MOBILE I/O

ULN-2
User's Guide

METRIC HALO

Metric Halo Distribution, Inc.
M/S 601 • Building 8
Castle Point, NY 12511-0601
phone: (845) 831–8600
fax: (603) 250–2451
tech support: (845) 831–8600 x 203
tech support email: support@mhlabs.com
www.mhlabs.com

Mobile I/O ULN-2 User's Guide.

Printed in the USA.

Mobile I/O User's Guide – Table of Contents

ULN-2: Introduction and Overview

Thank you for purchasing a Mobile I/O ULN-2™, the ultimate FireWire®–based professional audio interface. Your ULN-2 provides an array of functions that allow you to record and mix with unprecedented quality – Anywhere, Anytime.

What it is

ULN-2 is the result of a dream to create a piece of audio gear that provides unbelievable audio quality while at the same time offering a degree of mobility and convenience that until very recently was simply not possible. The successful integration of world-class analog stages, excellent A/D/A conversion, and the amazing digital mixing, routing and FireWire connectivity that has already made the Mobile I/O line famous, places the ULN-2 in a unique position among computer audio interfaces.

ULN-2 is a portable, high–quality, modular FireWire–based multi-format audio converter, interface, and processor for professional audio applications. The ULN-2 is equipped with two balanced analog inputs on Neutrik™ combo connectors, two channels of Digital I/O (AES/EBU and S/PDIF), two balanced analog outputs (1/4" TRS), two balanced monitor outputs for connecting directly to power amps and self powered monitors, as well as word-clock in/out and 2 IEEE 1394 FireWire connectors that support 400 Mbs operation. All inputs and outputs are capable of 24-bit/ 96kHz operation.

What it has

- 4 simultaneous input channels and 6 simultaneous output channels
- full 24 bit/96kHz audio
- 2 independent channels of high gain, low-noise mic-pre with switchable phantom power
- Fully Portable Capabilities – Bus and Battery Powerable
- Rack Mount Kit
- 44.1, 48, 88.2, 96kHz Sampling Rates
- 24 bit 110 dB Dynamic Range A/D converters
- 24 bit 120 dB Dynamic Range D/A converters
- Selectable stereo Digital Inputs (AES/EBU or S/PDIF)

- Stereo Digital Outputs (AES/EBU and S/PDIF)
- Sample Rate Conversion (SRC) on Digital I/O
- Built-in 80-bit, fully interpolated, multi-bus mixer for near-zero latency foldback of all input channels and all DAW busses simultaneously
- Full cross point router for I/O management
- Word Clock 1x, 256x
- Front Panel Metering for Analog Inputs and main Outputs
- Full console metering of every channel and mix bus
- Total recall of every console parameter

What you need to use it

- Computer:
  - a Power-PC based Macintosh with a FireWire Port and OS 9.1 or newer (G4, Powerbook G3 or G4 recommended). Mac OS 9.2.2 and Mac OS X recommended.
  - 128MB of RAM
  - a monitor that supports 1024x768 resolution or better
- Peripheral FireWire Adaptor:
  - OHCI compliant PC-Card or
  - OHCI compliant PCI card
- Software:
  - an ASIO 2 or CoreAudio compatible host (such as Cubase, Nuendo, Logic Audio, Digital Performer, Deck or Peak)
  - for ASIO direct monitoring-based foldback and overdubbing, an application that supports ASIO direct monitoring

What comes with it

Your ULN-2 package contains the following items:

- One ULN-2 unit

Figure 1: ULN-2 Unit

- One IEC Power Cord appropriate for your area

Figure 2: IEC Power Cord

- One 24-volt 48-watt world-ready external power supply

Figure 3: External Power Supply

- One 0.5 meter IEEE 1394 6-pin FireWire Cable

Figure 4: 0.5 meter 6-pin 1394 cable

- One 4.5 meter IEEE 1394 6-pin FireWire Cable

Figure 5: 4.5 meter 6-pin 1394 cable

- Two Rack Ears w/ fasteners

Figure 6: Rack Ears

- MIO Software CD-ROM
- This ULN-2 Users Manual
- Warranty/Registration Card

If any of these items are missing from your package when you open it, please contact Metric Halo or your dealer immediately for assistance.

Acknowledgements

The Mobile I/O product line is the result of a tremendous amount of work to deliver a world-changing product. ULN-2 would not be what it is if not for the efforts of a brave few who worked tirelessly and selflessly to test Mobile I/O. We especially want to thank the folks who did focused beta testing on the early releases of the hardware and software and found and helped to resolve a number of nasty issues:

Ed Abbott, Philip J. Harvey, Tom Cowland, Orren Merton, Mo Jen, Jon McBride, Hiro Honshuku, Henry Robinette, David Walters, John Leonard, Brian Peters, David Das, Darren Gibbs, David Morrison, Mark Ernestus, Craig Shepard, Al Evans, Steven Campbell Hilmy, Daniel Courville, Jeff Taylor, Matt Mora, Chris Frymire, Lewis Chiodo, Sean Witters, Erlend Myrstad, Floris van Manen, and V. Lewis.

Support Information

Warranty

The ULN-2 hardware is covered by a manufacturer's warranty against manufacturing defects. The details of this warranty are described in a separate enclosed document. This warranty is applicable to products purchased in the USA. Products purchased in other regions are covered by a warranty administered by the distributor for that region and the terms will be as set forth in a separately enclosed document.

Registration

In order to receive warranty service, you must register the product with Metric Halo. This may be done at any time with proof-of-purchase, but we strongly recommend that you register with Metric Halo as soon as you purchase your unit. There are a couple of practical reasons for this:

1. Your product will be registered with us and this registration can be used as proof of ownership if your product is ever lost or stolen.
2. Metric Halo aggressively updates ULN-2 on a regular basis and we will keep you informed of updates as they become available.

In order to register your ULN-2 you can send in the enclosed registration card via mail or fax:

Mail Address:

ULN-2 Registration
Metric Halo
Building 8 M/S 601
Castle Point, NY 12511 USA

Fax Number:

+1 (603) 250–2451

Alternatively, if you have internet access you can use our automated registration webpage at:

http://www.mhlabs.com/mio/register

Service and Support

If you have problems configuring or using your ULN-2 and you need help, please contact our support group. We offer support via email, as well as peer support on the Mobile I/O Users Group mailing list.

For email support, send email to:

support@mhlabs.com

To subscribe to the ULN-2 Users Group mailing list or to peruse the archives go to:

http://mail.music.vt.edu/mailman/listinfo/mobileio

Finally, you can always find the latest info and updates for ULN-2 at the Metric Halo website:

http://www.mhlabs.com/mio

Safety Compliance

This equipment has been tested and found to comply with the limits for a class B digital device pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio or television equipment reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by any combination of the following measures:

- Relocate or reorient the receiving antenna
- Increase the separation between the equipment and the receiver
- Plug the equipment into an outlet on a circuit different from that to which the receiver is connected
- Consult your dealer or experienced radio/television technician for additional assistance.

WARNING: Changes or modifications to this unit not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

Using the ULN-2 Hardware

ULN-2 Front Panel

Figure 7: ULN-2 Front Panel

The ULN-2 front panel provides ten-segment metering for the 2 analog inputs and the main outputs as well as knobs and switches to control the input, monitor and headphones sections. The meters are fast PPM peak reading meters with auto-resetting peak holds.

Each input channel has the following controls:

- Input gain knob
  - This is a 12 position gold-contact rotary switch which allows you to control the gain of the selcted input.
- Phantom Power enable switch
  - This is a push-button switch which enables/disables Phantom power. Push the switch IN to enable phantom power.
- Trim Enable switch
  - This is a push-button switch which allows you to control whether the attenuator trim pot is in the signal path or not. Push the switch IN to enable the trim pot. The attenuation range of the trim pot is -2dB to -20dB.
- Mic/TRS switch
  - This is a push-button switch which selects the input stage. The ULN-2 has two distinct input stages: The Mic Amp and the DI Amp.
  - The Mic Amp is optimized for high gain and very low noise with low impedance sources like microphones. This input is connected to the XLR portion of the Neutrik combo connector. Maximum gain is 72 dB. Push the Mic/TRS switch IN to select the MIC input.
  - The DI amp is optimized for high impedence sources like magnetic pick-ups. This input is connected to the TRS portion of the Neutrik

combo connector. Maximum gain is 63 dB. The Mic/TRS switch should be in the OUT position to use this input.

- Trim Pot
  - The trim pot controls a passive attenuator. The attenuator is buffered between the return receiver and the A/D converter so its operation is transparent with regard to sound quality. Push the the trim enable switch IN to enable the trim pot. The attenuation range of the trim pot is -2dB to -20dB.

The front panel also provides ULN-2 system status at a glance:

- Sample Rate (nominal 44.1, 48, 88.2, or 96)
  - The Sample Rate is determined from the currently selected clock source so it will accurately indicate the current sample rate, even when the clock source is being provided by an external device.
- Clock source:
  - **Internal** indicates that the system is internally clocked
  - **Wordclock** indicates that system is being clocked from the wordclock input
  - **256x WC** indicates that the system is being clocked from a 256x clock at the wordclock input
  - **Digital In** indicates that the system is being clocked from the selected digital input (AES or S/PDIF)
  - **256x WC + Digital In** indicates that the system is being clocked from the ADAT optical input
- Power:
  - Indicates that the ULN-2 is receiving power.
- FireWire
  - Indicates that the ULN-2 has been successfully connected to a FireWire bus and has detected the isochronous cycle required to transmit and receive audio.
- Locked
  - Indicates that the system clock recovery circuit is properly locked to the selected clock source. If this light is not illuminated, the ULN-2 will not be locked to a clock and will revert to its failsafe internal clock source. Even if

the Locked light is not illuminated, the actual sample rate will still be indicated on the front panel display.

- Digital I/O Section:
  - The AES and S/PDIF lights are mutually exclusive and indicate which of the the two input ports are feeding the Stereo Digital input of ULN-2. The Locked light indicates when the digital receiver is locked to the incoming digital audio signal.

The ULN-2 front panel provides access to the level control knobs for headphones and for the monitor outs. The headphone output jack is on the front panel and the monitor output jacks are located on the back of the unit.

The headphone output jack is a TRS 1/4″ jack that provides the Left Channel on the tip, the Right Channel on the ring and the ground return for the two channels on the sleeve. The Monitor output jacks are balanced TRS connectors.

ULN-2 Rear Panel

Figure 8: ULN-2 Rear Panel

The ULN-2 rear panel features:

- 2 channels balanced MIC/LINE/INSTRUMENT inputs on Neutrik Combo connectors. Each input has:
  - 24-bit 96kHz A/D converters (110dB SNR)
  - high gain, low noise Mic amps with up to 72 dB of gain (fed by the XLR connector)
  - high gain, low noise DI amps with up to 63 dB of gain (fed by the TRS connector)
  - switchable input impedance characteristics (Mic input 3.3k Ohms, DI input 200k Ohms)
  - switchable 48V Phantom power (on XLR connector)
  - balanced analog inserts (S1, S2, R1, R2 jacks) which are post preamp but pre A/D. You can use the inserts to patch in analog processing between the preamp and the A/D converter. The send jack can also be used to send a mult of the input signal to another device while still using the A/D section of the ULN-2. This allows the ULN-2 to be used as an active mic splitter.
- 2 channels balanced TRS main outputs. Each output has:
  - 24-bit 96kHz D/A converters (120dB SNR)
  - switchable +4/-10 level
- 2 Channels balanced Monitor output with front panel level control
  - Connect these outputs directly to power amps or self powered monitors
- 4-pin XLR power port for use with broadcast batteries
  - Compatible with any 4-pin XLR power system with the following characteristics: 9v - 30v DC, Pin 4 Hot, 15 Watts
- Wordclock input/output on BNC connectors
- 256x Wordclock input/output on BNC connectors
- Stereo S/PDIF input/output on RCA connectors

- Stereo AES/EBU input/output on XLR connectors
- 2 IEEE 1394 (FireWire) ports (400 Mbs)
- 1 2.1mm DC power jack (9v - 30v, center positive, 15 Watts)

Making connections to the ULN-2

There are five classes of connections you can make to the ULN-2 hardware:

1. Analog Audio
2. Copper-based Digital Audio
3. Clock Sync
4. FireWire
5. Power

Analog Audio Connections

The analog I/O connections on the ULN-2 have been engineered for maximum flexibility in that they support both balanced and unbalanced connections with a wide range of input and output levels and a wide range of matching impedances. This means that ULN-2 handles sources from mic level to line level and from mic impedance to guitar impedance. With that in mind, there are a number of aspects of the design that you should take into account when interfacing with ULN-2.

There are really three distinct analog input stages available in a ULN-2 input:

1. The Mic amp, which is fed by the XLR portion of the Combo connector.
2. The DI amp which is fed by the TRS portion of the combo connector.
3. The TRS return jack. This is a line level input which is the shortest path to the A/D converter.

Each input path is optimized for specific sources, but each is capable of handling a wide variety of sources. For example, both the Mic amp and the DI amp are capable of receiving Line level inputs. Additionally the DI input is capable of 63 dB of gain and can be used with dynamic microphones (phantom power is only available with the Mic Amp).

Feel free to experiment with the different input paths and choose the one which works best for a given application.

Whenever possible, use balanced interconnects with ULN-2. The performance of balanced interconnects is much higher and much more resistant to noise interference and electrical (power) wiring problems. The expense of balanced interconnects is not substantially higher than unbalanced connections, so if the gear that you are interfacing with supports balanced connection — use it.

If you cannot utilize balanced interconnects, there are connection schemes that you can use that will maximize performance.

On input, at line level, it is sufficient to simply use standard unbalanced (TS) connections. If you are interfacing with the ULN-2 XLR inputs, you will need to ensure that pin 3 is grounded in the unbalanced adapter cable. The ULN-2 XLR inputs are all wired pin 2 hot and the 1/4” inputs are wired Tip hot.

TIP: To use the ULN-2 TRS input with guitar or bass, you can simply use a standard TS guitar cable (patch cord) and it will work fine. However, you can take advantage of the balanced input design of the ULN-2 to get more noise rejection than you thought possible on a guitar input. In order to do this, you will need to make a psuedo-balanced telescoping shield guitar cable. This can be constructed with a TRS connector, a TS connector and balanced microphone cable. This cable will treat the guitar as a floating balanced source and provide a telescoping shield from the ULN-2 ground.

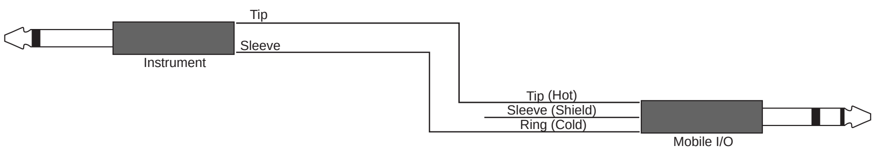

Figure 9: Telescoping Shield Cable for Instruments

If you want to use the TRS inputs with balanced microphones, you will need an XLR female to 1/4" TRS balanced plug adapter cable. These are available commercially, or you can construct one easily. The connections are Tip to Pin 2, Ring to Pin 3 and Sleeve to Pin 1:

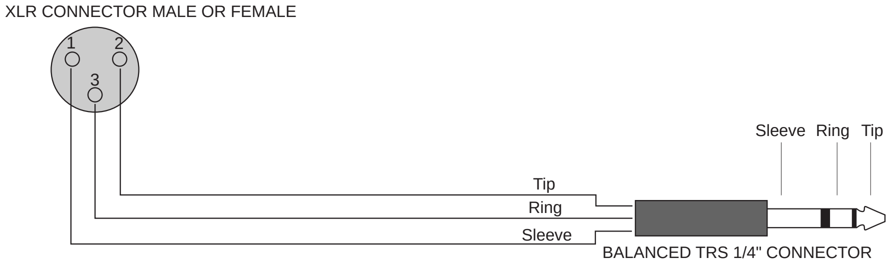

Figure 10: XLR to Balanced TRS Cable

On output, the situation is a bit more complex. If you are driving an unbalanced load, you will get the best performance by not connecting the ring of the TRS jack to ground. In order to do this, you can simply use a balanced TRS/TRS connector with the unbalanced gear. You can also construct a special cable with a TRS connector and a TS connector. In this cable, you just let the ring of the TRS connector float:

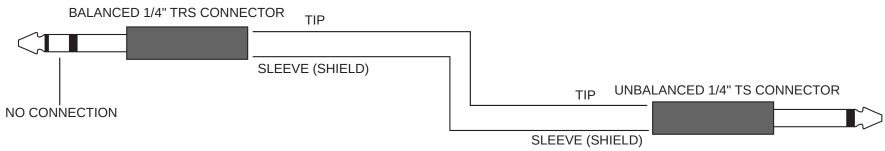

Figure 11: TRS to TS unbalanced cable

Alternatively, the TS connector can be replaced with an RCA connector for interfacing with gear that has RCA unbalanced interconnects.

Making the 1/4" connection

When you connect a 1/4" plug to a ULN-2 jack, insert it straight and firmly, ensuring that the plug is fully inserted into the jack. If the plug is not fully inserted you will get level shifts, phase flips, distortion, or no sound.

To disconnect a 1/4″ plug, firmly pull the plug straight out from the connector body. The connectors on ULN-2 are stiff, so you may have to exert some force to remove the plug.

Making the XLR connection

When you connect a Male XLR plug to a ULN-2 jack, ensure that you have aligned the pins with the connector body and insert firmly until the retention tab clicks.

To disconnect the plug, press the metal retention tab flush against the box, and pull the plug from the ULN-2.

Copper-based Digital Audio

ULN-2 supports 2 channels of digital audio over copper-based connections. These connections can be made using either S/PDIF interconnects with the RCA connectors or with AES interconnects using the XLR connectors. Even though only one of the AES or S/PDIF inputs can be active at any given time, you can have different digital sources connected to each of the input connectors at the same time – you use the MIO Console application to select the active input. Audio routed to the digital outputs will be mirrored by both S/PDIF and AES outputs. This allows you to send the same stereo pair to two devices at once.

We recommend that you use the AES interconnect mechanism to establish the digital communication between the ULN-2 and other digital devices. The jitter and electrical noise tolerance on AES interconnects is substantially better than with S/PDIF interconnects. The AES interconnect standard is equivalent to balanced audio interconnections. If you need to use S/PDIF interconnects, try to use the shortest cables you can and, if possible, use special purpose 75 ohm S/PDIF or video cables.

The RCA connectors used for S/PDIF are friction fit coaxial connectors. When you connect them, ensure that they are fully inserted and tight.

The XLR connectors used for AES are fully locking. When connecting to them, make sure that you align the pins and insert firmly. When you remove the connector, make sure that you release the lock by pressing the lock release button before you pull the connector out of the ULN-2.

Integrated SRC

Normally, when working with digital audio transport, you must take care to ensure that all devices communicating with one another are synchronized to the same audio clock. While this is still an important consideration with ULN-2, the hardware provides a special feature to simplify copper-based digital connections to the box. The digital input on ULN-2 has an optional asynchronous sample rate converter (SRC) that will automatically match the sample rate of the incoming audio to the sample rate of the ULN-2. This converter is enabled by default and you can disable it in the System section of the MIO Console. If you have synchronized the ULN-2 to the external source (using any of the extensive synchronization methods provided by ULN-2), you will generally want to disable the SRC in order to get 24-bit transparent audio transport over the digital input.

Clock Sync

Clock sync is a serious consideration in any digital audio system.

If you are recording analog sources with ULN-2, you can simply use the unit's high-quality internal clock source to drive the converters. This is the easiest case to deal with.

If you need to interface with other devices digitally or ensure sample accurate sync with video sources, the extensive clock synchronization capabilities of ULN-2 will prove to be more reliable (and better sounding) than most higher priced alternatives.

There are three different ways to get external clock information into the unit:

1. Sending a 1x word clock signal into the WC Input BNC.
2. Sending a 256x word clock signal into the WC Input BNC.
3. Sending an AES or S/PDIF signal into the Digital input.

The BNC word clock input port is a 75 Ohm terminated coaxial input. It should be driven by a 75 Ohm source driver and interconnected with 75 Ohm coaxial cable. If you do not use proper cabling and source drive, you will introduce reflections on the word clock cable which will propagate jitter into the recovered word clock. This is true whether you use the port as a

1x WC input or a 256x WC input, but becomes more important when the clock signal is 256x.

1x is generally appropriate for use with devices that provide a word clock output. If your device provides a 256x output, you may find that you get better results using that clock signal. The Digidesign® line of Pro Tools® products use 256x as their "Superclock™" clocking signal.

The AES recommended procedure for distributing clock is to use an AES clock signal. The AES clock signal is an AES digital audio signal with no audio activity. ULN-2 only uses the AES preambles for clock recovery, so it is immune to data dependent jitter effects. This means you can reliably use the Digital Input as a clock source with or without audio data.

FireWire

FireWire® is Apple's registered trademark for the IEEE 1394 High-Speed Serial Bus. FireWire started as an Apple technology to replace a variety of interface ports on the back of the computer. After promulgating a number of closed proprietary technologies in the early days of the Macintosh, Apple determined that open standards were better for the Mac, for the industry, and for Apple itself. On that basis they opened their technology for standardization under the auspices of the Institute of Electrical and Electronics Engineers, Inc. (IEEE), an international organization that promotes standards in the field of electronics. FireWire was standardized as IEEE 1394 and promoted for open licensing in the industry.

The first widespread adoption of the technology was for DV camcorders where space was at a premium and bus powering was not percieved as a real issue since all camcorders have batteries. Sony designed an alternative version of the standard 6-pin FireWire connector that provided 1394-based communication with 4-pins in a much smaller form-factor. This version of the connector sacrificed bus-power support and mechanical stability for reduced space requirements. Sony dubbed this version of IEEE 1394 "i.Link®." This became the de facto standard in the DV world, and was later added to the IEEE 1394 standard. Both i.Link and FireWire refer to the same underlying standard and are completely interoperable. Obviously, i.Link connectors and FireWire connectors cannot be used together without adapters.

ULN-2 uses the FireWire flavor of the IEEE1394 connector with 6-pins for bus power support. The unit ships with two 6-pin to 6-pin FireWire cables, one that is 0.5 meters long (about 18 inches), and the other 4.5m (about 14.5 feet) long. If you want to use ULN-2 with a 4-pin FireWire device, you will need to purchase a 6-pin to 4-pin adapter cable. These cables are available from a wide variety of retail sources. If you are using a 4-pin cable to connect any device to the computer with ULN-2, bus power will not be available.

The 6-pin FireWire connector is polarized by its shape, one end of the connector is pointed. The FireWire ports on ULN-2 point downwards toward the bottom of the box. It will be very difficult to insert the connector upside down, but it is possible if you force it. If the plug is inserted into the socket upside down, the socket will be destroyed. **NEVER FORCE A FIREWIRE CONNECTOR INTO A FIREWIRE SOCKET.**

Devices connected to the FireWire bus are autoconfiguring. You do not need to set IDs or DIP switches or in any way configure the devices in order to facilitate communication between devices or to configure of the bus.

FireWire devices on the same bus must be connected in a tree structure with no loops. This means that devices can be connected to each other in any order, and any device with multiple ports can act as a chain or a hub for other FireWire devices, but you should never be able to get from one device to another by more than one path. If you construct a loop in the bus, it will not operate properly and you will not be able access some or all of the devices on the bus.

Although you are able to attach devices in any order on the FireWire bus, the order of attachment will have an impact on performance. Most current model FireWire devices support 400 Mbs operation, but many older devices may only support 100 or 200 Mbs operation. These devices act as a bottleneck in the bus and limit the speed of any bus traffic that flows through them. In order to maximize performance, you want to ensure that low speed devices are not used to join high speed devices. In practice this generally means that you should attach your ULN-2 directly to your computer or through a high speed hub.

To connect ULN-2 to your computer simply plug a FireWire cable into the ULN-2 and into the computer. The FireWire bus provides a path for all communication between the computer and ULN-2 – audio, control and meter data.

ULN-2 audio transport takes advantage of FireWire's support for isochronous transmission, in which the ULN-2 can reserve a dedicated amount of bandwidth on the bus for moving audio samples. Since the audio **must** be transmitted on a regular basis to ensure continuous playback and recording , the isochronous model is perfect.

Control changes and meter data are transmitted using asynchronous transactions on the FireWire bus. This transmission approach makes use of the unreserved bandwidth on the bus and competes with things like FireWire hard disk accesses for time. Under normal circumstances this is completely transparent to the user. If the bus becomes overloaded, you may find that disk accesses and meter updates slow down. If you are experiencing bus overloads, you can always add a second FireWire bus with a third-party FireWire card (PC-Card or PCI card depending on your machine), and offload one or more devices to the second bus.

Power

One of ULN-2's great strengths is the flexibility of its power system. ULN-2 can be powered from any DC source (including bus power) in the range of 9V to 30V as long as it provides 12 Watts of power. The DC inputs on ULN-2 are a 2.1mm coaxial power connector, center positive and a 4-pin XLR connector Pin 4 Hot. So if you are powering the unit with a third party power source and it supplies 9V, the power source will have to provide 1.4 amps of current. If you are powering the unit with 12V, the power source will have to provide 1 amp of current, and so on.

The ULN-2 ships with a world-ready 24 volt, 2 amp power supply. You can plug this supply into any AC power source from 90V to 240V, 50Hz - 60Hz, using an appropriate IEC power cord, and it will supply the proper power to the ULN-2 on the 2.1mm coaxial power connector. ULN-2 will automatically supply the extra power to the FireWire bus. This means that the ULN-2 and its power supply can be used to power other bus-powerable FireWire devices including hard-drives, hubs, and other ULN-2 units.

Since ULN-2 is DC powered, you can also power up the ULN-2 using the FireWire bus or another DC source. The ULN-2 uses 12 Watts of power, so the device supplying the bus power must be capable of sourcing that much power. Most desktop Macs provide more than enough power for ULN-2 and one other low power device. Most laptops provide enough power for ULN-2, but not enough for ULN-2 and another bus-powered device at the same time. If you are using a Powerbook computer, you should not expect to be able to power both the ULN-2 and a hard drive from the computer. The power capabilities of individual computers vary, so you will have to test the complete system to determine exactly how much your computer can handle.

If you find that the computer is not capable of powering ULN-2 or does not provide enough run time, you may want to explore using an external power source with the ULN-2. Check with Metric Halo for details on different battery power solutions for ULN-2.

As with all electronic devices, when connecting an external power source to the ULN-2, you should first connect the power source to ULN-2 while it is in an unenergized state (e.g. not connected to the mains or switched off). After the connection to ULN-2 has been made, you should energize the power source.

If you connect an energized power source to the ULN-2's 2.1mm power connector you may see a small spark when you make the connection. This is due to surge current and is normal if you connect a power source in this way. While this will not damage the ULN-2 in any way, to avoid the spark just connect the power connector to ULN-2 before connecting the power source to the wall.

MIO Console

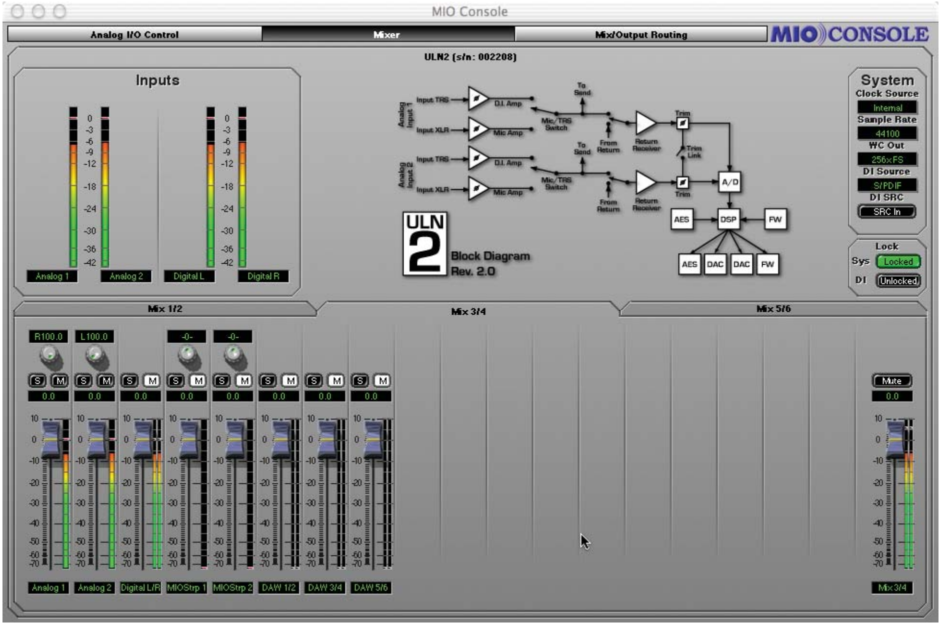

Figure 12: MIO Console

MIO Console is the nerve center of your ULN-2. Functioning as a standalone application in both MAC OS 9 and MAC OS X, MIO Console provides full control of every aspect of ULN-2. The console software allows you to rapidly and easily adjust the system sample rate, Digital I/O source, and system clock source. It also allows you to assign ASIO/CoreAudio output channels and hardware input channels to the integrated 80-bit, fully-interpolated, multi-bus, near zero-latency hardware mixer.

MIO Console's powerful mixer supports both mono and stereo busses, with solo and mute functions for all input and master channels. The mixer bus outputs are routable to any of the hardware outputs, allowing you to easily create multiple simultaneous mixes for send/return busses and multiple live main and monitor mixes. Various applications include foldback support for multiple performers, separate monitor feeds for studio, tape, and control room, and separate mixes for front of house, archive recording, and monitors for live shows.

MIO Console also contains a patchbay router, which allows you to quickly select the source being fed to any output. The patchbay provides easy configuration of stand-alone operation, mix mults, direct outs and various combinations thereof to suit the needs of the moment.

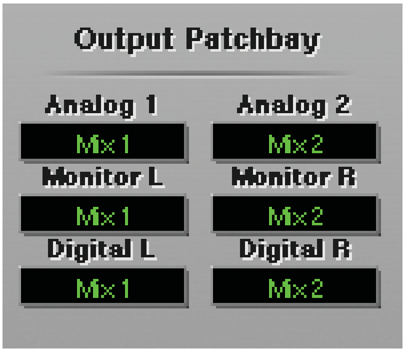

Figure 13: MIO Console Output Patchbay

In order to simplify work flow and optimize the extent of system control, MIO Console supports comprehensive preset management on both a global and individual control level.

The preset management popup controls within MIO Console allow you to configure various aspects of ULN-2 and save that configuration information for later recall. Various applications include storing routing configurations for monitor setups, mixer configurations for stem and scene recall, and storing analog level standards for interfacing with external gear and managing different mastering standards.

Global configuration snapshots allow you to save each and every aspect of ULN-2's configuration for later, total instant recall. This is useful for preconfiguring ULN-2 and bringing back the configuration once at the gig, managing separate location setups, or for saving complex studio routing setups for quick changeover.

MIO Console Overview

The MIO Console application consolidates all of the controls for the Mobile I/O hardware into one easy to use window. The ULN-2 has an extremely large number of user configurable parameters and it is very important that you have instant access to the ones that you need. The console provides a thinly layered interface to the entire system and keeps you from having to deal with "Window Overload".

The MIO Console window has a **view panel selector bar** that runs along the top of the window. This bar indicates which of the console view panels is currently active. You can tell which panel is active because the button in the bar is "pushed in." To switch to one of the other panels, simply click on the name of the panel you want to use. The view will change instantly to the one that you have selected.

Figure 14: View Panel Pane Selector Bar

Under the **view panel selector bar** is the currently selected view panel. You control the various aspects of the box with the controls in each view. The MIO Console has three main panels:

1. Analog I/O Panel
2. Mixer Panel
3. Routing Panel

ANALOG I/O VIEW

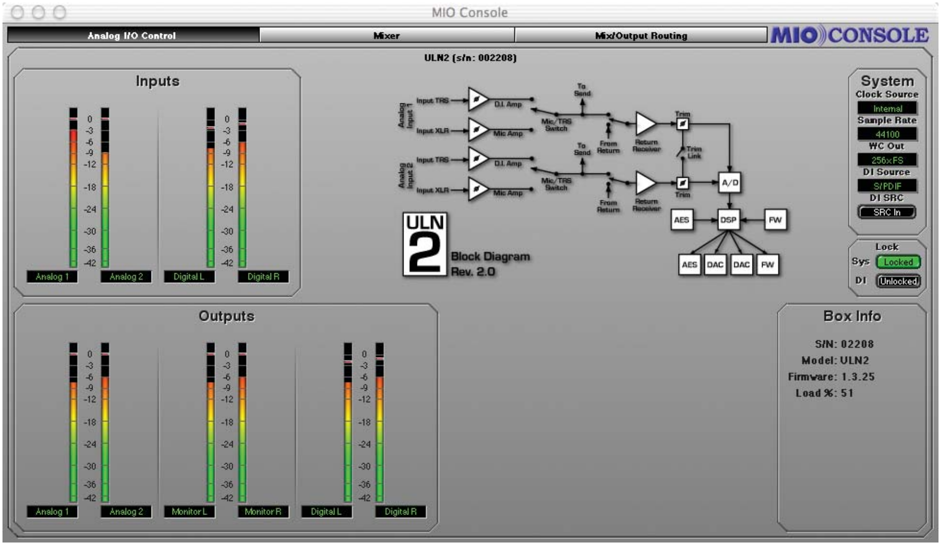

Figure 15: Analog I/O Panel

This panel provides full control and metering of all of the analog I/O that the box provides. The top half of the view is dedicated to inputs and the bottom half is dedicated to outputs. You access this panel by clicking on the "**Analog I/O Control**" button of the **view panel selector bar**.

1. **Channel Label** button

Figure 16: Channel Label

- This simply labels which channel is associated with the channel strip.

2. **Channel Level Meter**
    - This is a peak reading, high-resolution, fast PPM meter. It shows the post converter level of the input signal of the associated channel. The peak hold bar indicates the highest level seen on the channel since the last reset. You can reset the hold by clicking on the meter. These meters are simply high resolution versions of the meters shown on the front panel of the box – all the meter data is generated by the ULN-2 hardware.

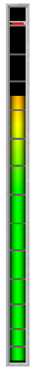

Figure 17: Channel Level Meter

Optimizing Input levels

The Analog to Digital converters (ADC) in most devices function best when the peak level is around -6 dBFS (lowest distortion, best sound). This is true of the ADCs in ULN-2. Since you have full level control of the input with the gain trim knob, you will find that you get the best quality recordings if you try to set the nominal peak level of the input at about -6 dBFS. In addition to providing the best recording quality, it has the added benefit that you will be operating with an extra 6 dB of headroom before clipping. There is no drawback to optimizing your levels in this way, and plenty of benefit.

Digital Input Meters

To the right of the Analog Input control section is the Digital Input Meter section. This group of meters provides level metering for all of the digital inputs on the ULN-2. These meters have the same response characteristics as the analog input meters and show you the audio activity on digital input channels 1-2, going from left to right.

BLOCK DIAGRAM

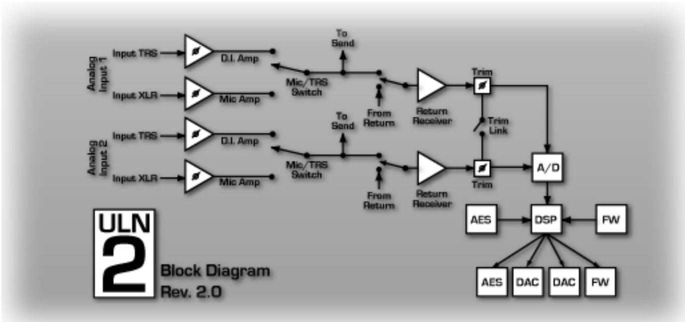

Figure 18: MIO Console ULN-2 Block Diagram

The **Block Diagram** shows the basic signal path of the ULN-2.

SYSTEM CONTROLS

Figure 19: System Controls

The **System** block provides controls that adjust various system level aspects of the ULN-2 hardware:

1. The **Clock Source** popup menu controls the system clock source used by the hardware for digital synchronization and driving the converters:
   - **Internal** causes ULN-2 to use its internal clock. You must select this if you want to set the sample rate from the ULN-2. If any other clock source has been selected, the console will not allow you to change the

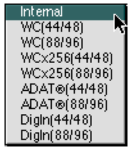

Figure 20: Clock Source Pop-up Menu

sample rate since the sample rate is determined by the external clock source.

- **WC (44/48)** directs ULN-2 to clock off of an external Word Clock Source at single rate (e.g. fs = 32k-50k)
- **WC (88/96)** directs ULN-2 to clock off of an external Word Clock Source at double rate (e.g. fs = 64k-100k)
- **WCx256 (44/48)** directs ULN-2 to clock off of an external 256fs Clock Source at single rate (e.g. fs = 32k-50k)
- **WCx256 (88/96)** directs ULN-2 to clock off of an external 256fs Clock Source at double rate (e.g. fs = 64k-100k)
- **DigIn (44/48)** directs ULN-2 to clock off of the selected stereo digital input at single rate (e.g. fs = 32k-50k). This allows operation of the digital input without SRC, and from devices that must supply clock.
- **DigIn (88/96)** directs ULN-2 to clock off of the selected stereo digital input at double rate (e.g. fs = 64k-100k). This allows operation of the digital input without SRC, and from devices that must supply clock.

2. The **Sample Rate** popup menu allows you to select the sample rate when you are using internal clock. The ULN-2 must be running on internal clock for the Sample Rate popup menu to have any effect. If the ULN-2 is running from an external clock source, you cannot select the sample rate since it is determined by the external clock source.

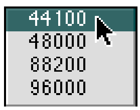

Figure 21: Sample Rate Popup Menu

3. The **WC Out** popup menu allows you to select the output clock signal the ULN-2 generates on its WC Out BNC connector. The available choices are 1x and 256x. The 1x signal is appropriate for

driving devices that accept a Word Clock signal. The 256x signal is appropriate for driving devices that accept 256x or SuperClock signals. Refer to the documentation for the external device to determine what is the most appropriate clock reference for it.

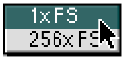

Figure 22: WC Out Popup

4. The **DI Source** popup menu allows you to select the active input for the digital input pair. The choices are AES and S/PDIF. This selector physically switches the input to the digital audio receiver between the RCA input and the XLR input.

Figure 23: DI Source Popup

5. The **DI SRC** button enables and disables the asynchronous sample rate converter (SRC) in the ULN-2 digital audio receiver. When the SRC is engaged (button illuminated yellow), the digital audio receiver will automatically synchronize the input signal to the ULN-2 system clock over a wide range of sample rate ratios.This allows you to, for example, digitally transfer a sample from a CD player into a 96k session without any clocking problems. If you want to make bit-transparent transfers, you will need to disengage the SRC and ensure that the ULN-2 and the external device are both using the same digital audio clock via one of the ULN-2 synchronization mechanisms.

Figure 24: DI SRC Buton

6. The **Lock** indicators show which elements of the ULN-2 clocking system are properly locked. The clocking system must be locked for the unit to behave as expected. If the system is not locked, audio will play at the wrong rate and will be distorted or noisy. Under normal circumstances, the system should always be locked, but if you have selected an external clock source and the clock signal is not present, corrupted or out-of-range, the system may unlock. There are indicators for the system and the digital input.

Figure 25: Lock Indicators

Analog Output Control

The bottom half of the panel is dedicated to the hardware outputs of the ULN-2.

Channel Label button

- This simply labels which channel is associated with the output.

1. **Channel Level Meter**
    - This is a peak reading, high-resolution fast PPM meter. It shows the pre converter level of the output signal of the associated channel. The peak hold bar indicates the highest level seen on the channel since the last reset. You can reset the hold by clicking on the meter. All the meter data is generated by the ULN-2 hardware.

Monitor Output Meters

To the right of the Analog Output controls is the Monitor Output Meters section. This group of meters provides level metering for the monitor and headphone outputs on the ULN-2. These meters have the same response characteristics as the analog output meters, and show you the audio activity of the monitor stream.

Digital Output Meters

To the right of the Monitor Output controls is the Digital Meters section. This group of meters provides level metering for all of the digital outputs on the ULN-2. These meters have the same response characteristics as the analog output meters, and show you the audio activity on digital output channels 1-2.

Box Info

The **Box Info** section of the panel, in the lower right-hand corner of the window, shows you information about the currently connected and selected ULN-2 unit. This section displays the Serial Number, Model Information and

Firmware revision of the connected box as well as the DSP load on the unit. All of this information can be useful in trying to track down any connection problems that may arise.

Figure 26: Box Info

If there is no information displayed in the Box Info section, the software is not communicating properly with the ULN-2 hardware, or there is no ULN-2 present on the FireWire bus.

If the FireWire light on the front panel of the ULN-2 is illuminated but the box information does not appear in the console window, it is very likely that the software has not been installed properly. If this is the case, please refer to the installation instructions for details on how to properly install the software. Also, please ensure that there are no copies of the MobileIO Driver in the same folder as the MIO Console application or any of the ASIO drivers that you will use with an ASIO host.

If, on the other hand, the FireWire light on the front panel of the ULN-2 is not illuminated, the box is not communicating properly with the computer. Please check the cabling of your ULN-2 and other devices on the FireWire bus and make sure that everything is connected correctly. If that does not properly establish the connection, try rebooting your computer. As a last resort, try connecting only the ULN-2 to the computer to ensure that communication can be established.

Mixer Panel

The Analog I/O panel provides all of the bread-and-butter functionality that you would expect from a flexible audio interface like ULN-2 - what you need to get the job done quickly and easily. The Mixer panel provides features that you won't find anywhere else – here is where things start to get *really* interesting.

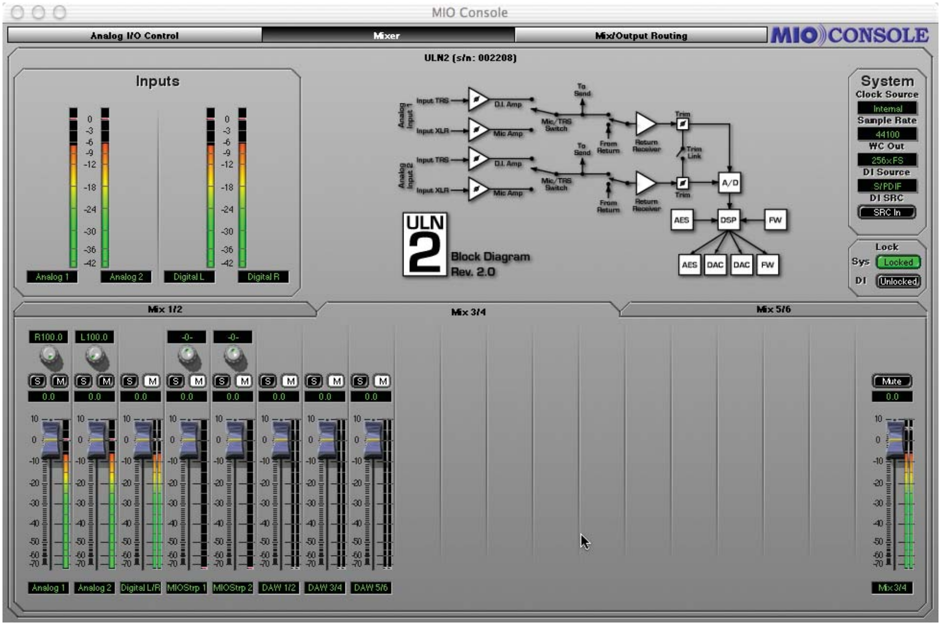

Figure 27: Mixer Panel

The controls in the top half of the mixer panel are the same analog input controls described in the section **"Analog I/O View**" on page 24. The bottom half of the mixer panel provides a control surface for the multiple integrated hardware mixers running on ULN-2. You access this panel by clicking on the "**Mixer**" button of the **view panel selector bar** at the top of the console window.

MIXER CONTROLS OVERVIEW

Each mixer has controls for Level (fader), Solo (button), and Mute (button) for each channel. Each stereo mixer also includes a pan control (knob) for each mono channel. Stereo inputs, such as the defaults of the digital inputs or your DAW output, will not have a pan knob. Instead, the two individual channels that make up the pair will be hard panned, and the fader meters will appear as stero meters.

At the far right of each mixer surface is the Master Fader. This fader will be Mono for Mono Mixers and Stereo for Stereo Mixers. Above the Master Fader is a bus mute button that allows you to quickly mute the entire mixer.

Each mix bus can have up to 10 inputs assigned to it. All of the analog and digital hardware inputs are available in each mixer. You can also assign any (or all) of the host playback channels from your DAW to each mixer. Since there are 4 hardware inputs and 6FireWire return busses that are routed from the host computer, you can mix up to 10 channels on each bus.

The default state of the hardware input channels in the mixers is faders at unity, mutes on, solos off and center panned. The playback channels from your DAW are unmuted in pairs per bus (e.g. DAW 1/2 will be unmuted on Mix 1/2, DAW 3/4 will be unmuted on Mix 3/4, etc.)

Mixer Pane Tabs

Use the tabs above the pan knobs to select the mixer you want to control. Each tab represents a mixer you have configured in the routing panel. When you click on a tab, the controls will be instantly updated to reflect the state of the associated mixer. In this way you can quickly switch back and forth between multiple independent mixes, each with a full mixing interface.

Figure 28: Mixer Pane Tabs

Each mixer also has a **Parameter Popup** control associated with it. Unlike the Input and Output controls, the parameter control popup does not have an explicit user interface element representing it. Instead, the popup is accessed from the mixer's tab itself. To pop-up the menu, you either click and hold the tab, or <control>-click the tab. The parameter popup menu allows you to maintain a library of standard mix configurations, scenes, and setups (see: "**Parameter Popup Controls**" on page 50). It also provides a very quick method for copying mixes from one bus to another.

Channel Faders

The Channel fader controls the relative level of the input channel in the mix. It works just like its non-virtual counterpart. The calibration numbers to the

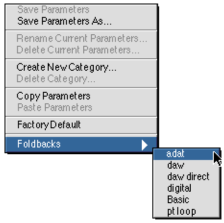

Figure 29: Mixer Parameter Popup

left of the fader knob provide an accurate guide to the amount of gain that will be applied by the fader. The label area above the fader knob lets you know which input channel the fader is controlling. The labels default to the console’s default names for the channels, but you can rename the channels to more meaningful names using the naming controls in the Routing Matrix (see: “**Configuring Channel Names in the Matrix**” on page 48).

The exact amount of gain for the fader is displayed in the small black window above the fader channel label. If you want to set a channel gain precisely, simply click in the small black window and type in the desired gain in dB numerically. Hitting the <return> key or clicking outside the entry box makes the new setting take effect. Hitting the <tab> key will make the new setting take effect and will move to the next numeric entry field in the mixer.

The gain range for each channel is from -∞ (muted) to +10 dB. The resolution of the gain control is extremely fine and gain changes are interpolated in the mixer, so there is no zipper noise when you make continuous (or even discontinuous) gain changes.

<Command> clicking the fader knob will allow you to make fine adjustments to the fader level by dragging the mouse up and down. <Option> clicking the fader knob will reset the fader to unity gain. Clicking the meter associated with the fader will clear the peak holds for that channel.

For mono channels, the current gain is applied to the channel before the pan. For stereo channels, the current gain is applied uniformly to both of the hard-panned input channels.

CHANNEL METERS

Each channel fader has a meter (or a pair of meters for stereo channels) associated with it. The meters are calibrated consistently with the fader calibration. Each meter is a peak reading, fast-PPM meter that is pre-fader. The peak hold bar shows the biggest peak since the last reset. Clicking on a meter will clear the peak holds for that channel.

CHANNEL PANS

Each mono input in a stereo mixer has a pan knob above the fader. The pan knob allows you to control the relative amount of the input channel that is placed into the two busses of the stereo mixer. Panning hard left (L100) means that the channel will appear at full volume in the left (odd) bus of the mixer. Panning hard right (R100) means that the channel will appear at full volume in the right (even) bus of the mixer. When the channel is center panned, the signal appears at a decreased volume (-3dB) in both channels, such that the volume of the total signal in both channels is equivalent to full volume in one channel. As you pan from left to right, the signal is distributed between the two channels so that the total volume remains constant.

As with the channel fader, the black window above the channel pan knob provides a precise readout of the current pan position, and clicking on the black window will allow you to type in an exact pan amount. Negative numbers (-100 to 0) indicate left pans and positive numbers (0 to 100) indicate right pans. 0 is center pan.

<Option> clicking the pan knob will return it to center pan, and <command> click-dragging the knob will allow you to adjust the pan position in finer increments.

MUTE BUTTONS

Each input has a Mute button (labeled "M") associated with it. When the mute button is engaged (illuminated white), the channel will be muted in the mixer. The mute is interpolated, so muting a channel will not cause audible clicks.

<Option>-clicking a mute button will set all of the mutes on the mix bus to the same state as the button you click. This allows you to quickly mute all the channels or quickly unmute all the channels on a bus.

Solo Buttons

Each input has a Solo button (labeled "S") associated with it. When the solo button is engaged (illuminated red), only channels that have been soloed will be mixed by the mixer. The other channels will effectively be muted. As with the mute button, the gain changes associated with soloing or unsoloing a channel are fully interpolated and will not cause audible clicks.

<Command>-clicking a solo button will exclusively solo the associated channel. The <command>-click will automatically clear the solo state of all other channels on the mix bus. <Option>-clicking a solo button will set all of the solos on the mix bus to the same state as the button you click. This allows you to quickly solo all the channels or quickly un-solo all the channels on a bus.

Mixer Master Fader

The Fader that appears on the right side of the mixer pane is the Mix Master Fader. This fader controls the overall bus level of the mix bus. Operationally, it is exactly the same as the channel faders (see: "**Channel Faders**" on page 32). Each Mix bus in the system has 24 dB of headroom above full scale. If the summing point of the mix bus is clipping, you can pull the mix out of clipping by dropping the Master fader, as long as the sum point is clipping by less than 24 dB.

Mixer Master Mute

Each mix bus has a master mute button above the master fader that allows you to mute the output of that mixer. This mute is interpolated.

WIDE Mixers

As was described before, the integrated mixers in ULN-2 are **WIDE** – that is they allow you to mix every available input channel (both hardware channels and FireWire channels) together. In the widest case, the mixer will have 10 faders to allow you to control the gains for all of the input channels.

The MIO console window is only wide enough to accommodate 18 faders plus the Master fader. If you have enabled more channels than will fit in the window, the MIO console will automatically display a scrollbar at the bottom of the Mixer pane. Use this scrollbar to control which faders are visible at any given time. The width of the scrollbar indicator shows you how many of the enabled channels are visible at any given time. The scrollbar will automatically be hidden if you reduce the number of inputs to the mixer below 19 channels or if you switch to a different mixer that has less than 19 channels enabled.

TIP: ULN-2's support of near-zero latency mixing of every channel opens up a huge variety of applications that cannot be achieved with standard interfaces or, at the very least, require external gear or major work-arounds to accomplish. Some examples are:

1. Stem-based mixing.
    - In this mixing technique, you mix disparate elements of the program to separate sub-mixes called stems. You might mix drums to one stem, instruments to another, and vocals to a third. Then the relative balances of the mix can be addressed later in a macroscopic way (during mastering, for example). This also enables remixing the project easily without having to go back to the multi-track master. Since you will be creating stems on individual busses in the DAW, you need to sum the stems for monitoring. This is easily accomplished with ULN-2's mixer.
2. Monitoring mixed-in effects sends when the external effects are unavailable.
    - You may find that you want to continue editing or mixing while you are away from the studio. You can use the WIDE mixer in ULN-2 to mix in DAW effects send busses for monitoring without having to reconfigure your DAW session.
3. Multichannel foldback mixes.
4. Near-Zero Latency monitoring of external effects.
    - Most singers need some reverb or other effects to get the feel right during their performance. With ULN-2 and the WIDE mixer, you can split off a send from the performer's input channel, send it to an external effects unit, and mix the effect return into the performer's foldback mix -- with virtually no latency.
5. Multiple WIDE mixes
    - Since ULN-2 supports multiple WIDE mix busses simultaneously, you can form multiple, individual foldback mixes for multiple performers at the same time, and best of all, each mix has its own complete mixer control surface, so you don't have to mess around with a million unreadable aux send knobs.

MIX/OUPUT ROUTING PANEL

The third panel in the MIO Console window is the **Mix/Output Routing** panel. This panel is what you use to access the powerful routing features that you won't find anywhere else. The Routing Matrix portion of the panel lets you dynamically control the configuration of the WIDE mixing engine and also configure the channel names for the hardware. You access this panel by clicking on the "**MIX/Output Routing**" button of the **view panel selector bar** at the top of the console window.

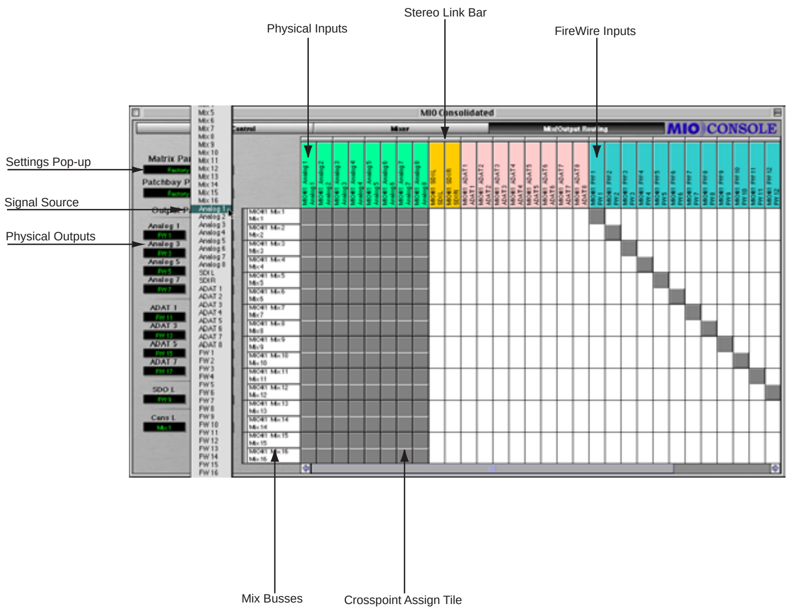

Figure 30: Mix/Output Routing Panel [for MIO 2882]

Figure 30 shows Mix/Output Routing Panel for a Mobile I/O 2882 unit. The ULN-2 has a smaller set of physical I/O and mixing resources, and the Mix/ Output Routing Panel automatically adjusts to the device it is displaying. See **Figure 31** on page 39 for the panel appearance when you are using a ULN-2. The Mix/Output Routing Panel has two main components:

1. The Output Patchbay on the left side of the panel.
2. The Routing Matrix on the right side of the panel.

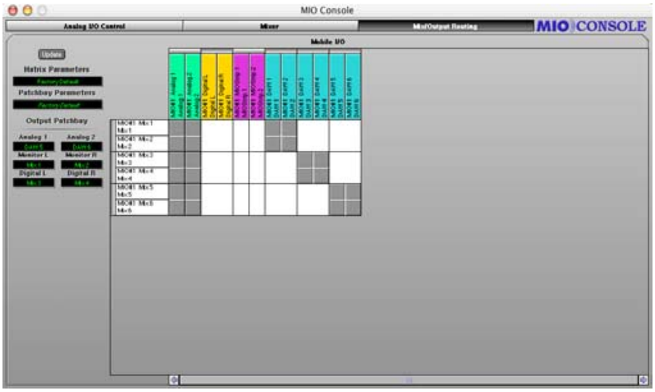

Figure 31: ULN-2 Mix/Output Routing Panel

The Routing Model

The Routing Panel provides a user interface for controlling the underlying routing architecture of the Mobile I/O. The routing architecture provides a powerful routing model to allow you to control the routing of signals between physical & virtual inputs and the hardware mixer & physical outputs.

Conceptually, the architecture is quite simple:

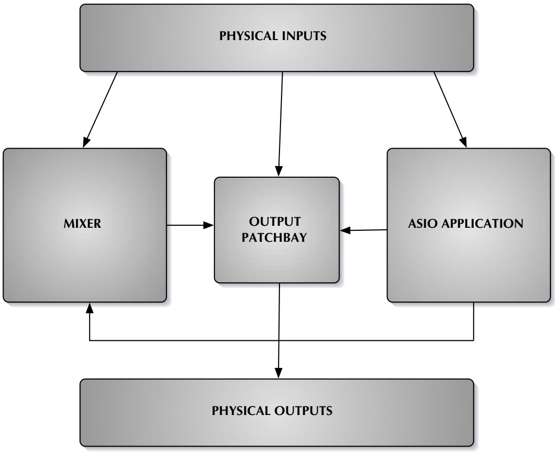

Figure 32: Block diagram of the Mobile I/O routing architecture

All of the physical inputs (e.g. Analog and Digital), all of the channels being transmitted over the FireWire bus from the host Application (e.g. DAW), and all of the outputs from the Mobile I/O WIDE Mixer are available to the Output Patchbay. The Output Patchbay can cross-point assign any of its inputs to any of the physical outputs (e.g. Analog, Monitor and Digital).

All of the physical inputs and all of the channels being transmitted over the FireWire bus from the host Application are also inputs to the Mobile I/O WIDE Mixer. Every bus has each of those inputs available for mixing. The mixer outputs are sent to the Output Patchbay for routing to physical outputs. The number of mix busses varies with the Mobile I/O hardware model and the sample rate. The Mobile I/O ULN2 supports 6 mono (3 stereo) mix busses at all supported sample rates.

Every physical input is mult'ed from the router/mixer section and sent directly over the FireWire bus to the ASIO host. Regardless of any mixing, routing, or mult'ing that you configure in the hardware, you can always record all of the inputs with your DAW.

As you can see from this simple, high level view, the Mobile I/O routing architecture supports direct routing of any input to any output and also mixing of any set of inputs to multiple mixers. The outputs of the mixers can be routed to any output or any set of outputs for hardware mults. All hardware inputs are available to the host.

As it turns out, the Mixer/Router in Mobile I/O is quite extensive and the simple user interface hides a lot of complexity (**Figure 33**).

As you can see from this schematic diagram of the mixer/router, the Mixer in Mobile I/O is a fully cross-pointed matrix. While we are presenting the mixer to you as a set of mono and stereo mixers, the actual underlying structure is a full matrix. This means that the mixer is ready for surround and multi-channel matrix mixing today. The schematic provides an accurate representation that shows the signal flow through the mixer and into the router with all analog and digital gain points represented.

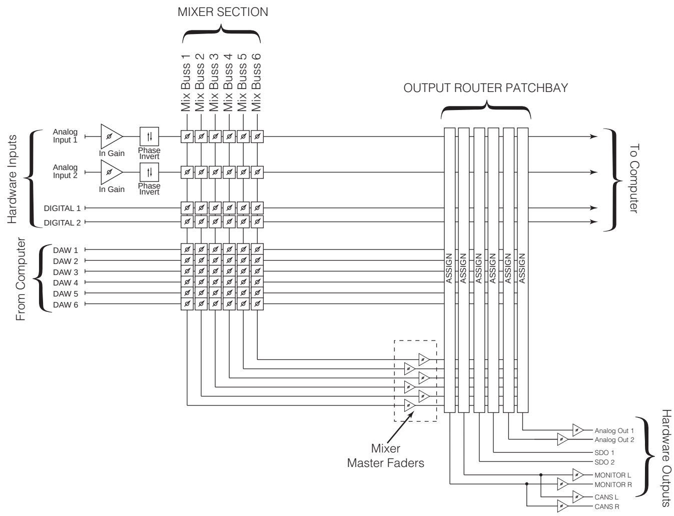

Figure 33: Complete view of the Mobile I/O Matrix Mixer

Output Patchbay Details

The Output Patchbay allows you choose the signal sources to send to the hardware outputs. You can feed any output from your choice of a mix bus, a hardware input, or a FireWire input from the computer.

Patchbay Popup Controls

The output patchbay is the set of popup menus on the left side of the panel. Each popup menu corresponds to the hardware output that is labelling it. The channel that you choose from the popup menu will drive the associated hardware output.

Since the output patchbay is fully cross-pointed, you can easily construct direct routes from hardware inputs to hardware outputs for stand-alone converter operation, routes from the host (using the DAW channels) to the outputs for direct dubbing and monitoring from the computer, and routes from the mixer to hardware outputs for foldback, monitoring, and effects sends.

The output patchbay even lets you send the same channel to multiple outputs, which makes it really simple to create channel mults. You can use this to:

- send multiple copies of an aux group to different effects devices
- manage monitoring on multiple monitor systems (big and small)
- monitor on analog outputs and record on digital outputs
- send the same monitor mix to multiple outputs for multiple performers
- record safety mixes on digital two-track recorders while multitrack recording to the hard drive

To choose a source for an output channel, simply go to the popup menu for the output you want to route to, click and select the source send. Each popup window is like the jack of a patchbay, and each name in the popup menu is like a patchbay cable that carries the named signal.

Patchbay Parameter Popup Control

Directly above the patchbay popup menus is the "**Patchbay Parameters**" parameter popup control. This control allows you to store patchbay configurations for later recall. See the section "**Patchbay Popup Controls**" on page 41, for more details about the parameter popup menu.

The factory default routing for Mobile I/O configures the box to behave as a direct-routed audio interface – all the inputs go directly to the computer, all the outputs from the computer are directly routed to the hardware outputs, and the Headphone output is fed by the stereo mixer on Mix 1/2. This makes the Mobile I/O act like a basic audio interface.

Other configurations that may be very useful include:

- direct routing of analog inputs to AES/SPDIF (2 channel Pre/DI/ADC)
- switching mults of the mixer outputs for alternate monitor configurations

This control allows you to maintain a library of frequently used routings and switch between them at will. Spend a little time to familiarize yourself with it – it is extremely powerful.

Routing Matrix Details

The **Routing Matrix** (Matrix) is the mixer assign matrix that is on the right side of the pane (see **Figure 30** on page 38). This control surface allows you to fully configure the structure of the multi-bus WIDE mixer in the ULN2 hardware. It also allows you to name all of the hardware and virtual channels that are accessible in the hardware.

You use the Matrix to assign hardware and FireWire inputs (DAW outputs) to mixers in the Mobile I/O. To assign channels to a given mix, you click on a crosspoint in the Matrix, darkening the associated Crosspoint Assign tile. Deselecting a tile removes the associated fader from the mixer and mutes the channel in the hardware.

The WIDE mixer allows you to assign all of the hardware input channels and all of the FireWire channels to each mix bus. By limiting the number of channels assigned to any given bus, you can reduce the complexity of the associated mixer interface.

Logical Description

Each mix bus and each input channel has a path label tile associated with it. These tiles are arrayed along the top and left edges of the Matrix.

The tiles along the top edge of the Matrix are the input path tiles. Each tile is color coded based upon the type of input it is:

- Green for Analog Inputs
- Orange for Digital Inputs
- Purple for Processed Busses
- Blue for FireWire Inputs (Outputs from the DAW)

Each tile shows the Mobile I/O unit it is associated with, the physical name for the path, and the user defined name for the path.

Above each pair of input tiles is a thin tile that is used to join two input paths into a stereo input channel. In the factory default configuration, the Digital Inputs are joined into a stereo pair and each pair of DAW channels is joined into stereo pairs. If the channels are joined into a stereo pair and the pair is assigned to a stereo mixer, the pair will be represented by a single stereo fader in the mixer interface, and no pan knob will appear.

The white tiles along the left edge of the Matrix are the Mix bus tiles. All of these tiles are white because they all represent mix busses. Each tile shows the Mobile I/O unit it is associated with, the physical name for the mix bus, and the user name for the mix bus.

To the left of each pair of Mix tiles are thin inset tiles that are used to join two mix busses into a stereo bus. If the busses are joined, the associated mixer will have pan knobs for each of the mono hardware channels. If the bus is a mono bus, there are no pan knobs associated with the input channels.

The interior of the Matrix is composed of a large number of square crosspoint assign tiles. The Matrix is too wide to fit completely within the MIO Console window; you may use the scrollbar that appears at the bottom of the Matrix to scroll the remaining DAW channels into view.

Each crosspoint assign tile indicates whether or not the input channel at the top of the column is assigned to the mix bus at the edge of the row. If the tile is filled in, the channel is assigned to the mix, and the controls for that channel will appear in the associated mixer. If the crosspoint tile is not filled in, the channel is not assigned to the mix, and the controls will not appear. The channel will be muted and un-soloed on the associated mix bus.

Stereo pairs are automatically assigned as a group to busses.

There are some keyboard shortcuts that you can use when making Matrix assignments:

- clicking in one tile, followed by <shift>-clicking in another tile will automatically select or deselect all of the tiles in between the end-points; the selection will match the new state of the tile at the second click location
- clicking and dragging in the Matrix will continuously apply the new assignment state to each tile you drag over
- <Command>-click dragging will scroll the Matrix

Configuring Channel Names in the Matrix

MIO Console has fully user configurable channel names. The names that you select for your channels will propagate to all of the other aspects of the MIO Console user interface. Due to limitations in the host driver specifications, the channel names will not propagate to your applications.

This allows you to name the channels in meaningful ways. The analog inputs can be named to match the sources. The Digital I/O can be named to match the effects device that you have patched. Mixes can be named by the fold-back monitor or effects send that they will feed.

To name a channel, click on the input path tiles (for input channels) or the mix bus tile (for Mix busses). The channel configuration window will appear above the MIO console window:

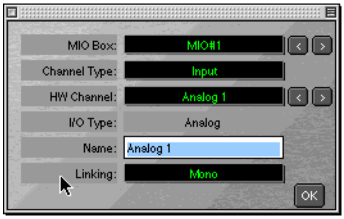

Figure 34: Channel Configuration Window.

All of the channel identification controls will be set for the tile you clicked on. To change the user selectable name of the channel, simply type the new name. The name will be updated in the console when you do one of the following things:

1. Hit the <return> key. This will update the name and dismiss the window.
2. Hit the <tab> key. This will update the name and switch to the next channel in the list.
3. Hit <shift><tab>. This will update the name and switch to the previous channel in the list.
4. Select any other channel using the popup menus or the “<“ or “>” buttons in the channel configuration. This will update the name and switch to the new channel selection.

You can dismiss the window without updating the channel name by clicking its close box.

You can also select the stereo linking state from **Linking** popup menu in the channel configuration window. The state will be updated in the console along with the channel name.

The channel identification controls identify which channel you are adjusting:

- The **MIO Box** popup menu allows you to select which Mobile I/O you want to configure if more than one Mobile I/O is present on the bus.
- The **Channel Type** popup menu lets you switch between input channels and mix busses.
- The **HW Channel** popup menu identifies which hardware channel you are adjusting.
- The **<** button associated with the MIO Box and HW Channel popup menus steps to the previous item in the associated list.
- The **>** button associated with the MIO Box and HW Channel popup menus steps to the next item in the associated list.

Matrix Parameter Popup

The entire state of the Matrix can be saved to and recalled from the Console Parameter Library System. The basic functions of the Parameter Popup control are documented in the next section (see "**Parameter Popup Controls**" on page 46). Since you are very like to have a number of tracking and mixing configurations you use over and over again, the Parameter Popup for the Matrix is a real timesaver. This control appears in the top left corner of the pane, above the Parameter Popup control for the Patchbay router. Each time you create a configuration that you are likely to use again, save it in the Parameter Library for instant recall when you need it next.

Parameter Popup Controls

The **Parameter Popup** control is MIO Console's unified mechanism for handling presets for the various sections of the Mobile I/O. Each element of the console that supports the Parameter Library mechanism has a parameter popup control associated with it. These elements currently include:

- Input Channels [2882 only]
- Output Channels [2882 only]
- Mixers
- Output Patchbay
- Matrix

Each instance of the Parameter Popup control provides the same commands and options for every section of the console.

Popup Commands

The parameter popup provides a hierarchical, categorized library of configuration presets for the associated section of the console. The menu is divided into three portions. The first portion consists of all of the items above the "Factory Default" item. The second portion is the "Factory Default" item and the third portion is the hierarchical items below the "Factory Default" item (see **Figure 35** on page 47).

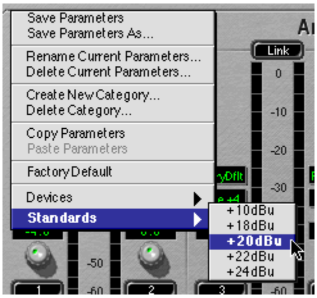

Figure 35: Parameter Popup Menu

The commands in the first portion of the menu allow you to save and manage the presets in the library. All of the presets are shared between like elements in the console. The preset commands are:

- Save Parameters

  use this command to save the current state of the associated console settings to the currently selected preset (will appear as "Save Parameters As…" if there is no currently selected preset)

- Save Parameters As…

  use this command to name and select a category to save the current state of the associated console settings as a preset into the library

- Rename Current Parameters…

  use this command to rename the currently selected preset

- Delete Current Parameters

  use this command to remove the currently selected preset from the library

- Create New Category

  use this command to add a new category to the library

- Delete Category

  use this command to delete the currently selected category and all of its associated presets

- Copy Parameters

  use this command to copy the current state of the associated console settings to the clipboard; you can use this to copy your settings from one block to another

- Paste Parameters

  If the clipboard contains compatible settings, this command will be available and will set the current state of the associated console settings to the settings on the clipboard. Use this with the "Copy Parameters" command to duplicate settings from one channel to another or from one mix to another

The "**Factory Default**" command will set the current state of the associated console settings to the default settings.

Popup Presets

In the third part of the menu, each of the categories will be listed as a hierarchical menu title. Each of the presets for each category will be listed in the submenu under the category menu. The currently selected category and preset are drawn in bold, so you will know what is currently active.

Selecting a preset from the menu will make that preset active and will set the current state of the associated console settings to the values contained within the preset. The name of the currently selected preset will be drawn in the popup area in the console window to indicate which preset is active.

If you change the settings in the console, the name of the preset will be drawn in italics indicating that the current settings differ from the selected preset.

For the Input and Output channels, you can hold down the <option> key while selecting a preset to automatically apply the preset to all of the other input or output channels.

To access the parameter popup for the mixers, either click and hold the associated mixer tab or <control> click the associated mixer tab.

We have provided an initial set of presets for the various parameter libraries. The presets for the output channels are relatively complete and give you an idea of the power and flexibility of this approach to parameter management. We will be adding presets on a regular basis – check the website for new presets.

Finder Management of Presets

The presets in the console preset library are saved as standard files on your hard disk. This means that, in addition to using the commands in the parameter popup menu, you can also manage the presets directly from the Finder. The presets are all contained in the "Parameter Library" folder that is in the same folder as the MIO Console Application. The "Parameter Library" folder in the Console Application folder may be an alias to the real "Parameter Library" folder – so you can move the Parameter Library folder to another location and simply make an alias to the folder and place it with the console application.

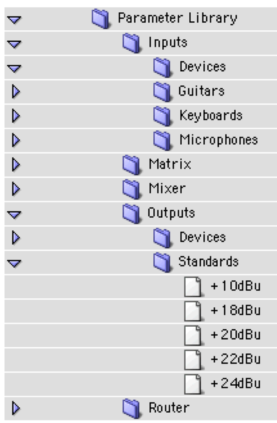

Figure 36: Parameter Libray Files in the Finder

You can create categories by creating new folders within the folders that appear within the "Parameter Library" folder. You can move presets from one category to another by moving the preset files within the hierarchy. You can also make copies by copying the files. You can delete presets by deleting the associated files.

Since the presets are stored as individual files, it is easy to share presets with other Mobile I/O users. By sending your preset files to another user, they can install your presets into their library by simply dropping the files into the appropriate folders in the Finder. You can rename categories by renaming the folders within the Finder.

Do **not** change the names of the "Parameter Library" folder or the top-level folders within the "Parameter Library" folder – these are used by the MIO Console library system to find the presets for the different sections of the console. The presets will not be found if those folder names have been changed (this includes the "Inputs," "Matrix," "Mixer," "Output," "Router," etc.)

You **cannot** rename presets from the Finder -- the parameter library will use the name stored in the preset file and ignore the file name in the Finder.

Any other changes you make in the Finder will not be recognized by the MIO Console while it is running – you will have to quit the Console and relaunch it for the changes to be recognized.

Note: In OS X, the Parameter Library folder is contained within the MIO Console Application bundle. This is likely to change in the future, with the Parameter Library Folder moving to the user preferences folder contained in the ~/Library/Preferences folder.

Persistent State Management

The ULN-2 hardware has support for Persistent State Snapshots. There are 10 snapshot slots in the ULN-2 that are recallable from the controls on the ULN-2 front panel. Each Persistent State Snapshot contains a complete description of the state of the box, including Sample Rate, Clock Source, Digital input source, Sample Rate Converter Enable, Patchbay routing, Mixer Configuration and levels. In other words, a snapshot saves every aspect of the configuration of the ULN-2.

The first snapshot slot is special as it is used by the unit to configure the hardware and the routing when the ULN-2 starts up. The other 9 slots are available for storing alternate configurations that can be selected "on the fly" after the ULN-2 is up and running.

When a computer is attached to the ULN-2, the front-panel controls to select snapshots are locked-out since the computer is actively controlling the configuration of the box.

If the computer is not attached, the two tact-switches on the left-side of the front-panel (between the status indicators and the meters) may be used to select the snapshot that you want to use to configure the ULN-2. These buttons are labled with up and down arrows. The currently selected snapshot is indicated by the column of LED's labled **C, 1, 2, 3, 4, 5, 6, 7, 8, 9**. When the ULN-2 turns on, the "**C**" indicator will be illuminated, indicating that the unit has booted up with the state that was stored in the "Boot Snapshot".

Figure 37: ULN-2 Front Panel Snapshot Controls

Pressing the up arrow will move to the next higher snapshot in the list (e.g. if you are currently on snapshot 3, you will move to snapshot 2). Conversely, pressing the down arrow will move to the next lower snapshot in the list (e.g. if you are currently on snapshot 3, you will move to snapshot 4). If you are at either beginning of the list and you press the up arrow, you will wrap around to the last item in the list. When you select a new snapshot, the new snapshot is applied to the box immediately.

In order to configure the boot state and snapshots for your ULN-2, you will need to utilize the MIO Console application. Configuring and storing snapshots in the box is very simple:

1. First, attach the ULN-2 to the computer and start up MIO Console.
2. Use MIO Console to configure the box. Set up all aspects that you care about. Once you have the configuration as you like it, you are ready to save the snapshot.
3. Choose the appropriate save command from the Utilities Menu:

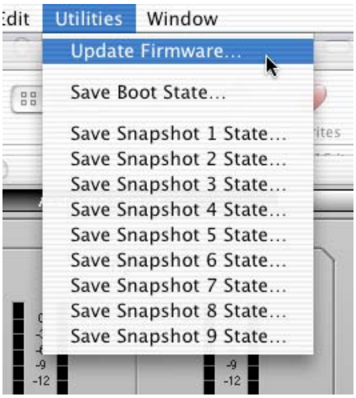

Figure 38: Utilities Menu

- To save the snapshot to the "Boot State" slot, choose the "Save Boot State…" item.
- To save the snapshot to one of the other snapshot slots, choose the appropriate "Save Snapshot x State…" item (where x is the appropriate number).

4. Save a copy of the current Console state to a file on your hard disk with an appropriate name (like "ULN-2 Snapshot 1" for the 1st snapshot) so that you have a copy of the state on the computer if you want to modify it in the future.

Routing Applications

Clocking Considerations

There are five ways that you can clock Mobile I/O:

1. Internal
2. Digital (AES/SPDIF)
3. Wordclock
4. 256X

Each choice is appropriate for a particular situation. In general, you should use Internal clock if you can because you are likely to find that it makes your other digital gear sound better. However there are some devices that either must be the clock master or work better if they are the clock master. For these devices, choose the clock port which is most appropriate for the device.

For example, many DAT machines distribute clock over their digital audio connection (AES or S/PDIF). For these machines you would connect to the AES or S/PDIF port on Mobile I/O and choose DigIn (44.1/48) for single rate or DigIn (88.2/96) for double rate in the MIO Console clock source popup. For more information about configuring the clock source, see "**System Controls**" on page 29.

Single speed clocking vs. double speed clocking

The difference between single speed (1x) and double speed (2x) clocking is handled as a fundamental mode change in Mobile I/O. The sample rate does not vary continuously between 48k and 88.2k but changes discontinuously when the double speed mode is set in the hardware.

When Mobile I/O is running on internal clock, this mode change is handled transparently when you specifiy what sample rate the box should use.

When Mobile I/O is running on external clock, it cannot determine in advance which mode to use, so you need to tell it what mode is appropriate for your external clock source. You communicate this information by selecting the proper external clock source. Each external clock source comes in two different flavors:

- xxxx(44/48)
- xxxx(88/96)

where the "xxxx" corresponds to the actual clock source. If you will be clocking off of a 1x source (e.g. 32kHz–50kHz sample rate) then choose the xxxx(44/48) flavor of the clock source. On the other hand, if you will be clocking off a 2x source (e.g. 64kHz–100kHz sample rate) then you will choose the xxxx(88/96) flavor of the clock source. If you choose the wrong flavor, Mobile I/O will not lock properly. When you select the flavor, it tells Mobile I/O which mode it has to run in, and the Mobile I/O can lock to the external clock in the appropriate frequency range.

Example Setup

Figure 39: Example ULN-2 Setup

In this section we will go step by step through an application that is possible with a ULN-2 configured as shown in Figure 39 on page 54. After reading this section you should have a feel for how to apply Mobile I/O's routing and mixing capabilities.

Which device is the Clock Master

The first consideration involved with a setup like this is which device should be the clock master. If the external device is happy slaving to Mobile I/O then make Mobile I/O the clock master by setting the clock source to internal in MIO Console. Set the external device to lock to its digital input, or wordclock if the external device has a wordclock input. If you use wordclock, make sure to connect a 75 Ohm BNC cable between Mobile I/O and the external device.

If you find that you are getting clicks and pops in the audio coming from the external device, try clocking in the other direction. Set the external device to use its internal clock and set the Mobile I/O clock source to digital or wordclock. Remember to choose the appropriate clock rate (single or double) for the sample rate you will be working at.

Now that all of the digital devices are playing nicely with each other, let's take a look at some possible applications with this setup.

Effects for Tracking

Many performers (especially singers) find it easier to get a good take if they have some sweetening effects in their headphone mix while tracking. These will usually be temporary effects and will not neccesarily make it into the final mix but they can make the difference between a good take and a bad take. In this example, we will be using an effects device connected to Mobile I/O's AES ports to add some reverb to the headphone mix we make in MIO Console.

Setting up the routing

The first thing we need to do is configure our mixers and routing. For this setup we will use two mixers in MIO Console: one for the actual headphone mix and one for the send mix to the reverb. To do this we need to bring up the Mix/Output Routing view in MIO console:

In its default state, the WIDE mixer is configured with only the Analog Inputs assigned to the mix busses, so we need to add the Digital inputs. To do this we simply click in the assign tiles under the Digital inputs for the mix busses we want. For this example we will use mix busses 1 and 2 for the Monitor mix and mix busses 3 and 4 for the effects send mix.

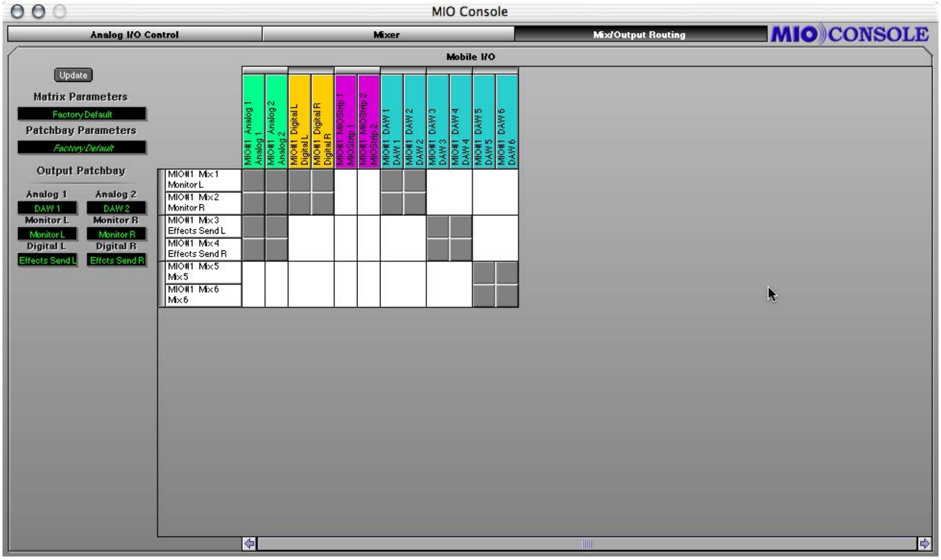

Figure 40: Configuring the Matrix

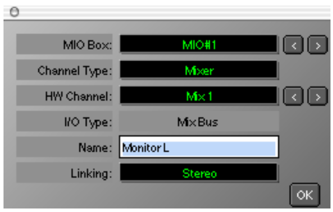

Figure 41: Naming the mix busses

We'll also assign DAW 1 and 2 to mix busses 1 and 2 for DAW playback. We can name the mix busses by clicking on the mix bus label. This will show the matrix configuration window (Figure 41 on page 56).

We'll name the headphone mix Monitor L/R and the send mix Effects Send L/R. You must name each mono bus separately, and the Console combine the names automatically. You can use the tab key to increment through the mix busses as you name them. When you're finished, hit the return key to close the configuration window.

The mixer bus names have propagated through MIO Console which means they will now show up in the Output Patchbay and the Mixer view.

Note that Digital L/R is assigned to the Headphone mix but not the Effects send mix. This is critical because Digital L/R is the effects return, and we definitely do not want to assign the return to the send. If we do, we'll get feedback and no one, especially the singer, will be happy.

To route the mixers to their intended destinations we use the Output Patchbay. We want the Headphone mix to feed the front panel headphone output of Mobile I/O so we'll set the Monitor section of the Output Patchbay to take its input from Monitor L/R.

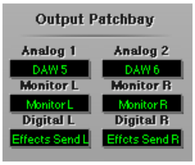

Figure 42: Making the Output Assignment

The Effects Send mix should feed the AES outputs so we'll set the Digital section of the Output Patchbay to take its input from Effects Send L/R.

The routing is now set. We can save the setup for the next tracking session using the **Matrix Parameters** and **Patchbay Parameters** popup menus (see "**Parameter Popup Controls**" on page 50 for details).

Now we'll switch to the Mixer View.

Notice the Mix Tabs on the left are now labeled Monitor L/R and Effect send L/R. Also notice the fader marked Daw 1/2 on the right. We'll use this fader to control the playback level of the DAW tracks we want to overdub against.

If you have signals running to Mobile I/O you should see activity on both the input meters and the mixer meters.

By default the mixer's faders are are at unity gain and the mutes are engaged. We can unmute all the mixer channels by holding the option key and clicking one of the mutes. All of the channels will unmute. But for the time being we'll keep the Digital L/R fader muted.

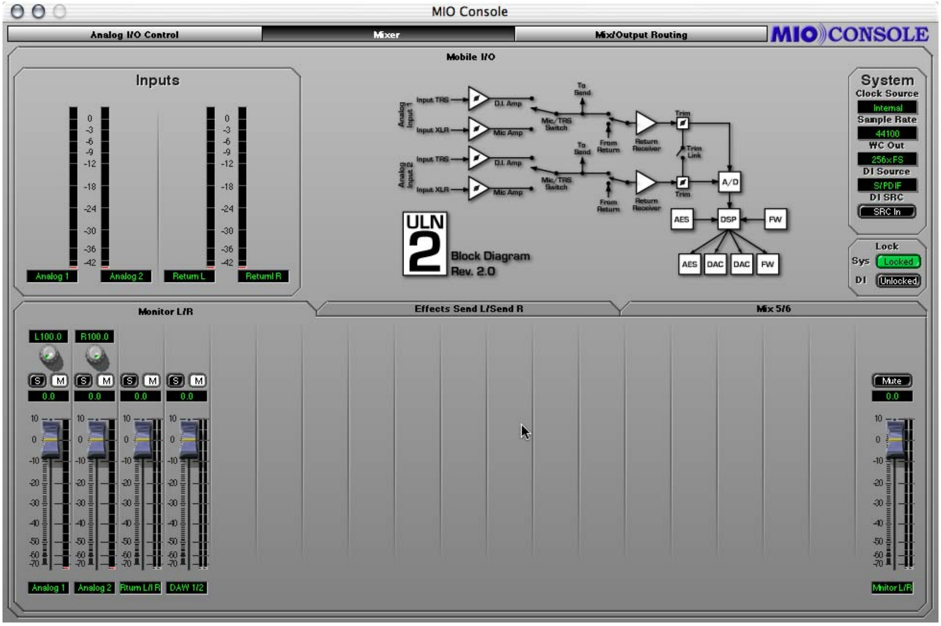

Figure 43: Headphone Mixer

Now we can set up a dry headphone mix. Once we have basic balances set, we can switch to the Effects Send mixer by clicking its mix tab. In the Effects Send mixer we can unmute each channel and set its send level to the reverb, or we can use the Mixer parameters popup (see "**Mixer Pane Tabs**" on page 36) to copy the mix from the Headphone mixer to use as a starting point.

To copy mix parameters from one mixer to another:

1. Click and hold the mix tab for the mixer you want to copy from. A popup menu will appear:
2. Choose Copy Parameters
3. Now click and hold on the mix tab you want to copy to
4. Choose paste parameters
5. The mix parameters have now been copied

Now we can can go back to the Monitor mixer and unmute the Digital fader. The reverb should now be heard in the headphone mix.

Figure 44: Mixer parameters popup

If you want to use this example as a tracking set up, choose **Save As...** from the file menu. This brings up a standard Macintosh save dialog and allows you to save the entire state of MIO Console as a setup document.

The preceding examples should give you a sense of the possibilities that are enabled by the routing and mixing features of Mobile I/O. While this is just a starting point, we have covered all of the basic operations required to manipulate Mobile I/O with complex routing. You should be able to build upon these scenarios to construct routings that suit your needs and your workflow.

Configuring ASIO Hosts [Mac OS 9]

About ASIO™ Technology

ASIO (Audio Stream Input/Output) is Steinberg Soft- und Hardware GmbH's technology standard for interfacing professional audio applications to professional audio interface hardware. Steinberg defined the standard and used it for all of their audio applications, and took the further step of making the specification of the standard available for use by other hardware and software vendors. In doing so, they established it as the de facto standard for making high-resolution, multi-channel, low-latency connections between audio hardware and audio applications on Mac OS 9 and Microsoft Windows.

Virtually every pro audio computer application provides support for communicating with audio hardware via ASIO. As such, it was the natural standard for Metric Halo to support for interfacing with Mobile I/O. Applications (programs) that communicate with hardware via ASIO drivers are called ASIO Hosts.

The ASIO standard is quite rich and provides many possible options for variations in the specification. There are many aspects of the spec that are not supported by all ASIO hosts (e.g. ASIO direct monitoring).

Metric Halo has done extensive testing with the major ASIO hosts, and some of the minor ones, and has worked to ensure maximum compatibility with all hosts. Even if you are using a host that is not specifically discussed here, you are unlikely to encounter problems. If you do, please file a bug report with both Metric Halo and the developer of the ASIO host.

ASIO Basics

How the ASIO Driver Works

The Mobile I/O driver is broken into two components. The first component is composed of the "**MobileIO Driver**" and "**MobileIO Enabler**" files, which are installed in the **Extensions** folder in the **System Folder**, and are the core of the Mobile I/O driver (the enabler file allows the Mac to recognize the Mobile I/O). The **MobileIO Driver** file is responsible for connecting the computer to the Mobile I/O hardware, managing the driver state, facilitating the communication between MIO Console and the hardware, and transporting audio between the hardware and clients. The "**Mobile IO™ ASIO**" file, which is installed in the "**ASIO Drivers**" folder of each ASIO host, is a client of the **MobileIO Driver**, and provides a gateway between the ASIO host and the Mobile I/O driver. This allows us to change and enhance the **MobileIO Driver** without having to have third parties update their applications to support new functionality in Mobile I/O – as long as the functionality can be mapped onto the commands specified by ASIO. With the current version of the Mobile I/O driver, only one ASIO client can be attached to the driver at any one time. This may change in the future.

Where the ASIO Driver File Goes

Most ASIO hosts use an "**ASIO Drivers**" folder located in the same folder as the host application. If your host does not use the standard mechanism for organizing and locating ASIO drivers, you will have to refer to the documentation that accompanies your host for information about the appropriate location for the **Mobile IO™ ASIO** driver file.

ASIO Transport and Sample Rates

The ASIO specification supports a wide variety of audio transport standards, but most applications only implement a small subset of the possibilities. As a practical matter, ASIO supports multichannel transport of 24 bit audio at virtually any sample rate. Many hosts only support a small number of sample rates and may not support all of the sample rates that are available with the Mobile I/O hardware.

Channel Names

ASIO provides a mechanism for the driver to tell the host the names of the channels. This mechanism is not dynamic, so the Mobile I/O driver cannot update the host's names as you adjust the Output Patchbay router. As a result, the input channel names are always accurate, but the mapping between the ASIO output names and the hardware outputs can be adjusted on the fly, and the host may not match the actual hardware routing.

The Output Channel names reported to the host are the direct mapped hardware outputs. With ULN2 there are 6 ASIO output channels are mapped to the hardware in the following order:

1. DAW 1 ⇒ Analog 1
2. DAW 2 ⇒ Analog 2
3. DAW 3 ⇒ Monitor 1
4. DAW 4 ⇒ Monitor 2
5. DAW 5 ⇒ Digital 1
6. DAW 6 ⇒ Digital 2

You can use the table above as a guide to the mapping between ASIO outputs and DAW FireWire channels when you want to adjust the direct routings in the Output Patchbay Router.

Channel Enables

ASIO supports enabling and disabling channels. Some hosts do this automatically (e.g. Digital Performer), some hosts provide manual controls (Cubase, Nuendo), and other hosts do not provide explicit control for enabling and disabling channels. In order to reduce CPU load, Mobile I/O’s ASIO driver takes notice of disabled channels by skipping processing for those channels. For applications that provide manual control over enabled channels, you will get the best performance if you only enable the channels you need. For applications that automatically maintain the enabled channels, you will get the best performance if you only assign channels to outputs you really want to use, and only record enable channels that you intend to record on.

ASIO Buffers

Audio channels are transported individually in buffer sized chunks. The size of the audio buffers has an effect on the CPU load of the audio application, as well as the round-trip latency from input to output when the audio is routed through the host application for monitoring or processing.

Generally, the CPU load increases as the buffer size decreases (this is not true for all hosts; emagic’s Logic Audio is a notable exception). On the other hand, the latency decreases as the buffer size decreases.

Since, in general, you want the lowest CPU load **and** the lowest latency, you will have to make trade-offs.

The WIDE mixer engine in Mobile I/O helps substantially with this issue, because for the common critical monitoring configuration (monitoring while tracking external sources), the Mobile I/O mixer removes all of the ASIO latency from the monitor path and allows you to decouple the latency from the buffer size.

In the case that you are trying to perform with a softsynth running in your host, Mobile I/O's mixer does not help decrease the latency since the signal is being generated on the computer. In this case you'll want to minimize the output latency by selecting the smallest ASIO buffer size possible. This will depend on your computer hardware, the amount of processing you are doing, and the ASIO host you use.

Setting the ASIO Buffer Size

The methods used to set the ASIO buffer size will vary from host to host. Some ASIO hosts provide direct controls for adjusting the buffer size and others do not. If your host supports setting the buffer size directly, you should take advantage of that facility. Most hosts that support controlling the buffer size will automatically save your selected buffer size with your session.

Mobile I/O supports buffer sizes that are powers-of-two. This means that you can use buffers that are 64, 128, 256, 512, 1024, or 2048 samples long. Some hosts that allow you to change the buffer size do not properly follow the ASIO spec and will let you specify buffer sizes that are not supported by the driver. If your host allows you to select a buffer size, please make sure that you choose one of the sizes that Mobile I/O supports (Digital Performer is an example of a host that lets you set the size, but lets you set it to values that are not valid).

If your host does not support setting the buffer size, it will generally use the buffer size that the Mobile I/O driver recommends. The default buffer size for Mobile I/O is 1024 samples per buffer.

The "Driver Options" dialog supported by most ASIO hosts provides a popup menu to allow you to set the default buffer size reported by the Mobile I/O driver to the ASIO host. Most hosts will take notice of a new setting immediately, but some will not. If your host does not appear to notice that the buffer size changed (e.g. the latency does not change), you will have to select another ASIO driver and then switch back to the Mobile I/O driver for the changes to take effect (Cubase VST is an example of a host that will not notice the change automatically).

Sample Size

The Mobile I/O ASIO driver provides the ASIO host with 24-bit samples. If you record with a 16-bit session, there is no mechanism for the ASIO host to inform the Mobile I/O driver that it wants 16-bit samples. It is the responsibility of the host to dither the incoming audio to 16-bit samples before recording them. If the host does not dither the samples to 16-bit, they will be truncated by the host when they are recorded. For best recording quality, use your host's 24-bit recording option.

Clock Sources

The ASIO specification provides the capability for hosts to control the hardware clock source. Some hosts provide a user interface to do this, others do not. If the host does provide an interface to do this, you will be able to select one of the Mobile I/O external clock sources directly from the host. If the host does not provide an interface, you will need to use the MIO Console to select the external clock source. If you have selected an external clock source using either the host or the console, you will not be able to control the system sample rate from the computer. MIO Console will automatically reflect the clock source and sample rate set by the host.

ASIO Direct Monitoring

The ASIO2 standard added the capability for ASIO hosts to control hardware monitor mixing in ASIO hardware. The Mobile I/O ASIO driver maps ASIO Direct Monitoring commands to the Mobile I/O WIDE hardware mixer. The current implementation of the driver does not share mix configuration data with MIO Console, so if you enable ASIO Direct Monitoring, you will be able to change the state of the MIO Mixer without the changes being reflected in the MIO Console user interface.

For basic multitrack foldback, MIO Console provides detailed control and metering and may provide a more streamlined interface for controlling foldback mixes.

The exciting part of ASIO direct monitoring is that it enables "tape-type" cross-faded punch-ins when doing overdubs. Hosts that support tape-type monitoring and ASIO Direct Monitoring will automatically mute the input channel in the monitor mixer when the transport is in playback, and then will unmute the monitor channel when the engineer punches in. The Mobile I/O's interpolated mixer provides clickless, cross-faded transitions for punch-ins and punch-outs, allowing you to accomplish clean, transparent overdubs.

In order to take advantage of the direct monitoring features, you will need to configure both your ASIO host and your Mobile I/O:

1. Enable ASIO Direct Monitoring in your host. Refer to the documentation that comes with your host for details. Later in this chapter we provide details about the ASIO implementation of some of the primary ASIO hosts. That documentation includes details about ASIO Direct Monitoring.
2. Use the Output Patchbay (see "**Output Patchbay Details**" on page 41) in MIO Console to route the appropriate Mix Busses to your monitor output path (e.g. Analog 1/2 or Monitor L/R).
    - The ASIO Direct Monitor will assign your monitored input channels to the mixer that matches the output bus that the input channels are assigned to in the Host.
    - If the input channels are being mixed onto Analog 3/4, the ASIO Direct Monitor will assign the input channels to Mix 3/4 – your mixes must match your outputs for the Direct Monitor to work.
3. Enable the appropriate Monitoring mode in your host software if it supports multiple monitor modes.
4. Make sure that the mixer does not have solo'ed channels and that the Mixer master mute is not enabled – if either of these gotchas are in force, you will encounter problems controlling the mixer from ASIO.

Configuring Nuendo

Nuendo is a Steinberg application, and it uses ASIO as its only audio interface standard. The ASIO driver is selected from the VST Multitrack section of the Device Setup dialog:

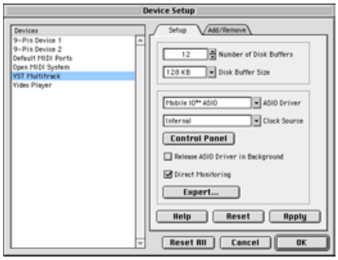

Figure 45: Nuendo VST Device Setup Dialog

If you want to enable ASIO direct monitoring for Nuendo, you enable the "Direct Monitoring" checkbox in the Device Setup dialog.

The "Control Panel" button accesses the Mobile I/O driver options dialog, which allows you to set the ASIO buffer size. Nuendo applies the new buffer size when you click the Apply button or the OK button.

There are some advanced settings in the Nuendo Device Setup Dialog that you need to select for proper operation. You can access these settings by clicking the "Expert…" button. The VST Engine Expert Settings Dialog will appear:

Figure 46: Nuendo VST Expert Options

You should make sure that the Lower Latency and Compatibility Mode checkboxes are enabled – if they are not enabled, you will encounter problems. The other elements of the expert options are not critical, but you may want to use the settings we show here as they have been tested extensively with Mobile I/O.

The Device Setup Dialog also provides access to Mobile I/O's external clock sources from the "Clock Source" popup menu. This menu contains the same options that appear in the MIO Console clock source popup menu.

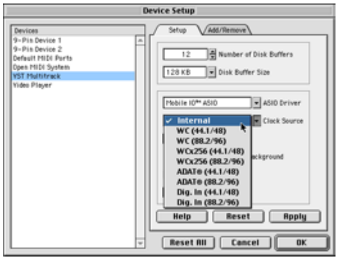

Figure 47: Nuendo Clock Source Popup

Nuendo's VST Inputs window provides access to the controls to name the ASIO input channels. It also allows you to enable input channels. The port names that appear on the left hand side of the window are the physical input ports reported to Nuendo from the driver. The names on the right side of the window are user configurable. The "On" buttons down the middle of the window control which channel pairs have been enabled in the driver. You need to enable these to assign input strips to the hardware channels, but under normal circumstances, you should only enable the channels you need. Especially if you are mixing and pushing your processor to the limit, you can gain some extra processor power by enabling the minimum number of input channels.

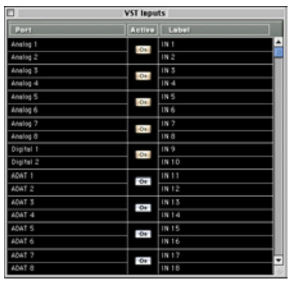

Figure 48: Nuendo VST Inputs

The VST Outputs window allows you to enable which physical outputs you want to mix to. As with the inputs, you can save some processing power by using the minimum outputs required to do the job.

TIP: Enabling and disabling an output bus causes Nuendo to reset the driver. If you encounter any driver synchronization problems, this is the fastest way to reset the driver and force a resync.

Figure 49: Nuendo VST Outputs

Configuring Cubase

Cubase is a Steinberg application that uses ASIO as its only audio interface standard. The ASIO driver is selected from the "ASIO Device" popup menu of the "Audio System Setup" dialog:

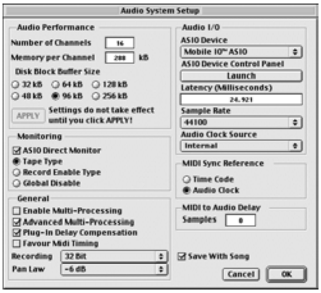

Figure 50: Cubase System Setup

If you want to enable ASIO direct monitoring for Cubase, enable the "ASIO Direct Monitor" checkbox in the Audio System Setup dialog.

The "Launch" button accesses the Mobile I/O driver options dialog, which allows you to set the ASIO buffer size. Cubase does not apply the new buffer size automatically. You will need to switch the ASIO Device popup to another driver and then back to Mobile I/O for the buffer size change to take effect.

The Audio System Setup dialog also provides access to the sample rate and clock source settings for the Mobile I/O driver.

Cubase's VST Inputs window provides access to the controls to name the ASIO input channels. It also allows you to enable input channels. The port names that appear on the left hand side of the window are the physical input ports reported to Cubase from the driver. The names on the right side of the window are configurable by you. The "Active" buttons down the middle of the window control which channel pairs have been enabled in the driver. You need to enable these to assign input strips to the hardware channels, but under normal circumstances, you should only enable the channels you need. Especially if you are mixing and pushing your processor to the limit, you can gain some extra processor power by enabling the minimum number of input channels.

Figure 51: Cubase VST Inputs

The VST Outputs window allows you to enable which physical outputs you want to mix to. As with the inputs, you can save some processing power by using the minimum outputs required to do the job

TIP: Enabling and disabling an output bus causes Cubase to reset the driver. If you encounter any driver synchronization problems, this is the fastest way to reset the driver and force a resync.

Configuring Logic

Logic Audio supports multiple driver types in addition to ASIO. In fact, Logic supports using multiple driver types at the same time. Refer to your Logic documentation for information on how take advantage of this feature.

Configuring Logic to use Mobile I/O as one of its I/O devices is done from the Audio Driver section of the Logic Preferences. This section of the preferences window displays a checkbox for each of the supported driver types. You must enable the ASIO driver type by checking the ASIO checkbox.

Make sure that the other options for the ASIO driver are displayed as shown in ***Figure 52 on page 70***.

To select Mobile I/O as your ASIO driver, choose "Mobile I/O ASIO" from the Driver popup menu.

To adjust the buffer size, click the "Control Panel" button to access the Mobile I/O driver preferences. Logic does not automatically reset the buffer size. In order to have the change take effect, you need to reset the driver. You can do this by cycling the state of the "Larger Process Buffer" checkbox.

You can choose an external clock source from the "Clock Source" popup menu.

t

Figure 52: Audio Driver section of the Logic Preferences

Finally, to enable ASIO Direct Monitoring, you must **uncheck** the "Software Monitoring" check box in the Audio Driver preferences.

Please note that, as of this writing, the Logic implementation of ASIO direct monitoring does not appear to be fully compatible with the MIO ASIO driver, and you may get erratic behavior if you enable ASIO Direct Monitoring.

Configuring Digital Performer

Digital Performer supports multiple driver types including ASIO through the MOTU Audio System (MAS). MOTU has provided an ASIO wrapper for MAS that allows MAS to use ASIO drivers as if they were built into the MAS system, albeit with some extra overhead. Unlike the other ASIO Hosts described in this chapter, MAS appears to have significant sensitivity to ASIO buffer sizes – running MAS with 128 sample buffers will double the CPU load over 1024 sample buffers. As a result, you have to carefully trade-off the total amount of processing against output latency when you work with Digital Performer.

SELECTING MOBILE I/O

Configuring MAS to use Mobile I/O as its I/O device is done from the **Configure Hardware Device** dialog of Digital Performer. In order to select Mobile I/O as your interface, you must first select ASIO from the popup menu at the top of the window (see ***Figure 53 on page 71***). After you have selected ASIO, you will be able to choose **Mobile I/O ASIO** from the ASIO Driver popup menu. MAS may warn you that the driver has not been tested and may cause problems. **You may safely ignore this warning**.

The **Configure Hardware Device** dialog provides a user interface to configure the clock source, sample rate, and buffer size for the Mobile I/O ASIO driver. All of these settings are saved with your session and therefore are the appropriate mechanism for controlling the driver settings.

Figure 53: DP/MAS Configure Hardware Driver dialog box

If you select an external clock source from the Clock Source popup menu, the external source will determine the sample rate and you will not be able to control the rate from the computer.

Clicking the **Control Panel** button will bring up the Mobile I/O Driver preferences dialog. In general, you do not need to use this dialog with MAS since MAS provides a user interface for controlling the buffer size.

The "**Override Buffer Size**" checkbox enables MAS to set the buffer size. If the checkbox is unchecked (unfilled), MAS will use the driver specified default buffer size. If you check the "**Override Buffer Size**" checkbox MAS will set the ASIO buffer size to number in the text box to the right of the checkbox. MAS allows you to set that number to any arbitrary value. The MAS/Mobile I/O Driver combination will not work correctly unless you specify one of the valid buffer sizes (64, 128, 256, 512, 1024 or 2048).

WARNING: Since the buffer size is saved with the session in Digital Performer, and using smaller buffer sizes increases the CPU load, you may encounter problems opening heavily loaded sessions on slower machines. If you overload a machine due to having saved a small buffer size with a session, MAS will warn you that the CPU has overloaded and will allow you to disable your MAS plug-ins. If this happens, open the **Configure Hardware Device** dialog and increase the buffer size. When you dismiss the **Configure Hardware Device** dialog, the plug-ins will be re-enabled.

Enabling Channels

MAS automatically enables and disables ASIO channels based upon the assigned output busses and the record-enabled channels and Aux inputs. You can use this fact to conserve CPU power by only record-enabling channels when you intend to record on them, only turning on hardware aux inputs when you are using them, and only mixing to hardware busses when they are being used.

ASIO Direct Monitoring

Digital Performer does not support ASIO Direct Monitoring.

Configuring SpectraFoo

In order to use ASIO drivers with SpectraFoo or SpectraFoo Complete, you must have the Rad3f18 release or newer. The f18 release is a free upgrade available from Metric Halo.

SpectraFoo is a multi-driver application that supports all ASIO devices using SpectraFoo ASIO support. To choose an ASIO driver, you select ASIO from the SpectraFoo "**Audio I/O**" menu.

After ASIO has been selected, you configure the ASIO driver from the "**ASIO Driver Configuration**" dialog. Select the the "**Configure Hardware…**" command from the "**Audio I/O**" menu to show the "**ASIO Driver Configuration**" dialog:

Figure 54: SpectraFoo Audio I/O menu

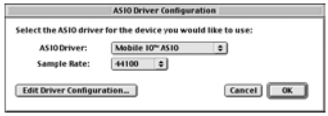

Figure 55: SpectraFoo ASIO Driver Configuration dialog

To use Mobile I/O with SpectraFoo, select "**Mobile IO™ ASIO**" from the ASIO driver popup menu. After you have selected the driver, you can set the Sample Rate from the Sample Rate popup menu. SpectraFoo supports all of Mobile I/O's sample rates.

Use the "**Edit Driver Configuration…**" button to adjust the ASIO buffer size. SpectraFoo adjusts the buffer size as soon as you change it.

SpectraFoo currently uses all the available hardware channels for instant matrixing, so there is no way to enable or disable ASIO channels.

Configuring CoreAudio Hosts [Mac OS X]

About CoreAudio™ Technology

CoreAudio is Apple® Computer's technology standard for interfacing applications to multichannel audio hardware with professional quality. Apple defined the standard and made it the primary interface for audio in OS X. It provides the mechanism for making high-resolution, multi-channel, low-latency connections between audio hardware and audio applications on Mac OS X.

All Mac OS X computer applications provide support for communicating with audio hardware via CoreAudio. As such, it was the natural standard for Metric Halo to support for interfacing with Mobile I/O. Applications (programs) that communicate with hardware via CoreAudio drivers are called CoreAudio Hosts.

The CoreAudio standard is quite rich and provides a number of places for hosts to not support the specification correctly. There are some hosts that were implemented early that did not get support for multichannel/multi-stream devices implemented correctly. Most of these hosts have been fixed or are in the process of being fixed now. If you encounter any problems with specific hosts, please let us know about it – but also please let the host venodr know about it. CoreAudio puts many more requirements on hosts than it does on drivers, so it is very likely that any such problems are in the host. Finally, CoreAudio is very new and still evolving – many functions that exist in ASIO (like Direct Monitoring) are not available in CoreAudio yet. As the standard evolves, the MIO driver will evolve to keep pace with it.

Metric Halo has done extensive testing with the major CoreAudio hosts, and some of the minor ones, and has worked to ensure maximum compatibility with all hosts. Even if you are using a host that is not specifically discussed here, you are unlikely to encounter problems. If you do, please file a bug report with both Metric Halo and the developer of the CoreAudio host.

CoreAudio Basics

How the CoreAudio Driver Works

The Mobile I/O CoreAudio driver is provided by Mac OS X KEXT. The KEXT is a Mac OS X kernel extension. This extension enhances the Mac OS operating system to provide support for communicating with the Mobile I/O hardware. The Mobile I/O driver is implemented as a KEXT due to the requirements of CoreAudio.

The KEXT is provided in a Mac OS X bundle called "MobileIODriver.kext". This bundle is installed in the /System/Library/Extensions folder. Since this folder is managed by the system, you will have to have administrator access on the computer to install the driver.

The CoreAudio driver provides the required information for CoreAudio to discover and control Mobile I/O. Once the driver has been installed, CoreAudio will automatically find Mobile I/O units as they are attached to the computer and will publish the availablity of the hardware to all interested CoreAudio hosts.

CoreAudio is inherently a multiclient interface -- more than one CoreAudio host can communicate with the hardware at the same time. Mulitple hosts can recieve the audio from a Mobile I/O at the same time, and multiple hosts can send audio to the Mobile I/O at the same time. When multiple hosts send audio to the hardware at the same time, CoreAudio will automatically mix the audio before it is sent to the Mobile I/O. While this mutlticlient operation is a very cool feature of CoreAudio, and can be very helpful for many operations, you must be careful about unintended interactions. In particular, it is very easy to set up the system such that sounds from programs like email clients and other productivity tools will be mixed into your main audio stream (this happens when you set up the default audio output path so that you can use iTunes with the Mobile I/O). If you are not careful, you can check your mail and have the "Mail recieved" sound printed into the bounce that you are doing in the background.

CoreAudio Transport and Sample Rates

CoreAudio supports a wide variety of audio transport standards. As a practical matter, CoreAudio supports multichannel transport of 24 bit audio via floating point streams at virtually any sample rate. Many hosts only support a small number of sample rates and may not support all of the sample rates that are available with the Mobile I/O hardware.

Channel Names

CoreAudio provides a mechanism for the driver to tell the host the names of the channels. Some hosts do not use this information and "make up" their own names for the channels. This mechanism is not dynamic, so the Mobile I/O driver cannot update the host's names as you adjust the Output Patchbay router. As a result, the input channel names are always accurate, but the mapping between the CoreAudio output names and the hardware outputs can be adjusted on the fly, and the host may not match the actual hardware routing.

The Output Channel names reported to the host are the direct mapped hardware outputs. With ULN2 there are 6 CoreAudio output channels are mapped to the hardware in the following order:

1. DAW 1 ⇒ Analog 1
2. DAW 2 ⇒ Analog 2
3. DAW 3 ⇒ Monitor 1
4. DAW 4 ⇒ Monitor 2
5. DAW 5 ⇒ Digital 1
6. DAW 6 ⇒ Digital 2

You can use the table above as a guide to the mapping between CoreAudio outputs and DAW FireWire channels when you want to adjust the direct routings in the Output Patchbay Router.

Channel Enables

CoreAudio supports enabling and disabling audio streams. This is a relatively new feature of CoreAudio, and many hosts do not yet support it. It is not clear exactly how the user would control this functionality at this time. Please look at the section on Channel Enables in the chapter on ASIO to see how representative hosts deal with this in OS 9 under ASIO. It is likely that hosts will support a similar mechanism under OS X once they have implemented support for stream enables. The MIO driver is aware of stream eanbles, and will take notice of disabled channels by skipping processing for those channels once hosts take advantage of this CoreAudio feature. For applications that provide manual control over enabled channels, you will get the best performance if you only enable the channels you need. For applications that automatically maintain the enabled channels, you will get the best performance if you only assign channels to outputs you really want to use, and only record enable channels that you intend to record on.

COREAUDIO BUFFERS

Audio channels are transported individually to the host in buffer sized chunks. The size of the audio buffers has an effect on the CPU load of the audio application, as well as the round-trip latency from input to output when the audio is routed through the host application for monitoring or processing.

Generally, the CPU load increases as the buffer size decreases (this is not true for all hosts; emagic's Logic Audio is a notable exception). On the other hand, the latency decreases as the buffer size decreases.

Since, in general, you want the lowest CPU load **and** the lowest latency, you will have to make trade-offs.

The WIDE mixer engine in Mobile I/O helps substantially with this issue, because for the common critical monitoring configuration (monitoring while tracking external sources), the Mobile I/O mixer removes all of the ASIO latency from the monitor path and allows you to decouple the latency from the buffer size.

In the case that you are trying to perform with a softsynth running in your host, Mobile I/O's mixer does not help decrease the latency since the signal is being generated on the computer. In this case you'll want to minimize the output latency by selecting the smallest CoreAudio buffer size possible. This will depend on your computer hardware, the amount of processing you are doing, and the CoreAudio host you use.

SETTING THE COREAUDIO BUFFER SIZE

The methods used to set the CoreAudio buffer size will vary from host to host. Some CoreAudio hosts provide direct controls for adjusting the buffer size and others do not. If your host does not support setting the buffer size directly, you will have to use the host's default buffer size.

Sample Size

The Mobile I/O CoreAudio driver provides the CoreAudio host with 24-bit samples in 32-bit floating point streams. It is the responsibility of the host to dither the incoming audio to 16-bit samples before recording them, if you record 16-bit. If the host does not dither the samples to 16-bit, they will be truncated by the host when they are recorded. For best recording quality, use your host's 24-bit recording option.

Clock Sources

The CoreAudio specification provides the capability for hosts to control the hardware clock source. Some hosts provide a user interface to do this, others do not. If the host does provide an interface to do this, you will be able to select one of the Mobile I/O external clock sources directly from the host. If the host does not provide an interface, you will need to use the MIO Console to select the external clock source. If you have selected an external clock source using either the host or the console, you will not be able to control the system sample rate from the computer. MIO Console will automatically reflect the clock source and sample rate set by the host.

Troubleshooting Guide

Computer Does Not See Mobile I/O

If you attach Mobile I/O to your computer, and the computer is unable to communicate with the Mobile I/O hardware there are five basic possibilities for the source of the problem:

1. The Mobile I/O is not powered up
2. The Software is not installed properly
3. The FireWire bus did not reset correctly
4. The FireWire cable is bad
5. The FireWire hardware has been damaged

Mobile I/O is not powered up

The first thing to check is that the Mobile I/O is, in fact, powered up.

If Mobile I/O is powered up and booted properly, the Power, Sample Rate, and Locked front panel indicators will be illuminated. If these indicators are not illuminated, the Mobile I/O is not powered properly or the unit's firmware has been corrupted. If you determine that you are powering Mobile I/O properly and the indicators are not illuminated, you will need to contact Metric Halo support.

If you are bus powering the Mobile I/O, there is a possibility that you have overloaded the power rating of the power source. Please see the troubleshooting section "**Not enough power on the bus**" on page 87 for details on troubleshooting this problem.

If the Mobile I/O is properly powered, then check the next possibility.

Software Is Not Installed Properly

In order for the computer to properly communicate with the Mobile I/O the various components of the driver software need to be installed correctly. If the software is not installed correctly, the communication between the computer and Mobile I/O will fail in various ways.

FOR OS 9

1. If the "MobileIO™ Enabler" file is not installed in the "System Folder:Extensions" folder, the system will not register the attachment of a Mobile I/O device to the bus.
   - The symptom of this is that the Front Panel FireWire indicator is illuminated, but the ASIO driver reports that the Mobile I/O is not present and

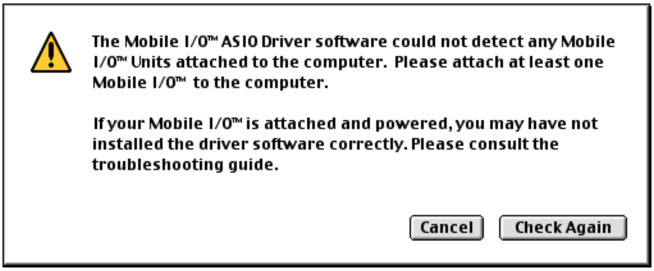

Figure 56: Mobile I/O not found dialog

   the Box Info section of the MIO Console does not register the presence of the box.
   - This symptom is also present if a copy of the "MobileIO Driver" file is installed in the wrong location (see below).
   - To correct this condition, make sure the "MobileIO™ Enabler" file is installed in the "System Folder:Extensions" folder and then reconnect the Mobile I/O to the computer.
2. If the "MobileIO Driver" file is not installed in the "System Folder:Extensions" folder, the MIO Console will not start-up and the ASIO driver will not connect properly.
   - To correct this condition, make sure the "MobileIO Driver" file is installed in the "System Folder:Extensions" folder and then reconnect the unit to the computer.
3. If you have accidentally put a copy of the "MobileIO Driver" file in the same folder with the MIO Console application or with any of the ASIO driver files, the symptoms will be similar to what you see with condition #1. The Console or the ASIO driver will appear to run properly, but it will not see the box connected to the computer.
   - To correct this condition, check for a copy of the "MobileIO Driver" file in any location other than the "System Folder:Extensions" folder. If a copy of the file exists, delete it from your disk or, at the very least, ensure that it

is not in the same folder with the MIO Console application or any of the ASIO Driver files.

4. If the "Mobile IO™ ASIO" file is not installed in the appropriate folder for each of your hosts, the corresponding host will not show the Mobile I/O as an available ASIO driver. Most ASIO hosts use a folder called "ASIO Drivers" located in the same folder as the host application. If your specific host does not use this convention, you will have to check the documentation that accompanied your host or contact the manufacturer of the host for assistance.
   - Note that you need to install a copy of the "Mobile IO™ ASIO" file in the appropriate folder for ***each*** host you will use on your computer. In general, ASIO Drivers are not shared between hosts.

FOR OS X

1. IThe MobileIODriver.kext is not properly installed in the /System/Library/Extensions folder of your computer. In this case
   - The symptom of this is that the Front Panel FireWire indicator is illuminated, but the Mobile I/O does not appear as a Sound Output device in the Sound panel of the "System Preferences" application, nore does it appear in the Apple Audio/MIDI Setup Application.
   - To correct this condition, make sure the MobileIODriver.kext file is installed correctly, reboot and then reconnect the Mobile I/O to the computer.

If the software is installed properly, check the next possibility.

THE FIREWIRE BUS DID NOT RESET CORRECTLY

When a device is plugged into the FireWire bus, a FireWire bus reset occurs automatically. The bus reset interrupts bus activity and reconfigures the bus so that all devices on the bus become aware of all the other devices on the bus. Sometimes the reset does not complete successfully, and the bus becomes partially hung. In this case, the "FireWire" indicator on the front panel of the Mobile I/O will not be illuminated. When the "FireWire" indicator on the front panel is not illuminated, the Mobile I/O cannot transport audio over the FireWire bus.

Generally, this condition can be fixed by disconnecting the Mobile I/O from the bus and reconnecting it.

If the disconnect/reconnect cycle does not fix the problem, another device on the bus may be interfering with the proper operation of the bus. If you have other devices on the bus, try disconnecting them from the bus and only using the Mobile I/O.

If removing other devices from the bus solves the problem, it is likely that there is a problem with either one of the devices you removed or with one of the cables connecting the devices. You'll need to isolate the problem component.

If removing the other devices from the bus does not fix the problem, check the next possibility.

The FireWire Cable is Bad

Metric Halo provides two high-quality overspec'ed FireWire cables for use with Mobile I/O and we recommend you use them. For various reasons you may decide to use other cables than the ones provided by Metric Halo. Under ideal circumstances all FireWire cables will provide years of service. However, cables will and do go bad. Cable failures can be difficult to track down. If you are experiencing problems with connecting or bus powering Mobile I/O you should try swapping the cable with another known-good cable.

If the FireWire cable is not the source of the problem, check the next possibility.

The FireWire Hardware has been Damaged

If all else fails, it may be that the FireWire hardware on either the Mobile I/O or the computer has been damaged. While this is an exceptionally rare occurrence, it is a possibility. The FireWire hardware can be damaged in the following ways:

1. If you insert a FireWire cable into a port upside down, it will damage the FireWire port and/or the connector. It is difficult to insert the connector upside down, but it is possible to force it. Never force a FireWire connector!
2. It requires significant pressure, but it is possible to force a FireWire connector over a male XLR connector pin. If you do this, the con-

nector will be shorted and it will destroy the port on the other end. Again, never force a FireWire connector.

3. Some devices that are bus-powerable and conform to the IEEE1394 standard will return power to the remote FireWire port if a power ground fault occurs. If the remote port is protected against this situation, nothing will happen. If the device does not use bus power, nothing will happen. But, if the device is fully compliant, uses bus power, and the remote device is not protected and supplies a high enough voltage on the bus, the remote device port will self-destruct.

If the FireWire hardware on the computer has been damaged, it will not communicate with any FireWire devices. Be sure that you are not checking this case with a bad cable, as a bad cable can make it seem like the FireWire hardware has failed since it will consistently keep devices from connecting properly to the computer. If the computer is damaged, you will need to contact the manufacturer for a repair or, as a stopgap measure you can use a third-party FireWire adapter card.

If the FireWire hardware on the Mobile I/O has been damaged the MIO will not communicate with any other devices. In this case, please contact Metric Halo support for help in getting your Mobile I/O hardware repaired.

When I Switch to MIO Console, Playback Stops [OS 9]

Some ASIO Hosts will automatically stop playback when the host is switched to the background. This occurs when you switch to MIO Console (or any other application). All of these hosts have an option to continue playing in the background; you need to ensure that the option is enabled. In addition to the host's own preference, many of the hosts that support MIDI sequencing will stop playing if the MIDI system (OMS or FreeMIDI) has been configured to release the MIDI port when in the background. While it is a bit confusing, you need to ensure that the MIDI system is configured to not share or release the MIDI port.

A more subtle issue is that MIO Console has an option to optimize its drawing operations which can be interrupted by background application activity. If the "**Allow Background Apps to Run**" item in the MIO Console's **Edit** menu ***does not*** have a diamond mark (◊) next to it, the console will lock other applications out while it is the active application. This is not a problem for most ASIO Hosts, but it does affect Digital Performer, which will stop playing after 10-20 seconds. To resolve this issue, make sure that the "**Allow Background Apps to Run**" item in the MIO Console's **Edit** menu ***does*** have a diamond mark (◊) next to it.

Where Are the {Extensions, ASIO Drivers} Folders

The **Extensions** folder is a system wide location for all third-party extensions to the Mac OS. It is located within the currently active **System Folder**. You can generally find it by using Sherlock to find files named "Extensions". Consult the documentation for your Macintosh for more information. Please note that localized versions of the Mac OS use different, translated names for both the **Extensions** folder and the **System Folder**.

The **ASIO Drivers** folder, by convention, is the location that each ASIO host uses to manage ASIO drivers. Most hosts follow the convention that ASIO drivers are stored in a folder named "ASIO Drivers" that is located in the same folder as the host application. Each host has its own **ASIO Drivers** folder; they are not shared. You need to install the Mobile IO ASIO driver in the **ASIO Drivers** folder for each host you want to use with Mobile I/O. If your host does not follow the convention, you will have to consult the documentation provided with the host to determine where to install the ASIO driver.

Ground Loops

Audio systems, in general, are susceptible to ground loop problems. Digital Audio Interfaces for computers are even more susceptible to grounding issues since they must interface with the computer's system ground, which tends to be much more dirty than the ground used by audio gear. By taking care when you connect the various components of your audio system you can avoid the hums, buzzes, and noises that characterize ground loops and other grounding problems.

First of all, most grounding issues go away if you utilize balanced interconnects between your audio gear. Balanced interconnects inherently reject ground differentials and common mode interference introduced by grounding problems. Balanced connections are not much more expensive than unbalanced connections and solve so many problems that if both ends of the connection support balanced interconnect, you should not even consider using unbalanced cables.

You may get the idea that we hate unbalanced connections. You're right. We do. You should too.

If you have to use unbalanced connections, or if any ground-related problems remain, you will find that the key to the issue is ensuring that you have a common hard ground between all the gear that you are interfacing. This is commonly referred to as a technical ground. A technical ground is characterized by a consistent low impedance path between each device and a common reference ground, ideally connected directly to earth ground. The above is sometimes difficult due to electrical wiring problems in the house, studio, or stage you are using. In the extreme case, you may need to hire a qualified electrician to untangle and correct electrical service problems in your working environment.

Unbalanced connections are a fact of life when interfacing with guitar amps, and, paradoxically, guitar amps are extremely sensitive to grounding issues since they use so much gain to achieve the effect of a "Guitar Amp". If you will be interfacing with guitars and guitar amps, you need to be very careful about grounding.

Common electrical wiring approaches to residential installations, and sub-par studio and stage installations use daisy-chained grounds for ease of installation and economy. Unfortunately, daisy-chained grounds can introduce signficant ground differentials between sockets, and these differentials can vary depending on other loads (like refrigerators, TV's and other household appliances) on the circuit.

Other problems with electrical service installations are improper wiring of power phases to the three-phase service and improper connections between the safety ground and hot legs of the three-phase service. These types of problems tend to be characterized by loud 60Hz hums in the audio system. Unfortunately, these types of problems extend well beyond noise in your audio system to genuine safety hazards. If you determine that your electrical wiring has problems beyond a simple daisy-chained ground, you should consult a licensed electrician immediately, as ignoring these problems can damage either you or your gear.

If you do not have a well implemented technical ground, you will want ensure that all of the devices in your audio system are plugged into the same phase and same ground. You can generally accomplish this by running all your gear off of the same socket (using a power strip or power conditioner) if your gear uses less power than is supplied by a single circuit from your premises wiring (generally 10-15 amps in residential installations and 20 amps in commercial installations).

It is usually a bad idea to put some devices in your system on a power conditioner and other devices on a separate strip, socket or conditioner, unless you have a technical ground. The power conditioner can introduce a ground differential.

The power supply provided with Mac laptops does not have a hard ground. This means that if the laptop is plugged in, it will dump high frequency buzz into the ground. That ground is shared with the Mobile I/O Firewire cable. If Mobile I/O will be connected unbalanced to other audio gear, the ground buzz can contaminate the signal if the Mobile I/O is not hard-grounded to the same ground as your other audio gear. To hard ground the Mobile I/O you will need to use a 3-pin power cable on the Mobile I/O power supply and power the Mobile I/O with the power supply. Plug the 3-pin IEC power cable into the same circuit and same ground as your other gear.

On the other hand, if you are encountering ground loop problems while operating with the Mobile I/O's power supply, you may find that lifting the Mobile I/O's ground resolves the problem. This can be accomplished by using a 2-Pin IEC cable (without the third ground pin), or by using a ground lift block (generally available in hardware stores, also known as a 3 pin to 2 pin converter). In general, it is better to resolve the fundamental grounding problems in your system, but this is a quick fix that may help. There are no hard and fast rules for solving this type of problem other than fixing the fundamental grounding issues, so if you go this route, you will have to experiment with lifting various grounds in your system until you find the magic combination. Or switch to balanced interconnects.

Finally, the Apple Cinema Display has a known issue with its backlight dimmer. If you run the Apple Cinema Display with its backlight at anything other than full brightness, the backlight dimmer will introduce a midrange buzz into the system ground which will appear in unbalanced interconnects (input and output) with Mobile I/O. This issue affects other devices that connect to the computer's system ground. The work around is to run the display at full brightness, or use balanced interconnects.

Firmware Update Problems

For details on updating the firmware of Mobile I/O refer to Appendix 1.

It is possible for firmware updates to "not take". This appears to be related to DSP loading issues in the Mobile I/O, other devices on the FireWire bus, and the state of the FireWire system software on the Mac. If you have problems with updating the firmware, try the following procedure:

1. Remove all devices from the FireWire bus
2. If your Mobile I/O is using external power, disconnect the power
3. Reboot your computer
4. Attach the external power supply to the Mobile I/O **while holding down the front panel Mute button;** this will boot the Mobile I/O into the safety boot firmware
5. Connect the Mobile I/O to your computer
6. Run the firmware updater

Since Mobile I/O implements safe-boot and safe firmware update, you should always be able to use this procedure to update the firmware, even if something goes horribly wrong (like losing power during an update).

Bus Powering Mobile I/O

If you are bus powering the Mobile I/O, there is a possibility that you have overloaded the power rating of the power source.

Not enough power on the bus

While all Macintosh computers with built-in FireWire supply bus power, some models do not provide enough power on the bus to power Mobile I/O. If this is the case, you will generally find that the Mobile I/O will boot on initial connection, but will then lose power or will reboot repeatedly after a short period of operation.

Some Mac models provide enough power if they are plugged into the wall, but will not provide enough power while running on batteries. If the computer does not provide enough power, you will need to use an external power source with Mobile I/O.

The external power supply provided with Mobile I/O is the perfect solution if you are using Mobile I/O in an environment where AC power is available. The external power supply will actually provide power to the bus and can be user to power other bus-powered peripherals (see "**Other Bus Powered Devices**" on page 89).

If AC power is not available, you will need to use an external battery-based power source to power Mobile I/O. Any source that provides 9V-30V and can support 12-15W of power consumption will work well with Mobile I/O. Check with Metric Halo for specific recommendations.

When using an external battery source, DO power Mobile I/O directly from the battery – not through an inverter. DON'T power the computer with an external battery and use the computer to power Mobile I/O; doing so will not resolve your bus power problems, and it will give you more limited run times. If you need to use the external battery with the computer use two batteries or split the DC supply at the battery and power both the Mobile I/O and the computer.

Screen Backlight Issues

When Mobile I/O is attached to FireWire bus without any external power applied, there is a small current surge as the Mobile I/O power supply charges up. Depending on the firmware revision and model type of your PowerBook and what other devices are attached to or installed in your computer, this inital connection may trick the power manager in the computer into thinking that it is about to lose power. In this circumstance, the power manager will shut-down the computer's backlight. The computer has not crashed or gone to sleep; only the backlight has been shut down.

There does not seem to be anyway to turn the backlight back on without putting the machine to sleep. The workaround for this condition is to put the machine to sleep (either by closing the lid or pressing <command><shift>0 [zero]) and then waking it up again with the Mobile I/O still attached. This will not happen if the Mobile I/O is externally powered.

With early Titanium PowerBooks, especially with early versions of Apple's firmware, spinning up the CDROM or hard disk while bus powering Mobile I/O may trigger the screen backlight issue. We recommend that you remove all media from the CD/DVD drive and set the hard disk to never sleep when you are bus powering Mobile I/O with these machines.

Other Bus Powered Devices

Mobile I/O consumes enough power that it is very unlikely that you will be able to successfully bus power Mobile I/O and any other bus-powered device (except for a hub) from the same computer. If you plan on using other bus-powerable devices with your computer, you will need to either self-power your other devices or self-power the Mobile I/O. It is probably best to use the Mobile I/O's power supply in this situation since Mobile I/O will then provide approximately 30 Watts of power to the bus (roughly 3x what most Portable Mac's will supply). This will allow you to power all the rest of your devices without any concern of running out of power.

Appendix I – Updating your Firmware

The Mobile I/O is a complex device with a complex DSP–based signal processing and control architecture. One of the major strengths of Mobile I/O's design is that the operating system of the box can be upgraded at any time by updating the firmware. The firmware provides data to the hardware upon system boot that configures both fundamental aspects of the hardware and the operating system for the box. This data is stored in a memory device on the Mobile I/O motherboard. The data can be updated at any time, but it will be maintained indefinitely, even without any power being applied to the Mobile I/O.

Since the hardware itself can be reconfigured by the firmware, this approach allows Metric Halo to make major enhancements to Mobile I/O without any physical changes to the hardware. In the past we have used software deployed firmware updates to increase the FireWire access speed, provide independent heaphone channels, and improve the converter sound quality over its already exceptional character.

Since the firmware updates exist simply as data, they can be sent to you in a variety of ways, whether via CD, email or download from our website. The MIOConsole Application provides a built-in tool to update firmware directly from the console. The following section describes how to use the built-in tool. If you need information about the previous version of the Firmware Update tool (which was only available in OS 9), please check the very end of this Appendix.

You may have had the experience of updating the firmware for your computer in the past. As you may know, this can be a stressful procedure, since there is a moment while the old firmware is being replaced by the new firmware, and if the process is interrupted you may be left with no firmware at all. Metric Halo has addressed this issue with a "safe firmware update" procedure. The Mobile I/O uses a dual-boot procedure. The first boot happens in the first 100ms (about 0.1 seconds) and has been extensively tested. It is smart enough to do two things:

1. It can boot the secondary boot image
2. It can update the secondary boot image over the FireWire bus

Actually, the primary boot firmware is much smarter than that. The box is completely functional on the primary boot, but all of the more advanced features of the box are enabled by the second boot. The firmware revision of the primary boot is **1.1.00**.

As soon as the primary boot image has booted, it checks the secondary boot image, and if the secondary boot image is installed and not corrupted in any way, the system immediately boots the secondary image. If the secondary image is corrupted or if you have held down the front-panel **Mute** button during the initial boot process, the Mobile I/O will not boot the secondary image and will stay in "**Safety Boot Mode**". This is a mechanism you can use if you install firmware that has problems and you need to back up or install a newer image.

The safe boot mechanism allows you to back out of a firmware update if you find that the new firmware has some problem that was not present in a previous version of the firmware. Metric Halo support may ask you to do this during troubleshooting if you encounter any problems.

In the future, Metric Halo may change the methods or tools used to find, download, and accomplish firmware updates. If the tools change, they will be accompanied by an updated version of this Appendix.

Installing a firmware update

In order to install a firmware update, follow these steps:

1. If there is an associated driver update, install the new driver:
   - [OS 9] place the MobileIO Driver and MobileIO Enabler in your extensions folder inside your system folder.
   - [OS X] Double click the driver installer package and use the Apple installer to update the MIO Driver package.
2. Reboot your computer.
3. Make sure your Mobile I/O is powered up and connected to your computer.
4. Run the MIO Console.
5. Select "**Update Firmware...**" from the "**Utilities**" Menu. You will see the following choose file dialog:

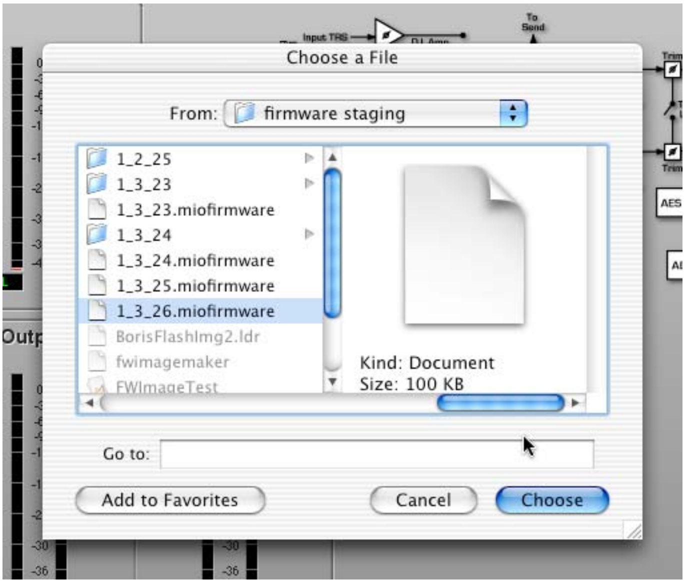

Figure 57: Update Firmware Choose Dialog

6. Select the new Firmware file to install on your Mobile I/O hardware, and click the **Choose** button. Firmware files will be supplied by Metric Halo with a name that contains the firmware version number in the following format: <firmware_version_number>.miofirmware. Only valid firmware files will be selectable in the dialog box. (In OS 9, only valid files will be displayed in the dialog box).

The Console will find your Mobile I/O on the FireWire bus and begin sending commands to it. While the update is taking place, the console will freeze, and when the Console is finished updating the firmware it will begin responding again. We recommend that you quit the Console and run it again. The Console should show the new firmware revision information in the Box Info area of the console window.

In OS 9, copy any included Mobile I/O ASIO Driver into the ASIO folders of the audio applications you want to use, reboot your computer, and you're ready to go.

Rolling back your firmware

If you find that you have problems with any given release, you can always go back to a previous release by downloading a package from

http://www.mhlabs.com.com/mio

and following the update instructions in that package. Please do not roll back the firmware to an earlier version than was originally supplied with your unit unless instructed to do so by Metric Halo.

Instructions on the OS 9 only MIOBootInstaller Tool (obsolete)

Metric Halo provides a firmware update tool that allows you to download an updated firmware image to the Mobile I/O from your computer. The following section describes how to use the previous, Mac OS 9 only version of the firmware update tool, called **MIOBootInstaller.**

You may have had the experience of updating the firmware for your computer in the past. As you may know, this can be a stressful procedure, since there is a moment while the old firmware is being replaced by the new firmware, and if the process is interrupted you may be left with no firmware at all. Metric Halo has addressed this issue with a "safe firmware update" procedure. The Mobile I/O uses a dual-boot procedure. The first boot happens in the first 100ms (about 0.1 seconds) and has been extensively tested. It is smart enough to do two things:

1. It can boot the secondary boot image
2. It can update the secondary boot image over the FireWire bus

Actually, the primary boot firmware is much smarter than that. The box is completely functional on the primary boot, but all of the more advanced features of the box are enabled by the second boot. The firmware revision of the primary boot is **1.1.0**.

As soon as the primary boot image has booted, it checks the secondary boot image, and if the secondary boot image is installed and not corrupted in any way, the system immediately boots the secondary image. If the secondary image is corrupted or if you have held down the front-panel **Mute** button during the initial boot process, the Mobile I/O will not boot the secondary image and will stay in "**Safety Boot Mode**". This is a mechanism you can use if you install firmware that has problems and you need to back up or install a newer image.

The safe boot mechanism allows you to back out of a firmware update if you find that the new firmware has some problem that was not present in a previous version of the firmware. Metric Halo support may ask you to do this during troubleshooting if you encounter any problems.

In the future, Metric Halo may change the methods or tools used to find, download, and accomplish firmware updates. If the tools change, they will be accompanied by an updated version of this Appendix.

Installing a firmware update

In order to install a firmware update, follow these steps:

Figure 58: MIO Package

1. If there is an associated driver update, place the MobileIO Driver and MobileIO Enabler in your extensions folder inside your system folder.
2. Reboot your computer.
3. Make sure your Mobile I/O is powered up and connected to your computer.
4. Double click on MIOBootInstaller.

You will see the following:

Figure 59: MIOBootInstaller Window

MIOBootInstaller will find your Mobile I/O on the FireWire bus and begin sending commands to it. When MIOBootInstaller is finished updating the firmware, it will say "Wait 10 seconds then hit return..." After you hit return, MIOBootInstaller will find your Mobile I/O again and display your unit's serial number and the firmware revision which should match the firmware revision identified with the package.

Copy the Mobile I/O ASIO Driver into the ASIO folders of the audio applications you want to use, reboot your computer, and you're ready to go.

Rolling back your firmware

If you find that you have problems with any given release, you can always go back to a previous release by downloading a package from

http://www.mhlabs.com.com/mio

and following the update instructions in that package.

Index

Numerics

A

B

C

M

N

O

P

R

S