Review: E-MU® 1820M
by Rick Paul - *12th May 2005 -*

It's not unusual with technology-based products to see products at any given price point getting more and more powerful over time. Neither is it surprising to see products with a certain level of power and functionality get less and less expensive. Every once in awhile, though, a product comes along that shakes things up a bit, raising the power and/or functionality bars significantly while coming in at a price point that seems too good to be true, thus making you wonder if something is "wrong with this picture." Oftentimes our natural suspicions in this area are well founded, and serve to demonstrate the truth of the old saying, "if something sounds too good to be true, it probably is." Occasionally, though, the universe decides to mess with our minds and expectations, and a revolution, or at least a new evolutionary branch, is born.

When E-MU® introduced their Digital Audio Systems (DAS) family of products awhile back, I might have been inclined to pass the products off as yet another line of cost-effective audio interfaces had it not been for two specific points mentioned in their announcements. One claimed to offer "mastering-grade 24-bit, 192 kHz converters -- the same A/D converters used in Digidesign®'s flagship ProTools® HD I/O Interface" in several of the products in the new line. The other was the pricing, which topped out at a list price of under $700 and street pricing around $500. That's right, I said, "topped out," not "started." (The lowest end DAS product to feature those same converters comes in at just under $300 list and approximately $200 street.) When you consider that ProTools HD I/O Interfaces' list prices start above $2,000, and getting up to 192 kHz means a list price of just under $4,000, the significance of this announcement becomes much more apparent. When I looked more closely at the other features included in the DAS family products, things looked even more interesting. This was especially true of the high-end 1820M, which included extensive audio and MIDI I/O and synchronization options, a couple of mic preamps with Mic/Line capabilities and phantom power, and on board hardware DSP.

E-MU had definitely captured my attention. Were we talking "too good to be true" or a dramatic change in the power and functionality versus price curve that could dramatically affect the baseline quality level for home and project studios from this point forward? Needless to say, when the opportunity to take a hands-on look at the E-MU 1820M arose, I was eager to check it out. Let's take a look.

Background

The E-MU 1820M is the flagship entry in E-MU's Digital Audio Systems (DAS) family of products. Its configuration is of the "everything but the kitchen sink" variety, so it's fair to say that most home studios, and many project studios, won't need everything it has to offer. However, the relatively low price for the product makes it accessible to those who are just starting out, but would like to have built-in room for growth. The facilities offered are particularly well suited to project studios working in the post-production area due to the extensive synchronization capabilities, built-in surround monitoring connectivity (for up to 7.1 surround), and connectivity for many different types of gear and audio sources.

The 1820M is delivered with three main hardware components:

The E-MU **1010 PCI Card** is the brains of the system, including its DSP horsepower. It also includes an EDI (E-MU Digital Interface) port to connect the computer-based end of the 1820M to the AudioDock M breakout box, coaxial S/PDIF in and out, ADAT in and out (192 kHz-compatible, and switchable to optical S/PDIF), and a FireWire® port.

The **Sync Daughter Card** is used for synchronizing with external gear such as video equipment, master clocks, and other DAWs. It includes Word Clock in and out, SMPTE Sync in and out, and MTC (MIDI Time Code) out.

The **AudioDock M** breakout box is where the main physical action will be for most users, with its inclusion of all the analog audio hookups and more. The front of the AudioDock M includes two TFPro™ mic preamps with Mic/Line inputs with 40db of gain/attenuation and 48V phantom power. Also on the front of the unit are a MIDI input/output pair, an optical S/PDIF output, a stereo headphone output (the package includes a Y-cable to allow use for this output with two pairs of headphones) with volume control, and indicator lights for MIDI, sync source, and sample rate. The back panel includes six balanced 1/4" analog inputs, eight balanced 1/4" analog outputs, a stereo turntable input with RIAA preamp and ground, another MIDI input/output pair, four stereo 1/8" computer-style speaker outputs (configurable from stereo to 7.1 surround), and the other end of the EDI connection. No external power is required -- the AudioDock M gets that from the computer via the EDI cable.

The hardware side of things may be the place to go for external connectivity, but it is the software side, specifically E-MU's PatchMix DSP applet, that ties all the bits and pieces together. We'll look at PatchMix DSP in more detail below. The quick synopsis is this is where you determine what inputs and outputs will be activated in a particular configuration, how signals will be routed between inputs, outputs, and software, what zero-latency, DSP-based effects are inserted in the signal chain, and for sample rates, bit depths, and so on. It is part mixer, part monitor, and part system control center. PatchMix DSP provides up to 32 channels of zero latency hardware mixing and monitoring.

Speaking of effects, the 1820M features E-MU's E-DSP chipset, which features a hardware-accelerated effects processor with 28 effects plug-ins. Furthermore, there are 600 presets, many of which use multiple individual effects in their chain. These presets are categorized by type (e.g. Distortion and Lo-Fi, Guitar, Vocal, Drums and Percussion, etc.) for ease of browsing. The E-DSP effects processor allows up to sixteen simultaneous effects to be run without impacting your CPU. PatchMix DSP keeps track of how much more DSP processing capacity is left so you don't have to worry about that -- it only makes available effects that can be run within the remaining capacity.

WDM, DirectSound®, and ASIO™ 2.0 drivers for Windows XP and Windows 2000 are included for compatibility, and low latency performance, with most audio and sequencer applications. Notably absent are GSIF drivers for use with TASCAM's GigaStudio sampler.

Rounding out the software side is a bundle of third party applications, including Cubase VST 5.1, WaveLab V 2.53, and SFX Machine LT. It is also worth noting that the 1820M is included in the Emulator X Studio bundle, which includes E-MU's flagship Emulator X software-based sampler and a sample library. Emulator X will be the subject of a future CakewalkNet review, though, so it will not be covered further in this review.

Paper documentation includes a *Quick Start* guide and "read me first" type notes. A more detailed *Operations Guide* comes in on-line Adobe Acrobat (PDF) form.

The current list price for the 1820M is $699, with street prices as of the time of this article coming in right around $500. The Emulator X Studio package, which adds the Emulator X sampler and sample libraries, lists at $799, with street price around $600.

As is usually the case with CakewalkNet reviews, most of the testing done for this review focused on using the E-MU 1820M with SONAR, specifically SONAR Producer Edition V4.0.2. However, to get a feel for how the various drivers worked, and just in the course of general use of the 1820M over a period of approximately three months, the 1820M was also used with Project5 1.5, Sound Forge 7.0b, the various bundled applications included with the 1820M package, Windows Media Player 10, and several other applications. The version of the E-MU DAS drivers used was V1.60.

Getting In and Hooking Up

The basic 1820M hardware installation was well documented and, frankly, about as easy as multi-board hardware installations get. Just open your system up and find two consecutive PCI slots, insert the 1010 PCI Card in one (preferably one that won't be sharing IRQs with other devices), connect the cable that goes between the 1010 PCI Card and the Sync Daughter Card to both boards, insert the Sync Daughter Card into the second slot, connect an available internal power cable to the power connector for the 1010 PCI Card, then close your system. Next, put the rubber feet on the AudioDock M box (or mount it in an optional rackmount fixture), connect the EDI cable between the AudioDock M and the 1010 PCI Card, and you're ready to start on the software side of the installation.

One note here is that the EDI cable supplied is 3 meters (just over 9 feet) in length. Depending on your studio setup, that may not be enough for your optimal location for the AudioDock M. In my case, I didn't have the desk space to put it on my computer desk (and wouldn't want all the cables coming out of it there anyway). The cable also wasn't long enough to reach from my computer to my rack (which has plenty of open spaces these days thanks to my going all softsynth). Thus, I ended up setting the AudioDock M on top of a storage container under one side of my desk, and having to run audio and MIDI cables between that and my audio rack (which contains my power amplifier and a few tape decks) and MIDI keyboards. The documentation indicates that maximum EDI cable length is 10 meters (over 30 feet), which would likely have been enough for my needs. However, the only cable available from E-MU's on-line store is a replacement 3-meter cable, and I have yet to find any other sources of longer cables.

The basic software side of the installation was also quite straightforward. On the first system bootup after connecting the hardware it was necessary to cancel out of Windows' automatic hardware detection process, then simply insert the Software/Driver Installation CD-ROM and follow the prompts. This installed not only the drivers and PatchMix DSP, but also SFX Machine LT, WaveLab Lite, Cubase VST 5.1, and the Fraunhofer IIS MPEG Layer 3 ACM Decoder Component. At the end of the software installation process, a reboot was required.

Registering the software via the net was extremely quick. Name and e-mail address were required. While it also prompted for address, gender, and date of birth(!), those were optional. It also gave the possibility of signing up for e-mail lists, but did not separate out tech support and marketing lists. From the e-mails I've received from this since registering, though, this is no big deal as the number of e-mails is relatively small. Once the form was filled out, the actual registration process took only a second or so.

I also immediately went to the E-MU site to check for driver and other updates. There was indeed an update available -- I didn't pay that much attention to the initial version of the drivers since I wanted to start with the latest ones, but the new drivers were V1.6. It was quick to download and install the drivers.

Next up was setting up a configuration in the PatchMix DSP software. We'll get to the details of PatchMix DSP below, but, glossing over those for now, I set up a simple configuration to work with SONAR in ASIO mode and various other software in Wave/WDM mode, and was ready to go. Well, sort of... I'd followed the instructions to the best of my abilities, and could even see the meters moving properly in all the applications I was trying. I also thought I had the configuration set up correctly to send a signal to the headphone output, but, try as I might (and, no, I didn't forget to turn the headphone volume up, either), I could hear nothing. I must have spent an hour or two (figuratively) beating my head up against the wall, searching the internet for possibilities, trying things, rebooting, etc., and thinking perhaps I'd received defective hardware, when I finally came across a post on "The Unofficial E-MU Forum" that gave me something to try. There is a "Restore Defaults" program included among the applications installed. Running that, and doing the required reboot after running it, got me back up and running with the same PatchMix DSP configuration file I'd been using earlier. A subsequent (unrelated) situation that caused me to have to reinstall Windows XP, along with all my drivers and applications, demonstrated that the loss of signal wasn't a fluke -- i.e. it happened the second time I upgraded from the default drivers supplied with the product to the V1.6 drivers -- but running the "Restore Defaults" program did the trick again. I have also lost signal to the outputs (both headphones and line outs to studio monitors) for unexplained reasons on a few other occasions, and the "Restore Defaults" program has done the trick for fixing things on each occasion. (NOTE: E-MU recently discovered the source of this problem and will be fixing it in a future release of the DAS drivers. In the interim, they have indicated a much quicker way to restore the sound, which does not involve a reboot, is changing the sample rate, for example, from 44.1kHz to 48kHz and back.)

Most of the magic of hooking the various signals up between physical inputs, physical outputs, and software-based inputs and outputs is accomplished in PatchMix DSP. Before we get to that, though, it is worth making a few observations on the physical side of the 1820M's connections.

The 1820M features dual MIDI input/output pairs. The first pair is on the front panel of the AudioDock M, and the other is on the AudioDock M's back panel. However, if you want to use certain Sync Card functions, specifically the SMPTE input and the Sync Card's software control panel, the MIDI 2 (i.e. back panel) input on the AudioDock M must be disabled. In that scenario the MIDI 2 output will mirror the MIDI 1 (i.e. front panel) output, and MTC (MIDI Time Code) will be sent out a special MIDI port on the Sync Daughter Card. All MIDI connections are via standard MIDI cables (a special adapter cable is supplied to adapt the special MTC port to a standard MIDI cable). The MIDI connections on the AudioDock M unit I was supplied were exceedingly tight, to the point where I had a difficult time inserting MIDI cables, and initially thought the MIDI ports simply weren't working due to my not having fully made the connections. The upside of these tight connections is that the cables aren't likely to get accidentally disconnected once inserted.

I would have preferred that the primary MIDI I/O pair be on the back, rather than the front, of the unit. That way, if synchronization functions that require disabling the MIDI 2 port are needed, it would not be necessary to use the front panel MIDI input, thus running cables where they could conceivably be in the way. My feeling is that MIDI connections in a home or project studio scenario are likely to be of the relatively permanent variety, unlike connections for a guitar or microphone, which may vary depending on the phase of a project. Plugging a cable into the front panel is useful for easy accessibility with temporary connections, but results in extra clutter if the connection is to be more permanent. This can particularly be a concern if placing the AudioDock M on top of a desk, where, in addition to the extra front panel clutter, the cable could be in the way, though this could also be inconvenient in a rack.

On the analog audio side, the layout of the AudioDock M is highly ergonomic, with six line inputs and 8 line outputs, all capable of balanced or unbalanced connections, on the back panel, and the two multi-function inputs (see below) on the front panel. The back panel also includes a special stereo set of inputs for a turntable, complete with the requisite RIAA preamp. Also on the back panel are four stereo 1/8" (computer-style) speaker outputs.

The 1/8" stereo outputs carry the same audio signals being sent to the eight 1/4" analog line outputs. These are provided as a convenience, primarily for use with computer-style surround speakers -- up to 7.1 -- though there are no surround decoding functions built into the PatchMix DSP software or 1820M's drivers, so support will be application-dependent. These should work nicely with SONAR 4 Producer's surround sound support for anyone on a budget and needing to get into a relatively inexpensive surround speaker setup.

Moving over to the front panel, the two multi-function inputs feature Neutrik connectors. For those unfamiliar with Neutrik connectors, they do double duty, allowing you to plug in either 1/4" balanced (or unbalanced) phone jacks (e.g. for synthesizers or guitars) or XLR-style cables (e.g. from mics and certain higher end line-level connections). Each has its own independent Trim/Gain control, with a whopping 40dB range! A relatively minor niggle is that, with that much range comes a relatively coarse degree of control, where small changes can make a lot of difference. For example, my vocal mic level had to be around the one quarter range of the gain control, and little movements up and down could make a big difference in the levels coming into the PatchMix DSP software (and, ultimately, into SONAR). Also, while there are some gradation-style markings on the trim/gain controls, they aren't exactly easy to read in dim light and inconvenient locations (e.g. on top of a container under a desk in my case). This makes it somewhat tough to set the gain controls repeatably and match levels for stereo miking applications.

The two front panel analog inputs are the connection points for the 1820M's TFPro mic preamps. For those not familiar with TFPro (I wasn't), the "TF" stands for Ted Fletcher, designer of most of the classic Joe Meek equipment line. These particular mic preamps use TFPro's CurrentSensing techology, which is currently (pun sort of intended) used only in TFPro and E-MU equipment. The CurrentSensing technology allows the mic preamp to sense the best impedance to run at for the microphone or instrument currently plugged into it. The idea is to match the impedance to the mic, to avoid the scenario where you need different preamps for different mics to match mics to preamps. Needless to say, this notion is particularly attractive in home studios where mic and preamp budgets are likely to be highly limited.

The mic preamps are supplied with 48V phantom power. Note, though that there is only one phantom power switch which applies to both preamps. This might be a concern in some situations where you need two mics to run concurrently, where one needs phantom power and the other may not work properly, or might even be damaged, by application of phantom power. This is consideration is likely to be of only minor practical concern for the typical budget studio, though. (Most of us aren't well equipped enough to have, for example, expensive ribbon microphones that could be damaged by application of phantom power.)

Rounding out the analog I/O, also on the front panel of the AudioDock M, is the headphone output, which includes its own volume control, and is separately assignable from the analog line outputs. E-MU thoughtfully includes a short headphone splitter cable. While this won't give you separate control over what each set of headphones is hearing, either at the volume or mix content levels, it is extremely helpful for those home studio situations where you are mostly a solo shop, but occasionally have one other musician or vocalist in overdubbing miked parts. Those who need more headphone outputs, or more control over what each set of headphones is hearing, will need to take advantage of the extra analog line outputs (or perhaps the ADAT digital output) in conjunction with a headphone distribution amp. Those of us with more modest needs, however, can save the money and space a headphone amp would consume and just plug in the splitter cable when we need an extra set of cans. Note that the headphone jack is capable of ear-splitting volumes, so make sure and start with it turned all the way down and turn it up gradually for comfort and ear protection.

Sticking with the AudioDock M's connections for a moment, but moving over to the digital side, the front panel also includes a single (stereo) optical S/PDIF output. This jack is intended for use with portable DAT, MD, and other digital recorders, and, by default, echoes the audio signal sent to the first two analog line outputs. However, it can be configured to put out other signals via PatchMix DSP.

The rest of the digital audio connections are found on the 1010 PCI Card. Coaxial S/PDIF input and output jacks are provided, and work with word lengths up to 24 bits and at data rates up to 96kHz (they are disabled for 192kHz operation). These can also be used with AES/EBU connections by means of an adapter cable (not included). ADAT input and output jacks, which can also be used as optical S/PDIF jacks are also provided, and can work with word lengths of 16 or 24 bits. When used in ADAT mode, they provide up to eight channels of digital audio (in each direction) at data rates up to 48kHz. At higher data rates, industry standard S/MUX interleaving is used to interleave the data across multiple channels. For example, at 96kHz, each channel of audio would take two 48kHz channels, so you would only get four channels of audio. At 192kHz, each channel would be interleaved across four 48kHz channels, so you would only get two channels of audio.

Also on the 1010 PCI Card is an IEEE 1394-compliant FireWire port. While it does not support FireWire audio "at this time" (perhaps E-MU is leaving the door open to this with future driver upgrades?), it can be used for connections to external storage, digital video cameras, scanners, and other OHCI 1.1 compliant devices.

Moving over to the Sync Daughter Card, Word Clock input and output connections allow you to either sync the 1820M to an external master clock or use the 1820M as your studio's master clock, respectively. SMPTE (LTC) input and output connections similarly allow synchronizing the 1820M with external video equipment or external audio recorders that understand SMPTE. When SMPTE is in use, the Sync Daughter Card also puts out MIDI Time Code (MTC), which is a form of SMPTE that can travel over MIDI cables, for use with external sequencers.

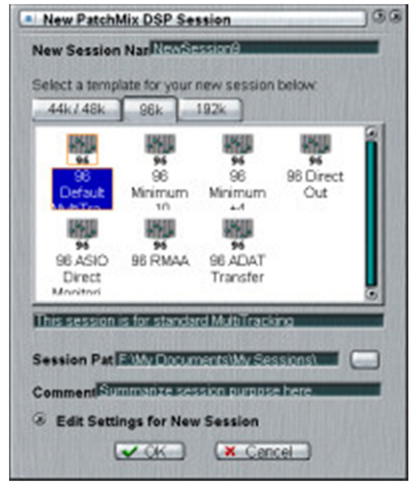
While most home and other budget studios won't need all this connectivity, it is certainly nice to know it is there if/when you do need it. In my case, I used two of the analog outputs to go to my studio amp and monitors via balanced cables. On the input side, I took two analog ins from a cassette deck for conversion of old demo tapes, and used another pair of inputs from one of my keyboards to use for practicing for shows (I don't use it as a sound source in recording these days, just as a MIDI controller). I also used the ADAT input for hooking in an old blackface ADAT for dumping of archival projects. On the MIDI side, I hooked both of my MIDI controller keyboards up using the two MIDI inputs to avoid having to turn both keyboards on if I was only going to want to use one (and I like the flexibility of having both available, for example, to use as dual manuals with NI's B4 organ). Those were the semi-permanent connections. I also plugged a vocal mic into the first mic input as needed, and played around with plugging in an electric guitar from time to time, just for fun (you **really** don't want to hear my guitar playing). I also used headphones for tracking and after hours monitoring, and really appreciated the headphone splitter cable when working with a talented female vocalist. While I don't have any need for the sync functions in the near term, I do appreciate knowing that they're there as I do harbor a few aspirations of doing music for video work in the longer term.

Patching in the Virtual World

Most, if not all, audio interfaces come with some type of control panel-like application to allow you to configure latency settings, set sample rates, and perform various other low level setup operations. Some audio cards include software that goes beyond the basics, providing some level of direct monitoring control, signal routing, and other mixer- or patch bay-like operations. The E-MU 1820M takes things a step further, providing a user configurable, modular virtual mixer, called PatchMix DSP, complete with effects inserts, pre- or post-fader console strips, sophisticated patching capabilities, and more.

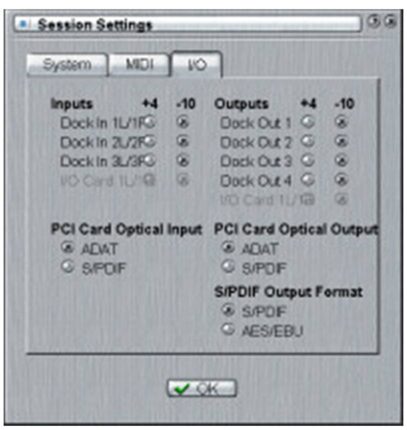
Before we dive into the details of what PatchMix DSP offers, it is worth mentioning that, when you first encounter the 1820M and PatchMix DSP, it may seem a bit daunting. "Where do I start?" is the question that comes to mind. In fact, it is not unlike encountering SONAR for the first time. That is, there is a lot of power there, and, once you get used to a few basics, it is pretty easy to use the product, and, as you grow into the product, there is much more you can do. However, until you do understand a few basics, it is easy to just sit there staring, trying to figure out where to begin.

E-MU provides help on this front, via a number of pre-configured "Sessions", which are E-MU's term for PatchMix DSP configuration files. Actually, they are more than just configuration files, as they can encompass everything from I/O configurations to fader levels, to effects chain and effects unit settings. In fact, the information held by these files is very much like what you would expect to be contained in a snapshot of virtual mixer settings, so "Sessions" is an apt name. However, for someone who is used to working with SONAR, or another DAW, and considering any "session" data to be what is stored in the SONAR project files, the name may also be confusing. This becomes more apparent when you consider it is likely you will use the same PatchMix DSP "session" file for multiple SONAR sessions or projects. Whatever the wording, though, the important thing to know up front is that PatchMix DSP is where you tell the 1820M what physical inputs and outputs should be configured, how signals will be routed to and from your DAW, which signals will be monitored directly (with zero DAW-induced latency), and so on.



For those who may be somewhat squeamish about the notion of configuring a virtual mixer from scratch, the session file templates provided by E-MU will cover many common bases. Simply press the New Session button in PatchMix DSP, select the tab for the sample rate you want to use, then browse the list of templates available for that sample rate. Odds are pretty good that, if your use is a fairly common one, you'll find something that will do the trick. For example, at each sample rate you'll find templates for default multitracking setups, minimal multitracking setups at +4dBu and -10dBV, ADAT transfer, surround monitoring, and more. Even if none of the templates does exactly what you need, you can probably find something close, then use that as a starting point for making whatever minor modifications you need to get you the rest of the way.

If I may be so bold, though, I am going to suggest you forgo the quick start route, pick a blank template, and learn to configure a session from scratch. This is not just for the sake of adventure. Rather, it takes configuring a session from scratch to truly understand what is going on inside PatchMix DSP. Understanding what is going on is useful for, and maybe even important to, configuring a system that truly meets your needs as optimally as possible given the uniquenesses in your environment, the task at hand, and various other factors that ultimately play into how productive you and your system will be with any task you throw at the 1820M. So let's give our mouse hand a little exercise, shall we?



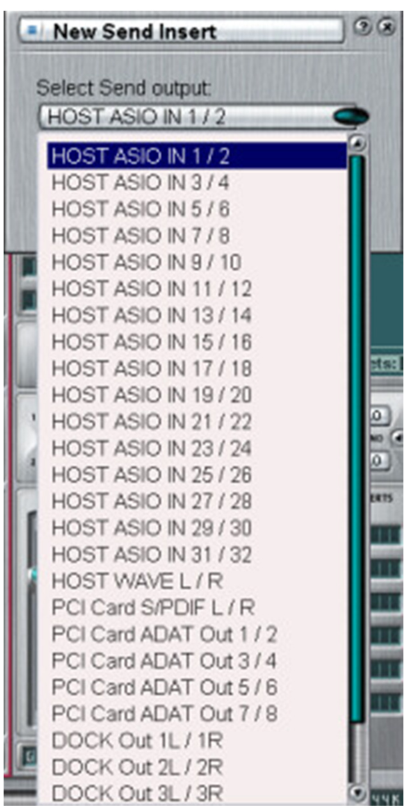
Once you choose your bit rate, select a blank template for that rate, and fill in a few "housekeeping" fields (e.g. where the session file is to be stored and any comments on its purpose), you will be asked to choose some general session settings. For example, will the clock be internal or external? If it is external, will the clock signal be coming from Word Clock, SMPTE, or ADAT? On the MIDI side, will you be using the AudioDock M's MIDI 2 input or do you need the Sync Daughter Card enabled? What about audio I/O? Will each analog signal be +4dBu or -10dBV? Will TOSLINK optical I/O be configured for ADAT o r S/PDIF use? Will S/PDIF outputs be configured for S/PDIF or AES/EBU? Once you've made your choices, a simple "OK", and you're on your way. PatchMix DSP will configure a default "WAVE L/R" mixer strip for you to handle audio from any Windows applications whose stereo audio output is assigned to the 1820M's Wave (or WDM) device, and will route it to the default monitor outputs (DOCK 4L/4R) and headphone outputs. If you want anything beyond that, though, including recording inputs for Windows applications, ASIO input and output devices for SONAR, etc., you'll need to configure them yourself. (Of course, you can still go back to the idea of just loading a pre-configured template, but this is no time for wimping out!)

Let's take a look at a simple SONAR configuration, where I'd like both of the AudioDock's front panel audio inputs to be available for mic and/or line use, and only need stereo outputs, plus the headphone output, for monitoring. While you can use the 1820M's Wave/WDM interface within SONAR, it is highly recommended that you use the ASIO interface instead, as it is further developed (e.g. the WDM interface is limited to fixed latency and stereo only). I'm going to be working at 44.1kHz since my ultimate output will intended for audio CDs. However, most of what I will be talking about here applies equally if I am using 48kHz. There are some additional considerations for the 96kHz and 192kHz sample rates, which I will get into a bit later on.

Starting with the input side, we will use the Create Strip button to create a new mixer strip. In this case we will be bringing in inputs from the physical AudioDock hardware, so we'll choose from the "Physical Source" list, and the first one we want is the DOCK Mic/Line A input, which is the leftmost of the AudioDock's front panel inputs. We could finish configuring that first strip now, but let's repeat the exercise to add a PatchMix strip for the DOCK Mic/Line B input while we're at it. All this is doing thus far is configuring these two mono inputs to bring their signal into PatchMix DSP itself. If you were to go into SONAR now to see what ASIO inputs were available, you would find that none are (i.e. unless you have another audio card with ASIO drivers configured on your system). Before we configure these for use in SONAR, though, down near the bottom of each strip is a "scribble strip", where you can label the each mixer strip with something meaningful to help you remember what it is. It gives you up to 8 characters. For my relatively simple setup, I am expecting to use the first input for mics, mainly vocals, and the second for a guitar, so I will label the left and right inputs "Mic" and "Guitar", respectively. Note that these names are only used in PatchMix. In particular, you will not see them in SONAR.

As I mentioned above, what we've done to this point doesn't bring the signal into SONAR. It does, however, make the signal available for direct monitoring and mixing inside PatchMix. We will use that aspect of it for zero latency monitoring of the signals we are recording later on. Before bringing the signal into SONAR, I may wish to configure meters to help in setting signal levels via the preamps' physical gain control knobs. Just below the portion of each mixer strip that shows the input's name, is a section for placing multiple inserts. Six slots are visible. If you need more than six inserts a scroll bar will appear once the seventh insert is configured, and you can scroll to see any six consecutive inserts at a time. For example, you might want to use more inserts if you are configuring a multi-effect guitar patch since these inserts can be use for running the 1820M's E-DSP effects. In our case, we want a peak meter in the first insert slot of each of the newly-created strips, and making that happen is as simple as right clicking on the relevant slot and choosing "Insert Peak Meter" from the popup menu.

Next we have a decision to make: Do we want to take signal directly from the preamps into SONAR, or will we want to preprocess that signal with EQ or other effects prior to bringing the signal into SONAR? If you choose the latter route, you will choose the effects modules you want to use at this point, and put them in the next set of insert slots until you have all modules you want to use inserted. Personally, however, I prefer to save effects decisions for later on, even if I want to monitor through effects (e.g. for that singer who just has to hear her voice with reverb on it to capture a great performance, or to give a guitar player going direct amp-like feedback while tracking, even though we will not use the "amped" signal in SONAR, but, rather, will use an amp simulator plug-in later on). Thus, my next step will be to send the incoming signal to SONAR (or any other DAW sporting an ASIO interface).



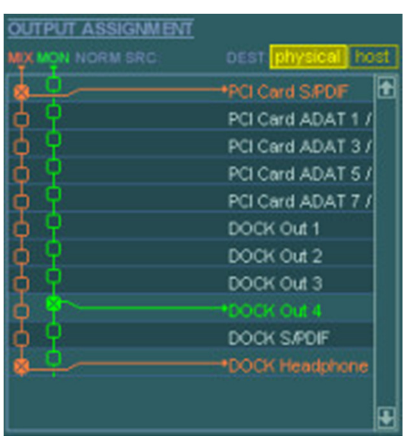
Right clicking on the second insert slot, I will choose "Insert Send (Output to ASIO/WAVE or Physical Out)". This terminology, especially the words "Output" and "Out", may seem a bit confusing since what we want to do here is configure an input to SONAR. The key is we are going to be sending the signal out of PatchMix so it can be made available for patching into SONAR. Thus, from the PatchMix perspective, it is an output. On picking that menu choice, a dialog box pops up, asking us to select the Send Output. We want to make this output available as an ASIO input to SONAR, so we will pick the first available ASIO output, which is labeled, perhaps somewhat confusingly, "HOST ASIO IN 1 / 2". Just remember, though, when you cable up real gear, you connect an output from one piece of equipment to the input of another. In this case, you are doing much the same thing, "virtually cabling" a send output from PatchMix DSP to an ASIO input of SONAR (or another ASIO host). Do the same thing with the input strip for the second mic preamp, this time choosing "HOST ASIO IN 3 / 4", and we now have both inputs connected for use with SONAR, but with zero latency monitoring via PatchMix DSP, independent of the latency setting within SONAR itself.

At this point, if you'd like to monitor the input signals through effects, you can use subsequent insert slots for that purpose. We'll talk about the effects available later, but the important thing to remember is that, since they are appearing after the send output to SONAR, they will not be included in what is recorded in SONAR. If you wanted to record them into SONAR, besides just putting them before the send, as mentioned above, you could insert a second send, recording them to a separate track in SONAR, thus having the flexibility of using the dry signal, the effected signal, or a mixture of both.

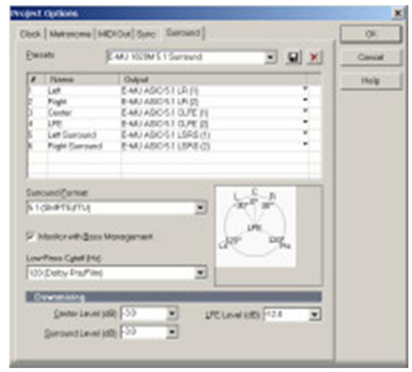
Before moving on to configure outputs from SONAR back to PatchMix, it is worth mentioning that, in addition to meters, sends, and effects, PatchMix also makes available a Trim Control and a Test Tone/Signal Generator. The Trim Control won't be too exciting if you're using the AudioDock's mic preamps, because you are better off getting the signal level right with the physical gain/trim knobs. However, it could be of interest if you are feeding in a line level signal from a piece of gear without an output volume control, or for use within an effects chain to adjust the level flowing from one effects module into another. The Test Tone/Signal Generator can be very useful for troubleshooting, and for calibrating your system. It provides an oscillator, a white noise generator, and a pink noise generator.

Setting up an output from SONAR into PatchMix is simple. Just as we did when we created a physical input, we create a new mixer strip, this time choosing a Host Source instead of a Physical Input. Choose any ASIO output from the list box of choices (we'll go with "ASIO OUT 1 / 2" since this is the first output we're creating), say "OK", and we've got a strip ready to go. Optionally, put a useful name in the scribble strip for the new mixer channel -- note that the name will be reflected in SONAR this time, so I'm going to use "2ch Mix" to make it clear this is my stereo mix bus -- and we're good to go. Oh yeah, remember to save the PatchMix configuration to your session file.

That wasn't so hard, was it? Want more? The mic preamp inputs are only mono, and maybe you'd like to add a stereo input for dumping from cassette or 2-track reel-to-reel, or even for capturing the output of a stereo synth, into a single stereo track in SONAR. Just repeat the strip adding process for another physical input -- perhaps DOCK IN 1L / 1R for one of the aforementioned possibilities or DOCK IN 3L / 3R(PHONO) if you're hooking in a turntable, insert a meter, if desired, and a send to ASIO, and you're there. I'll add an input from cassette on DOCK IN 1L / 1R. Did that mess up your thinking of the logical order of the strips? Just drag them around to get the order you like. You can see the resulting PatchMix configuration at right. I've placed all physical inputs on the left, with inputs from hosts, one for Wave and the other for ASIO, on the right. If you need more channels coming from the host, for example for use with SONAR 4 Producer's surround sound, additional host inputs can be added just as easily. Similarly, if you need to record in a non-ASIO application, you can add a send to "HOST WAVE L / R".



What about getting the signal back out to the 1820M's physical outputs, be it analog outputs on the AudioDock M, ADAT outputs on the 1010 PCI Card, or otherwise? If all you need is stereo monitoring, it couldn't be simpler. PatchMix provides an Output Assignment matrix that allows you to send either the mix, a separately mutable monitor bus, or neither to any of the available physical outputs. While the mix and monitor buses contain basically the same audio content -- i.e. the output of the mixer strips, including any effects on those strips and on the main bus, the monitor bus does has a few extra goodies. Perhaps most important is a mute control. If you're a one person/one room shop, where you are doing everything in a single room, this would allow you to, for example, make the stereo monitor bus be your speakers, use the mix bus as your headphones, then use the monitor mute switch to effectively turn the speakers off when it comes time to track vocals or an acoustic instrument, while not affecting the signal going to the headphones. There's nothing saying, though, that you can only have one monitor or mix output. For example, come mix time, you may want to be feeding a DAT and reel-to-reel tape in parallel. Or, if you have a more complex studio than our one-person operation, you may wish to feed monitor signals to multiple destinations. This will allow that single monitor mute control to mute any physical outputs designated as receiving the output of the monitor bus. The monitor bus also has its own volume and pan controls. Beware, though, that the monitor's volume control, which can only trim, not add gain, comes after the mix bus' fader. Thus, you won't be able to turn the mix bus all the way down but still send signal to the monitor bus. You can, however, mute the monitor bus or turn it down, while still sending a full signal to the mixer outputs.



If you need more than stereo outputs, though -- e.g. perhaps you need to monitor in surround sound or send different signals out to a headphone mixer for creating "more me" mixes for individual musicians, or to dump stems to a multi-track digital tape recorder -- things get slightly more, but not overly, complex. The way this works is very much like the above-mentioned sends to your DAW. That is, you use an insert to tap into the signal wherever you need to access it, then, instead of sending it to a DAW's input, you send it to one of the 1820M's physical outputs. Let's take the case of a simple setup to allow monitoring in either stereo or surround with SONAR 4 Producer Edition. We'll use the first three of the AudioDock's 1/8 inch stereo outputs with powered computer-style speakers for 5.1 surround, then use the two quarter inch outputs running to our control room amplifier, which is attached to non-powered speakers, for stereo monitoring. The first thing we'll need to do is configure some additional host ASIO outputs from SONAR to PatchMix DSP to get the total of 8 channels (or 4 stereo channels) we need. We're no longer in a simple stereo monitoring configuration, so we'll need to use sends inserted in each of these new channels to send the signal from the mixer channel (which we will feed from SONAR's ASIO outputs) to the relevant physical outputs. In this case, we want them to go to the DOCK 1, DOCK 2, and DOCK 3 stereo pairs, which are feed to both the first six 1/4 inch analog outputs and the first three stereo 1/8 inch outputs. I would also enter meaningful names (e.g. "5.1 LR", "5.1 CLFE", and "5.1 LSRS") into the scribble strips for the new channels so it is easy to tell in SONAR which outputs are which. Last, but not least, we need to enable the extra channels in SONAR, then set up SONAR's surround configuration at the project level to use the ASIO outputs we've just set up in PatchMix. The resulting PatchMix configuration might look something like the screen shot at left. Whee! We're having some fun tonight!

There is much more that PatchMix can do, but we've already covered most of what you'll need to know for a typical SONAR (or other DAW) configuration of the "mixerless studio" variety. For those whose needs go beyond that, though, let's mention just a few relevant bits and pieces.

As good as software effects have become in the last few years, some people still have favorite hardware processors. Perhaps yours is a high end reverb in a pro studio, or some really funky processor you picked up on eBay at an impulse purchase price, or even just running your mix through a tape loop and back to simulate mixing to tape. If so, you'll be happy to know PatchMix makes it easy to configure a physical send/return loop for use in its insert slots. Just create the insert, tell it which physical output to send on, and which physical input to return the processed signal on, and hook up the processor. If you want that processed signal going back into SONAR, you could insert another send to an ASIO output, and then could just use the other end of that as an input into a SONAR track. Of course, if you did things that way, latency would be introduced via the round trip of audio to and from SONAR. A better option comes in the form of an E-MU-supplied VST plug-in called E-Wire. E-Wire takes care of the delay compensation for you, just like if you were using software-based effects with plug-in delay compensation (and assuming the hardware effects box does not add latency of its own). E-Wire works in conjunction with an ASIO output in PatchMix, where you can have a chain of insert effects comprised of E-MU's E-DSP effects and/or hardware effects via send/return inserts followed by a send back to ASIO. Inside the E-Wire VST plug-in, you simply have to tell E-Wire which device to use for the send and which to use for the return. The system then takes care of the rest. Okay, so it won't let you automate parameters or save and recall configurations like with software plug-ins, but were you expecting miracles? Of course, if your hardware effect can be automated via MIDI, you may able to take care of that side of things with a MIDI track.

For those synchronizing with other hardware devices, such as video gear or an analog tape deck, or using a master clock with multiple digital audio devices (e.g. digital mixer, outboard converters, etc.), PatchMix is also used to set up some sync parameters (e.g. SMPTE frame rate, whether Word Clock termination is terminated or not) and perform SMPTE striping operations. I'm no audio for video guy, and it's been years since I've striped time code to analog tape to sync up a sequencer (and I think it was just FSK at the time), so it's tough for me to say much about this part of the software. From my "just knowledgeable enough to be dangerous" perspective, though, it looks like it will cover the key things people would need to do, and my quick search through the "Unofficial E-MU Forums" suggested this area of the 1820M performs largely as expected.

Of course, PatchMix is also where you deal with selecting sample rates, such as 44.1kHz, 48kHz, 96kHz, and 192kHz. (Note that 88.2kHz is not currently available.) This might be a trivial enough bit of information to not even warrant mention here, except that going above 48kHz carries with it a few limitations and other considerations. Perhaps the main limitation once you go beyond 48kHz is that the E-DSP-based effects are no longer available. (You can, however, use the hardware output/input-based sends and returns.) I/O is also reduced based on bandwidth considerations. For example, at 96kHz, instead of having 8 channels of ADAT I/O, you get only 4 channels of I/O, and the ADAT interface becomes stereo at 192kHz. Also, at 96kHz, you lose 4 channels of inputs overall, so, if you use the 4 channels of ADAT input, you will only have 4 channels of analog input, losing (stereo) line inputs 2 and 3, for a total of 12 inputs. All outputs are still available, but the number of ADAT outputs is reduced to 4 as mentioned above. At 192kHz, S/PDIF functionality is lost since the S/PDIF specs are not specified to work at 192kHz. Total inputs are reduced to 4, which could be a combination of 2 ADAT and 2 analog or 4 analog only. At 192kHz, outputs are also decreased, with the total number of ADAT outputs down to two, the loss of S/PDIF, and a reduced number of analog outputs. At both 96kHz and 192kHz, the headphone output is also no longer directly assignable, and simply reflects whatever is routed to analog line 4, which is designated as the monitor output.

These limitations and considerations are all reasonable when you consider the increased bandwidth and the increased work that has to be done to feed more data in and out of the computer. However, they can make for a few surprises if you aren't prepared for the differences. For example, I was wondering why the headphones "no longer worked" when I switched to 96kHz, and it turned out the reason was because I'd been using (stereo) line output 1 as my monitoring output, with line output 4, which E-MU designates as the monitor out, not connected to anything. It wasn't that the headphones weren't working, it's just that they weren't showing up in PatchMix since they weren't directly assignable, and I wasn't routing anything to the output that needed to be used under that scenario. Once I found that out, it was easy to correct my "user error" in not paying attention to the labels on the hardware (okay, my excuse is it's dark in my studio, and I'm just used to using the first pair of outputs for monitoring from my old hardware), hook things up how the 1820M prefers, and get my headphone signal back. This also explains why it is important to change sample rate configurations from PatchMix DSP prior to trying to use a different rate in SONAR or another DAW -- i.e. you're not talking about a simple switch of rates, but, essentially, a whole new configuration.

Before moving on, I'll summarize PatchMix DSP briefly by reiterating that there is a lot of power and flexibility there. If you only need a very simple setup, it is easy enough to configure what you need, save it, and forget about it afterward. On the other hand, if you need lots of flexibility for different types of projects, you'll want to learn as much as you can about PatchMix DSP so you can create different virtual configurations to suit whatever needs you might have at the moment. Of course, you can save any configurations for later re-use, too, but it's also nice to know you can just configure something custom on the fly if the need arises.

That's a Nice Effect

The 1010 PCI Card, which is a core component of the 1820M, includes the E-DSP 32-bit Multi-effects Processor, and E-MU currently provides 28 hardware-accelerated effects modules to take advantage of this power. What's more, E-MU provides upwards of 600 presets, including both presets for individual modules and multi-effects chain presets, and provides interfaces to these effects both directly in PatchMix DSP and via a VST plug-in interface called PowerFX.

The core E-DSP-based effects modules available include 4 equalizers (single band shelving and parametric, and 3- and 4-band with shelving end bands and parametric middle bands), auto-wah, chorus, compressor, distortion, flanger, frequency shifter, leveling amplifier, 2 reverbs (lite and stereo), 11 delays (6 each mono and 5 stereo, with various maximum delay times), phase shifter, rotary speaker simulator, guitar speaker simulator, and vocal morpher. Most of these are just what you'd expect based on their rather generic names, but there are a few that might require a little elaboration. For example, the frequency shifter is not a pitch shifter. Rather, it shifts every harmonic in the spectrum by a fixed frequency, thus messing with the relationships between harmonics. Depending on how much shift is applied, the effect can range from relatively subtle to extremely strange. The leveling amplifier is modeled based on behavior similar to vintage optical compressors. The vocal morpher allows selecting two vocal phonemes to morph between using an LFO, and also allows each of the selected phonemes to be tuned up or down up to two octaves in semitone increments. This can impart talkbox-like or alien-style effects on a vocal. Suffice it to say this isn't the kind of thing you'll use every day, but it might be just the ticket when you're looking for something out of the ordinary in some futuristic context. Of course, just because it is called "vocal" morpher doesn't mean you have to use it on vocals. It can also used to impart interesting periodic sound shifts, almost like talking effects (picture having your guitar say, "ow, ow, ow", for example), on instrumental sounds. You can probably figure out the rest.

All in all, this is a reasonable selection of effects if you need the basics to get started. Of course, SONAR Producer Edition users already have the excellent Sonitus:fx suite, which covers most of the basic "must have" effects, and, to my ears, does this with better quality. The Sonitus:fx suite doesn't include distortion, speaker simulators, or some of the more esoteric effects, though -- SONAR does include AmpSim which does some guitar amplifier simulation. You may be wondering if these effects are of value to SONAR Producer Edition users. The answer is a definite "maybe".

First, there are the esoteric effects, which complement what SONAR has to offer. The rotary and guitar speaker simulators also complement SONAR nicely. While SONAR does have AmpSim for guitar amp simulation, this is one area where I feel the E-DSP-based effects are better. Even if you were to disagree, though, it's always nice to have additional flavors. In fact, I have a number of higher end guitar amp simulation plug-ins, including IK Multimedia's AmpliTube, iZotope's Trash, and Alien Connections' ReValver. While I'd probably prefer AmpliTube or Trash if I could only choose one amp simulator, I like having the additional flavors offered by having more than one, and the E-DSP-based suite provides some nice alternatives. In fact, even were I using, say, AmpliTube on a final guitar (or simulated guitar) track, I might turn to the E-DSP-based effects for monitoring while tracking. The reason is that I can run those with the zero latency monitoring, running the guitar signal directly into SONAR through one of the AudioDock M's front panel preamp inputs. This would be especially useful if the SONAR project in question already had some heavy duty CPU use going on and I was thinking about using Trash, which is probably the ultimate CPU consumer of all the guitar amp simulators I've tried, and which can easily send SONAR to its knees in a heavy CPU use situation. By contrast, even a complicated guitar effects chain using E-DSP places zero load on the CPU. That's right I said zero! Which brings me to my next point...

Did I mention the E-DSP-based effects use none of your computer's CPU for their processing? There may be some insignificant CPU usage to deal with the VST plug-in overhead, but it didn't even measure 1% of the CPU on my Athlon XP 1600+, even with long, involved effects chains running via the PowerFX VST interface. This is great for monitoring effects for tracking, using them directly in the PatchMix DSP inserts. However, it can also be very useful if you've got a busy mix going on, using lots of CPU, and the E-MU effects will suffice for less prominent uses while you use your native effects, CPU hogging or otherwise, for more critical tracks.

Of course, you may find that some of the E-MU effects happen to be just the right flavor for creative uses in the context of your recording. After all, if it were always a best/worst distinction between, say, two models of compressors, you wouldn't see so many flavors out there in common use. If all you happen to have is what came with SONAR, with or without the Sonitus:fx suite from Producer Edition, simply having another flavor would be nice, too.

On the other side of things, while tastes in sound quality are extremely subjective, I suspect not many would prefer the E-DSP-based effects to the comparable Sonitus:fx suite plug-ins, nor to more established brands from the likes of Waves, UA, TC, and others. Of course, you are talking about effects that can cost as much as, or even more than, an E-MU audio interface, depending on which E-MU interface and which third party effects package you happen to pick. Thus, you can rightly consider the E-DSP effects to be a bonus or freebie in the context of the overall value provided by the 1820M or other E-MU Digital Audio System interfaces. In that context, it is hard to argue with the value provided, whether you just use them for occasional monitoring, occasional different flavors, or whatever.

On balance, I find the E-DSP-based effects to be a useful inclusion in the 1820M for monitoring purposes, and that is really the only purpose I've used them for to date. However, to put this in perspective, I have a large selection of plug-ins, with multiple high quality entries in most common processor categories. For someone with a less extensive set of plug-ins, these might be even more useful, assuming the rendering and other reliability issues work in their favor, be it due to a different configuration or future software updates.

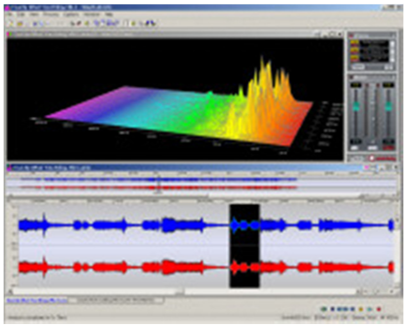
I might add that the E-DSP capability is, at least on paper, extremely valuable. What I would really like to see here is E-MU encouraging other plug-in vendors to port their plug-ins for use with E-DSP. Cakewalk and Waves are you listening? (I think the Sonitus:fx and Renaissance Maxx collections would be two wonderful starting points. Oh, yeah, and don't forget Lexicon Pantheon while you're at it!) Such a development could provide some nice, extremely cost effective, alternatives to ProTools TDM and HD-type systems for those who really want hardware assists, but don't have high-end budgets. I'm talking mainly for the case where people wouldn't be prime prospects for even a UAD-1 or PowerCore, but need a high quality audio interface anyway (and the E-MU range definitely qualifies there) and would like to be able to use the E-DSP capabilities for more than just the E-MU effects suite. Can you imagine even being able to run the Sonitus:fx equalizer built into SONAR's Console View there? I'll stop drooling now.

What's in the Goody Bag?

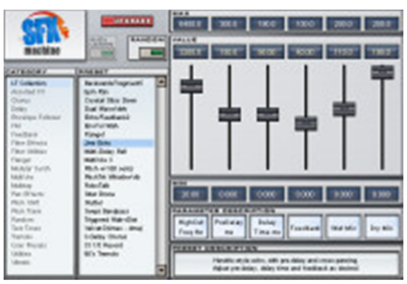
Bundles of software extras that accompany a hardware or platform-type software product are decidedly a mixed bag. You can generally be guaranteed you won't be getting the latest, greatest commercial offerings, unless perhaps in a stripped down "lite" version or time-limited demo form. Nevertheless, sometimes what is included may be useful, especially to users who are just starting out and haven't already built up an arsenal of high-end tools. And sometimes, it is just plain fun to experiment with "toys" you might not have considered otherwise.

What currently comes in the box with the E-MU 1820M is Steinberg's Cubase VST 5.1, Steinberg's WaveLab Lite 2.53, and The Sound Guy's SFX Machine LT. I actually received the Emulator X Studio package, which, in addition to including E-MU's high end Emulator X software sampler as part of the main attraction, adds a number of sample libraries to the overall suite. We will be covering Emulator X in a separate review (timing TBD), though, so I will focus here on the "goodies" included with the basic 1820M hardware package.

Cubase VST 5.1 is a fully featured DAW, albeit one that is several generations behind current mainstream DAWs like SONAR 4, Cubase SX 3, and others. For users who are just getting started out, are budget-strapped, and don't yet have a DAW, Cubase VST 5 has a fair amount of power -- remember, at one time it was actually Steinberg's flagship DAW. Unfortunately, its user interface is far from intuitive, and the on-line documentation doesn't help much in getting past that issue. Nevertheless, users who don't yet have a DAW package, and have sufficient patience to get past the initial learning curve, will be getting a powerful enough package to let them do most basic, and many advanced, MIDI and audio recording chores. Cubase 5.1 includes the ability to host VST and ReWire instruments and VST and DirectX effects plug-ins, and comes with a basic set of VST instruments and plug-ins. Of course, it can also host the E-MU PowerFX plug-ins. Most CakewalkNet readers are likely already running SONAR, maybe even SONAR 4 Producer Edition, so Cubase VST 5.1 will be of limited interest. In fact, it really isn't even a very good way to check out what is going on in the modern Cubase world since the interface and capabilities of Cubase SX 3 have come a long way from Cubase VST 5.1.



WaveLab V 2.53 Lite is a stereo audio editor and recorder. It is also a several generations old (the current state of the art is WaveLab 5), and a lite version to boot. Still, SONAR users who don't already have a modern audio editor, such as Sound Forge or a more current version of WaveLab, may find this of some use. For example, if you're recording stereo samples for sound effects-type uses, it probably won't be all that convenient to record and clean up the samples directly in SONAR. SONAR is really meant to be a larger project environment, and isn't particularly conducive to recording many short, unrelated (or only semi-related) stereo or mono audio files. WaveLab Lite could come in handy for recording and editing such samples, and the ability to apply VST plug-ins (unfortunately, not DirectX plug-ins) at the stereo mix level, in addition to doing most common audio editing tasks, provides a fair amount of tweaking power. The final WAV files could then be imported into SONAR projects as needed. WaveLab Lite also makes it very convenient to import audio files from CD and save those to WAV files, a trick SONAR has not yet learned, but which can be extremely useful in certain types of SONAR projects. I wouldn't go so far as to suggest WaveLab Lite 2.53 would be a good stereo mastering, or pre-mastering, program for SONAR users, though. While it can do many tasks applicable to pre-mastering projects (there is no built-in CD burning, so actual CD mastering would require another application for that stage), and its FFT analysis window provides some very useful visual feedback, its lack of built-in dithering options would seem an important oversight for possible use in this area. Of course, you could use external dithering, for example from a plug-in by Waves, iZotope, or others, for that, but users who had those types of plug-ins would also likely have applications more suited for pre-mastering tasks than this version of WaveLab. For SONAR 4 Producer Edition users in particular, the built-in POW-R dithering would make SONAR a more attractive environment for pre-mastering.



SFX Machine LT is a "lite" version of The Sound Guy's SFX Machine RT plug-in, with the key limitation's of the lite version being that only a very small subset of the full product's effects are available. What is SFX Machine? At its most basic, it is a VST effects plug-in whose actual effects can take many forms. These range from traditional delays, choruses, and flangers, to pretty darn outlandish special effects -- e.g. one of the effects included in SFX Machine LT transforms a voice into robot-like form, and another gives an effect similar to talking into the strings of a piano with the sustain pedal held down. It turns out that the underlying technology is a modular synthesizer, so this may explain the wide variety of effects. While the palate of effects included in SFX Machine LT is limited to 21 presets, albeit "presets" whose parameters can be tweaked (the actual parameters vary according to the type of effect represented by the preset), what is there may be useful on its own. It also provides a good idea of what to expect if you are interested in the full version of the plug-in. While there are certainly general musical uses for SFX Machine LT (and SFX Machine RT), I suspect this plug-in will be of most interest to those with creative sound design needs, such as for electronic music, video games, science fiction movies and videos, etc.

Playing Around

After an extremely straightforward hardware and software installation, albeit with a few temporary snags and logistical cabling/location challenges (see the "Getting In and Hooking Up" section above), it was time to use PatchMix DSP to tailor the 1820M configuration to my needs. Configuring the PatchMix DSP software involved a bit of a learning curve. I could have just short-circuited that by using some of the the pre-configured templates, but I tend to like to know why things work the way they do, and also like to tailor my environment to my needs. Thus, I didn't mind the few front-end "user errors" enroute to tying things into the various applications I would subsequently use with the 1820M. One thing I did note in the process of getting things cabled and "PatchMixed" up was that even running the line outputs of the AudioDock M at -10dBV, into my Hafler P1500 power amp set at its minimum gain setting, could run the volume up to fairly ear splitting levels in my small home studio (and I thought I'd run +4dBu from my Mackie 32*8 mixer into the P1500 at one time). Thus, I ended up reducing the PatchMix DSP master fader by 10dB for my normal listening level. I can increase the level a little if I want to monitor at high volume temporarily. I can also reduce it quite a bit further if I want to monitor at even lower volume. It's nice to have easy software control, including retaining settings across boot sessions, over the mixer and monitor outputs to avoid any rude surprises, while also cutting down on the number of times I need to physically access the hardware (e.g. if changing monitor level required turning a hardware knob).

Once I had the basic hardware and software routing set up, I tried the 1820M with a variety of software packages, and in a number of different types of projects, over the course of just under three months of using the interface between installing it and writing this review. Before giving some details on the applications and software, it is worth mentioning that I took an all or nothing approach to migrating from my previous audio interface, a MOTU 2408 mkII, to the 1820M. I removed the MOTU interface from my system, and put the 1820M's 1010 PCI Card in the same slot the MOTU PCI-324 card had been occupying. This let me use the same IRQ, which had been carefully selected to avoid conflict with other devices (thankfully, there was an open adjacent slot for the Sync Daughter Card). I even ended up deinstalling the MOTU drivers, though that was not initially planned, but, rather, the result of unrelated disk troubles I had around the same time and which caused me to reinstall Windows XP from scratch. While I kept the MOTU hardware handy for awhile, "just in case...", I was never once tempted to go back.

Software applications I tried with the 1820M include Cakewalk SONAR 4.0.2, Cakewalk Project5 1.5 (Project5 2.0 was delivered just prior to publication, and seems to work as well as, or better than, Project5 1.5 based on my initial tests), Sony Sound Forge 7.0 and 8.0, Cool Edit 96, Microsoft Windows Media Player 10, MusicMatch Jukebox, Apple QuickTime 6.5.1, various Native Instruments standalone softsynths, E-MU's own Emulator X 1.5.1, and the versions of Cubase VST and WaveLab Lite supplied with the 1820M. Where ASIO mode was available (e.g. SONAR, Project5, Sound Forge 8, the NI softsynths, Emulator X, etc.), that was used. Where it was not (e.g. Sound Forge 7, the various media players, WaveLab Lite), Wave (i.e. WDM or MME) mode was used. In general, everything worked as expected. There were a couple of points worth noting, however.

With the Wave interface-based applications that provided recording functionality, it was necessary to configure an explicit send in PatchMix DSP, similarly to how a send was configured for ASIO use. While this makes sense, most of the default configurations included a setup for Wave interface output, but not for sending input to a Wave interface-based application. This also makes sense in that the Wave interface for the 1820M only supports a single stereo channel, and E-MU can't predict which physical interface you are likely to want to use. However, those of us used to multi-channel WDM interfaces (the 1820M is only multi-channel in ASIO mode) could easily overlook the need to explicitly configure inputs and to have to select which inputs will be used at the sound card level, as opposed to simply at the application level.

When an ASIO-based application crashed, the 1820M's ASIO drivers remained hung until after a system reboot. Wave drivers were still usable. Thankfully, most software I use, including SONAR 4.0.2 is quite stable. However, there are a few not-so-well-behaved plug-ins that happen to be among my most frequently used software instruments. One of those in particular will hang or crash SONAR upon closing a project if it is still connected. Previously, this was no big deal as I'd simply restart SONAR (killing it via Task Manager if it was hung) and be on my merry way within a few seconds. With the 1820M interface and this issue, though, the few seconds turned into several minutes to include the time for rebooting. Still, this was somewhat manageable, as I often work on a single project all day, and could just close the project at the end of the day. In a second case, though, the application would crash SONAR or Project5 while auditioning different internal effects presets within the plug-in's user interface. This was not as predictable, thus making it harder to work around. Also, its nature was such that it would be more likely to happen more frequently than the other issue, and potentially at highly inconvenient times. E-MU has acknowledged the problem with ASIO driver hangs, and is working on a solution, which is expected to be available with the next update to the DAS drivers, currently expected for June 2005 availability.

Over the course of the almost three months I've been using the 1820M, I have used the 1820M on numerous projects. Most of those would fall in the category of general use with Wave or ASIO playback and/or recording, and there are too many cases to go into detail here. However, let's take a look at a few specific cases that touch on various elements of the E-MU hardware and software:

Perhaps the aspect of the 1820M I was most curious about was its vocal recording chain through its mic preamps. For a number of years -- probably since roughly 1996 -- I have used a CAD Equitek E200 microphone through the preamps on my Mackie 32*8 mixer. Initially this was feeding into a blackface ADAT for recording to ADAT tape, then a bit later the ADAT's A/D converters to digitize the signal for use with a Sonorus STUDI/O board to bring the digital audio into computer-based recorders from Cakewalk and other vendors. For the last roughly 3 years, though, the Mackie preamps fed the A/D converters on the MOTU 2408 mkII, which then fed the digital audio into SONAR. Could the 1820M's TFPro preamps and touted A/D converters improve upon the Mackie 32*8 preamp/MOTU 2408 mkII A/D converter combination? You bet! Let's back up a bit, though.

Plugging my mic into the first of the AudioDock M's mic preamps, engaging the phantom power, and setting levels, the first thing I noticed was that the preamps have an amazing amount of gain. Of course, these preamps are meant to be used with lots of different types of mics, as well as with line level instruments. For my mic, using it with both my own vocals and the vocals of a female country singer I recorded in one session, I didn't even have to turn the preamp up one quarter of the way to get solid levels, and could drive it up into the red well before reaching the halfway point. The good news is that there is an immense amount of gain available if you need it. The bad news is that, when you don't need so much, it makes it a bit tricky to set the position of the gain knob because small movements on the lower end of the scale can end up translating to pretty big gain adjustments. On balance, if you're recording at 24-bit, which I am, that is no big deal -- just set the levels at a safe point, and don't worry too much if they're quite a bit under what a normalized signal would end up hitting. It might be a bigger concern if recording at 16-bit, where every bit of resolution can matter. In that case, you may need to be a bit more careful when turning the gain knob. In my view, you'd be defeating the purpose of working with high quality converters if you were only using them at 16-bits, so perhaps this is a moot point.

As for the quality of the recorded signal, we are talking a significant improvement. I don't know how much of that is due to the TFPro preamps and how much is due to the A/D converters. However, several people noticed significant improvements in my recorded vocal quality when they heard the results, and the preamp and A/D converters were the only difference in my normal vocal processing chain. In fact, the results were so good that one of my colleagues who'd heard the female vocal results I'd recorded with this same setup was asking about my overall recording chain. He'd wondered whether I had a specially treated room for recording vocals, what mic and preamp I was using, and so on. I think he was mildly shocked to hear I was working in an untreated room, with an under $500 (i.e. back in 1995 or 1996) mic less than 6 feet away from my computer (complete with fans and hard drives spinning), and I was running that mic through the 1820M's built-in preamps. Of course, it helped that the singer was superb (for confidentiality purposes, I won't mention her name, but she made it to the semi-finals of American Idol in one of the past seasons). I'd also used Antares' excellent Mic Modeler plug-in to optimize the virtual mic choice to her voice. Still the difference in my own recorded vocals suggested that the 1820M played a key role in quality of the results.

Another area I was curious to explore was plugging an electric guitar directly into the AudioDock M's preamps. (NOTE: E-MU's original specifications for the AudioDock M indicated the preamp inputs also function as Hi-Z inputs. They have since indicated that these inputs do not technically meet Hi-Z specifications, and they will be removing the term from their specifications. The practical difference is that very high impedance instruments, like a vintage Strat, may lose some high end, and thus better results would be obtained by using a direct box. However, I found out about this after conducting my tests, and and don't have a vintage Strat anyway.) Not being a guitar player myself, I didn't have any direct boxes or guitar amps. My wife does have an electric guitar, though, and at one point a few years back I'd tried playing on one of my recordings by simply hooking the guitar up to my mixer's preamps and cranking the Trim knob up. The result was a very noisy recorded signal, which really didn't sound good even through the amp simulators I had available at the time. (It was also really bad guitar playing, but let's not dwell on small details.) Since then, I've used a variety of software amp simulators in conjunction with various software-based sample players, so I was curious about the possibility of using the 1820M's preamps like a direct box in conjunction with these amp simulators. The idea was to record the dry signal, leaving ultimate flexibility in shaping the tone for later on, but I also wanted to monitor through an amp simulator to give at least reasonably decent feedback while tracking. (While it was just me playing the guitar, and my guitar playing hasn't improved measurably over the years -- i.e. it still sucks -- this was meant to be a conceptual experiment, and something I could then use with real guitar players in the future.)

The first step was to plug a guitar in and set up levels. That was easily done. Next I thought I'd try out SONAR's built in guitar tuner plug-in, rather than trying to tune the guitar by ear. That was a little trickier in that I had to temporarily turn on SONAR's input monitoring, which I wouldn't be using after tuning, and I'd had no real experience tuning a guitar with a pitch sensing tuner before (I'd only done it by ear with a piano for reference). But my fights with guitar tuning had nothing to do with the 1820M, so let's move on. (I don't know how you guitar players survive all that retuning -- thank God for solid state MIDI keyboards!)

Turning input monitoring back off, I thought I'd give the 1820M's E-DSP effects a whirl for monitoring while tracking. Bringing up PatchMix DSP's Effects Palette, and going to the Guitar category, I was faced with upwards of 50 preset effects chains! We're not talking just different amp simulations, but combinations of effects and the speaker simulator. I was already tempted to start playing around, but gritted my teeth and just picked one -- I think it was either the Foxy Haze or Foxy Haze 2 preset -- and the preset's name inspired me to try playing a solo version of "The Star Spangled Banner". (Jimi's classic version is not in danger of being upstaged.) At the end of that, I wanted to try something chordal, so I limped through the chord changes for one of my original songs. I recorded both parts into SONAR dry, while monitoring with the effects chain preset in PatchMix at zero latency. SONAR's latency was set to something like 50ms. Cool! (My guitar playing was decidedly not, though.)

While tracking the guitar part, I'd heard a fair amount of noise in the signal, not unlike a live guitar with stomp boxes and an amplifier. I was wondering if the recorded result would be as noisy as I'd remembered my previous attempt, using my mixer's preamps with no direct box, to have been. On the SONAR side, though, the recorded signal was extremely clean, so the noise had obviously been coming from the effects chain, which was placed after the send to SONAR's ASIO input. Nice. In SONAR, I experimented with a number of guitar amp simulators, including the E-DSP-based PowerFX chains mentioned above, IK Multimedia's AmpliTube, iZotope's Trash, and Alien Connections' ReValver. The sound is highly subjective -- I could get settings I liked, as well as ones I didn't like, with any of them (even if none of them could make me sound like a good guitar player). What isn't so subjective is the percentage of CPU usage, on my Athlon XP 1600+, measured by SONAR's meters with the various amp simulators. For example, the native simulators increased the CPU usage roughly 5-11% for AmpliTube, 6-14% for ReValver, and a whopping 24-44% for Trash, depending on what preset was being used in each. With PowerFX, though, no matter how complex the preset being used was, the additional CPU usage was always 0% of my CPU -- yes, that was zero percent, or no measurable CPU usage! I might add that, guitar sound tastes being as subjective as they are, this is an area of the E-DSP-based effects I found to be one of the more interesting ones. When it comes to guitar sounds, there isn't really a right or wrong, and individual players' tones can vary so widely, and I find the E-MU-provided presets to be quite usable on average. Thus its saving so much CPU seems doubly nice.

While playing around with the effects chain presets I stumbled into another problem. From time to time, after clearing one set of PowerFX inserts and dragging in a new preset chain, the PowerFX plug-in would start behaving as if it were a mute switch. That is, if PowerFX were enabled, the sound from the channel on which it was inserted would be muted; if it was bypassed, then you could hear the sound of the channel (but without the processing from the PowerFX chain). Further testing with PowerFX revealed this to be a more general issue on my system (E-MU have not yet been able to reproduce this in their test labs), and it was also experienced in Project5 (also using Cakewalk's VST Adapter). Closing and restarting SONAR would consistently bring back the sound through the PowerFX plug-in. Closing the current SONAR project and reopening it would bring the sound back most of the time, but not always -- the difference seemed to be related to how long after the first project was closed the second one was opened. Sometimes just changing ASIO latency (through SONAR's Options/Audio command then calling up the ASIO Control Panel) would get the sound back, but that worked only occasionally. My further testing suggests that the odds of seeing this problem go up as project CPU usage goes up and/or there are more instances of PowerFX in the project.

Yet another problem was discovered when trying to render PowerFX-based guitar amp simulation to an audio track. Simply trying an "Apply Audio Effects" operation or using SONAR's Freeze operation, resulted in something that sounded like clicking. If I turned SONAR's Fast Bounce off in the Freeze Options, doing the Freeze again got something closer to the original sound, but it was still badly mangled, sounding something like you might expect to hear if you were listening to the sound through a fan. Trying something similar with other PowerFX effects chains, I did find I could sometimes get effects to render in a manner that seemed at least close to what I would expect, but, even then, there might be artifacts occasionally in the course of the entire length of the rendered clip. As of this point in time, I have not gotten to the bottom of what is going on and if there is any consistently successful workaround for the problem when using normal SONAR rendering operations. However, I did find a real-time workaround using PatchMix's ASIO send and return capabilities. In particular, routing the output of the track with PowerFX on it to an ASIO output in SONAR, then routing that back to a SONAR with a send in PatchMix, and recording the result onto a new track in SONAR, using the ASIO input from the PatchMix send, resulted in a perfect recording of what I'd heard when simply playing back the track with PowerFX on it in SONAR. It's certainly nowhere near as convenient as having rendering work as expected, but it'll do in a pinch.

Most of the work I've done in the recent past, and continuing into the present, has been at 24-bit and 44.1kHz. This is true of most of the projects I worked on with the 1820M, too, whether multi-track tracking/editing/mixing projects in SONAR or Project5, stereo pre-mastering projects in SONAR or Sound Forge, or simply stereo listening to reference tracks in Sound Forge or a more consumer-oriented media player. You could say 24-bit/44.1kHz represents my norm, and, as such, was what was in use for the large majority of projects undertaken during my roughly three months with the 1820M. Aside from the above-mentioned issues with PowerFX, which were tried mainly for test purposes (there were no problems using the E-DSP effects directly in PatchMix, such as for monitoring while tracking, which is the main practical use I made of those effects), and with the 1820M's ASIO drivers hanging if SONAR or another ASIO-based application crashed, the operation of the 1820M was largely uneventful. As mentioned above, the quality of the preamps and A/D converters was a definite plus, and an improvement over my old setup, and, to my decidedly non-golden ears, the D/A converters seemed at least as good as what I'd previously been using. Because I'd removed my old interface to install the E-MU interface, it was not possible to do any A/B listening tests to try and compare D/A converter quality. Thus, the only evidence I have, beyond my highly subjective "sounded at least as good as..." observation, that might attest to the D/A converter quality is that projects I've mixed during these last few months have gotten better-than-average feedback compared to projects I'd mixed prior to that period. It is at least conceivable that this might attest to an improvement in the D/A converters, and their effect on my monitoring accuracy. There are, however, two areas where I noted decided differences:

The availability and flexibility of PatchMix DSP meant it was easy to choose zero-latency monitoring through PatchMix DSP over SONAR's input monitoring for tracking vocals. (This would also apply to tracking any other acoustic instruments, of course, but my normal projects tend to be 100% softsynths other than the vocals. The electric guitar experiment mentioned above was designed specifically to try out the 1820M's preamp inputs.) In the past, depending on the project, I'd sometimes had to settle for non-optimal monitoring latencies while tracking vocals through SONAR's input monitoring. I preferred getting nominal latency down to 1.5ms (actual latency would be a fair bit higher due to the path in both directions between SONAR and the audio hardware), but tended to end up going with 2.9ms more frequently. Any higher than that could be disconcerting and likely to make me consider simply taking one ear of the headphones off and not using any actual monitoring of my vocal. With PatchMix's direct monitoring, however, I could set SONAR's latency at a decidedly conservative 50ms, while still getting zero latency monitoring on my vocal. This allowed me to not even worry about rendering softsynth parts for tracking, something I'd often had to do in the past to get input monitoring latency down far enough. Additionally, I could adjust the vocal monitoring level directly in PatchMix without monkeying with SONAR track volumes at all. While I don't like to monitor with reverb while tracking my own vocals, this setup also allowed me to add E-DSP-based reverb after the send to SONAR when working with another vocalist who was more comfortable tracking while hearing her voice with added ambience. With SONAR's input monitoring, that would have added additional CPU load, thus possibly requiring even higher latency. (While my old MOTU interface did have a mixer application that allowed for direct monitoring, I never quite got the hang of using it, and it did not include any effects, be it for monitoring or otherwise.)

When recording softsynths in SONAR with the 1820M, it was necessary to use SONAR's monitoring to hear the softsynths. I am generally not as sensitive to low nominal latency settings with softsynths as I am for vocals. This is partly because there is only a one way trip of the audio from SONAR through the driver and D/A converters then out to the analog monitors, rather than the two-way trip needed for audio recording. Additionally, some instruments have relatively slow attacks anyway, and aren't as sensitive to minor timing variations (a key exception would be percussive instruments, where getting as low-as-possible latency can be critical). In this category, the 1820M generally delivered acceptable latency, but tended not to get quite as low as what I'd previously experienced with the ultra-efficient MOTU drivers. For example, in a case where I might have been able to set latency at 2.9ms with the

MOTU interface, I might have to go with 4ms on the 1820M. Some CPU-hogging softsynths (I won't mention any names to protect the guilty) could get up into territory where it would have been extremely useful to have ASIO latency settings between the 10ms and 20ms marks that the 1820M offers. (At a 44.1kHz sample rate, the 1820M provides settings of 2ms, 4ms, 5ms, 7ms, 10ms, 20ms, 40ms, 50ms, and some additional, higher settings. For practical purposes, though, some softsynths don't like very high settings, so the specific settings mentioned here tend to be the most likely ones to be used, and 20ms is too high for most tracking uses -- one exception might be something like a slow-building pad.)

One other area of application I had a chance to try was related to rehearsing for my live piano/vocal shows. While I go 100% softsynth for instrumental tracks when recording, my live shows use a real keyboard with a quality piano sound on board. Normally I don't even have the audio outputs of that keyboard hooked up in the studio, but rather use the keyboard purely as a MIDI controller. However, when I'm rehearsing for a show I need some way to monitor its sounds. I also like to put a bit of ambience on the keyboard to simulate being in a room with similar size and acoustical makeup to the venue where my show will be. I'd previously used SONAR's input monitoring for this, putting a reverb in the FX bin for the track I had armed to allow my keyboard's sound to be heard. With the 1820M and PatchMix DSP, though, I didn't even bother using SONAR. I simply set up the appropriate analog inputs in PatchMix, used one of the E-DSP presets for room simulation, and monitored directly from PatchMix.

While 24-bit/44.1kHz is my norm, I did try a few projects at other sample rates, both to explicitly try out some of the higher sample rates and to check out other aspects of the 1820M's capabilities as they related to my studio environment:

Back in the 1990s, from the introduction of the ADAT to sometime in early 1998, I recorded all my songwriting demos to ADAT tape, exclusively at 16-bit/48kHz. Those projects are pretty much all old song demos, which I'd be averse to playing to pretty much anyone these days given the significant progress of my production skills over the year (not to mention my songwriting skills in some cases). However, there is a part of me that wants to preserve those old demos for historical/archival purposes, and perhaps for "just for fun" remixing, before I toss the ADAT deck out of my studio for good. Thus, I thought I'd try dumping one project as a test for how this might work with the 1820M's ADAT interface. While SONAR makes you jump through a few minor hoops to start a new project at a sample rate other than the one you've set as your default, once I got that squared away, and started a new PatchMix session directly using a template designed for dumping 8 48kHz ADAT tracks, I was mostly in business. I did have to configure my monitoring setup as the PatchMix session template did not do that (it wouldn't know what you would intend, and driving all 8 channels would almost certainly overload the master out if they were all routed there without fader trims), but this was easy to do on the SONAR input monitoring end, and the actual digital dump of the tracks from ADAT came off without a hitch.

The higher sample rates, 96kHz and 192kHz, are mostly curiosities to me at this point due to their heavily increased demands for disk space and commensurate resource use hikes when you start talking about processing all that extra data through plug-ins, streaming it off disk, and so on. However, I thought I'd try a few quickie stereo projects just to get a feel for how this would go, including how my system might hold up, whether there would be any audio artifacts due to performance issues, etc. In this case, I chose some audio cassettes for dumping to the computer via the 1820M's analog line inputs. At 96kHz, everything went smoothly recording to both SONAR and WaveLab Lite. Working with the resulting files after recording, the CPU load needed for plug-ins added up quickly. Somewhat surprisingly, I could get much farther in SONAR than I could in Sound Forge. I couldn't compare apples to apples with WaveLab Lite due to its support for VST plug-ins only. My gut reaction was that this might be a useful format for stereo mastering-type work, but I'll definitely need a significant system upgrade before I could consider using it for anything else. The other question in my mind was whether it made a difference on the audio quality side. Perhaps my exercises' having been converting from cassettes was a poor test case here, or perhaps my less-than-golden-ears were a limitation, but I honestly couldn't hear any significant difference at a purely subjective level between 24-bit/44.1kHz and 24-bit/96kHz. It is an area, though, where I would like to experiment somewhat more objectively in the future.

As for 192kHz, I tried one more cassette dump at that rate, using SONAR as the recorder. The audio recording made it to disk with no artifacts. Once I had it there, though, I really couldn't do much of anything with it as simply playing the recording back appeared to take sufficient system resources to preclude doing any serious processing. In fact, while I could get SONAR to the play audio file back reasonably the first few times I tried, I couldn't get it to do that 100% of the time. I'd probably lucked into just the right settings the first time out. When I tried doing the same thing in Sound Forge 8, I couldn't get solid playback going at all with either the ASIO (in Sound Forge 8) or Wave (in Sound Forge 7 or 8) drivers. I think I'll need to upgrade my system before I do any further experimentation with the 192kHz sample rate. Of course, at something like 462MB of data for a stereo track that came in under 7 minutes (i.e. at 24-bit/192kHz), not being able to practically use that high of a sample rate isn't exactly causing me any heartache.

To summarize my hands-on use of the 1820M, the 1820M did a good to excellent job of handling most of my reasonable, normal uses over the three months I was using the card. This use crossed over a number of different applications, though SONAR was my most frequent application (Project5 probably came in second, then Sound Forge 7). Some of my less ordinary uses, in particular the projects at 96kHz and 192kHz, were more iffy, but not unreasonably so given the specs of my computer system and the demands of what I was asking it to do. My biggest practical frustration was the ASIO driver hang after an application crash, but, thankfully, the applications I use regularly are solid enough to make application crashes a relative rarity, thus making this issue relatively minor in the grand scheme of things. While I couldn't get PowerFX to work very well in practice, I probably wouldn't have used them much, if at all, anyway, so the problems there were pretty artificial -- i.e. confined to test projects -- in terms of my real uses of the 1820M. The underlying E-DSP effects worked just fine for my more likely use of them for monitoring while tracking. Most significantly, people I trust to provide feedback on my recordings noticed a decided improvement in my vocal recording quality, and the general results I was achieving with my mixes were getting more-positive-than-usual feedback. The latter might be a coincidence, but, then again, it might not be. In any case, I'm pretty sure the former is at least partly attributable to the 1820M's preamps and A/D converters. Perhaps the most telling statement I can make, though, is that, after using the 1820M for approximately two months, I sold my old audio interface, thus cutting out any possibility of a return to my past status quo. Yep, the 1820M is here to stay!

Closing Notes

As someone who keeps reasonably abreast of technology, in both the computer and music arenas, I've grown used to expecting that the product I purchase today will give me lots more capabilities than the product I might have purchased at the same price a few years back. What is much more unusual, though, is to find a product that I purchase today at roughly half the price of the one I purchased a few years back which is still a significant upgrade from the older product. It usually takes a couple of technological generations, sometimes more, to get to that point. From my perspective, though, that is exactly what E-MU has made possible here.

Sure, there are some niggles, a few of them of non-trivial concern. If I were running a commercial studio, where time is money and directly impacts customer satisfaction, I might be tempted to settle with a lesser product, or a much more expensive one if budget allowed, to avoid the risk the most serious of these (i.e. the ASIO hang on a DAW crash) posed. Then again, if I were running that type of studio, I'd be all the more careful about what plug-ins I allowed on my system, and one that sounded great but had a high risk of crashing my DAW probably wouldn't be in operation in front of paying customers.

For my environment, though, time decidedly does not translate to money (unfortunately). Thus, quality of results and functionality, including providing for future growth -- oh yeah, and keeping budget as tight as possible -- are much bigger concerns. In that scenario, which I'd guess applies to a fairly significant portion of CakewalkNet readers, the E-MU 1820M has raised the expectations bar. That is really good news for DAW users, though probably decidedly less so for E-MU's competitors.

Future Goodies

At Winter NAMM 2005, E-MU announced that a new E-MU Production Tools software bundle would ship with all of its Laptop (Cardbus) and Desktop (PCI) Digital Audio Systems beginning in April 2005. While there has been some delay in cutting in the bundling change, and the final schedule is still to be determined, but likely to be during 2Q2005, it is worth noting the changes, as they are significant in terms of potential value to SONAR users and other potential customers.

E-MU's Production Tools bundle includes E-MU's new Proteus X LE with over 1,000 sounds, as well as Cakewalk's SONAR LE, Steinberg's Cubase LE and WaveLab Lite, Ableton Live 4 Lite for E-MU, IK Multimedia AmpliTube LE and T-RackS EQ (not included with E-MU 0404), Minnetonka diskWelder BRONZE (all except E-MU 0404, which will include diskWelder BRONZE Trial Edition), and SFX Machine LT. SONAR 4 users already have the diskWelder BRONZE trial edition, but will appreciate the update to the full retail version if purchasing any of the E-MU cards other than the 0404. The lite version of Ableton Live 4 may be primarily a curiosity for dedicated SONAR users. For those who don't already have IK Multimedia's AmpliTube and/or T-RackS plug-ins, though, the lite version of AmpliTube and EQ portion of T-RackS (with all DAS systems except the 0404) could be quite useful. Of course, SONAR users already have a full working copy of SONAR, but the inclusion of SONAR LE will provide a great way to introduce non-SONAR users to the world we SONAR users know and love.

Of stronger interest for SONAR users is the lite version of Proteus X with its large starter bank of sounds. As of this writing, I have not had a chance to take a hands-on look at Proteus X LE. However, E-MU indicates the sound set is the same Proteus Composer bank supplied with Emulator X. That bank has sounds in most commonly used categories, as well as a large variety of creative sounds, and should provide a nice set of sounds for anyone wanting to augment an existing softsynth collection. For those, just starting out, though, who perhaps have only the softsynths included with SONAR, adding this particular sound bank will likely feel like they've hit the sound jackpot!

While these goodies will come standard for new users of E-MU cards once the new bundling kicks in, existing users won't be left high and dry. E-MU indicates an upgrade to the new software bundle will be made available to registered users for a nominal cost. (I won't spill the beans on what that cost will be in case it changes between now and when it becomes available. Let's just say that, if the figure mentioned to me becomes official, I suspect most users will want to jump on the upgrade.)

**Rick Paul is a songwriter living in Southern California. You can contact him at http://www.RickPaul.info.*