7 Appendices

I. Results of the Tests

The test methodology for the latency tests is to use a simple technique that can be implemented on a sample-accurate sequencer, rather than requiring to use software developers' tools (eg Million Monkeys from the Apple SDK). It is the additional latency (if any) of the routing software that is the focus of these tests. The results were obtained using Logic Pro v8. CPU load tests were not originally considered but it became apparent that some of the audio routing software places a considerable workload on the CPU. As the computer will be running at least two other audio applications at the time the routing software is active, this could not be ignored.
Initial Testing was done on a MacBook (2 GHz Intel core 2 duo, OS 10.4.11, 3GB RAM). Other computer models and/or operating systems used are indicated in the following tables.
Housekeeping routines and verification of permissions were checked before the tests were run.
Test files are on CD1. The results of testing various combinations of audio software are in Appendix IX.

Latency[4] Test Methods

The test setup was to record the relative latencies of each of the audio routing programmes against the others. The absolute system latency is not considered. As a reference, latency was also recorded when using a loopback cable. In either case the 'round trip' latency[5] is measured (i.e. input + output), where:

$$\text{Latency (in mS)} = 1000\ [\partial \text{ sample count / sample rate}]$$

The testing procedure was as follows:
A transient signal was used to generate a pulse. In the case of the first method the source was a Test Oscillator plugin using a 1kHz needle pulse waveform. This was inserted on a mono channel with the output panned hard left so as to only appear on the L side of the output channel. A second mono channel with input set to input 1 (L) was panned hard right and sent to the output channel. A series of stereo channels were created to record both these signals. The tone was manually pulsed on for three bursts using the channel mute control while the signal was re-recorded onto a stereo channel.
The recording was analysed using the sample editor, which gives a sample count reading between the L and R signals. Results were averaged on readings from the leading and trailing edge of the three tone-bursts.
The second method used a wavefile as the source. A percussive electronic signal was used, and the setup was otherwise similar. The wavefile was trimmed in the sample editor so the leading edge of the envelope was exactly at 0 time. The wavefile was then put in an audio track and copied on the beat for two bars. Once each recording was done the new track was merged with the original track and analysed in the sample editor, using the same technique as above. All recordings were taken at a sample rate of 44.1kHz.

---

[4] for a description of latency, see Appendix VI
[5] here, Latency is defined as the minimum time needed for the computer to store a sample from an audio interface into application memory and copy the same sample from application memory to the audio interface output.

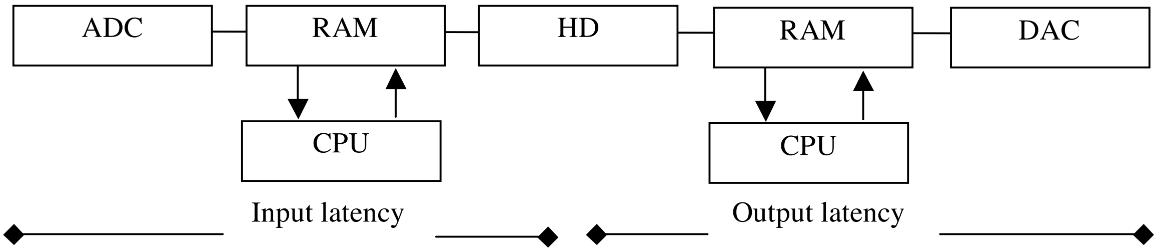
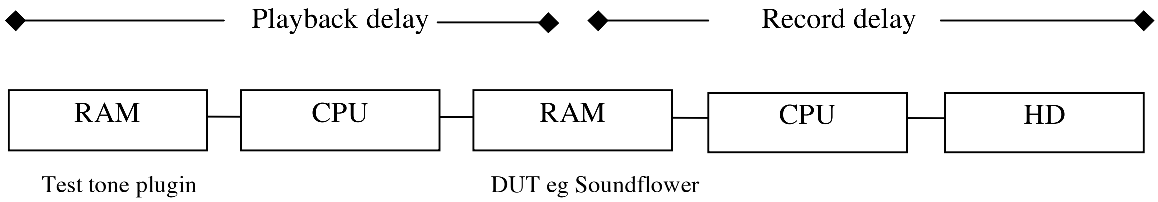


Fig. 9a Overall system latency is the combination of input and output delays.

Latency Measurements

i) Software Routing:



ii) Cable:

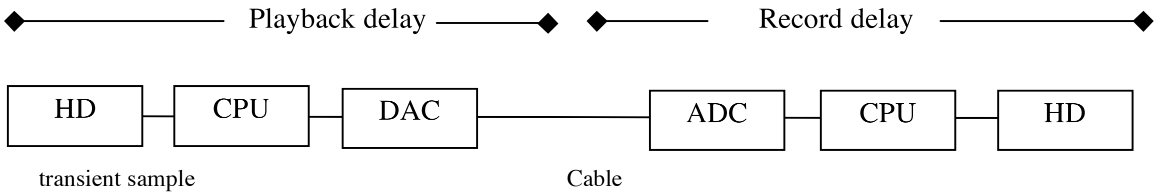


Fig. 9b The methods used for testing latency: i) software, ii) via loopback cable

Fig. 10 The latency test setup. A narrow width pulse waveform is used to get a fast risetime.

Latency Test Results

The best-case latency test results are shown below. These values are taken from the following tables. A result for WireTap Anywhere was not possible, as it does not allow loopback of the audio for a single application.

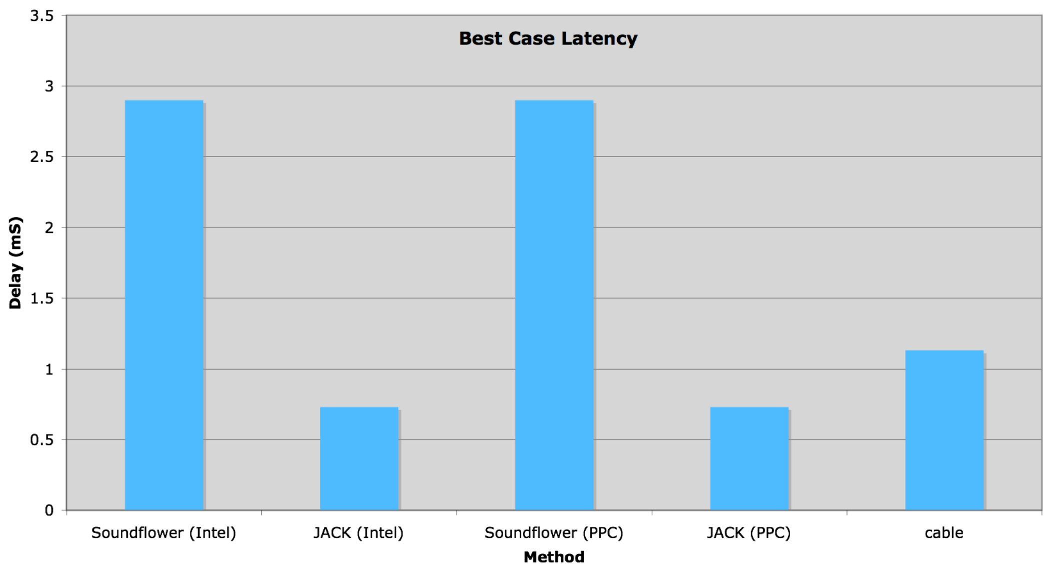


Fig. 11 Lowest latency comparison

Notes: 1. tests using test tone are in the folder: Testing/Latency Toneburst Tests.
2. tests using the wavefile are in the folder: Testing/ Latency Reference Tests.

A. Latency for MacBook (2 GHz Intel core 2 duo, OS 10.4.11, 3GB RAM)

Table 1. Reference Tests using a Test Tone plugin as the source signal

Device Tested	Logic Audio buffer size	Samples	Delay (mS)	Soundflower buffer size
loopback cable	1024	3720	84.35	N/A
	32	744	18.14	
	32*	720	16.33	

* with System Preference (Energy Saver) set to “Better Performance”

Table 2. Reference Tests using a time-aligned+ percussive wave file as the source signal

Device Tested	Logic Audio buffer size	Samples	Delay (mS)	Soundflower buffer size
loopback cable	32***	50	1.13	N/A
	32*	112	2.54	
	32**	50	1.13	

* with Safety Buffer enabled
** with Delay Compensation on
*** with Delay Compensation off
\+ sample truncated so the signal attack begins at exactly 0 mS in Logic Audio.

Table 3. Tests using a Test Tone plugin as the source signal

Device Tested	Logic Audio buffer size	Samples	Delay (mS)	router buffer size
Soundflower v1.2.1	64	128	2.9	512
	1024	2048	46.44	512
	32	64	1.45	512
	32	64	1.45	2048
	32	64	1.45	64
	32*	64	1.45	64
Jack v0.77	512	512	11.61	512
	32	512	11.61	512
	32	32	0.73	32
	512	32	0.73	32

* with System Preference (Energy Saver) set to “Better Performance”

Table 4. Tests using a time-aligned+ percussive wave file as the source signal

Device Tested	Logic Audio buffer size	Samples	Delay (mS)	Soundflower buffer size
Soundflower v1.2.1	32	50	1.13	64
	32	50	1.13	2048

B. Latency for iBook (1.42 GHz PowerPC G4, OS 10.4.11, 1GB RAM)

Table 5. Tests using a Test Tone plugin as the source signal

Device Tested	Logic Audio buffer size	Samples	Delay (mS)	router buffer size
Jack v0.77	1024	512	11.61	512
	256	512	11.61	512
	1024	32	0.73	32*
	32	32	0.73	32*
	32	64	1.45	64
Soundflower v1.2.1	32**	64	1.45	64
	1024	2048	46.44	64
	1024	2048	46.44	2048
	32**	64	1.45	2048
	64	128	2.90	2048

- recording was completed with no noticeable degradation, but brought up a sync error (“error trying to sync audio between send and receive devices”).

** dropouts of recorded signal occurred (with or without I/O Safety Buffer enabled).
All tests done with System Preference (Energy Saver) set to “Better Performance”

C. Latency for iMac (2.0 GHz Intel Core 2 Duo, OS 10.5.4, 3GB RAM)

Table 6. Tests using a Test Tone plugin as the source signal

Device Tested	Logic Audio buffer size	Samples	Delay (mS)	router buffer size
Jack v0.77	512	512	11.6	512
	512	32	0.73	32
Soundflower v1.2.1	32	128	2.9	64
	32	128	2.9	512
Soundflower v1.3.1	512	1024	23.2	512
	32	128	2.9	64
	32	128	2.9	2048
	1024	2048	46.4	64
	32	128	2.9	512
	64	128	2.9	512
	128	256	5.8	512

CPU Load Test Results

Like many other audio tasks, audio routing can take a considerable amount of the available processing use. These figures are indicative only as they often fluctuate. Where considerable variances occurred between the peak and steady values, these have been listed separately. Figures were obtained by observing the % CPU usage in the Activity Monitor utility over a period of one minute.

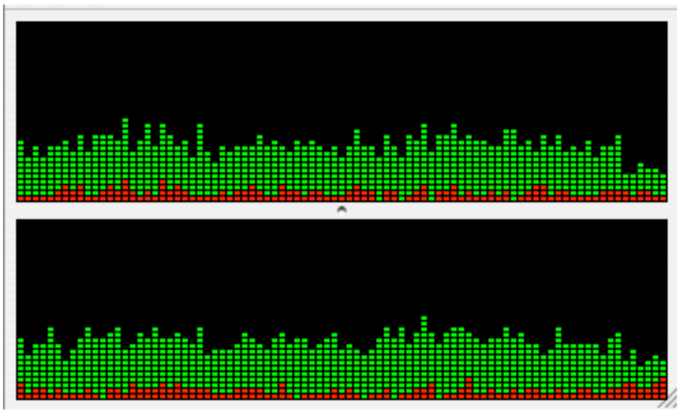


Fig. 12 The Activity Monitor CPU Usage window. Each graph is for an individual CPU core. Green indicates User %. Red is System %. The window is swept about every 1.5 minutes. This graph is showing the CPU usage while streaming audio from Reason into Logic on a MacBook

CPU usage is also shown on the JackPilot so I originally used those figures when testing JACK. They vary considerably from those in Activity Monitor, generally reading lower values. These are included for interest only and the unbracketed numbers should be used for comparison purposes. The CPU meter in JackPilot represents the sum of the real-time audio thread CPU use of all Jack clients. Thus it gives a partial picture of total CPU use, since it does not take into account any GUI impact on CPU use of either JackPilot or of the Jack clients (Davis P, 2008)

A. CPU Load Table

Device Tested	Logic Audio buffer size	router buffer size	Computer	CPU usage (%)
Jack	N/A	64	PPC*	20 (20)
Jack	N/A	32	PPC*	30 (29-50)
Jack	N/A	512	PPC*	8 (5)
Soundflower#	64	N/A	PPC*	0+
Jack	N/A	32	Intel**	20 (14)
Jack	N/A	64	Intel**	12.5 (8.5) +++
Soundflower#	64	N/A	Intel**	0++
WireTap Anywhere	N/A	N/A	Intel***	6I, 6U, 22P
Jack	N/A	32	Intel***	13 (1.5) I, 14 (6) U
Jack	N/A	512	Intel***	6 (0.2) I
Rewire	N/A	N/A	Intel***	1.2 I, 5 U^
Audio Hijack Pro	N/A	32	Intel**	2I", 10U

Table 7. CPU usage for various software, hardware, and buffer sizes

*iBook (1.42 GHz PowerPC G4, OS 10.4.11, 1GB RAM)
**MacBook (2 GHz Intel core 2 duo, OS 10.4.11, 3GB RAM)
***iMac (2 GHz Intel core 2 duo, OS 10.5.4, 3GB RAM)

Jack readings are the combined CPU load of JackPilot and Jackdmp (the Jack Server).
bracketed results were as shown in JackPilot
+ Logic Pro CPU usage was 17.5% when idle and 55% when recording.
++ Logic Pro CPU usage was 6.5% when idle and 24% when recording.
+++ Logic Pro CPU usage was 8% when idle and 25% when recording.
" Audio Hijack idle usage was 0.2% with hijacking inactive.
Soundflower is a Kernel extension but showed no noticeable increase in kernel_task activity
^ Pro Tools CPU usage was 16% when idle and 40% when streaming audio
I= idle, U= in use (i.e. while streaming 2 channel audio), P= peak

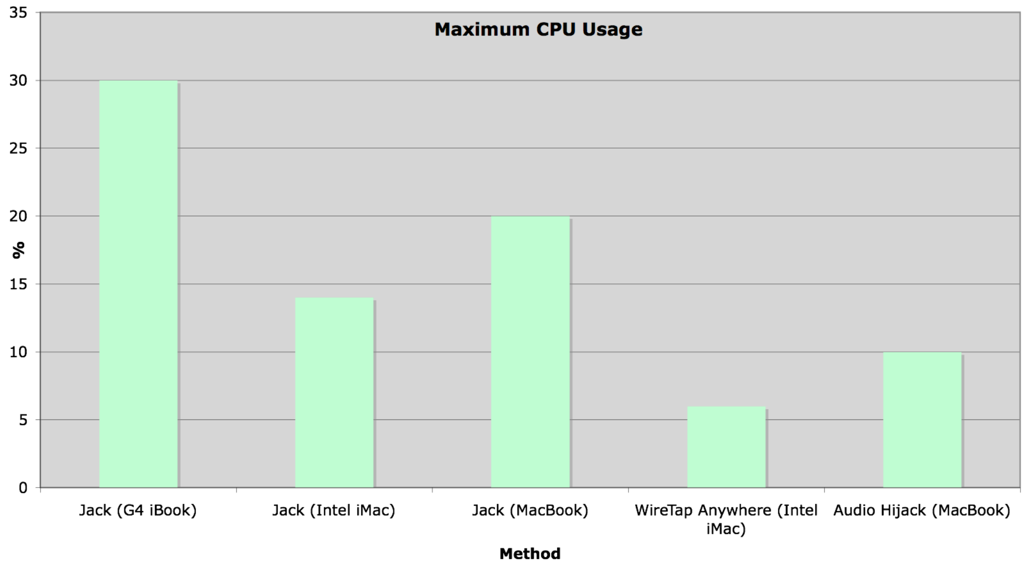


Fig. 13. Worst Case CPU Load comparison. These results are from Activity Monitor.

B. CPU Load vs. buffer size for JACK on a MacBook (2 GHz Intel core 2 duo, OS 10.4.11, 3GB RAM)

buffer size	Activity Monitor (%)	JackPilot (%)
32	18.8	1.2
64	12.5	0.9
128	9.5	0.7
256	7.3	0.28
512	6	0.2
1024	5.5	0.13
2048	5.1	0.09

Table 8. CPU load vs. buffer size for JACK

The JACK FAQ states, " the only impact of using JACK is a slight increase in the amount of work done by the CPU" (Davis, 2006). While it is true that the JackPilot test results are in the order of 1% and can be disregarded, Activity Monitor results are much higher. This would suggest that the overall CPU use is significant.

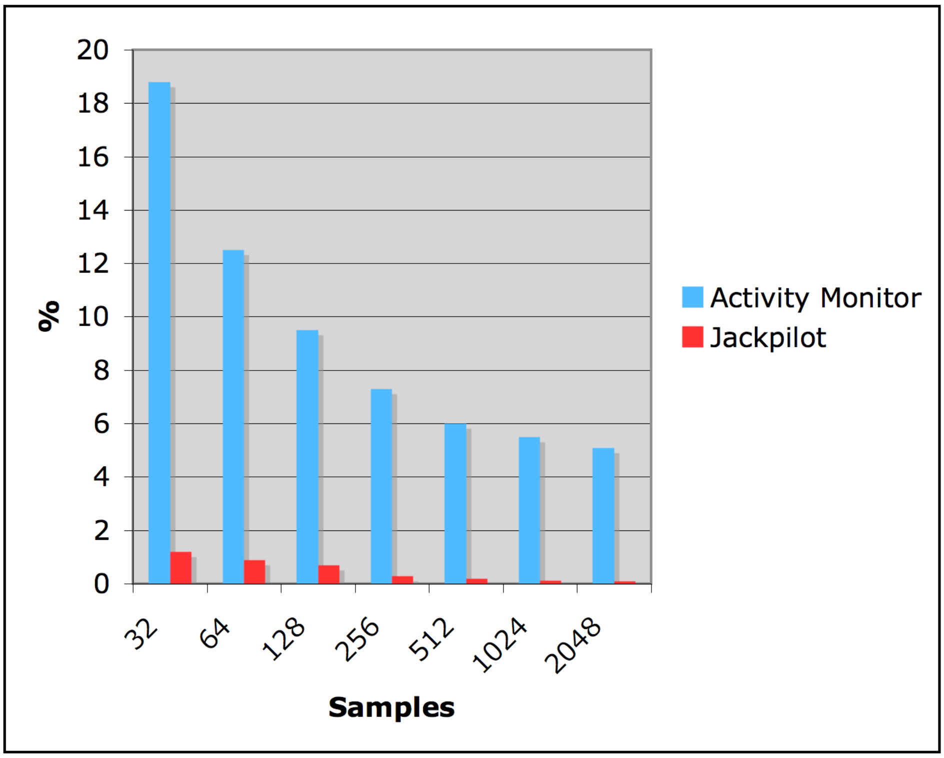


Fig. 13 CPU usage vs. buffer size for JACK. The sample rate is 44.1kHz

C. CPU Load vs. Sample Rate for JACK on a MacBook (2 GHz Intel core 2 duo, OS 10.4.11, 3GB RAM). The buffer size is 32 samples.

Sample rate (kHz)	Activity Monitor (%)	JackPilot (%)
44.1	17.3	1.4
48	19	1.5
96	57	5

Table 9. jackdmp CPU Load vs. Sample Rate for JACK

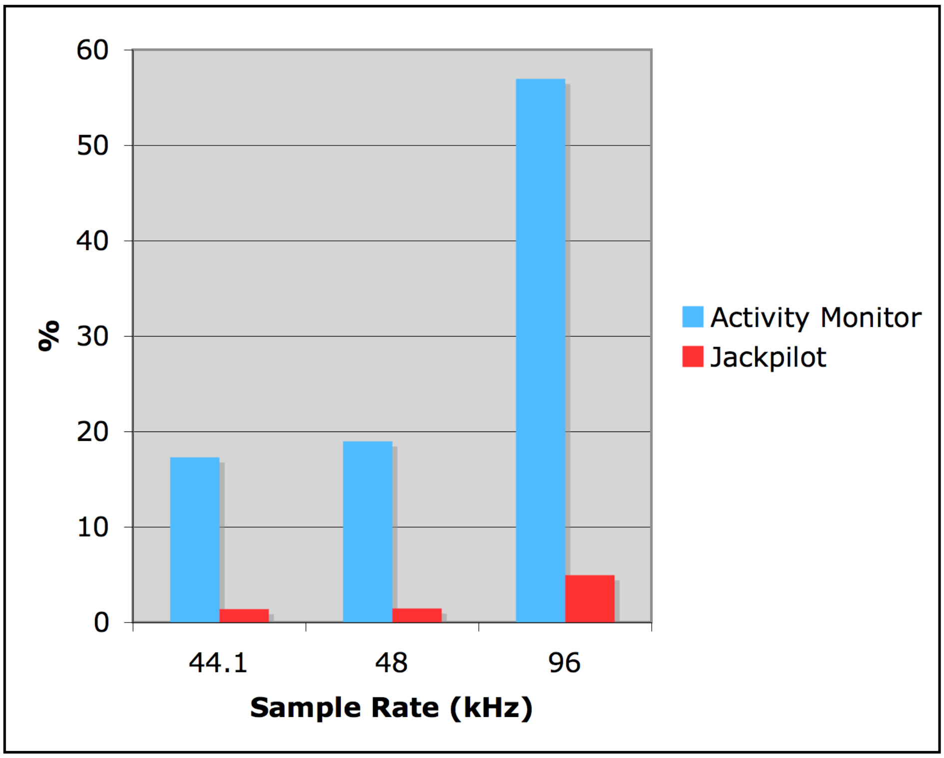


Fig. 14 CPU usage vs. sample rate for jackdmp

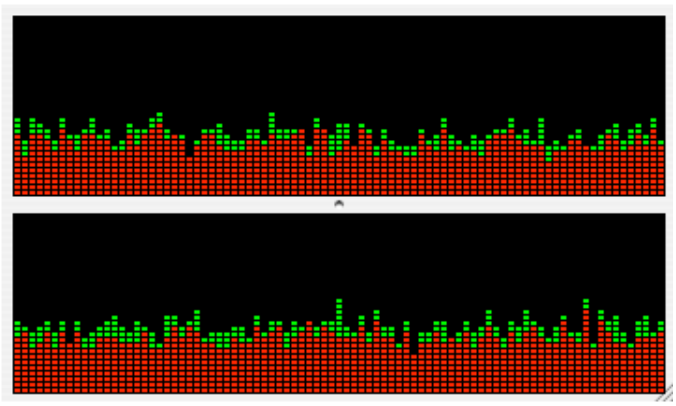


Fig. 15 System CPU usage (red) is high when running JACK at 96 kHz on a MacBook. Total average usage is over 40%, most of which is being used by JACK

II. Comparison Chart

The following table provides a quick way to compare the features and specifications of software tested in this report. Gaps in the table exist where a function could not be tested, or no data could be found. More detailed explanations of each software capability and performance are in the text of this report.

<u>Table Legend</u>

#	separate Jack parts are: Jackdmp 0.71, JackRouter 0.8.7, JackPilot 1.6.3
?	stability is evidenced by running without crashing (eg quitting, freezing, not responding). Reliability was appraised during normal operation (eg connecting, disconnecting, streaming)
$$	where CPU loads were variable; I= idle, U= in use, P= peak
*	for details see the latency performance charts
**	prone to high levels of digital noise on some interfaces (eg M-Box)
***	for 2 channels in and out, as measured in Activity Monitor
****	for an average user: simple = can be easily setup with little or no instructions, average = can be easily setup with simple instructions, complex = can only be setup by following detailed instructions
NA	not applicable
++	not tested; data is from the developers website
+	for further information refer to Combination test results in Appendix IX

Comparison Chart

	Soundflower	Jack	WireTap Studio	Audio Hijack	AUNetsend/ rec	Detour	cable
Version	1.2.1	0.77[#]	1.0.4	2.8.1	1.4.0, 1.4.1	1.5.5	N/A
Ease of Installatio n	simple	simple	simple	simple	Auto (with OS)	simple	simple
Documentation	minimal	50 page manual	help file	82 page manual	minimal	help file	N/A
Support	minimal	good				no	N/A
Ease of Setup	simple	complex****	simple	simple	simple	simple	simple
Reliability[?]	very good	very good	good	good	good	good	very good**
Ease of Un-installation	Uninstaller provided	Uninstaller provided		simple	N/A		N/A
Features: No. of audio channels No. of MIDI channels Buffer size settings (samples)	 2 or 16 0 64 - 2048	 unlimited 0 32 - 4096	 2 0 N/A	 2 0 32-6144, 384-2[18]	 2 0 N/A	 2 0 none	 2 (on built in audio) N/A
Latency (measured)	very low*	0			0		low*
Internal Bit depth		32 bit floating decimal point			16, 24, 32 FP		
Global or specific solution	global	global	global	global	semi-global	Semi-global	global
System Architecture	Kernel Extension	Synchronous Server/Client	Kernel Extension				
Compatibility[+] OS versions tested Digidesign Core Audio driver Conflicts with other utilities Hardware: PPC Intel Macs	 10.4.11 Y Y	 10.4.11 Y Y	 10.4.11 Y Y	 10.4.11 N Y Y	 10.5.4 Y	 10.4.11 Y N Y N	OK on most Mac models. Can only be used on an iBook with an external I/O as there is no line-in.
CPU load***$$	0%	12.5%	1%I, 22%U, 100%P	2%I, 10%U		0%	0%
Cost	free	free	US$69	US$32	free	free	$10 cable
URL	www.cycling74.com	www.jackosx.com	www.ambrosiasw.com	www.rogueamoeba.com	www.apple.com	www.rogueamoeba.com	N/A
Utilities accessed from menu	AMS	Sound Prefs AMS	none	AMS	none	none	N/A
Stability[?] (no. of crashes)	0	1	0	0	0	0	0

Comparison Chart

	WireTap Anywhere	DLS MusicDevice	Rewire	Soundsource	Sound Menu	QuickTime	Core Audio
Version	1.0.1	1.4.0	1.7	2.0	1.5.1	7.5.5	3.1.0
Ease of Installation	simple	Auto (with OS)	simple	simple	simple	Auto (with OS)	Auto (with OS)
Documentation	good on-line help	none	little	Readme file	Readme file	Developer docs	Developer docs
Support	online tutorials		online tutorials				
Ease of Setup	simple	simple	average	N/A	N/A	simple	simple
Reliability[?]	good	limited	good	excellent	excellent	excellent	excellent
Ease of Un-installation	Uninstaller provided	N/A		Uninstaller available	simple	N/A	N/A
Features:							
No. of audio channels	2	2	256	2	2	Unlimited	Unlimited
No. of MIDI channels	0	0	4080	0	0		N/A
Buffer size							Set by client
Latency	30mS++			NA	NA		Very low
Internal Bit depth						Sample-dependant	32 bit floating decimal point
Global or specific solution	global	specific	specific	global	global	specific	global
System Architecture			Server (host application)/ Client			Container format	Synchronous execution-via-callback API
Compatibility[+]							
OS versions tested	10.5.4	10.4.11, !0.5.4	10.4.11	10.4.11	10.4.11	10.4.11, 10.5.5	10.4.11, 10.5.5
Digidesign Core Audio driver			Y			Y	Y
Conflicts with other utilities	N			N	N	N	N
Hardware: PPC	Y			Y	Y	Y	Y
Intel Macs	Y	Y	Y	Y	Y	Y	Y
CPU load	6%						
Cost	US$129	Included with OS	bundled	free	free	Included with OS	Included with OS
URL	www.ambrosiasw.com	www.apple.com	www.propellerheads.se	www.rogueamoeba.com	www.aspirine.li	www.apple.com	www.apple.com
Utilities accessed from menu	N	N/A	N/A	Sound Prefs, AMS	Sound Prefs	N	Sound Prefs
Stability[?] (no of crashes)	0	0	1	0	0	0	0

III. Setup Procedures

1. Routing QuickTime audio into Logic, using Soundflower (refer to screen movie 1):

1. In the AMS set the default output to Soundflower (2ch)
2. In Logic set the record channel I/O= No output
3. In the Logic Preferences: Audio/Devices, set the CoreAudio Device to Soundflower (2ch)
4. Adjust the record volume in QuickTime
5. Record to an audio track in Logic
6. In Logic return the channel I/O to Output 1-2 and disable the track record
7. Playback the recording in Logic

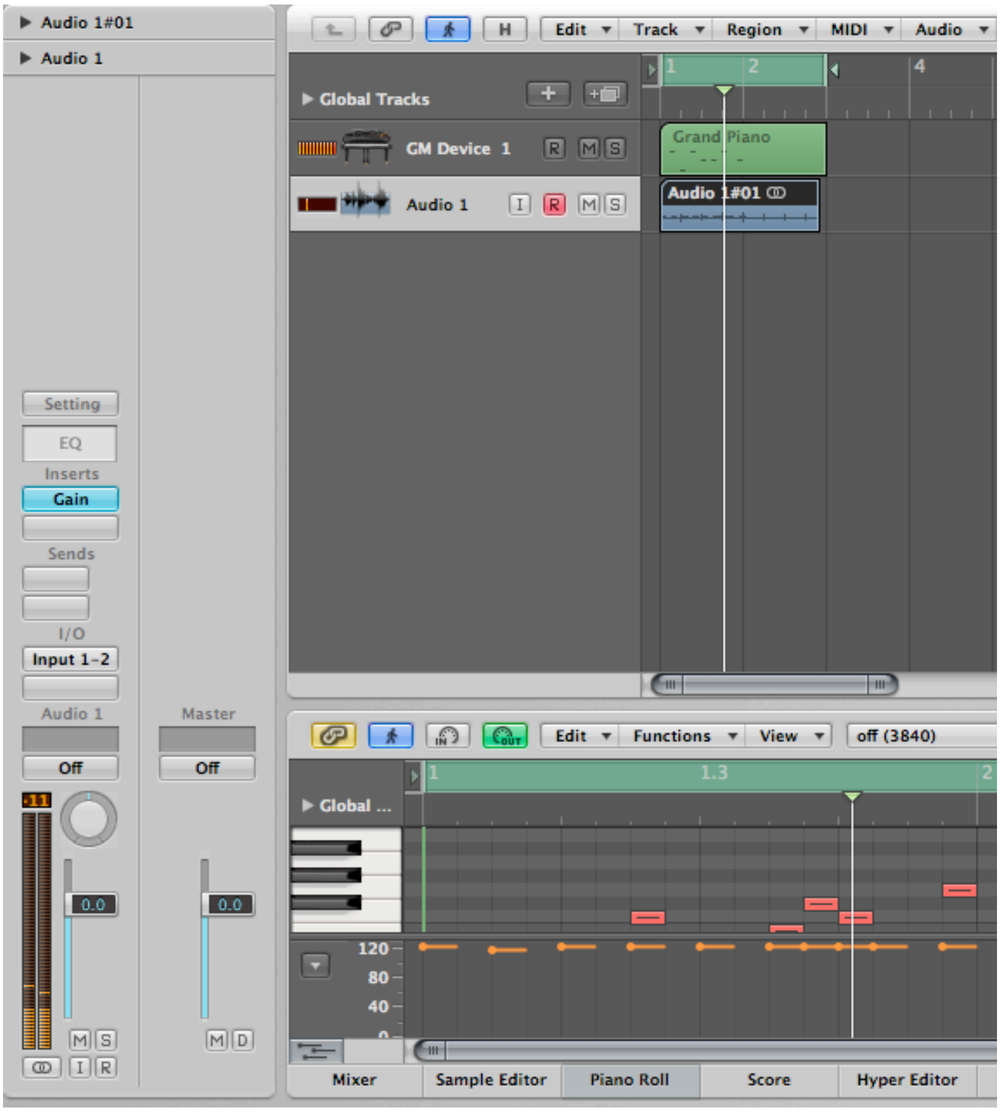


Fig. 16 Re-recording MIDI as audio in Logic, using Soundflower

Audio from iTunes can be recorded in a similar way (using the iTunes volume to set the record level). This also works for QuickTime Music Synthesiser (i.e. playing a MIDI file), however the level is quite low and cannot be adjusted with the QuickTime volume. This can be fixed by putting a Gain plugin into the record channel and setting the gain to around 10dB.

2. *AMS and Sound Preferences operation:*

When an input or output is selected in either the other one will automatically update to the new setting.
The input and output 'volumes' can only be adjusted in the Sound preference pane. These controls also set the level for recording and playback. If both are set to around 66% this will give unity gain on loopback.
A feature that is not apparent is that the output volume has two separate settings: headphone (when there is a plug in the jack, and another for the speaker volume.

3. *Setup Procedure for recording QuickTime into Logic Audio (Intel Mac), using JACK (refer to screen movie 3):*

1. In the Audio MIDI Setup create an Aggregate Device from the Audio menu. From the list select 'Built in Input' and 'Built in Output'. The Clock setting is not important, as Jack will synchronise the audio. Name the device eg Full Duplex (refer to screen movie 2)
2. Start JackPilot
3. In JackPilot Preferences select this Aggregate device as the Interface. Also check that the Virtual Input and Virtual Output channel numbers are set to 2 and that the buffer size is 512. Deselect 'Auto Connect with physical ports.
4. Start the Jack Server using the Start button on JackPilot.
5. In the Audio MIDI Setup select JackRouter as the Input and Output devices.
6. Start QuickTime and start playing a file. Check that QuickTime appears in the Send Ports list of the Connections Manager, which is accessed from the Routing button on JackPilot.
7. Start Logic Audio and in Preferences/Audio/Devices/CoreAudio select JackRouter as the device. Set the buffer size to 512. Click on the Apply Changes button. A pop up 'initialising CoreAudio' window should appear momentarily.
8. In the Connections Manager Logic Audio should now be showing on the Receive Ports list. Click **once** on QuickTime in the Send Ports list (this will highlight it in blue) and **double click** on Logic Audio in the Receive Ports list (it will turn red). The connection will now show in the Connections list on the right. To delete a connection double click on either the Send or Receive device again.
9. In Logic record enable a stereo audio track. The QuickTime audio should now be seen on the channel meter.
10. To hear output, Logic Audio needs to be connected to the System Device. In the Connections Manager click on Logic Audio in the Send Ports list, and then double click on system in the Receive Ports list. To save this setup, go to the JackPilot file menu (save studio setup…).

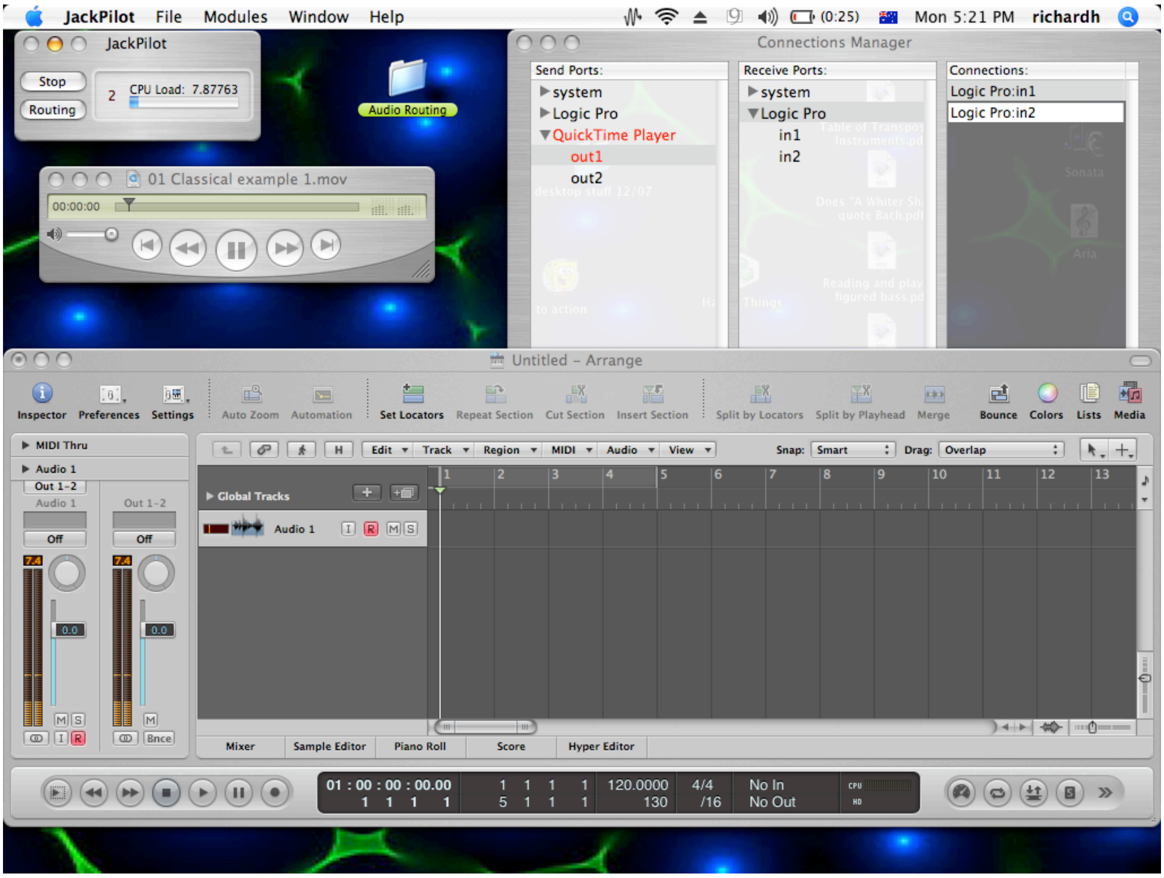


Fig. 17 Routing QuickTime to Logic using JACK. The setup for a PPC Mac is similar except step 1 can be ignored.

4. Routing Logic audio and MIDI into QuickTime Pro, using WireTap Anywhere (refer to screen movie 4):

1. In WireTap Anywhere create a device for Logic and name it Logic
2. Select this device (it will turn blue)
3. In QuickTime Pro Preferences/Recording set Microphone: WireTap: Logic and Quality: Device Native.
4. In QuickTime Pro select File/ New Audio Recording
5. In Logic push play. Check that audio is showing on the meter in the QuickTime Pro Audio Recording window.
6. Push record and then push play in Logic.
7. Push stop on Logic and then stop on the QuickTime Pro recording
8. A .mov file will be saved onto the desktop
9. Use File/Export, Export: Sound to Wave to create a .wav file.

IV. Other Audio Routing and Control Utilities

There are several other audio utilities that offer useful routing and control functionality. The two main purposes are to capture and record audio within the computer, and connect or select applications and audio devices. The software companies Rogue Amoeba and Ambrosia are most active in this area.

WireTap

The original WireTap was a free utility from Ambrosia Software and provided the ability to record any audio playback in OS X. Recordings are saved to file, either compressed or uncompressed.

Fig. 18 The WireTap GUI (Ambrosia Software, 2003)

WireTap Studio provided extra functionality and a more advanced GUI. Features added include adding audio effects, an audio library window, comprehensive on-line help, and scheduled recording. The main audio addition was the ability to source from two audio devices simultaneously and adjust the relative levels of each.

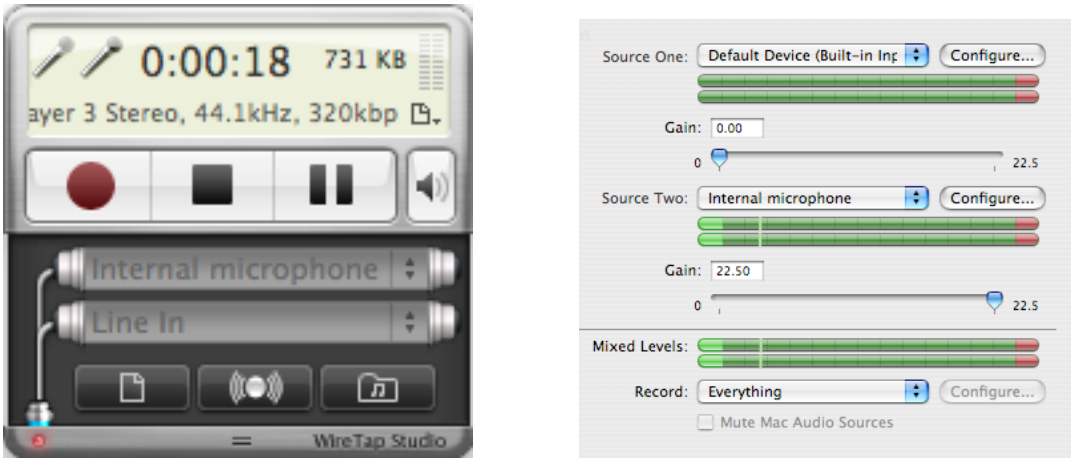


Fig. 19 The WireTap Studio GUI (floating controls, source tab) (Ambrosia Software, 2006)

Ambrosia has just announced (3Q, 2008) the release of its third generation WireTap product; WireTap Anywhere. The GUI is a System Preference pane where audio devices are configured. Up to 12 virtual devices can be created and each can capture the audio of up to 16 applications simultaneously (dominic, 2008). An uninstaller is provided with the user able to select the files with or without the associated kernel extension. The reason for this is that the kernel extension (AmbrosiaAudioSupport) is also used by other Ambrosia software, namely Snapz Pro X, and WireTap Studio.

Fig. 20 The WireTap Anywhere GUI (Ambrosia Software, 2007)

WireTap Anywhere was tested on a 2 GHz Intel iMac with 3GB of RAM. The operating system was Leopard 10.5.4. CPU power used was 22% peak and 6% while streaming 2 channel audio. This utility is quite easy to setup and use and has an excellent on-line help file. There are also six tutorial videos on the Ambrosia website which cover all the features. Unlike previous versions of WireTap, this utility is an audio patchbay and does not have any record to file function. Recording is done in any other chosen application (eg QuickTime Pro). An advantage of WireTap Anywhere is it will route audio to another application for recording without disconnecting the audio stream to the sound output. This enables monitoring of the audio while setting up and recording. Under test WireTap Pro performed very smoothly with no signs of crashing or glitching the audio. Devices can be selected and the power switched on or off while the audio is streaming without any adverse effects. WireTap Anywhere also has the option to run in AU mode in which case it is not necessary to create audio devises. For this to work the audio application must be AU Generator compatible. I was unable to test the latency of WireTap Anywhere, as it would not allow audio loopback. According to Dominic Feira from Ambrosia Software the latency is about 30mS at a 44.1kHz sampling rate for the AU Generator, and 32mS for the AUHAL plugin. Lowest latency is 27mS (http://www.ambrosiasw.com/forums).

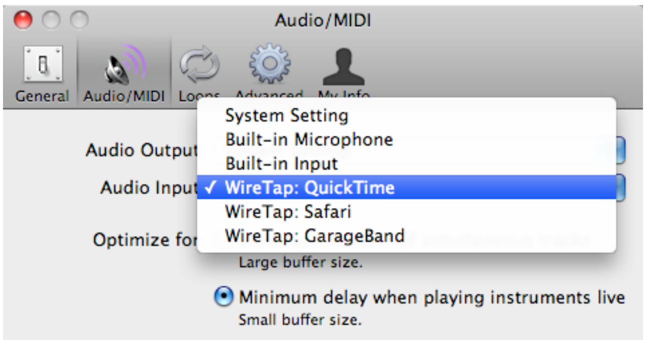


Fig. 21 WireTap devices as they appear in the GarageBand Preferences (Apple, 2008)

Audio Hijack Pro v2.8.1

This application performs the same function as WireTap. It can record audio from any source on OS X (applications, audio devices, internet streamed audio), and it is possible to combine these into a single recording. With the addition of Soundflower it can also record system audio. Audio can be recorded in four formats: AIFF (16 bit or 24 bit), ALAC, MP3, or AAC. A Quick Record feature allows hijacking audio without having to use all the settings for a full session. A convenient Split button allows the user to start a new file while audio is streaming. The input buffer size can be set between 32 and 6144 samples, and the output buffer can be set between 384 and 262144 samples. An optional install (Instant Hijack) lets this programme hijack audio from an application that is already open. Otherwise Audio Hijack must be run first.

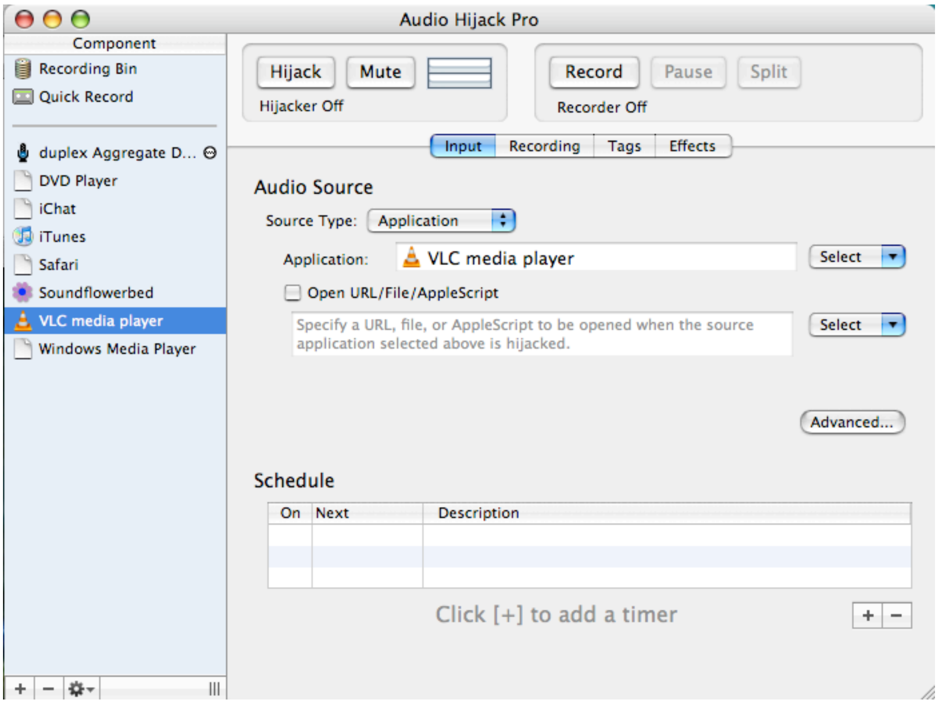


Fig. 22 Audio Hijack Pro GUI (Rogue Amoeba, 2008). Source Type can be set to Application, Audio Device, AM/FM Radio, or System Audio.

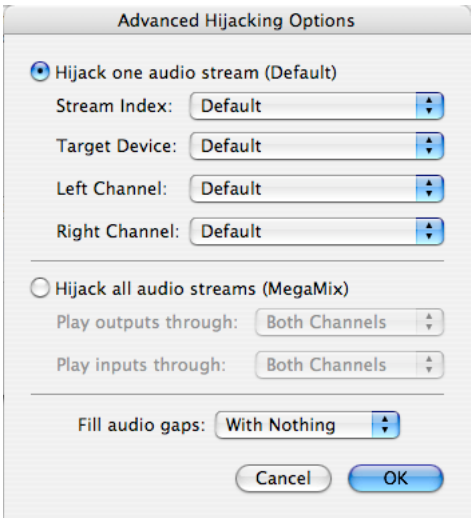


Fig. 23 Advanced options for the Application as a source setting (Rogue Amoeba, 2008). Stream Index has 8 levels to "allow you to receive audio from applications which output audio in non-standard ways". No further description of this feature is given.

Audio Hijack Pro 2 has a considerable number of extra features such as adding AU effects, using the Apple Automator function, running it using Applescript, CD burning, and recording podcasts using the inbuilt timers. These fall outside the scope of this paper so they will not be explained here.

Detour v1.5.5

Detour is a system preference pane, allowing application-independent audio routing, with independent volume control, to any sound devices enabled on a PPC Macintosh. Development of Detour has been discontinued and the final version is 1.5.5, which was released in 2005. All of the application-redirects and most other features can be accessed from the menu-bar Detour menu.

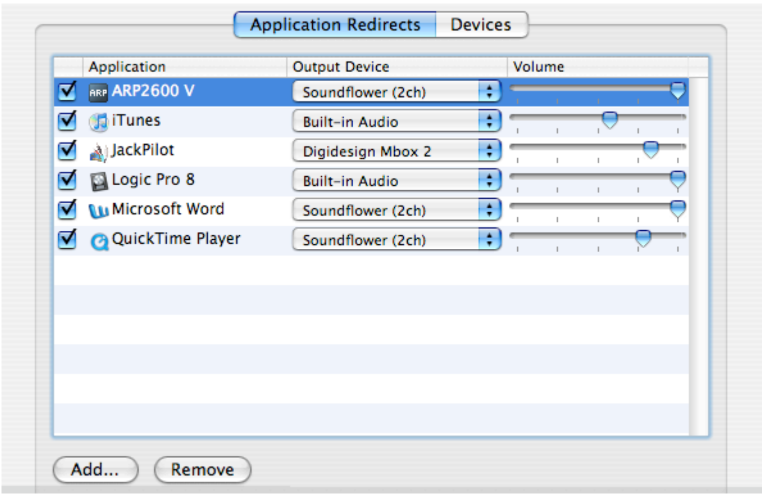


Fig. 24 The Detour Applications Redirects window (Rogue Amoeba, 2005)

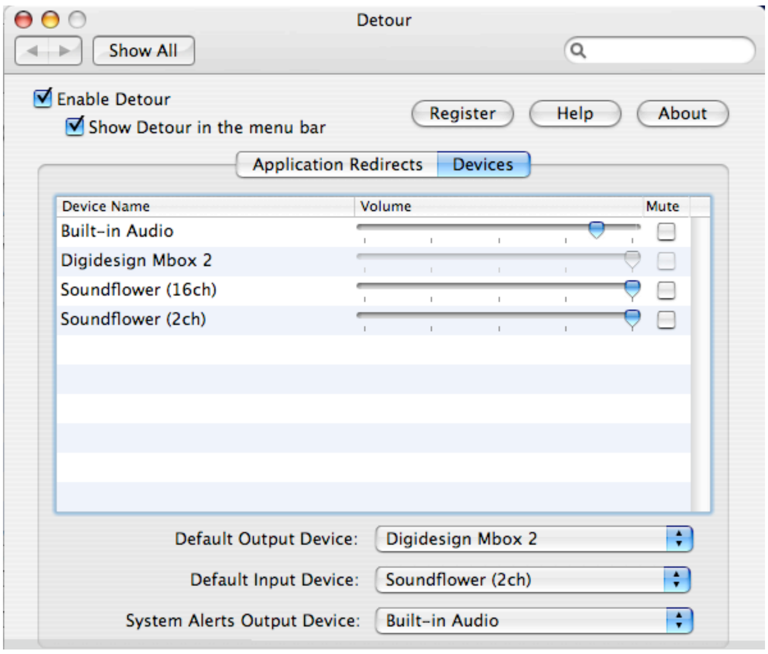


Fig. 25 The Detour devices window (Rogue Amoeba, 2005)

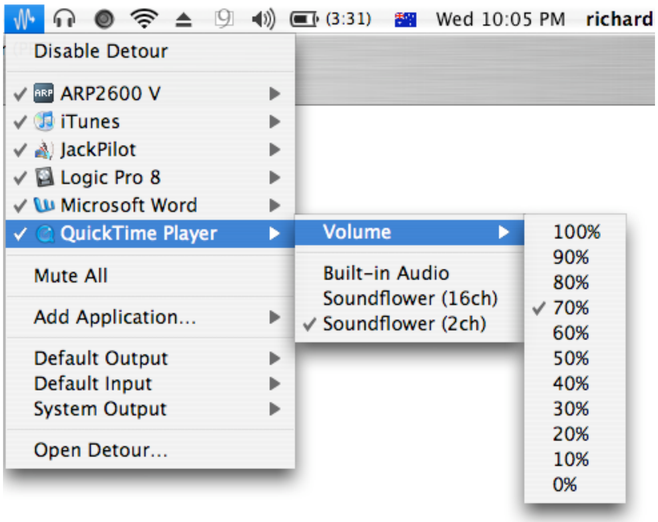


Fig. 26 The Detour menu showing QuickTime set to Soundflower with a volume of 70% (Rogue Amoeba, 2005)

SoundSource

Sound Source is an audio utility allowing menu-bar access to control selection of the input, output and system audio devices.

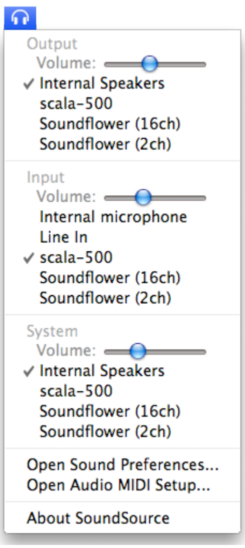


Fig. 27 The SoundSource menu (Rogue Amoeba, 2008). This is the Leopard version; volume sliders do not appear on the Tiger version.

A limitation of most Macintosh models is the inability to isolate sound output between the headphone (audio out) jack and the internal loudspeaker. The Mac Pro does offer separate output control and SoundSource features an ‘auto-switch to headphones’ menu item to mimic the way other models switch the internal speaker off if the headphone jack is in use (Rogue Amoeba, 2008).

Sound Menu

An alternative to SoundSource, this is another menu-bar utility to allow access to the sound devices. The features are similar, but direct access to the AMS is not provided. It does have an indication of mute being on which is shown as a cross through its menu-bar speaker icon.

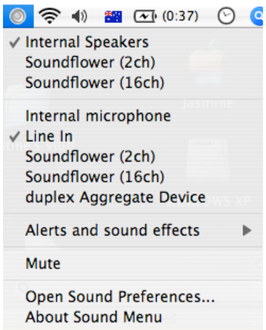


Fig. 28 The Sound Menu GUI (Aspirine, 2008)

Line In

This utility allows any input audio to be directly routed to any output device. The advanced tab brings up a second window where input buffer size can be set between 32 and 6144 samples, and the output buffer can be set between 384 and 262144 samples. The minimum output buffer size automatically sets to twice the input buffer size. Tests on a MacBook revealed the minimum output buffer size allowed was 512 samples, which gives a latency of 11.6 mS. Below these values the thru function was muted.

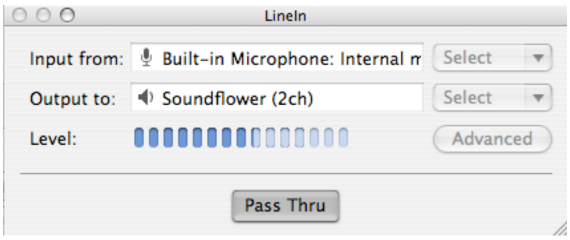


Fig. 29 The Line In GUI (Rogue Amoeba, 2008)

PTHVolume

This is a system preference pane, which allows individual volume control for each audio device from the menu bar. It requires OS 10.5. Custom keyboard shortcuts can be set individually for any device that allows volume control.

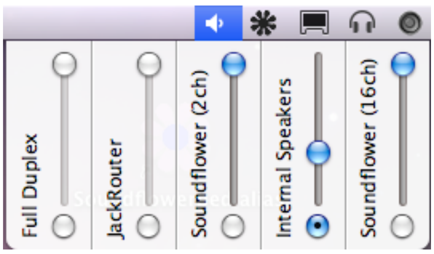


Fig. 30 The GUI for PTHVolume (PTH Consultants, 2008). Faders can be shown either vertically or horizontally.

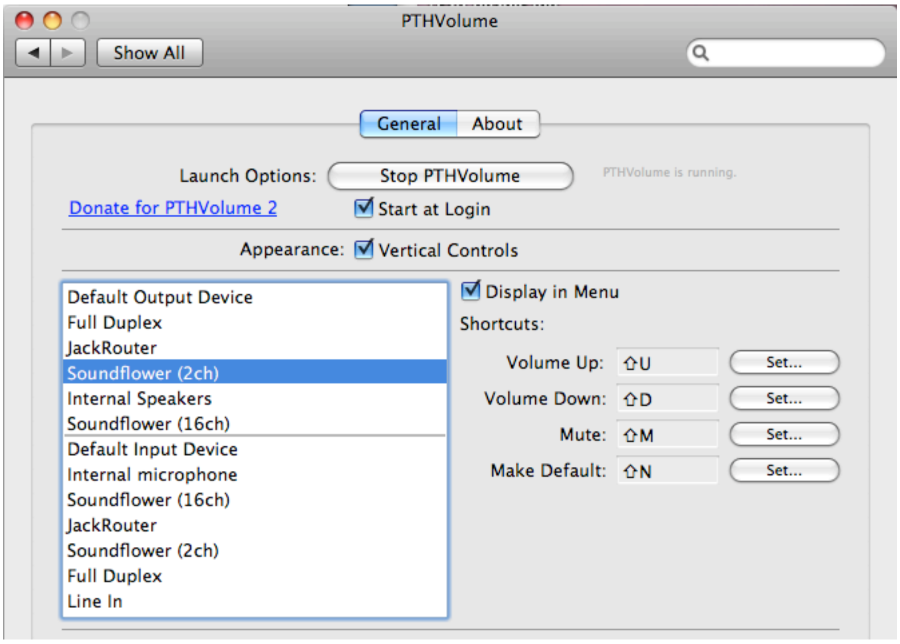


Fig. 31 The preference pane for PTHVolume showing keyboard shortcuts set for Soundflower (PTH Consultants, 2008).

Distributed Audio Processing

VST System Link is a distributed computing system, which allows audio and MIDI to be transferred in real-time between a number of computers. It was introduced in 2002 and is proprietary software of Steinberg so it requires Cubase or Nuendo as the host. The LSB of the 24 bit audio word is used to maintain sample accurate synchronisation. “A typical configuration might include a keyboardist with many virtual synths operating on one computer that does not affect the mixing engineer's computer running with many VST effects and plug-ins.” (Rudolph, 2003).
Apple’s Logic Audio Node is a similar system for distributed audio processing using Ethernet and the built-in networking capabilities of Mac OS X.
With the increased processing power available with the Intel CPUs it will be unnecessary for many users to require extra processing on remote machines. The ability to freeze tracks has also reduced CPU usage of virtual instruments to acceptable levels for a lot of situations. Neither of these systems was tested as this paper is concerned with audio communication on a single computer.

AUNetSend v1.4.0 and AUNetReceive v1.4.1

These two AU plugins work together and are also intended to transfer audio between two computers using the Bonjour protocol to communicate over Ethernet. It is also possible to transfer 2-channel audio between applications on a single machine. I was able to successfully stream audio from Logic into Audio HiJack Pro.

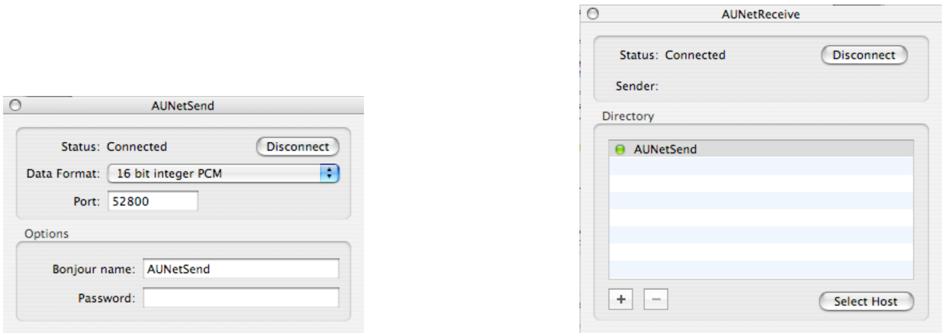


Fig. 32 *The two AU Network windows (Apple, 2008)*

I also tested routing audio from a Logic audio track via a channel insert and back into the input of an audio instrument channel (and from there to mix out). This worked without any problems with the AUNetReceive automatically connecting when it senses audio playing.

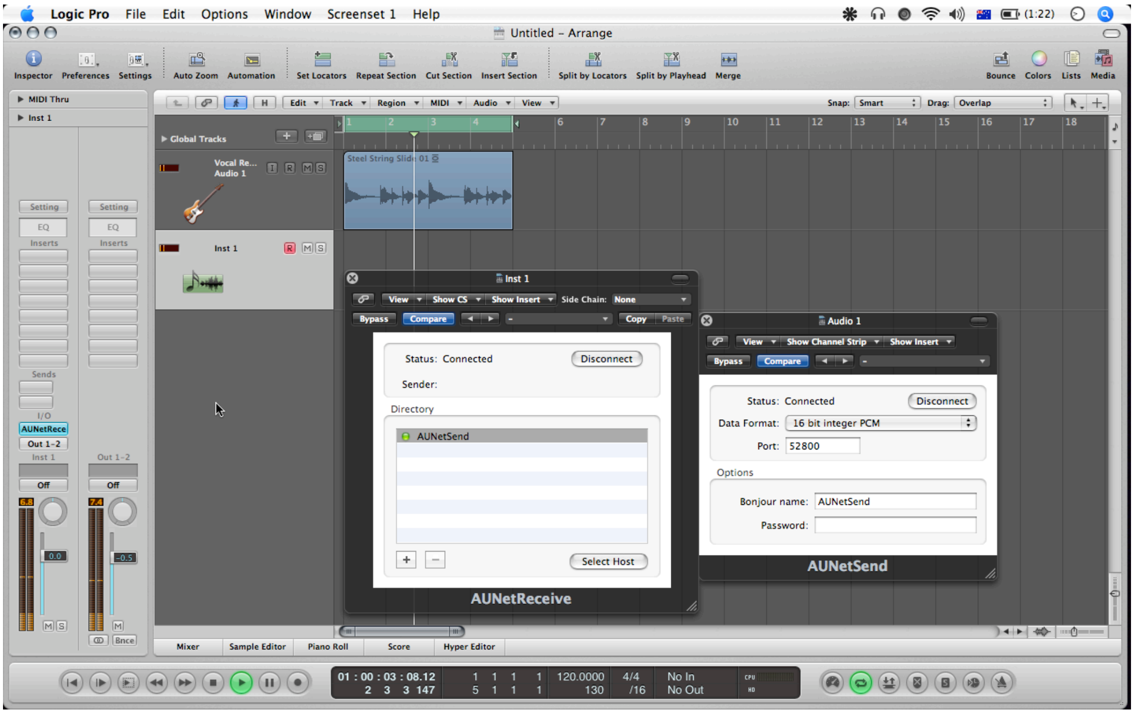


Fig. 33 *The setup for testing the AUNetwork plugins in Logic*

By bussing the incoming audio stream onto a new audio track I was able to check the latency which was found to be 0 samples. The AUNetSend plugin allows the user to set the data format. 16 bit, 24 bit and 32 bit floating integer are presented as well as AAC (32 to 128kb/s), µ-Law, and IMA 4:1 data compression.

DLSMusicDevice v1.4.0

This is an Apple AU plugin, which allows a MIDI sound device to be inserted into an audio instrument channel of a sequencer. By default this will be the QuickTime Music Synthesiser output but any Soundfont2 device can be selected if it resides in the Library/Audio/Sounds/Banks folder. The DLSMusicDevice will recognise most of the GM2 MIDI commands including velocity, pitchbend, volume and bank select, reverb, chorus, and program change (Shaffer, Rosenzweig, 2004). I found it worked well for a small number of tracks but when tried with 16 tracks of MIDI the audio glitched badly (testing was on a 2GHz MacBook).

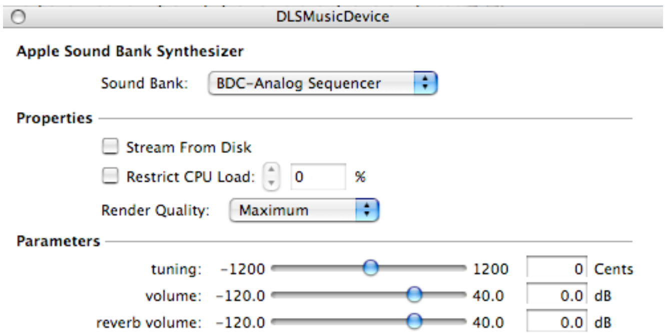


Fig. 34 The DLSMusicDevice edit window (Apple, 2008)

Digidesign CoreAudio Driver

This allows any CoreAudio enabled application to access Pro Tools hardware, such as the Mbox, or Digi 002. A separate driver is required for the Mbox 2 series. It provides for full duplex recording and playback of audio up to 24 bit and 96kHz. Up to 18 channels of I/O are possible with a Digi 002. It is also possible to use it on a TDM Pro tools system but only the first 8 channels of I/O are available. Buffer sizes can be set from 128 to 2048 samples, depending on the Digidesign hardware used.

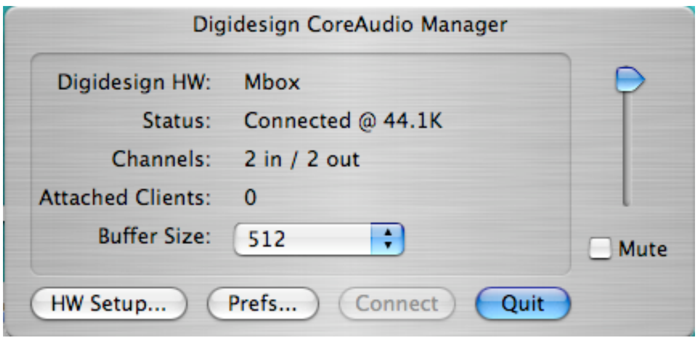


Fig. 35 The Digidesign CoreAudio Manager (Digidesign, 2008). On a MacBook or Intel iMac the buffer size can be set to either 512 or 1024 samples.

The correct order of connection is important when using the Digidesign CoreAudio Driver. It is:

1) Start Digi CoreAudio Manager
2) Select "Digidesign HW" in Sound Prefs or AMS
   (if Sound Prefs is already running; quit and reopen it)
3) Start the audio application, open an audio file, and play or record