Review: PSP EffectsPack
by Rick Paul - *30th May 2007* -

PSP Audioware's early reputation was established based on plug-in processors oriented toward mixing and mastering. From the stereo field manipulation and analysis tools of PSP StereoPack to the analog-flavored processors of PSP MixPack, PSP's classic plug-ins excelled at making significant improvements through subtle manipulations. More recent PSP MasterPack plug-ins such as the PSP MasterComp mastering compressor and PSP Neon linear phase equalizer continue in that tradition.

The PSP EffectsPack bundle exhibits a very different side of PSP. The name of the game here is creative sound design, with individual plug-ins ranging from software emulation of a classic hardware delay unit to a multi-stage filter plug-in with a wide range of modern and traditional algorithms. Let's take a look at what PSP brings to the table in this area.

Background

PSP EffectsPack consists of four individual plug-ins, which are available separately or for a heavily discounted bundled price. Lexicon PSP 42 is a digital stereo delay and phrase sampler based on the legendary Lexicon PCM 42. PSP 84 is like a PSP 42 on steroids, adding independent controls over right and left delay line channels, resonant filter, and plate and spring reverbs, while preserving the basic sound quality and look and feel of the Lexicon PSP 42. PSP 608 MultiDelay takes things even farther, providing an eight-tap multimode delay processor with a variety of processing possibilities for each of its taps. Finally, PSP Nitro is a dedicated multimode, multi-stage filter.

Windows plug-in formats supported include VST and RTAS. (NOTE: The versions of the plug-ins reviewed here also supported the DirectX format. However, after this review was written, PSP Audioware released updated versions of all of these plug-ins, dropping DirectX support in the process. PSP indicated that there is only one plug-in host that remains DirectX only, with all others providing VST support as either primary or at least equal to DirectX.) AudioUnit, VST and RTAS are supported on the Macintosh under OS X. Lexicon PSP 42 and PSP 84 also support VST and MAS on the Mac under OS 9, but OS 9 is not supported by PSP Nitro or PSP 608 MultiDelay. For purposes of this review, I checked out the VST versions of the plug-ins under SONAR Producer Edition 6.2.1.

List prices for each of the individual processors is $149, which makes the PSP EffectsPack bundle, at a list price of $389 a nice deal. If you don't need the entire EffectsPack, but do want more than one of the plug-ins, PSP's on-line store also provides discounts each time a plug-in from the EffectsPack bundle is added. Most of the retailers I usually check for street prices did not carry these products. However, I did find one retailer who carried the products and shaved a bit off the list prices, with Nitro and 608 MD going for $139 apiece and a $349 price tag on the overall bundle. That retailer also carried a bundle of PSP 42 and PSP 84 for $179, but did not carry those plug-ins individually.

Software protection is either via a user name and authorization key or via i-Lok authorization. Since most companies don't give the user a choice of software protection methods, this is a novel concept. I went with the user name plus authorization key for my tests.

Each of the plug-ins is installed with a separate setup program, and you can choose which plug-in formats get installed. I installed both the DirectX and VST versions on my Windows XP Home (SP plus all latest Windows Updates) system. My general intention was to use the VST versions, and to only use the DirectX versions for comparison in case of troubles in one or more of the plug-ins. I did not encounter any reasons to switch plug-in formats.

Documentation comes in the form of a detailed PDF operation manual for each plug-in. With the exception of the Nitro manual, all of those manuals used the bookmarks feature of the Acrobat format and reader to make it particularly easy to navigate around the document. I hope PSP will consider adding bookmarks to the Nitro manual in the future. The documentation is quite thorough, as has historically been the case with PSP manuals.

Let's move on to the individual plug-ins.

Lexicon PSP 42 (V1.5.0)

As soon as you insert the Lexicon PSP 42 (abbreviated "PSP42" hereafter), before even getting the opportunity to try presets or tweak settings, there is something that pops out as being unique among all PSP Audioware's plug-ins. I am, of course, talking about the Lexicon name. PSP42 is modeled after the classic Lexicon PCM 42 (abbreviated "PCM42" hereafter) hardware delay unit and phrase sampler, apparently closely enough that Lexicon was willing to allow PSP to use its name on the product. The PCM42 is no longer available new, and used units tend to go for upwards of $1,000 on eBay, thus making the less than $150 PSP42 seem like a real bargain. That is especially true when you consider being able to use as many instances as you'd like in parallel, which would be cost prohibitive in the hardware world, and that even a single instance of PSP42 improves upon some of the capabilities of the original PCM42.

At the high level, PSP42 is a stereo digital delay with a variable sample rate. Depending on the sampling frequency, up to 9.6 seconds of delay may be available. Delay times can be synchronized to host tempo or set to explicit amounts of time, and delay time can be continuously controlled without causing artifacts such as clicks and other glitches. Saturation algorithms, based on those used in PSP's line of mastering plug-ins, allow PSP42 to emulate the warmth of tape-based delay units. A modulation section allows modulating both the delay time and sample rate. These modulation capabilities allow PSP42 to go beyond simple digital delay applications, also addressing such delay-based effects as doubling and flanging. A phrase sampler feature is provided for loopers. Host-based and MIDI-based automation are available to control PSP42's settings in real time.

The visual elements of PSP42's user interface largely clone those of the PCM42. Perhaps the most noticeable difference is that, where the PCM42 has a single Level knob, used for adjusting the input level, on its front panel, PSP42 has both Input and Output level knobs. The PCM42 also has an output level knob, but it is located on the back panel of the unit. Let's take a look at the controls provided.

On the far left of the interface is a headroom meter that sits just after the saturation processor, which in turn sits after both the input and feedback portions of the signal. To it's right is a "Delay X2" button, which is used to double the actual and maximum delay times by halving the sampling rate. Even if extended delay times are not needed, this button can be used for lo-fi delay effects. Next up are Input and Output knobs, which are largely obvious in their function in controlling levels. What may not be quite as obvious is that, because the Input knob sits before the saturation processing block, it can be used to control the harmonic characteristics of the signal. A traditional Feedback knob, which controls the amount of the wet signal fed back to the input of the processor (also before the saturation block) is next. To the right of the Feedback knob are two buttons related to the feedback signal. A "Hi Cut" button, if engaged, activates a low-pass filter to roll off 4 dB per octave above 4 kHz. A "FB INV" ("Feedback

Invert") button allows inverting the phase of the feedback signal. Next up is a "DLY INV" ("Delay Invert") button, which inverts the phase of the wet signal, if engaged. Dividing the first section of PSP42 controls from the next section is an "Output Mix" knob. As you might guess, it controls the balance of wet and dry signals in the output of PSP42, with the midpoint representing an equal blend of wet and dry signal, with each component signal attenuated by -6 dB.

The next section of controls on PSP42 controls the basic delay parameters. First up is an "∞ RPT" (literally "Infinity Repeat", but more accurately "Infinite Repeat") button, which engages PSP42's phrase sampler mode. This is a more or less real-time control in that what is repeated depends on when the button is pressed. When PSP42 is not synchronized with the host's clock, that will be the current delay buffer, and the button has immediate effect. I say "more or less real-time" because, when PSP42 is synchronized to the host's clock, the repetition is activated at the next clock cycle synchronization point, thus ensuring capturing a perfectly synchronized loop. A short (10 ms) crossfade is applied to ensure smooth transitions between loops.

To the right of the "∞ RPT" button are two sets of controls. Up on top is the "DLY<-->CLK" ("Delay<-->Clock") slider, which determines whether to set delay times in milliseconds or whether to lock the delay time to the host clock and set it in tempo-related values, respectively. Just below the slider are two buttons, one with a down arrow and one with an up arrow, that can be used to adjust the delay time in either case. In DLY mode, the meaning of the buttons is fairly obvious -- i.e. decrement or increment, respectively, the delay time -- though Shift and Ctrl keys can be used to make the increment sizes 10 and 100 milliseconds, respectively, instead of the default of 1 ms. In CLK mode, the down arrow button cycles through the allowed numerators, while the up arrow button cycles through the allowed denominators. Note that the fraction is relative to a measure, not a beat. Thus, for example, a setting of 1/4 (i.e. numerator of 1 and denominator of 4) in 4/4 time would represent a quarter note delay. To the right of these controls is the delay time indicator, which will either show the delay time in milliseconds for DLY mode or the numerator and denominator set in CLK mode. However, this delay time indicator also doubles as an invisible slider control for setting delay time quickly without using the buttons. Just to the right of the delay time indicator is a Manual knob, which allows setting the internal delay buffer sample rate in relation to the host sample rate. Settings range continuously from half the host's sample rate to 1.5 times the host's sample rate. Note, though, that the actual internal delay buffer sampling rate is also affected by LFO and envelope detector settings (see below).

The remaining controls, over on the right side of PSP42's interface, govern the modulation of the delay time and sample rate by the LFO and envelope detector. Keeping with typical modulation setups, the Depth knob controls to what extent the internal sample rate, which determines both delay time and playback pitch, is affected by the modulation signal. The Waveform knob switches between sine and square wave LFO waveforms while controlling the proportions of the LFO and envelope detector signals blended into the final modulation signal. This is a fairly clever setup with the 100% envelope setting smack dab in the middle of the knob and the sine and square wave LFO settings on each end of the knob's travel. If the knob's position is somewhere on the left side, it is blending between a sine wave LFO and the envelope detector. If it is somewhere on the right side, it is blending between the envelope detector and a square wave LFO. Finally, the Rate knob sets the LFO frequency continuously between 0.1 Hz and 10 Hz.

Never having had a PCM42, I found it took me awhile to get used to PSP42's user interface. Some of the user interface concepts were clearly designed with the limitations of hardware, such as front panel space and cost, in mind. For example, that clever LFO/envelope detector blending knob I mentioned could easily have been served by a button to choose the type of LFO waveform and a knob to control the blend between LFO and envelope detector. Of course, that would have meant an extra button and more front panel real estate. Similarly, the up/down buttons for setting delay time in DLY mode are relatively logical, but not so logical when a down button is toggling numerator values and an up button is toggling denominator values. While I might have preferred if PSP had at least offered the option of an interface that took advantage of the plug-in's being software, I did ultimately get used to using the PSP42 controls. There is one upside to the space and control conservation, though. In particular, PSP42 takes up relatively little screen real estate compared to most modern plug-ins.

One area of capability that the PCM42 did not include was presets. PSP42 includes PSP's standard preset management system down near the bottom of its user interface. In addition to providing access to file-based banks of plug-in presets, the preset management system includes a MIDI learn facility intended to allow mapping MIDI

continuous controllers to the various controls on PSP42's user interface. It is incorporated in all four EffectsPack plug-ins, though the MIDI learn facility is relocated in PSP608, and Nitro and preset management is relocated in Nitro.

Like the original PCM42, PSP42 packs a significant amount of functionality into a single delay line and a relatively small space. Some other basic digital delays provide an extra tap or three, or perhaps provide variable frequencies for their low-pass filters, but they may not provide the modulation options PSP42 provides or the flexibility for setting delay time increments when in host clock-synchronized mode. In general, though, you can do most of what you'd want to do with a basic delay, and a whole lot more, with PSP42.

Capabilities are nice to have, but once you get past basic level of capability, the real question is how a plug-in sounds. Of course how something sounds is highly subjective, and talking about sound isn't very useful? Was PSP42 warmer than other delays? Perhaps? Smoother? Maybe. Etc. What struck me most about the sound of PSP42, though, was that it just seemed to sound good, at least to my ears, with less work on my part than with other delays. Also, even after doing the work to get other delays closer to what PSP42 seemed to do effortlessly, PSP42-processed tracks seemed to sit better in the mix. That really is the bottom line, isn't it?

PSP 84 (V1.5.0)

The visual design and name of PSP 84 (abbreviated as "PSP84" hereafter) may make it tempting to think of PSP84 as a dual PSP42. Tempting as it may be, it would also be also incorrect. In fact, PSP84 can do pretty much everything PSP42 can do -- the key exception is phrase sampling -- and it does have independent controls over the right and left channel delay lines, whereas the PSP42 only has a single set of controls for both channels. That is where the double PSP42 analogy runs out, though, while the functionality of PSP84 keeps on going. PSP84's enhancements over the PSP42 fall into two categories: additional flexibility and new features.

There are several features of PSP42 that get additional flexibility in PSP84. Whereas in PSP42, the saturation algorithm is fixed, and controlled mainly through the use of the Input knob to drive the input and feedback signals to the point where saturation starts affecting the output signal more, the PSP84 also adds a Drive knob to control saturation depth. PSP84 also allows turning the saturation block off entirely in case you'd prefer a more digital flavor. Where PSP42 provides a fixed high cut filter at 4 kHz, PSP84 provides a much more flexible filter module that not only allows varying the cutoff frequency, but also changing the type of filter, and controlling its shape, what it affects, and even allowing modulating the cutoff frequency. We'll get into more detail on the filter block when we go through PSP84's controls.

PSP84 also includes some brand new features not present at all in PSP42. Of course, the independent controls for each delay line channel are the biggie, and perhaps the main thing that differentiates PSP42 from PSP84. The addition of the second set of controls also mandates providing a pan control for each delay line, as well as a stereo link capability. PSP84 also adds a reverb block to the processing options, with both spring and plate reverb algorithms provided.

Visually, it is easy to tell that PSP84 and PSP42 are of the same lineage. Not only is the overall aesthetic identical, but the various controls look and behave similarly on each module. I'll avoid redundancy by referring you back to the PSP42 section for an explanation of the common controls. However, differences in layout of the two modules, as well as the new controls added to PSP84 do warrant some discussion.

If you considered the notion of PSP84 as two vertically stacked PSP42 modules, perhaps the biggest layout difference is that, though the headroom meters are all the way to the left of the modules, the input-oriented controls and output mix controls have been moved all the way over to the top right corner of the PSP84. This initially caused me some head scratching as I was looking for the Input and Feedback knobs next to the headroom meters since those knobs govern the values displayed by the meters. Instead, the Input knob, Mix knob, and Output knob are in a

Main section on the other side of the plug-in's interface. This makes perfect sense for the Output knob, which controls the signal level output from the plug-in. It also makes reasonable sense for the Mix knob, which governs the wet and dry balance of the output signal, though it may cause a slight navigation slowdown for someone who is bopping back and forth between PSP42 and PSP84. It is far less intuitive for the Input knob, which would logically be the first control in the signal processing sequence. It seems fairly obvious that this relocation was made to keep the overall interface conforming to hardware layout aesthetics, since it would have been unbalanced (i.e. since there is only one input knob but two headroom meters and sets of delay controls) over on the left side of the interface. This is one of those cases where it would have been interesting to see what PSP might have come up with had they just thrown away the hardware model and started from scratch on their interface design. (Or course, saying that, I risk getting ahead of myself. See the PSP 608 MultiDelay section below for more.)

Moving back to the left of the interface, the two stacked delay modules are next up after the meters. There is one module for the left channel and another for the right. I should clarify that these modules correspond to the left and right input channels as what ends up in the left and right channels on output is under the user's control. In particular, there is a new "FB Pan" ("Feedback Pan") control in each of the delay modules to allow panning the feedback for that channel anywhere in the stereo spectrum. If the delay modules are linked for stereo operation, by engaging the Link button, the Feedback Pan controls will result in equal, but opposite direction, movement, while all other controls will move identically for each module. Most of the rest of the delay module controls look and work similarly to PSP42. The "DLY<-->CLK" switch on PSP42 has been renamed to the more intuitive Mode switch with "Time" and "Note" as its possible settings. There is also a new Range switch, which governs the time range available (possible values are 100 ms, 1000 ms, and 5000 ms) when the delay units are in time mode. If you don't need long delay times, setting this range to lower number allows finer granularity in setting delay times.

To the right of the top (i.e. left channel) delay module is the display. By default, it displays the host tempo when you are in Note mode or whatever tempo you set when you are in Time mode. However, it also works as a general-purpose display to show the value of various controls when you mouse over or manipulate those controls. This is a very useful enhancement over PSP42.

Continuing to the right in the top layer of the virtual rack module, the "FLT" ("Filter") module is next. The filter in PSP84 is quite flexible. The "FLT" label itself serves as a switch to either turn the module on or bypass it to conserve CPU power. A Mode switch determines whether the filter will be applied to the input signal prior to any processing ("IN"), both the feedback and wet signals ("FB"), or the wet signal only ("FX"). An "FLT TYPE" ("Filter Type") switch selects between low-pass ("LP"), band-pass ("BP"), or high-pass ("HP") operation. A Cutoff knob allows setting the corner frequency for low-pass/high-pass filters or the center frequency for the band-pass filter between 50 Hz and 16 kHz. A "RES" ("Resonance") knob allows for the possibility of emphasizing the frequency around the cutoff frequency to varying degrees. Finally, the "MOD" ("Modulation") knob allows modulating the cutoff frequency within a plus or minus two-octave range. It controls modulation depth and phase of the modulation waveform.

To the right of the Filter module is the "DRV" ("Drive") module, otherwise known as the user controls over the saturation block. As with the FLT module, the DRV module's name serves as a bypass switch to turn the saturation processing off if desired. There is also a Drive knob, which allows setting saturation gain in the range from 0 dB to 24 dB. Note that a setting of 0 dB does not turn saturation off. Only bypassing the DRV module does that.

Moving to the bottom portion of the virtual rack module interface, the "MOD" ("Modulation") section sits to the right of the right channel delay module. The MOD module's name also serves as a bypass switch to activate or deactivate the modulation processing. A Mode switch allows setting the LFO rate in Hz ("Freq") or tempo-related rhythmic values ("Note"). Up and down buttons under the mode switch operate just like the similar switches in the delay modules, including exhibiting the different behaviors in "Freq" and "Note" modes. Waveform labels, which serve as waveform selection switches, allow setting the LFO waveform to one of five possible selections: sine, square, triangle, sawtooth, or "random" (i.e. synchronized to the start of a measure if the VST host is capable of sending current song position information). A Phase knob sets the phase offset between the LFO waves applied to the left and right channels in a range of plus or minus 90 degrees. A Source knob controls the blend between the LFO and envelope follower signals in controlling the modulation signal. A "Sens" ("Sensitivity") knob sets the

envelope follower sensitivity, and a Speed knob controls envelope follower attack/release time within the range of 1 millisecond to 1 second.

The VCO ("Voltage-Controlled Oscillator") section is next up. This section, which is also enabled or disabled by clicking on its name, determines whether PSP84's internal delay lines operate at the same sample rate as the host, or whether they are controlled by the VCO. The Manual knob functions as in PSP42. A "MOD" ("Modulation") knob controls the depth of the sample rate modulation.

Last, but certainly not least, is the "RVB" ("Reverb") section, which can also be enabled or disabled by clicking on its name. A Mode switch allows choosing whether the reverb is applied only to the wet signal ("FX") or to the entire output signal from the plug-in ("OUT"), including the blended-in dry signal. A type switch is used to select either spring or plate reverb. A Damp knob controls reverberation damping, and thus effective reverb time, and an "AMT" ("Amount") controls the proportion of reverb signal mixed into the output of the reverb module (whose content varies depending on the setting of the Mode switch).

The preset management system of PSP84 is the same as for PSP42. However, PSP provides significantly more presets -- 60 on PSP84 versus 16 on PSP42 -- to demonstrate the broad range of additional applications for PSP84. Those presets range from simple delays to doublers and flangers to designer echoes (e.g. "Shimoni Caves") to ambient noise generators (e.g. "Rain Forest").

As someone who primarily thinks of digital delays in musical contexts such as vocal slapbacks, automated doubling, flanging, and so on, I found it fairly amazing what this "simple" dual line delay could do. Sound designers should love it. Of course, it still handles the simple things well, too, with the same warm sound quality as PSP42, at least if you want that same warm quality -- you have more of a choice with PSP84. In my book, the real key as to whether to use PSP42 or PSP84 for a musical application comes down to whether you need dual delay lines, or one of the other additional controls provided by PSP84. If not, PSP42 tends to make the "correct" decisions for most practical applications, thus making it very quick to get to the results you want to achieve. If you need that extra flexibility, though, PSP84 significantly extends your capabilities, from minor additional tweak territory up into some serious sound design capabilities.

PSP Nitro (V1.1.0)

PSP describes PSP Nitro (abbreviated as "Nitro" hereafter) as being a multimode filter plug-in. What exactly does that mean? What do you want it to mean?

Though the above may seem evasive, the notion I am trying to get across is that what Nitro actually does depends heavily on what you want it to do. It brings a whole raft of capabilities to the table, from classic high-pass and low-pass filters to bit-crushers and comb filters. However, even providing a list of all the filter types available in Nitro would not convey its capabilities, because Nitro allows combining up to four individual filters, which PSP refers to as "operators", in parallel and/or series, with feedback loop capabilities, to create a complex filter.

Before I elaborate on Nitro's capabilities, it's full disclosure time. I was briefly involved in the late beta testing of PSP Nitro 1.0. Prior to that beta test, my only exposure to filters came from some limited analog-style synth programming, and I found it hard to even come to grips with what a plug-in filter would be, no less how I would use it. However, the presets that were already available at the time gave me a general feel for Nitro's sound shaping potential, and, once I got past a slight learning curve, I was even able to participate in developing a few presets that got included in the final Nitro 1.0 product.

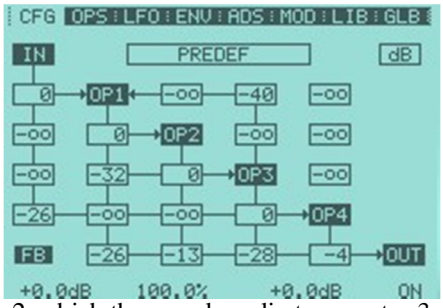
When you first insert Nitro in your project, you will see its faux hardware interface, complete with several knobs, sliders, and switches, as well as a pair of meters, some LED-style labels, and a fairly big central LCD-style display.

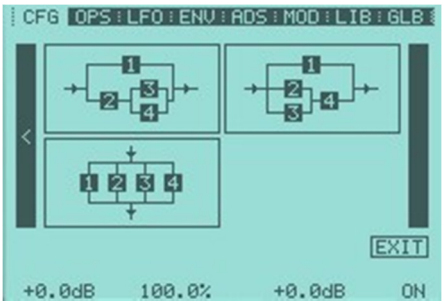
The screen is divided vertically into three main sections. The center section contains the multi-purpose LCD-style display, which we'll discuss further below, as well as the meters, a "power" (or bypass) switch, and three knobs (for adjusting input level, wet/dry mix, and output level). The left and right portions of Nitro's interface have basic controls for each of the four "operators", or individual filter modules, plus labels for the two operator-specific parameters that can be modified through this part of the Nitro interface. The common controls for each operator include an On (or bypass) switch to take the operator's processing in and out of the signal chain while preserving routing, a Solo switch to take any other operators that aren't soloed out of the signal chain, an M/S switch to toggle the operator between left/right and mid-side processing, and a Level slider to control the output level of the operator. There are also two additional knobs for each operator whose meanings will depend on the filter algorithm being used in that operator. An LED-style label under each of those knobs indicates what parameter the corresponding knob adjusts. For example, a filter module might provide access to cutoff frequency and resonance, while a lo-fi filter might provide access to bit-depth and sample rate.

While the basic Nitro interface I've just described provides access to quick adjustment capabilities, it neither gives you any information about the signal flow between operators nor provides sufficient information to tell you what type of filter is in use in each operator. Nor does it allow you to tweak anything beyond the two exposed parameters from each operator and the basic operator and global controls described above. For this you'll need to use the multi-purpose LCD display.

At the top of the LCD display is a row of screen names. The default screen that comes up when Nitro is first loaded is the "LIB" ("Library") screen. It basically provides a list of presets, facilities to load and save presets, the ability to switch banks, and a few other capabilities. To see the how the operators in the current patch are configured, we'll want to use the "CFG" ("Configuration") screen. The configuration screen shows a matrix of operators, plug-in input and output, and connections between all of these. Reading the configuration matrix is a bit tricky at first, but the basic gist of it is that, to tell if an operator sends audio to another operator (or to the output of Nitro), you look in the column of the first operator and the row of the second operator. In the cell that forms the intersection of that row and column, you will either see "-∞" (i.e. negative infinity) or some number between zero and negative infinity. If the cell contains negative infinity, then the operator in that column does not directly send audio to the operator in that row. If, on the other hand, there is a number in the cell, then the operator in that column does send audio data to the operator in that row, attenuated by the number of decibels displayed in the cell.

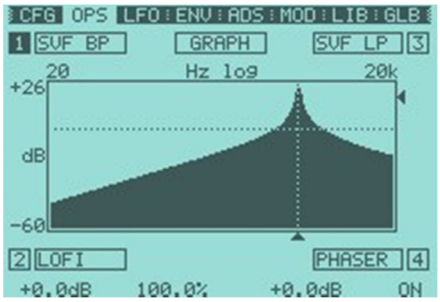
For example, if you look at the matrix to the right of this paragraph, you will see a fairly complex signal routing configuration. I won't go through all of it, but, for example, if you look at the column under the input at the left of the matrix, you will see the row corresponding to operator 1 has contains "0" (zero), and the row corresponding to operator 4 contains "-28" (i.e. negative twenty-eight), while the rows corresponding to operators 2 and 3 contain "-∞" (i.e. negative infinity). This means the input of Nitro gets routed directly to operators 1 and 4, with no attenuation of the input in routing its audio to operator 1, but attenuating the input by 28 dB before routing it to operator 4. Note, also, that operator 1 sends audio directly to operator 2, which then sends audio to operator 3, which further sends audio back to operator 1, creating a feedback loop, thus resulting in the "FB" symbol at the lower left of the matrix. I will confess to having created this particular signal routing by fairly arbitrarily making connections (which are created by dragging the mouse up over a cell with a negative infinity in it) to see what would happen in terms of the connection diagrams. Surprisingly enough, even with this mess of connections, the actual Nitro patch, which was processing an electric guitar signal, actually sounded fairly good! I assume that was helped along by severe attenuation on some of the routings.





The matrix is the only way to understand what connections are made in a patch you aren't creating from scratch, and a useful way to change connections or adjust them while being cognizant of the signal flow between operators.

However, a simpler interface is provided for just getting started in setting up the signal flow for a new patch. Clicking on the "Predef" button in the Configuration screen brings up several screens of predefined signal routing configurations. Choosing one is as simple as clicking on the one that routes things the way you prefer. It also provides a nice way of getting a quick glimpse at how powerful and flexible the signal routing capabilities of Nitro are, even if you can create even more complex routings in the matrix.



Having a view of the signal routing still isn't enough to tell you what is going on inside Nitro because you don't yet know what each of the operators is actually doing. Each one of them could be acting as any one of the seventeen available filter types provided by Nitro, and the operator could also be turned off. The "OPS" ("Operators") screen is used to select which filter type will be used in each operator, and can also be used to configure parameters, either graphically or by numerical values, for the currently active operator. The sample screen at right, which comes from the "Walkie Talkie" preset, indicates that operator 1 is a high-pass filter of the state variable filter (SVF) variety, operator 2 is a lo-fi filter, operator 3 is a low-pass SVF filter, and operator 4 is a phaser. The current operator (indicated by the black highlight around the operator number) is operator 1, and its filter graph is shown. The filter graph can be manipulated by dragging the little triangles on the bottom and right sides of the graph to manipulate cutoff frequency and resonance, respectively. Filter types available include state variable filters (high-pass, low-pass, band-pass), bi-quad filters (high-pass, low-pass, band-pass, band-reject), Moog filters (high-pass, low-pass, band-pass), a comb filter, a phaser, a lo-fi filter (i.e. for simulating bit-depth and/or sample rate reduction), a saturation effect, a stereo width/balance module, a filter for independently panning right and left channels, and a "Glide" filter (a fractional time delay line). When you add all the filter types to the ability to combine up to four of them in most any routing configuration you'd like, you may begin to understand why my answer to the question as to what Nitro can do is, "what do you want it to do?" There is still more, though.

Nitro also provides the ability to modulate pretty much any parameter of any operator, as well as its global parameters, via two LFOs (Low Frequency Oscillators), an Envelope Detector module, an ADSR (Attack-Delay-Sustain-Release) envelope generator, or most MIDI messages. The LFO, ENV, and ADS pages, respectively, are used to configure the first three types of sources. A MOD ("Modulation Matrix") page is used to direct individual modulation sources to the parameters they will control. We won't get into the details of these areas of Nitro's interface. Suffice it to say these modulation options add significantly to the overall possibilities offered by Nitro for dynamic sound processing.

Nitro comes with a bit of a learning curve if you want to dive into programming your own patches. However, it also makes it fairly easy to experiment by biting off a little at a time. For example, you can start by simply tweaking the parameters that are accessible in the main Nitro window via the multi-function knobs, then progress later to changing filter algorithms, rerouting signal flow, and adding or changing modulation routings and settings. If you just want to get a quick overview of what it can do, though, simply set up a looping part and play with moving around in Nitro's extensive library of presets. I didn't make a count, but I'm pretty sure the number is well in excess of 150 presets, and these range from simple processors to extremely creative dynamic effects. That dynamic aspect can be particularly useful when trying to add life to a static sound. As a result, Nitro has become a go to plug-in for me when I'm starting to feel like a sound, though in the ballpark of what I'm seeking, is feeling a bit stale. Rather than go through vast libraries of synthesizer presets or sampled sounds, looking for something else that is in the ballpark, but a bit more lively, I'll often just plug Nitro in after the basic sound, then go through its presets until I find something that is even closer to what I'm seeking. Of course, I can then do further tweaking at the Nitro level if necessary, but often just flipping through the presets will provide multiple options that resolve any staleness in the original sound.

It may be tough explaining what exactly Nitro does because, in fact, it can do so many completely different things. It is pretty easy, however, to predict that, if you have wide ranging sound shaping needs, you are likely to find Nitro will become a valuable tool for meeting those needs.

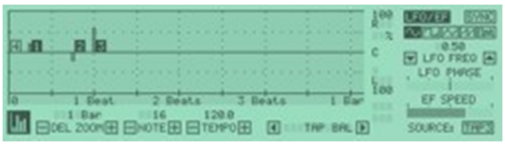
PSP 608 MultiDelay (V1.1.0)

PSP 608 MultiDelay (abbreviated "PSP608" hereafter) is PSP Audioware's newest EffectsPack plug-in. At its heart, PSP608 is a multi-tap delay unit with up to eight taps of up to eight seconds apiece. Each tap features independent control over feedback, stereo image and position, delay time, multimode filter type and settings, modulation blend and depth, drive/tape saturation, and reverb level. Various other options with respect to signal routing configurations provide an immense amount of creative control. Let's take a closer look.

As soon as you open up PSP608's user interface, you will see that this is quite a different animal from PSP42 and PSP84. While PSP608 features a hardware rack look, the top two virtual rack spaces are dominated by a large, multifunction LCD-type display, while the bottom five virtual rack spaces are occupied by modules reminiscent of mixer channel strips. Where PSP42 and PSP84 economized on knobs and buttons as if each one had a real hardware cost, PSP608 is generous in allocating dedicated hardware controls to the majority of the plug-ins functions, while the LCD display provides additional capabilities via a more software-like interface.

The differences between PSP608 and the other two PSP delays go beyond the look and feel of the user interface, though. The variable internal sample rates of PSP42 and PSP84 are gone. Instead, PSP608 focuses on signal processing options and routing flexibility. That isn't to say PSP608 hasn't benefited from PSP's development of PSP42 and PSP84. For example, the spring or plate reverb and saturation block will be immediately familiar to PSP84 users, and many of the controls that are common to two or three of the plug-ins behave similarly. No doubt there are other shared technologies beneath the hood, as well. However, where PSP42 sets out to fairly faithfully recreate a classic piece of hardware in software while adding a few enhancements, and PSP84 builds from there in adding features, PSP608 feels a lot more like PSP started fresh, without any history-imposed limitations.

Looking more closely at PSP608's user interface, the top two virtual rack spaces are dominated by the LCD display. It is subdivided into three sections. The left half of the LCD is the MultiDisplay. As the name suggests, it is home to multiple controls and information displays. These include global level meters, context-sensitive information on the parameter being adjusted at any given time, and a number of global controls, such as for setting the mode of the delay time (i.e. time in milliseconds, note fractions, or note fractions synchronized to the host's clock), MIDI learn functionality, and the basic note unit value (equivalent to the denominator setting in PSP42 and PSP84) to be used for setting delay times in note or sync mode. The other half of the LCD is divided into two sections. The "Tap Params" ("Tap Parameters") section provides a bar graph for viewing and editing the same parameter from multiple taps as an alternative to adjusting the hardware-look controls in the tap-specific strips in the central section of PSP608's user interface. On the far right, the Modulation section provides the controls for the LFO and Envelope Follower blocks, which are not duplicated in the hardware-look portion of the interface.



The LCD display also has an alternate display mode called Graph Mode. When this mode is active, the MultiDisplay and Tap Params sections of the LCD are taken over by a graph that shows the relationships between the same parameter of multiple taps. Which parameter is displayed depends on the most recently edited parameter. The graph is display-only -- i.e. any editing must be done via the hardware-look interface for the individual taps.

To the left of the LCD are the Main controls (i.e. Input level, wet/dry Mix, and Output level), as well as the Power button, which serves as a bypass switch for PSP608's processing when disengaged. On the right-hand side of the LCD, up in the top corner, is a large Tap Pad to provide "tap tempo" functionality. The Tap Pad works in two modes. In Tap mode, the time between clicks sets the delay time for the selected tap. In Tempo mode, the time between clicks sets the tempo of the plug-in when the plug-in isn't synchronized to the host's clock. Below the Tap

Pad are the controls for the spring or plate reverb, which are very much like those for PSP84. Note that the controls in this section only set the general reverb parameters. The contribution of each tap to feeding signal to the reverb is set below in the tap-specific parameters.

Let's skip now to the very bottom of PSP608's main user interface, just above the preset management system. This is the Master Feedback section. On the far left is a switch to turn Master Feedback on or off. Just to the right of that is the master feedback Mode display and toggle, which allows switching between MultiDelay and MultiTap modes. In MultiDelay mode, each of the individual taps of the delay has its own feedback buffer, and so can have feedback that is independent of any other tap. All taps can send their signal into the master feedback section, but only one of those taps will be the source for master feedback processing. That tap is chosen via a Source parameter toggle next to the Mode switch. In MultiTap mode, there is only one feedback buffer for all taps, and the FB ("Feedback") button from an individual tap strip selects which track is used. Next up to the right is the master feedback Gain control and independent Pan controls for the Left and Right channels, with a Link switch between them. Note that the link switch for the master feedback pan controls does not do an inverse link like the similar control for PSP84, where the channels would eventually cross over each other if panning far enough. Rather, it links them in the same direction of travel, preserving the width between the two controls until the point where there is no farther for the outermost control to go. After that the width gets narrowed until the controls are both at the extreme, after which they will travel together if panned back in the other direction. Further over on the right are the master feedback's Filter and Saturation sections, both of which have controls similar to those for the individual taps (see below) -- the master feedback Filter section has a few less controls -- and a Panic button. The Panic button will turn off all feedback when pressed, so serves as a safety valve in case of runaway feedback loops.

The middle section of PSP608 is where most of the action will take place. It contains eight half-height virtual rack modules, one for each tap of the delay. PSP doesn't give these a specific name, but I will refer to them as "tap strips" because their physical look reminds me of track strips from a mixing console, albeit horizontally-oriented and with the "fader" (here used for setting delay time for the tap) in an unusual place.

Over at the far left of each tap strip is an LED-style tap number. By default, these numbers are unlit, meaning the tap is turned off. Clicking on the number turns that tap on and illuminates the LED. Right clicking on the number solos (or unsolos) the tap. Just to the right of the tap number, and still part of the LED mechanism are two on/off LEDs that are activated or deactivated by clicking them. The topmost one is the "FB" ("Feedback") button. In MultiDelay mode, it turns the feedback mechanism for that tap on or off in. In MultiTap mode, it is used to select which tap is sent to the Master Feedback section for processing. The bottom button is used to include that tap in any linking that might be done between track strips. The former is obvious, but the latter could use some explanation. This is not a stereo link, but rather a link between the controls of any linked tap strips, such that, if you move a control on one of those strips, the corresponding control on another strip will move. There are various rules and modifier keys to govern exactly how the corresponding strip will move. For example, in time mode, tap delay sliders have a relative link, which maintains the ratio of timings between the strips. For example, if one linked slider is set to 25 ms and a second is set to 50 ms, and you move the first of those to 35 ms, then the second will move to 70 ms. However, if you use the shift key in conjunction with moving the faders, this keeps the distance, in time, between them constant. In that case, the same fader move would result in the second fader's landing at 60 ms. Use of the Alt key temporarily overrides the linking, to allow adjusting only one of the faders without affecting the other. There are other rules for other modes and types of controls, but, in general, the most intuitive linked behavior occurs with no modifier key.

Next up to the right is the Feedback section of the tap strip. In MultiDelay mode the "AMT" ("Amount") knob controls the level of each tap's signal that is sent to the Master Feedback section. In MultiTap mode, the value of the Amount knob for the selected tap interacts with the value of the Amount knob in the Master Feedback section to determine how much signal is sent into the Master Feedback section. Next up a Post button determines whether a tap will be fed back before or after the processing section when in MultiDelay mode (it has no effect in MultiTap mode). A Tap Input Gain knob controls the signal level being fed into processing section for that tap. Stereo field Width and Balance knobs are next up, then the most visible element of each tap strip, the Delay Slider. The Delay Slider is used to set the delay time or delay note, depending on the mode of the delay. If operating in note or sync mode, it effectively sets the numerator for the delay time fraction. If operating in time mode, it sets the delay time in milliseconds. A Delay Zoom Setup control in the LCD screen can be used to fine tune the resolution of the slider.

To the right of the Delay Slider is the Filter section for the tap. Eight types of filters are provided including high-pass, low-pass, band-pass, and various shelving and peak filters with resonance. Controls are provided to turn the filter on or off, set the type of filter, and to adjust cutoff frequency, resonance (or filter bandwidth for non-resonant filters), and gain (for all except low-pass/high-pass filters). The Modulation section is next with its "SRC" ("Source") button to blend between modulation via the LFO or the Envelope Filter, "PHS" ("Phase") button, and Depth knob. That is followed by the Saturation block with its on/off button, pre/post-post filter button, and Drive knob, then the Reverb section with its on/off button and Send knob.

If PSP608's interface looks daunting at first, keep in mind that the eight tap strips dominate most of its surface, and each of those is identical to the other one. They simply control different taps. If all you need is a single tap reverb, you can just turn all taps off except the one you need and set that one as you would for a simpler delay. Building more complex delays can be achieved by adding a tap at a time. There is lots of flexibility there if you need it, but you don't have to use every feature if you don't need it, either. The only real downside to all the controls' being there is that PSP608 does take up a good deal of screen real estate -- 3.5 times that of PSP84 and seven times that of PSP42.

Once I got used to PSP608's interface, it felt more intuitive than the other PSP delays. I especially appreciated all context-sensitive information in the MultiDisplay, which made it very easy to tell what I was doing with my knob movements and helped me avoid the kinds of user errors that are all too easy to make on a user interface that features a lot of similar-looking controls. I also very much appreciated the one parameter/one control layout of the hardware-look portion of PSP608 as compared to other multi-tap delays that provide a single set of controls and make you choose which tap has the current focus. In addition to helping cut down on problems with mistakenly working on the wrong tap, PSP608's layout makes it much easier to see what is going on quickly by examining the knobs, buttons, and sliders. Oh yeah, I should mention PSP608 sounds great, too.

Playing Around

While I've used PSP Nitro for a long time, all the other PSP EffectsPack plug-ins were new to me at the time I started preparing for this review. I generally try to test each product in the context of a real life project, to get an idea of what practical considerations might arise outside of a lab-type environment, and how a product does facilitating the job at hand, rather than simply in a job handpicked to feature its capabilities.

In this context, I had no problem finding potential uses for PSP42 -- I simply used it as a vocal delay instead of the Sonitus:fx Delay I'd been using in that role for several years. It slotted in nicely once I got used to the different meanings of the same fractions in the Sonitus:fx Delay and PSP42. (If you'd like to hear a recently completed mix with PSP42 as vocal delay, you can find it here.)

Since the projects I had on tap were relatively simple, acoustic-leaning ones, coming up with ideas for PSP84 and PSP608 was more challenging. Adding PSP84's independent controls for the left and right channel delay lines to the picture wasn't too big a stretch. Sometimes I like to have either one short delay for fattening alongside a rhythmic delay or one short rhythmic delay, such as a sixteenth note, alongside a longer rhythmic delay, such as a quarter note. However, I have to confess I was a bit overwhelmed by PSP608 at first. Besides being somewhat daunted by its interface prior to coming to grips with how it worked, I was a bit befuddled as to what I'd do with a eight-tap delay with all the sound shaping options PSP608 provided.

Somewhere along the line I came up with the idea of experimenting with a song I'd had on my back burner for a long time, which I thought would suit some creative sound design. I might normally reach for Spectrasonics' Stylus RMX for creative drum loops for a project like that. However, I decided I'd set up ToonTrack's EZ Drummer instead, using its individual outputs to allow me to process each drum mic differently. If it would work, the big advantage would be that I could use whatever rhythms I wanted, without worrying whether drum sounds could be mixed and matched, because they would be using the same (acoustic) drum set. My processing would be responsible for molding the sounds, and rhythms, beyond that point.

The name of the game was really experimentation, partly to explore what the various EffectsPack plug-ins could do to sounds and rhythms, and partly to see how this notion of processing acoustic drums to end up with a more electronic-sounding result would work. I started out trying various plug-ins on individual, soloed drum mics. Sometimes I had a preconceived idea of what I wanted. For example, I wanted to make the kick drum sound more like an electronic kick, but I didn't want to have it take up more rhythmic space by putting a delay on it. In that case, I reached for Nitro, then played around with presets until I found something that felt like what I was seeking. In other cases, I was much less certain, and sometimes went through all plug-ins, trying lots of presets, until I found something that added to the overall picture. It gave me a good excuse to see what PSP had put forth as examples of what these plug-ins could do. After awhile I made some choices that both sounded good when the whole loop was playing and held up when individual mics were soloed or muted. For example, I ended up with a 5-tap delay from PSP608 on the snare drum and a PSP 84 delay with a good deal of feedback and a filter sweep on the room mikes.

Unfortunately, I never did get very far on the song (nothing to do with the PSP plug-ins, other than to the degree that I may have spent an inordinate amount of time playing with them). However, just to give you an idea of what these plug-ins can do in this context, I thought I'd provide two MP3 file examples. The first example is a mix of the acoustic drums alone, playing a single loop for two and a half minutes, but with various drum mikes being faded in and out of the mix over time. It does have some drum bus processing on there -- PSP MixPressor for some compression, then PSP MixBass and PSP MixTreble to bring out various aspects of the sound. I would not have used the same processing had I going to be leaving the drums acoustic, but I wanted the EffectsPack plug-ins to be the only differences in my "after" example later on. Here is the "before" example:

Acoustic Drum Loop Experiment

Now here is the same experiment with each of the included drum mic outputs going through one of the EffectsPack plug-ins:

Processed Drum Loop Experiment

Of course, this doesn't even begin to scratch the surface of what the EffectsPack plug-ins can do in this context since I did not automate any of the plug-in parameters to change things up over time. Not to mention that, in a real life scenario, the actual rhythms behind a performance of this length would be varied over time, rather than just bringing drum mikes in and out of the mix.

Most of my uses of PSP Nitro in the past have been with either guitar or synthesizer sounds, where I wanted to animate the sound beyond the played part. With synth sounds, this would often be to animate a pad somewhat, without having to worry about any individual synth's unique programming interface. With guitar sounds, seeing I'm not a guitar player, this was often to take a guitar loop, or MIDI-played guitar sample, and try and give a little more life to it to help disguise that it was a sampled guitar. Thus, I was also eager to see how the various EffectsPack plug-ins might fare in a simple guitar loop scenario. I also added a drum loop to make the overall example somewhat more interesting, especially with respect to seeing how the processed parts would blend. For a drum loop, I turned to an ACID-format loop from Beta Monkey that had a nice rocking beat, which I compressed with PSP MixPressor to get a more driving feeling. For the guitar loop, I played a simple set of chord changes using Steinberg's Virtual Guitarist - Electric Edition. Here is the raw part I was using for my experiments:

Raw Guitar/Drum Experiment

I would mainly be experimenting with the guitar part, but I thought it would also be fun to process the drums slightly, if I could find something that helped in the context of the passage's mix. I used a basic PSP 608 dual tape delay preset that seemed to work quite well in context, keeping that setting constant through the rest of the experiments.

My goal for the guitars was largely to get settings on each plug-in that were in the same general ballpark on basic feel, but which added in some of the unique capabilities of each processor as I moved around the plug-in line.

Starting with PSP42, I set up a basic sixteenth note delay. Here is the PSP42 version of the passage, including the processed drums:

Guitar/Drum Experiment (Lexicon PSP 42)

Exchanging PSP42 for PSP84, I added a second tap with a delay time of 3/16 for a more complex feel, and also added spring reverb to the processing of the wet effect. Here is the result:

Guitar/Drum Experiment (PSP 84)

Next I replaced PSP84 with PSP608. Keeping to the same basic feel, with the delay times of 1/16 and 3/16, I added a third tap on the quarter note (4/16 in PSP 608 parlance). I also played around with the filters. I'd been using a 4 kHz low-pass (or high-cut) filter from PSP42's Hi-Cut switch, and had manually set up a 4 kHz low-pass filter on PSP84. On PSP608, I kept that sort of setup on the first tap, but then put a 700 Hz high-pass filter on the second tap (i.e. the one at 3/16) and a band-pass filter at 164 Hz on the third tap. I also experimented with different width and balance settings for each of the three taps, among other tweaks. Here is the result:

Guitar/Drum Experiment (PSP 608 MultiDelay)

While Nitro is a different animal than the delays, I wanted to keep the notion of the sixteenth note delay feel going so as not to get too far afield from the other examples. Rather than setting delay parameters manually, as I'd been doing with the other three plug-ins on the guitar part, I simply hunted for the first preset I found that gave me a sixteenth note feel. (It was called "Ancohuma".) It also happened to have some spacey-sounding self-oscillation going, which gave it an interesting texture during playback. Here is the Nitro version of the result:

Guitar/Drum Experiment (PSP Nitro)

The more I experimented with these plug-ins, especially for processing this simple, loop-based guitar part, the more fun I had. I began to appreciate what the different processing of each tap could do to shape the overall sound. At some point the dry loops started sounding really boring to me in comparison to the processed parts, and I found myself eager to have the opportunity to try out these new toys on real projects in the future.

Complaints? I have a few minor ones.

I could not get the MIDI learn facility of these plug-ins to recognize MIDI CC information from my keyboard under SONAR 6.2.1. I was able to use SONAR 6's ACT Learn facility to map PSP42 controls to my keyboard's continuous controllers, and ACT may be preferable to individual plug-in mapping facilities for SONAR 6 users anyway. Thus, this issue may be more of a concern to users of SONAR 5 and earlier SONAR versions, assuming it is not simple user error on my part.

I also did experience a few SONAR 6.2.1 crashes while using these plug-ins, at least two of which would seem to be related to PSP608, independent of whether the fault actually lies with the plug-in. In one case, SONAR literally complained that PSP608 had crashed, and, in another case, I was moving a PSP608 control when SONAR crashed. I also experienced a few other crashes or "poofs" (i.e. where SONAR disappears but no crash message is presented) while switching projects where both projects had been using one or more EffectsPack plug-ins. For most of the time I was using these plug-ins, though, I did not experience any instability. It is rare for me to switch projects frequently or have multiple projects open at one time, as I did when most of these specific crashes occurred. Thus, perhaps there is something special about having multiple projects open at a time that came into play here.

Finally, though the version history files for three of the four of the latest version of these plug-ins (all except PSP608) indicate the VST versions of the plug-ins should support a 64-bit host-to-plug-in interface, SONAR is only recognizing the host-to-plug-in interface as being 32-bit. PSP indicated the version history file was incorrect. They further indicated that a 64-bit host-to-plug-in interface would not be very useful for the EffectsPack plug-ins since

the internal processing in all the plug-ins is 32-bit float. However, users of SONAR 5 and onwards may still find this to be of concern, especially if they are considering using one of the EffectsPack plug-ins instead of one of the Sonitus:fx plug-ins, all of which feature 64-bit interfaces and internal processing, and which are included in SONAR Producer Edition.

Closing Notes

While the PSP EffectsPack plug-ins may well cover different territory than PSP's historical stronghold of mixing and mastering processors, they uphold PSP's reputation for quality products. When I first encountered PSP Nitro awhile back, I would have never predicted how much I'd end up using it over time. Nevertheless, it has found its way into quite a number of my projects due to it's wide range of filter types, the notion that it can be applied to any type of sound, be it synthetic or acoustic, and its sound quality. I expect the remainder of the EffectsPacks plug-ins will find their way into many of my future projects for similar reasons -- even PSP608, which I had no clue how I'd end up using when I started putting this review together.

Anyone who enjoys sound design, or who simply wants to liven up stock sounds and loops, would do well to check these plug-ins out. You may not technically need all the flexibility plug-ins like PSP Nitro and PSP 608 MultiDelay provide, but that doesn't mean they won't come in useful. Just beware that you could find yourself spending countless hours playing with filter choices, creative delay tap processing, and so on. It's hard to stop once you get started.

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