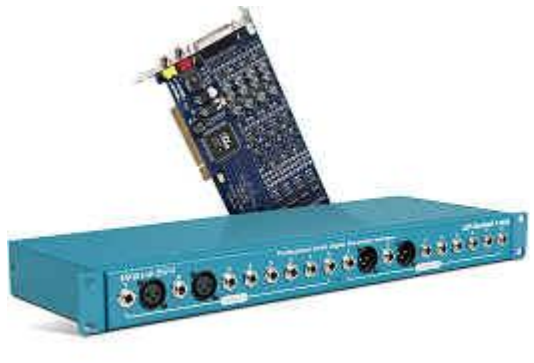


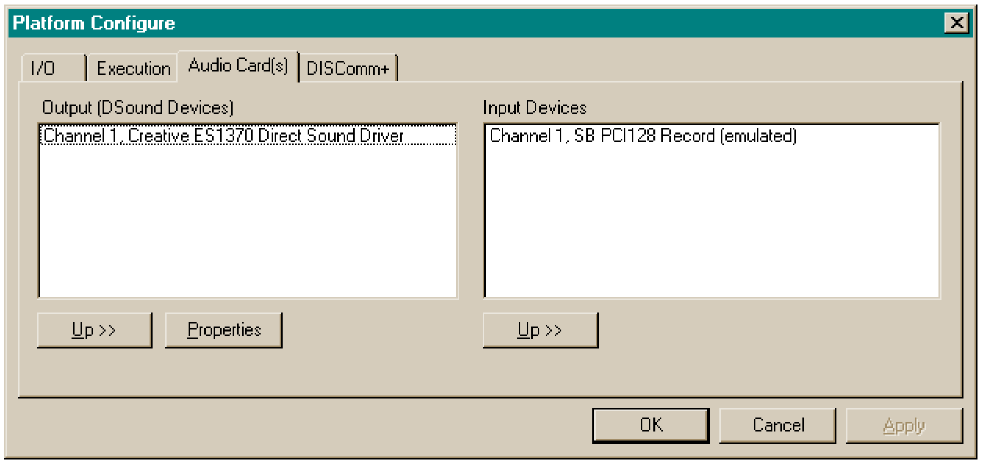
Figure 9: GadgetLabs Wave 8/24 Components

Another important sound card issue is FULL-DUPLEX capability, which is required when using VCOMM+, which uses both the input and output for communications. (VCOMM+ replaced DISComm+)

FXDIRECT CHANNELS VERSUS AUDIO OUTPUTS

The term "channel" within FXDirect refers to a stereo independent audio output or input. A stereo output actually has two electrical outputs and all FXDirect objects treat the output as a single output. However, by using two FXDirect objects, and the pan control that is available on some FXDirect objects, two independent outputs can be realized.

Therefore FXDirect channels are stereo pairs with two audio outputs. The following V+ runtime system shows a single audio card, although Channel 1 is actually two electrical outputs.



Output (DSound Devices): Channel 1, Creative ES1370 Direct Sound Driver

Input Devices: Channel 1, SB PCI128 Record (emulated)

Figure 10:V+ Platform Configuration Menu

This V+ design shows two sound players that can be set to the same "channel", although each can be generating a separate sound, since the balance controls are set for left and right outputs.

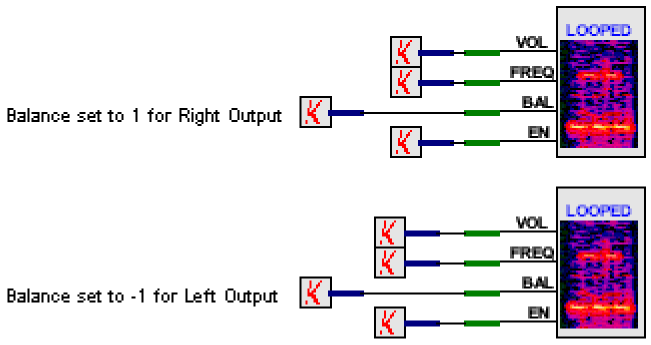
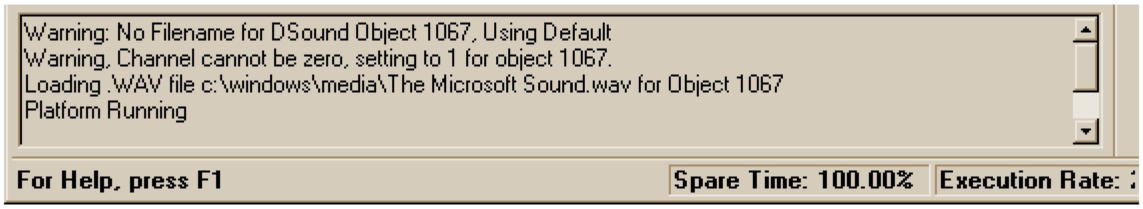

LOOPED
VOL
FREQ
BAL
EN
Balance set to 1 for Right Output
LOOPED
VOL
FREQ
BAL
EN
Balance set to -1 for Left Output

Figure 11: Using separate channels of PCI Audio Card

FXDIRECT MESSAGES

The V+ run-time window contains a message area that will display various system messages, both during initialization, and during run-time. FXDirect uses this area extensively for generating information, warning messages, and errors.

For example, if a single FXDirect object is loaded and executed with default filename and channel the following warnings are generated in the message area.



```
Warning: No Filename for DSound Object 1067, Using Default
Warning, Channel cannot be zero, setting to 1 for object 1067.
Loading .WAV file c:\windows\media\The Microsoft Sound.wav for Object 1067
Platform Running
```

For Help, press F1 | Spare Time: 100.00% | Execution Rate:

Figure 12: Typical FXDirect Warning Messages

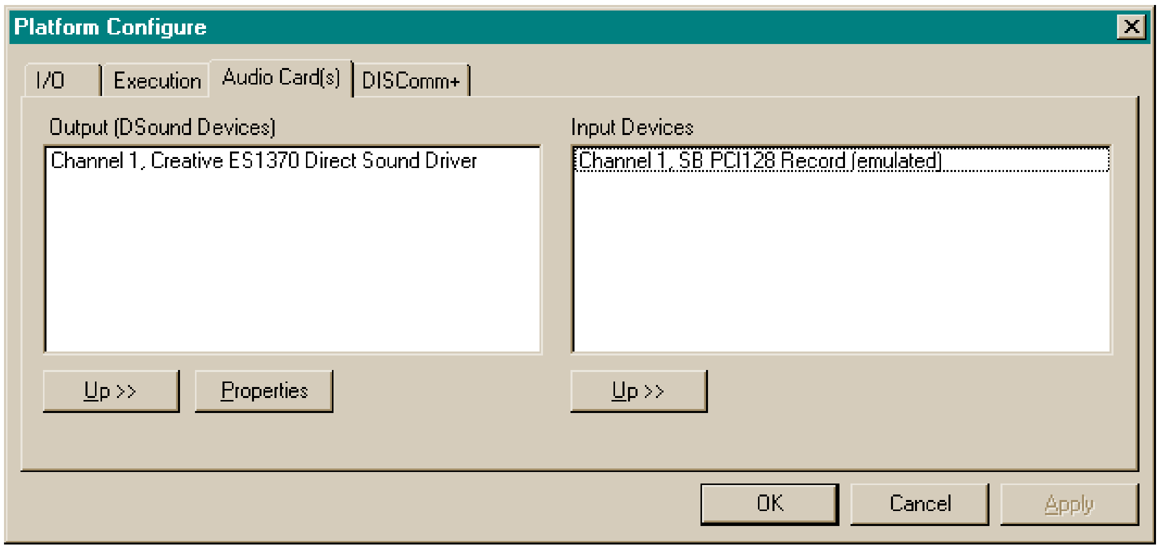
In the example offered in Figure 13, the first warning indicates that the filename was missing, so the system assigned a default filename to the object in order to execute the design. The default filename is always *"c:\windows\media\The Microsoft Sound.wav".* If your system does not contain this file, an error will be generated, and the system will not start.

The second warning indicates the channel number was zero, which is not a valid channel, and the system defaulted the channel to 1.

Next, the system is indicating the filename that was loaded.

TROUBLESHOOTING

First and foremost, in order to run FXDirect, your system must have a sound device. Otherwise, FXDirect objects will not execute, or even initialize. In order to determine which sound devices that FXDirect recognizes, click the CONFIGURE menu item in the run-time system, and click the Audio Card(s) tab as shown.



Output (DSound Devices)

Channel 1, Creative ES1370 Direct Sound Driver

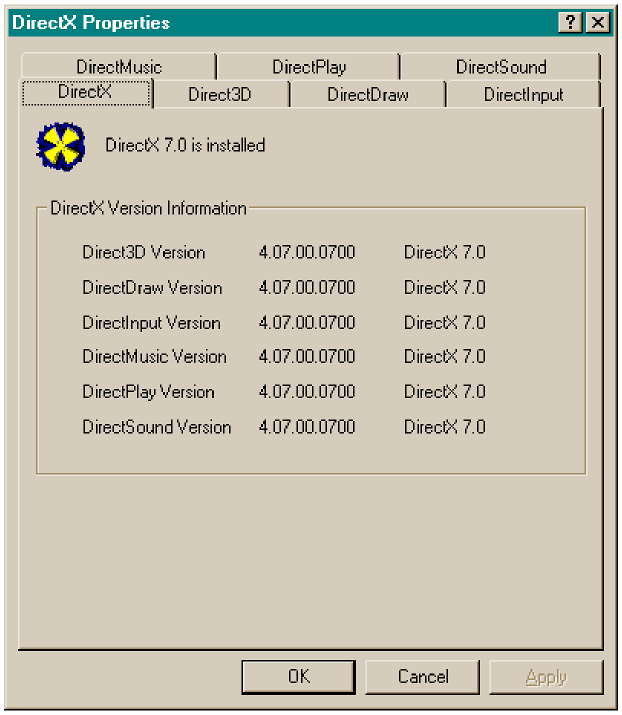
Input Devices

Channel 1, SB PCI128 Record (emulated)

Figure 13: V+ Runtime CONFIGURE Menu

Second, ensure you have an operating system that is running DirectX® 3.0 or later. See www.microsoft.com/directX for information regarding DirectX® versions and compatible operating systems. V+ is shipped with the latest version available of DirectX®. Determine the your version of DirectX® and load the latest version of DirectX® on your system if your out-of-date.

Open the control panel and open the DirectX® icon. The following screen will appear:


DirectX Properties

DirectX 7.0 is installed

DirectX Version Information

Direct3D Version	4.07.00.0700	DirectX 7.0
DirectDraw Version	4.07.00.0700	DirectX 7.0
DirectInput Version	4.07.00.0700	DirectX 7.0
DirectMusic Version	4.07.00.0700	DirectX 7.0
DirectPlay Version	4.07.00.0700	DirectX 7.0
DirectSound Version	4.07.00.0700	DirectX 7.0

Figure 14: Determining installed version of DirectX

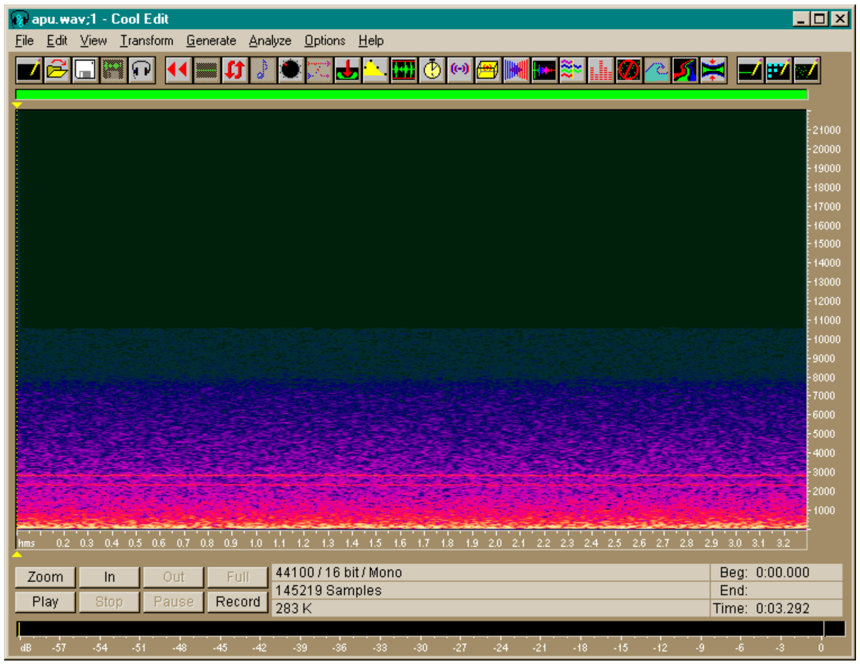
As shown in Figure 10, this computer is running DirectX version 7.0.

Next, ensure you have the latest version of V+. Check the SimPhonics web site at www.simphonics.com/vplus/ for the latest versions. Some older builds had problems with some sound cards and FXDirect.

WAV EDITORS

There are a number of audio editors on the market today, and most of them are adequate for sound editing. However, few of them support the generation of sounds from scratch, and lack visualization features.

SimPhonics uses Adobe Audition from Adobe Systems, Inc. The editor features a spectrograph view, which makes the job of analysis much easier when working with WAV files. Several other features make the product desirable for use with V+ and simulation of sounds, including synthesis capability. This document uses screen captures from this program.


apu.wav:1 - Cool Edit
File
Edit
View
Transform
Generate
Analyze
Options
Help
21000
20000
19000
18000
17000
16000
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
hms 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.0 3.1 3.2
Zoom
In
Out
Full
Play
Stop
Pause
Record
dB -57 -54 -51 -48 -45 -42 -39 -36 -33 -30 -27 -24 -21 -18 -15 -12 -9 -6 -3 0

44100 / 16 bit / Mono
145219 Samples
283 K

Beg: 0:00.000
End:
Time: 0:03.292

Figure 15: The Adobe Audition Sound Editor

OPTIMIZING FXDIRECT SYSTEM PERFORMANCE

Tests have shown that up to 200 FXDirect objects may be used simultaneously with a Pentium II 350 on a machine with 128M of RAM under Windows98®. After that the system is completely loaded. Since CPU performance is ever increasing, currently around 700mHz, performance will not be a factor in the future. However, you should be aware of the performance issues when using FXDirect.

Therefore, the most important performance consideration is the performance of the hardware. Use the highest speed possible, and ensure that your memory size is at least 128M.

Use an AGP video card if possible, since some video cards "hog" the PCI bus, which the sound cards use.

Use the latest version of DirectX® and the latest build of V+. As of this printing, DirectX 7.0a is available.

Compute the maximum size of your FXDirect sounds, to ensure you have at least twice that size in memory. Otherwise, the operating system will swap the files in and out of the virtual memory area, which resides on the hard drive.

USING SOUNDS WISELY

One of the best features of DirectSound® is its ability to play and control multiple audio tracks independently. While this is a real boon to sound designers, it doesn't come without cost. The cost is CPU cycles. Each secondary buffer you use consumes CPU cycles. Each processing operation such as frequency scaling consumes additional CPU cycles.

Determine the impact of sound use on overall performance. Premix sounds whenever possible to reduce the use of secondary buffers. For example, if you're creating summertime night ambiance with chirping crickets on one track and croaking frogs on another track, combine the two into a single file.

USING THE SAME FORMAT FOR THE SECONDARY AND PRIMARY BUFFERS

The DirectSound® mixer converts the data from each secondary buffer into the format of the primary buffer. This is done on the fly as data is mixed into the primary buffer. This format conversion costs CPU cycles. You can eliminate this overhead by ensuring that your secondary buffers (i.e. wave files) and primary buffer have the same format. In fact, due to the way DirectSound® does format conversion, you only need to match the sample rate and number of channels—it doesn't matter if there is a difference in sample size (8-bit or 16-bit).

USE A PCI SOUND CARD

If you can specify and use a PCI type sound card, do so, since this reduces the number of wait states on the CPU and increases system throughput to the sound card. Using ISA type cards involves the use of DMA channels, which reduces the number of hardware resources available to the system, and requires overhead from the main CPU to manage.

REDUCING DATA RATE OF PRIMARY BUFFER

Experiment with reducing the data rate requirement by changing the format of the primary buffer. The tradeoff here is obvious: performance versus sound quality.

PLAYING THE PRIMARY BUFFER CONTINUOUSLY THROUGH PERIODIC SILENCES

If the sound is not playing, turn it off using the enable pin on the FXDirect object. If amplitude is zero, turn off the enable pin, since the sound is still playing and consuming CPU cycles. However, if your model has frequent short intervals of silence, the overhead of starting and stopping the object each time a sound is played may be worse than the overhead if you kept the object active.

USING HARDWARE MIXING

Most sound cards support some level of hardware mixing if there is a DirectSound driver for the card. The following tips will allow you to make the most of hardware mixing:

- DirectSound will attempt to use hardware mixing when possible.
- Use the execution sequence feature of V+ to setup the sequence of execution, so that sounds you use the most (there's a limit to the number of buffers that can be hardware mixed) are initialized and played first.

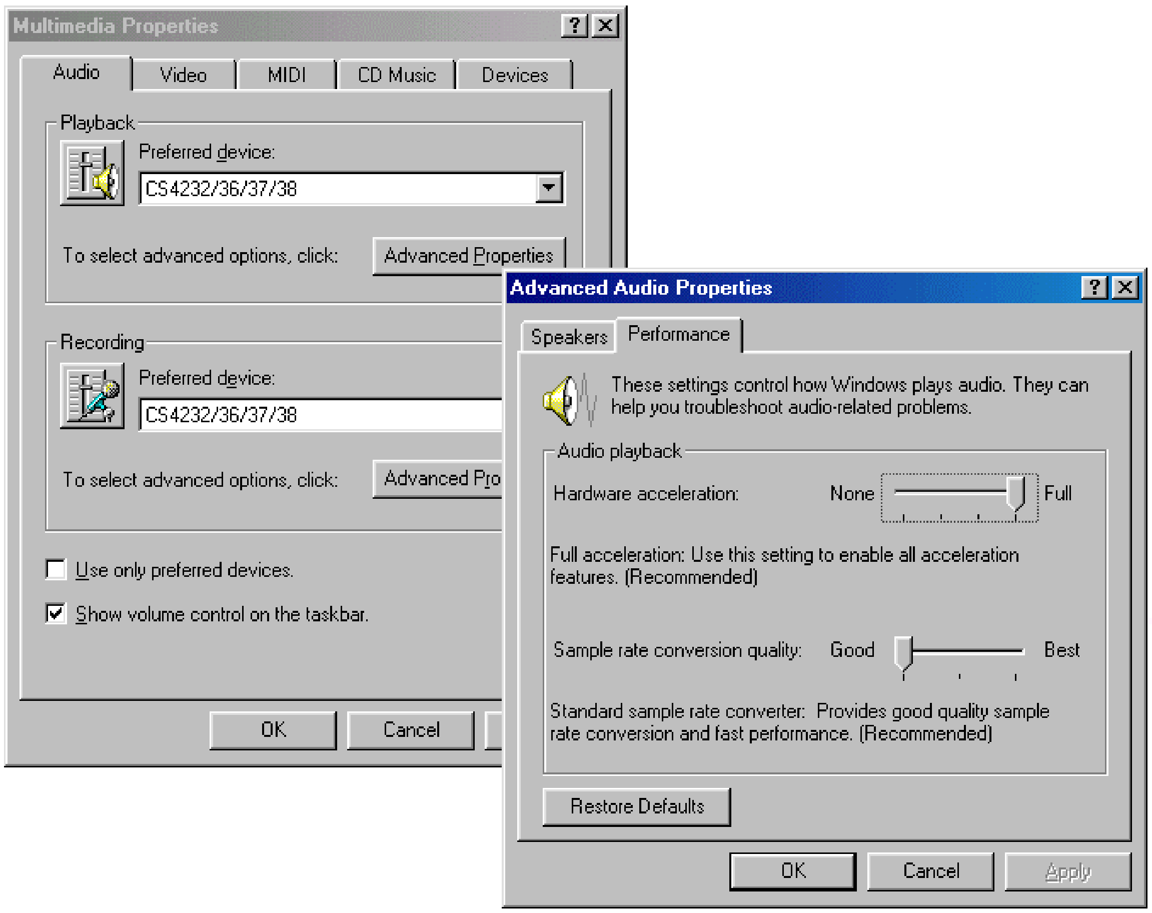
WINDOWS TECHNIQUES FOR SAMPLE RATE CONVERSION (SRC)

KMixer (Microsoft DLL associated with DirectSound®) can perform four different types of sample rate conversion. There are a number of rules that govern the SRC type used for a given application. The four types are as follows:

- **Linear Interpolation** provides reasonable results for similar sample rate conversions, for example, converting from 12 kHz to 13 kHz. This is typically the mixing type for Microsoft DirectSound®.
- **Multipoint Interpolation** delivers approximately 70 dB of signal-to-noise ratio (SNR) through a reduced complexity version of the following conversion type.
- **High-end Multipoint Interpolation** provides high fidelity by using a technique of extremely high oversampling. The performance exceeds 90 dB SNR for all conversions.
- **Add/Drop** is also known as truncation or nearest neighbor. This technique is only used as a fallback scenario when CPU utilization or other constraints prevent using one of the higher quality modes. Under normal circumstances, Add/Drop will never be used.

Policy for Sample Rate Conversion

The multimedia control panel gives the user some control over how the KMixer will behave when SRC is required. The following illustration shows this control panel.



Playback

Preferred device: CS4232/36/37/38

To select advanced options, click:

Recording

Preferred device: CS4232/36/37/38

To select advanced options, click:

[ ] Use only preferred devices.

[✓] Show volume control on the taskbar.

These settings control how Windows plays audio. They can help you troubleshoot audio-related problems.

Audio playback

Hardware acceleration: None [✓ Full]

Full acceleration: Use this setting to enable all acceleration features. (Recommended)

Sample rate conversion quality: [✓ Good] Best

Standard sample rate converter: Provides good quality sample rate conversion and fast performance. (Recommended)

Setting SRC Characteristics

Some of the rules governing which SRC type is used include the following.

- For DirectSound, the slider shown in the "Advanced Audio Properties" window maps "Good" through "Best" to linear interpolation, multipoint interpolation, and high-end multipoint interpolation, respectively. Linear interpolation is the default, which is consistent with Microsoft DirectX® 6.0 policy. SRC will only occur if necessary. Many applications that use several audio streams use the same sample rate for each stream. When that happens, the KMixer will not perform SRC on the streams.
- The Wave APIs contained in MMSYSTEM/WINMM streams always use the high-end multipoint interpolation if SRC is necessary. The control panel has no effect for these streams.
- Digital CD will always use high-end multipoint interpolation if SRC is necessary.
- The SoundBlaster emulator and software synthesizer always use high-end multipoint interpolation if SRC is necessary.

As a point of reference, the following table indicates each application's appropriate application programming interface (API):

Application	API
Windows 98/2000 Sound Recorder	Wave
Windows 98/2000 Media Player	Wave
Windows 98/2000 ActiveMovie™ Player	DirectSound
Most game titles	DirectSound
Most high-end wave editors	Wave

Output Sample Rate and Mixing Policy

The KMixer has a sophisticated heuristic to avoid unnecessary SRC and mixing. The following list details the process for the first and additional streams.

Starting the first audio stream:

- The system audio device initializes the KMixer and assigns it a sample rate corresponding to the higher of the following rates:
    - 44.1 kHz
    - The highest rate available on the device
- KMixer asks the hardware if it supports the incoming rate:
    - If yes, pass the incoming stream directly to the hardware
    - If no, stay at the current output rate

Starting another audio stream:

- If the incoming rate of the new stream is **equal** to the output rate:
    - Perform mixing only
- If the incoming rate of the new stream is **equal** to the rate of another input stream:
    - Mix with the other stream at the same rate
    - Take advantage of the existing conversion from the input rate to the output rate
- If the incoming rate of the new stream is **lower** than the current maximum input rate:
    - SRC to the output rate and mix
- If the incoming rate of the new stream is **higher** than the current maximum input rate:
    - Query the hardware to see if it supports the new input rate:
        - If yes:

        - Change the output rate to the incoming rate of the new stream
        - Switch the hardware to the output rate
        - SRC all the other streams to the output rate and mix it with the new stream
    - If no:
        - Stay at the current output rate and downsample the new stream

EXCEPTIONS:

- The KMixer will not adjust its output rate when a SetFrequency() call is made on a DirectSound® buffer. It only updates the output rate when the buffer is first played.
- The SoundBlaster emulation performs dynamic adjustments to its sample rate. The KMixer will maintain the output rate to be equal or higher than the emulator. The KMixer will not track the emulator when it decreases its sample rate.

REFERENCES

SimPhonics Web Site: www.simphonics.com
FXDirect Information: www.simphonics.com/fxdirect
FX-30 System: http://www.simphonics.com/fx30/fx30.htm
V+ www.simphonics.com/vplus
DirectX: www.microsoft.com/directx