



Review: Native Instruments Absynth 2 by Rick Paul - 31st December 2003 -



Anyone who has been around softsynths for any amount of time is likely to have encountered Berlin-based [Native Instruments](#). My first exposure to NI was with their excellent B4, which emulated a Hammond B3-type organ played through a Leslie-type cabinet. It has since gone on to emulate a few other vintage organs, via sets of replacement "tone wheels", but the gist of the product was, and is, extremely accurate emulation of a vintage instrument. B4 broke relatively new ground for software instruments when it came out due to the combination of the accuracy of its emulation and the resulting notion that a MIDI keyboard plus a computer and software just might be a reasonable substitute for something that had never quite been truly satisfactorily modeled in the MIDI hardware world.

NI continued the vintage hardware emulation tradition with their Pro-53 (originally Pro-52, which models a Sequential Circuits Prophet 5 analog synth) and FM7 (models the Yamaha DX7 FM synth line). In all these cases, NI went beyond cloning the originals to make improvements in areas where that was facilitated by a software- and computer-based architecture. For example, FM7 provides direct graphic envelope manipulation that never could have existed on the DX7 hardware. Still, that didn't change the basic nature of the products, which was to be new, improved versions of the hardware products they emulated.

Besides the vintage keyboard emulators, but still sticking with the general area of softsynths, NI has made a name for itself in samplers and sample players (e.g. Battery, KONTAKT, KOMPAKT, INTAKT) and with a modular tool kit for building softsynths and audio effects (REAKTOR). All of these areas have been covered, to varying degrees, by other software developers, though NI's products tend to be at or near the top of their respective classes. Most of these areas have also been covered, again to varying degrees, in hardware keyboards and/or sound modules, even if a software-based architecture may allow improvements beyond what can generally be found in hardware equivalents.

Somewhere amidst all this, along came Absynth, which has more recently been updated to [Absynth 2](#). Unlike the rest of NI's synth and sampler line, Absynth has no direct precedent in hardware. While various components of Absynth will be familiar to those with experience with other hardware and software synths and samplers, those components come together in a manner that would not have been practical to implement in the hardware world. Perhaps more significantly, though, the result is compelling enough that it might even convince a few "hardware only" types to take notice.

Basic Ingredients

So, what exactly is Absynth? From a technical perspective, it combines elements of synthesizers and samplers in a semi-modular hybrid instrument. Is that too vague? How about if we add that sound generation capabilities include subtractive, wavetable, frequency modulation (FM), amplitude modulation (AM), ring modulation, and wave shaping synthesis, as well as traditional and granular sampling. There are also 14 different types of filters, wave-based modulators, and delay-based effects. That narrows the field a bit, but may sound a little too much like a smorgasbord of softsynth/softsampler technology. Not that this comparison would be entirely inappropriate, mind you, but we're getting a little ahead of ourselves, and verging on talking about sound design in Absynth before having described what it is and what it does.

Therein lies the challenge, though. It's easy to describe a softsynth that emulates a known entity by saying it emulates a certain classic organ, analog synthesizer, FM synthesizer, drum box, or whatever. With Absynth, the raw technologies it brings to the table aren't really what is new. They overlap much of what is already out there from NI and other sources. What is new with Absynth is more about flexibility and capability in the area of how the ingredients get combined, and the potential this provides for creating and playing unique sounds. Yet the sounds that can be created with Absynth range from stock stuff you've heard from numerous different kinds of synths to sounds and patterns you've probably never heard before. Thus, it isn't enough to say it creates "phat" sounds or "piercing" sounds or "evolving textures" or "loop-like" sounds or ... (name your favorite kind of sound). Sure, it can do all of the above, but describing Absynth's "characteristic sounds", if there were such a thing, would be getting back to that smorgasbord description again, and I'd hate to be redundant.

Thus, I'm going to take a coward's way out, at least for the moment, and procrastinate on trying to describe what Absynth is and does, just leaving it at the notion that it is about providing new kinds of sound design possibilities, and deferring to the sections below that talk about sound design in Absynth to fill in missing details. What I will say now is Absynth is about power and flexibility, both at sound design time and when it comes time to actually play Absynth. I will also drop a hint that Absynth's powerful ways of creating and using envelopes are a key differentiator.

In the meantime, let's talk about what's in the box and on the software CD-ROM. First and foremost is the software CD, which contains the standalone version of Absynth for Windows XP (supports ASIO 2.0, DirectSound, and MME interfaces), Mac OS 9 (supports ASIO 2.0 and SoundManager interfaces) and Mac OS X (supports the CoreAudio interface). In addition to the standalone softsynth, there are plug-in versions including DXi2 (Windows only), VST 2.0 (Windows and Mac), and AudioUnits (Mac OS X only). RTAS support (Windows XP and Mac OS X only) is planned, but not yet available as of this writing.

The CD also contains 13 banks of patches -- on the order of 800 patches in total -- with names such as "Atmospherics & Weird", "Drums", "Percussion", "Strings & Pads", and "Synth, Bass & Acoustic". Sounds range from somewhat emulative sounds, to truly unusual evolving textures and loops. If you're looking for the equivalent of a General MIDI module, though, you won't find it in Absynth. Patch quality is extremely good. The usefulness of those patches will depend on your musical style, and possibly also on your willingness to tweak.

All this and a paper manual (see below for more on that) are included for the list price of \$299 (USD). As of this writing, low-end street price in the USA was just under \$220, with more typical street pricing in the \$230 to \$250 range.

I might note at this point that, while many of the comments in this article will be applicable to multiple versions of Absynth 2, and various DAW platforms, since this is CakewalkNet, the actual hands-on review has been made primarily using the DXi2 plug-in under SONAR 3.0 Producer Edition. I have also done some minimal testing with Cakewalk's Project5. The review process was begun with Absynth 2.0.1, but Absynth 2.0.4 was used for most of the review, and issues noted earlier in the process were checked against version 2.0.4 wherever practical.

Documentation, Installation, and Software Protection

The first thing I noticed upon opening the Absynth 2 package was a decently thick (98 pages of content plus table of contents and cover-type pages) manual, with an attractive, abstract design that looks three dimensional from certain angles. Knowing ahead of time that Absynth was a reasonably complex product, my first thought was the manual might actually be way too thin, because my past experience with NI products was the thickness of the manual was often misleading due to its containing basically identical content in four languages (i.e. English, German, French, and Spanish). Thus, I was pleasantly surprised to find that all 98 pages of the Absynth 2 manual were in a single language (thankfully, English).

The first roughly one third of the manual is dedicated to installation, software registration, and getting started in various versions of Absynth (i.e. standalone on Mac and Windows and the various plug-in formats). If you're new to softsynths and/or haven't already satisfied yourself on the prerequisites for running Absynth, that part may be worth reading. However, the meat of the manual is the roughly two thirds of it that follows, detailing Absynth's patch architecture and user interface. The manual is much more a reference manual than a tutorial or other "getting started" guide. However, it is relatively easy reading, and I heartily recommend having a read through it to familiarize yourself with Absynth's interface and patch architecture as there is no on-line help, and Absynth definitely has some little user interface quirks. Those "quirks" can be very productive once you understand them, but might well leave you befuddled on how to do some basic things, such as turning a sound generation channel on or off, if all you have to go on is Windows (or Mac) conventions and past experience with other softsynths.

My one key criticism of the manual is that there are sections of it where it seems clear that the manual was initially written for Absynth 1. Rather than integrating the Absynth 2 additions into the content, NI chose to just add those sections on in the relevant areas. Thus, for example, you get discussion of the subset of features of a particular portion of the interface that were applicable to Absynth 1, with a few pages immediately following that go into the new features of that same portion of the interface from Absynth 2. It can make for slightly disjointed reading, and even more disjointed referencing, especially given the lack of an index in the manual or context-sensitive help in the software itself. Also, some of the actual screen shots in the manual are not consistent with the software, in all likelihood because they were taken from Absynth 1 and were not updated when Absynth 2 added fields to those screens.

Installing the software (on Windows XP) was straightforward, with no problems whatsoever. I installed the standalone interface, as well as DirectX and VST plug-ins.

After installation, the software needs to be registered and authorized to allow for more than 30 days of operation. In practice, the registration procedure only took me a few minutes via the Internet, and then another half an hour or so of waiting for the authorization code to arrive, during which time it was possible to use Absynth since NI provides 30 days of operation before authorization. NI recently announced they have changed their authorization method to provide for immediate authorizations via the Internet, so that should speed things up even further.

For someone without an Internet-connected computer, there are other authorization methods, though those will be notably slower, and the 30-day period may end up being more necessary in such cases. While I know some people are almost religious about not connecting their music computers to the Internet, enough of the software I use either requires Internet connection for authorization, or just has built in conveniences of one kind or another that save time for users whose systems are connected to the net, that I can't imagine having a disconnected computer, at least if that computer is modern enough and powerful enough to run state-of-the-art softsynths like Absynth.

The hardware-dependent registration and authorization methods used in recent NI products, and products by other companies who use similar methods, do concern some users, both in terms of the inconvenience of reauthorizing in the event of system changes, be they intentional or due to hardware failures, and due to the concern for what might happen down the line if the vendor goes out of business. NI addresses the first issue by providing for two authorizations, with the ability to deauthorize those authorizations permanently if one or both of the system goes away, thus reclaiming the deauthorized authorization for another system. This should do the trick in all but cases where someone is frequently switching back and forth between three or more systems. As for what would happen

were NI to go out of business, and a new authorization were needed after that point, I don't think any vendors who use hardware based authorizations have solved that issue, and about the best we can do is hope the quality and innovation in NI's products helps keep them around for the long haul.

Besides the authorization requirement to use the software beyond 30 days, registration is also important for another reason, and that is to have access to updated versions of the software. When I initially installed the software, and updated to the latest patch available at the time (2.0.1), Absynth DXi was fairly unstable in SONAR when using multiple instances. Version 2.0.4 of Absynth came out a week or so into my review process, and stability has been greatly improved.

Getting Started



While I believe anyone who just picks up Absynth 2 to play off-the-shelf sounds will likely be missing the point, I would be remiss not to mention that it is certainly possible to do that, and it couldn't be simpler. Simply bring up the relevant Absynth plug-in in your favorite DAW, or play the standalone version directly from your MIDI keyboard and computer, and you're on your way with a library of sounds that would rival the number of sounds provided in all but the most patch-packed hardware MIDI modules. Most of the included sounds are of a fairly electronic-sounding nature, and many of those have evolving textures and/or rhythms, often synchronized with the softsynth host's tempo. If your tastes run toward the electronic, it would certainly be possible to produce tracks for an entire song using only Absynth 2 for sounds, though it is worth pointing out that Absynth is not multi-timbral, so you will need as many instances of the Absynth plug-in, up to a maximum of 16 as of version 2.0.4, as you have parts to assign in your sequencer. For this reason, Absynth would not be an ideal softsynth for standalone use with a MIDI keyboard outputting on multiple MIDI channels.

Absynth's main plug-in window, which will be the only window visible within your DAW, makes it very easy to browse banks of patches. Absynth can also respond to program changes (i.e. within a bank -- it does not respond to bank changes as banks must be loaded from files) from a MIDI sequencer or keyboard to make it easy to just play around with Absynth. Beware if you do just play it from a keyboard, though, because you may find yourself eating up large amounts of time just going through the various presets from the provided sound library. In fact, some of the patches are downright inspirational. Even if you're not typically a composer of electronic music, you may find some of the pattern-based patches encourage you to experiment with new compositional directions. Or at least that is the effect they had on me (and anyone who has checked out my music on the web knows that I primarily write country and pop/rock, both of which are about as far as you can get from the atmospheric, textural electronic music for which Absynth is particularly well-suited).

Besides bank and patch access, the main Absynth window provides access to polyphony settings, a volume fader, panning, a percentage of BPM (i.e. in the plug-in -- just BPM in the standalone version), transposition, and tuning/scales. Absynth can be relatively CPU hungry, so it is a good idea to keep polyphony as low as you can for the part you want to play. Many of the preset Absynth patches keep polyphony pretty low, often at 4 voices or less, and this makes having easy access to the polyphony count quite convenient. The percent of BPM control governs how any elements of Absynth's patch which are synchronized are matched up to the DAW's tempo, allowing for patterns which are fractions or multiples of the actual project tempo. Transposition is set in semi-tones and 1/1000ths of semi-tones (1/10ths of cents). The tuning or scales control provides for alternate tunings. Absynth comes with a pretty respectable collection of factory tunings, while also allowing the creation of user scales. Editing of factory tunings is not allowed in the current version of Absynth. That's probably just as well, as it could invite messing things up pretty badly if you didn't know what you were doing. However, it would have been nice to have a "clone factory tuning" capability to allow using a factory tuning as a starting point for custom scale creation.

The main Absynth window also shows as many MIDI controller sliders as are used in the individual patch. Thus, for example, a basic patch that uses only mod wheel, volume, and pan may only have three sliders while a patch configured for more types of continuous control could have considerably more. In practice, I found the dynamic

resizing of the main window to accommodate more or less controllers worked with in the VST version of the plug-in, but did not work in the DXi version. The DXi version fixed the number of controller slots at 6, independent of how many controllers were configured for use in the loaded patch. In practice, this may not be a serious limitation in that not a whole lot of patches actually configure more than six controllers. However, in cases where a patch does use extra controllers, this may favor use of the VST plug-in, which works in SONAR via the Cakewalk VST Adapter, despite DXi2's being the native softsynth format for SONAR.

Making Sounds



Playing sounds from Absynth 2's included sound library is a nice way to get a basic feel for what Absynth can do, and may keep you busy for awhile due to the sheer volume of sounds. Stopping there, though, would be somewhat akin to going to an authentic Chinese restaurant with a 15 page menu and just ordering the lunch buffet. Anyone who has gone the buffet route, but later gone back to explore the nooks and crannies of the longer menu, quickly realizes that the buffet tends to present a selection of dishes that appeals to a wide variety of tastes, but satisfying your own personal tastes is often better accomplished by ordering from the wider menu, or even requesting special touches when ordering ("make mine the 4 star curry, but please leave out the MSG"). Similarly, someone who just plays the included sounds with Absynth will only be experiencing a fraction of the overall user interface and capabilities of the product.

When you do decide to take Absynth farther, and hit one of the editing buttons in the main plug-in window, such as "Patch" or "Waveform", the first thing you will notice is that Absynth brings up a totally separate window -- i.e. one that is totally separate from SONAR or whichever other DAW you may be running -- to do the editing. This patch editing interface will also have separate sub-windows within it, each tailored for editing a specific aspect of an Absynth patch. If lots of overlapping sub-windows bother you, Absynth provides a setting to close other editing windows if another editing window is opened, but I tend to like being able to refer to multiple windows in parallel while working on a patch, and the ability to do that is something not many softsynth sound-editing interfaces provide. If you do like having the multiple editing windows open at once, but occasionally get into "messy desktop syndrome" by opening up too many windows at once, Absynth provides a convenient Navigation Bar, which can be oriented vertically or horizontally, and which can be used to bring any desired editing window to the foreground. Since the Navigation Bar stays on top, it never gets lost.



If you're creating an Absynth patch from scratch, or are looking to make broad changes to the nature of a patch's sound, you will likely encounter the Patch window. The patch window lets you see the basic architecture of Absynth patches and the signal flow of the patch being edited. The top portion of the Patch window displays three parallel sound generation channels, each comprised of an "Oscil" (no doubt short for "oscillator", though, depending on the sound generation method chosen, it may not technically be an oscillator) and, optionally, a Filter and/or Mod (assumedly short for "modulator") module. Individual modules within a channel are toggled on or off by clicking on the name of the module (or anywhere on the "plumbing" that forms its border, for that matter -- this is one of those user interface "quirks" I mentioned above that is very efficient once you realize how it works, but which was definitely not intuitively obvious prior to having read the manual). Toggling an Oscil module off toggles the entire sound generation channel off (i.e. since the channel can't make sound without some synthesis or sampling module's presence), but Filter and Mod modules can be toggled on and off independently, and the "plumbing" that indicates signal flow within the channel and overall Patch window is visibly rerouted to reflect the changes.

Below the three sound generation channels is a master channel comprised of "Waveshape", "Filter", and "Effect" modules, each of which can be toggled on or off independently. Signal flow is from the mix of the three sound

generation channels through the Waveshape then Filter then Effect modules in sequence, bypassing any modules that are disabled. Again, the graphic "plumbing" reflects the signal's bypassing any bypassed modules. We will get to the details of each type of module below.

Sound Generation Channels

Each of the up to three sound generation channels in Absynth 2 has up to three modules: an Oscil, a Filter, and a Mod. Let's look at what each of these does.

The Oscil module defines the type of synthesis or sampling used in each sound generation channel, and specifies any parameters needed to control and shape the sound generated. The user first specifies a mode of the Oscil, selecting from "single" (single oscillator subtractive or wavetable synthesis), "double" (dual oscillator subtractive or wavetable synthesis), "fm" (frequency modulation synthesis), "ringmod" (ring modulation and amplitude modulation synthesis), "sample" (i.e. traditional sampling), or "granular" (i.e. granular sampling).



For synthesis methods, the user will also pick a single cycle wave shape from the factory-defined library (factory waveforms include such classics as sine, triangle, saw, as well as more complex wave shapes such as one based on a female vocal or organ -- there's even a built in wave shape called "broken speaker", but I think I'll avoid that one for the time being!), or may choose to create his or her own wave shape, using the Waveform window. The Waveform window provides two editing interfaces, as well as a number of waveform

transform operations, for manipulating waveforms. One of the interfaces is a Waveform mode, which shows a single cycle wave graph, editable in the time domain, with the default's being a sine wave. The other is a Spectrum mode, which shows the first 64 harmonics of the waveform, editable in the frequency domain. (Harmonics above the 64th harmonic are preserved during Spectrum mode editing.) The waveform transform operations include, for example, the ability to load and save waveforms, apply DC offset,



invert phase, reverse the waveform, mix two waveforms together, and perform a number of basic to advanced processing operations. One particularly interesting one is "fractalize", which can take an otherwise bland waveform, such as a sine wave, and, while retaining the same basic overall waveshape at the very high level, add significant character. Visually, in the Waveform window, this looks like the wave has just become very jagged, while still retaining the basic overall curve flow -- think craggy mountain peaks versus rolling hills. For dual oscillator analog-style synthesis you make waveform choices twice, once for each of the two oscillators. For FM synthesis, you also make the choice twice, once for the carrier and once for the modulator.

Other settings available in the Oscil module for synthesis methods include how pitch of the sound source is determined (e.g. transposing based on the MIDI note played, a frequency ratio based on the MIDI note played, a fixed pitch specified in Hz, or fixed pitch specified by MIDI note number), whether or not phase is inverted, and any phase offset. For the single and double modes, there is also an Antialiasing button, which determines whether the oscillator channel is antialiased to achieve a smoother, more analog-like sound.

For sampling methods, instead of picking or creating a waveform, the user will specify a WAV or AIFF sample file. Pitch determination is set as noted for synthesis methods. Instead of a phase offset, there is a start time field to specify where in the sample file the sound is to start (specified as a percentage of the total length of the sample file). There is also a mono or stereo button, which determines if stereo samples are played back in mono or stereo.

For traditional sampling, there is an additional parameter to determine whether the sample is looped or treated as a one-shot. If it is to be looped, the user can choose between looping the entire sample or just a portion of it. In the latter case, two additional parameters will appear to determine the loop start and end points within the sample.

Granular sampling is a bit more complex by nature. Instead of playing a sample linearly, looping or not, granular sampling divides the sample up into smaller "grains" to be played back separately. This allows messing with the playback of each grain, for example, adjusting time and pitch independently. As the size of the grains get smaller, the sound of the sample playback gets further and further from the character of the original sampled sound.

Parameters available for tweaking the sound in granular mode include playback speed, the number of grains to be played back simultaneously, grain size in samples, and randomness factors for how often grains play back, pitch of the individual grains, and amplitude (volume) of the individual grains.

Once the sound in an individual channel is created, it can optionally be processed by a Filter module and/or a Mod module. If engaged, the Filter module is next in line after the Oscil. There are 14 different filter types, some which are slight variations of other types. I won't bore you with a list of all the individual modes, but suffice it to say that the basic food groups are covered, including lowpass, highpass, bandpass, notch, comb, and, newly in Absynth 2, 2-, 4-, and 8-pole lowpass and allpass filters. Those last two groups deserve a bit of elaboration. The 2/4/8-pole low pass filters are modeled after analog filters, for a smoother response across a wide frequency range, even with high resonance. With the allpass filters, the number of poles indicates the number of notches and peaks in the filtered signal's frequency spectrum. The difference between these newer filters and the closest Absynth 1 equivalents is somewhat akin to the differences between EQ models in the Waves Renaissance Equalizer and Waves Q10 Equalizer. That is to say, they do roughly the same job, but with at least subtly different characteristics.

Mod modules modulate the signal emanating from the Oscil and (optionally) Filter modules, with the possibility of mixing the input signal with the modulated signal based on setting a balance control. The Mod modules have settings very much like the ringmod mode of the Oscil modules, and their behavior is similar in modulating the signal.

Any of the three sound generation channels in the patch window that are active are mixed together prior to going to the master section modules. If the master section is set for stereo operation, the individual sound generation channels can also be panned separately prior to going to the master section. Absynth can automatically balance the volume levels of the three sound generation tables in order to preserve the overall volume level of the patch. As you slide the volume level higher on one channel, Absynth scales the levels for the other two channels down proportionately. For those who want more control over this, this automatic balancing can also be turned off, but I find it to be a useful capability for helping keep levels roughly in check while tweaking the balances that contribute to an overall sound.

The Master Channel

Still within the Patch window, but after mixing the three sound generation channels together, comes Absynth's master channel. This portion of the signal flow consists of three optional modules. In order of their placement in the signal flow, they are the Waveshape module, the Filter module, and the Effects module. There is also a mono/stereo button which controls whether the master channel modules are all in stereo or not, and thus dictates whether the individual sound generation channels can be panned.

The Waveshape module comes first. Its function is to distort the input signal according to a specified waveform. The waveform selection works similarly to in the Oscil modules, including the possibility of specifying a user-drawn waveform via the Waveform window. A mono/poly button determines whether the distortion is applied to each individual voice (i.e. of Absynth's overall polyphony), or whether playing more than one note will result in voices' interfering with each other. The Absynth manual compares the former, which is called "poly" mode, to having a separate guitar amplifier for each string of an electric guitar, whereas the latter, which is called "mono" mode, would be like the traditional model of sending the output of all strings of that guitar to a single amplifier. Besides the waveform and mono/poly mode controls, there are settings for input and output level adjustments and for offsetting the phase of the waveform.

Next up is the Filter module. Except that it operates on the mixed sound, and after the Waveshape module if engaged, the Filter module here is mostly just like the Filter modules in the sound generation channels, including

providing all filter types provided by those modules. The key exception is that it also features a mono/poly button that operates as noted above for the Waveshape module, but only if the Waveshape module is set in poly mode since the Filter module comes after the output of the Waveshape module.



Last up is the Effects module. This is the only one of Absynth's Patch window modules whose basic settings are not tweaked in the Patch window itself. Rather, the module, if enabled, contains solely the word "Edit", and clicking on that brings up the Effects window for specifying delay-based effects settings.

The Effect window provides three effects modes. "Multicomb" mode provides up to 6 independent delay lines with relatively short (i.e. less than three quarters of a second) delay times. "Multitap" mode provides a single delay line with three taps and up to 10 seconds of delay time. The Absynth manual suggests thinking of "pipe" mode as being similar to a string, where there is a speaker connected to it (the input position) to make it vibrate, and two pickups (i.e. as on an electric guitar -- these are the left and right output positions in pipe mode) to sense the sounds produced, and all three of these elements can have their positions be in motion, with that motion being controlled by an LFO. The motion produced via automating the positions of pipe mode elements via an LFO can produce an effect similar to that of a rotating speaker. Delay times in the multicomb and multitap modes can also be modulated by LFOs. Otherwise, those effects are fairly traditional delay-based effects.

It's All About Control...

As flexible as being able to combine several types of synthesis and sampling and create user-defined waveforms may be, what we've covered thus far of Absynth's capabilities may still not sound all that exciting. Perhaps it provides a wider palette of choices than many, or even most, of the softsynths out there, but it feels more like a logical extension of what is there than something truly new and different. So, just what is the big deal?

In a nutshell, the key to Absynth's uniqueness is how far and wide it goes in providing dynamic control. "Dynamic control over what?" you may ask. The short answer is, "just about everything that matters." Let's delve a little more deeply, though.



We've already touched briefly on Absynth's LFOs (Low Frequency Oscillators) above in discussing the pipe mode of Absynth's Effects module. Most of us are familiar with using LFOs to inject periodic time-based variations in pitch into a sound, for example to simulate vibrato, and the mention of using LFOs to manipulate effects output positioning in the stereo field may already have provided a hint that Absynth's use of LFOs go beyond what is typical. To quote a famous Carpenters' song, though, "we've only just begun."

If you've been paying attention thus far, you might already be guessing that each of Absynth's up to 3 LFOs can have it's own waveform, which can be picked from factory settings or defined by the user using the Waveform window. Each LFO can modulate multiple patch parameters at the same time. The effects of all three LFOs can be combined for manipulating individual parameters, too, thus providing even more complex modulation possibilities than even the user-defined waveforms would provide. In addition to the stock continuous waves that most LFOs provide, a sample and hold mode is also available. This mode can be thought of as a digital sampling mode, which takes a picture of the wave every so often and plays the value of the wave at that point up until the next point a sample is to be taken, after which it plays the new sample until the next time, and so on. Thus, if the LFO is modulating pitch, instead of continuously variable pitch, you get what sounds a bit like an arpeggiator hitting different "notes" depending on where along the LFO waveform the pitch lands at the point when a sample is taken.

LFO timing can be set in absolute time or in multiples (or fractions) of beats. Ditto for sample and hold timing, and these settings are independent of one another. A phase setting indicates the initial phase of the waveform if the LFO

is set to poly (versus mono) mode. (If the LFO is in mono mode the LFO runs continuously, so there is no practical effect unless the LFO is retriggered, for example, by a MIDI continuous controller.) Each LFO can modify Oscil pitch, Oscil amplitude, FM index/balance, Filter frequency, Effect time, and Effect panning. Where relevant these modifications can be selected for each individual sound generation channel's modules and/or the relevant master module. For example, for Filter frequency, it would be possible to have LFO 1 modify the frequency of Filter 1 and Filter 2, have LFO 2 modify the frequency of Filter 3, and have LFO 3 modify the frequency of Filter 3 and the main Filter module, thus creating complex modulations for Filter 3's frequency. Some LFO parameters can themselves be modified by MIDI continuous controllers. These include master LFO depth, LFO rate, and sample and hold rate. This could allow "playing the LFOs" in real time, or under the control of a sequencer. These are not your father's LFOs.



The capabilities of Absynth's LFOs are relatively simplistic compared to the capabilities of Absynth's envelopes, though. Where LFOs can manipulate just a handful of parameters, envelopes can manipulate pretty much any parameter that has a numeric value in the Patch window. Where LFO shapes are governed by a preset or user-drawn waveform, envelope shapes can be manipulated by setting up to 68 breakpoints (plotted in time and amplitude) and their slopes with numerical accuracy (or plotted by hand), and can also be synchronized with a grid of beat divisions. Each envelope can also have its own integrated LFO associated with it, and these LFOs can also have factory or user-drawn waveforms. These integrated LFOs can also be of the sample and hold form described above for the patch level LFOs. The whole envelope has a single LFO, but the depth and speed, and sample and hold speed if applicable, can be varied per breakpoint. The timing and amplitude of each envelope can also be assigned to separate (or the same) MIDI controllers, thus allowing you to "play the envelope" in real time. Of course, you can also just manipulate the envelopes on screen while sounds are playing through Absynth to hear the effects of the envelope changes in real time, too. (Note, though, that being able to record these movements into your sequencer will depend on your sequencer's capabilities, and I have not succeeded in figuring out a way to record Absynth user interface movements in SONAR.) Have you gotten the idea these capabilities run deep yet?

If all this envelope stuff is starting to sound like just so much technospeak, perhaps a quick example would be in order. Let's say I have a very simple patch that will be used to create a bass line. Not just any bass line, but one of those pulsating disco-type bass lines that has alternate octaves playing on the half beat, and perhaps has a filter opening up and closing over the course of a few beats or a measure. And perhaps there is a slide up to the higher octave of the note from the lower octave of the note each time a new note is struck. Oh yeah, and we want the pitch to not be too steady, but have some analog-style drift, which can be modified by an LFO with only a bit of pitch depth effect. Are you thinking about what might be involved in creating that in your favorite synth? I hope you won't mind if I don't think too hard on it as I'm not in the mood for a brainteaser at the moment.

Let's consider what it would take to do this with Absynth, though. Would you believe all of this can be programmed easily in Absynth using just two envelopes? The first will govern the pitch of the Oscil being used (you may use more than one such envelope if the patch will have multiple Oscils). Simply assign an envelope to the relevant Oscil's pitch, creating nodes at the right spacing for your beat divisions, and making the slope such that you get the right timing of glide between notes (or, alternately, use "step" spacing if you'd prefer not to have portamento between notes), and use an integrated LFO to get the sort of warble you like, using a low pitch depth for the LFO. The initial short segment of the LFO can also give a slide up to the first "note" (keep in mind you are only playing a single note and the Absynth envelopes are making the octave shifts for you for additional "notes"). The second envelope will be used to tie to a filter, either a master filter or one for a specific sound generation channel, depending on preference for the sound. For my quick test, I set a master channel Filter module to act as a low pass filter with a high cutoff frequency, and used an envelope to start it out low and raise it, then lower it back to the original amount at the end of a measure. This had the effect of having the sound start out kind of dull, brighten, then get duller again over the course of a measure.

Okay, so this is a pretty simplistic example, and it might be easier to just play the octaves while using an LFO to vary the pitch, and just use the filter automation via an envelope or tempo-synchronized filter plug-in, in this particular case. However, I'm hoping this simple example will give a better sense for the capabilities Absynth has for

manipulating parameters over time, whether synchronized with the beat or not, and how much power there is for making that manipulation itself dynamic (i.e. with the LFO's modifying the envelope). We're also only touching a few actual Absynth parameters in this example, and the example isn't even bringing, say, mod wheel or expression pedal control into the picture to add the playability aspect to the picture.

Envelopes can also be MIDI control driven, where, instead of evolving over time, their position on the envelope changes with the value of a MIDI controller. They can be linked to other envelopes, but with time, amplitude, and slope scaled from the other envelope, and amplitude offset from it.

There is much more, but time and space only allows for so much description, and I'm guessing your eyes are starting to glaze over by now anyway. This is stuff that begs to be played with, not described! I do hope, though, that you are getting an idea of the power and uniqueness of Absynth as it is these sophisticated control capabilities, especially the envelopes, that set Absynth apart from most of what else is out there when it comes to creating complex, evolving sounds.

... And More Control



Absynth 2's MIDI window is used to configure how an Absynth patch responds to what it is told to do, be it by a MIDI sequencer or an honest too goodness human being playing a MIDI keyboard or alternate MIDI controller. Another way to put it might be that this is the part of Absynth used to fine tune the response of the "instrument" represented by the particular Absynth patch. Areas covered include which MIDI controllers are used to control various types of sound parameters and how those parameters respond, how MIDI velocity affects certain types of patch parameters, how certain types of patch parameters respond to which note is played, and the tuning of the "instrument".

The MIDI window's MIDI Controllers tab allows configuring pitch bend range and "lag" (roughly equivalent to its response curve, though there is no visible curve, just a numerical parameter), glide time between notes and whether or not a "legato mode" (i.e. staccato playing doesn't result in gliding between notes, but legato playing does) is active, and to select which MIDI controllers are used to control various patch parameters and how those parameters respond to controller movements. Defaults for which controller to use tend to be obvious ones, such as having patch volume and pan respond to MIDI volume (i.e. CC7) and pan (i.e. CC10), respectively, but there's nothing saying you need to use those settings. Specific parameters available for modification include FM index/balance for any or all of the Oscils, Filter frequency for any or all of the individual channels and/or the master Filter, Effect time, pan, and volume. All of these except volume have associated depth and lag parameters -- volume only has a lag parameter. This makes sense in that a volume value would typically be some kind of fader-like controller where there is an expectation that one end is the loudest a sound can be and the other end is silence, whereas the other values would be scaling the effects of parameters, and it would be desirable to scale them based on the basic parameter value configured into the patch.

Perhaps the key limitation in this area is that, for FM index/balance and Filter frequency, both of which can be set for multiple sound generation channels, and also for the master channel in the case of Filter frequency, it is only possible to tie any parameters of the specific type you want to tie to a controller to the same controller or no controller. Thus, for example, if you want to tie the Filter frequency of the master channel and that of Filter 1 to a controller, and choose the mod wheel (i.e. CC1) as the controller for the master channel, then you can only use the mod wheel for Filter 1, too. Also, this same limitation extends to setting depth and lag of these parameters, and that could be more likely to be a practical issue in that you might want the range or curve of responses to a controller to be different for different sound generation channels. Mind you, Absynth has other ways of accomplishing more finely tuned MIDI control capabilities. This specific situation could be addressed using special types of envelopes called "control driven" envelopes for each parameter. With control driven envelopes, the envelope's horizontal dimension represents the full range of the continuous controller, and its vertical position represents the effect of the controller on the parameter being modified at that point. The control driven envelope facility provides much more

power than what is available in this area of the MIDI Controllers tab, but the MIDI Controllers tab is also much simpler to set up if what it does will suit your needs.

The MIDI window's Velocity tab is fairly self evident, and governs how individual Oscil amplitude and FM index/balance parameters, as well as individual Oscil and master channel Filter frequency parameters, respond to MIDI velocity. Since there are separate numerical fields for each individual parameter for each relevant module, the above noted limitation for controller settings is not an issue in controlling velocity response.

The Note Scaling tab allows modifying the effect of various patch parameters based on what MIDI note is played. Parameters that can be controlled are the same as for the Velocity tab, and the modification can be made in a broad brush way, such as by drawing a curve on the note scaling graph for that parameter, or as tightly controlled as selecting a specific MIDI note and entering a numerical value for the scaling percentage to be used for that note. Typical uses might include increasing the frequency of filters as higher notes on the keyboard are played. However, with some clever sound programming, and, for example, using single Oscil sounds, where Oscil amplitude is modified in extreme ways by note scaling, the capabilities here could even be used to simulate a keyboard split and multi-timbral response (Absynth is not actually multi-timbral) to different regions of the keyboard.

The MIDI window's Tune tab provides the interface for creating and editing custom tunings. One side of the interface has the list of all possible MIDI note numbers, and the other side has a set of parameters for affecting what actual notes result from playing that note number. Those parameters include a note value (which can be fractional -- for example, to enter a note midway between MIDI notes 60 and 61, you would enter 60.5 to achieve a quarter step interval), absolute frequencies in Hz (e.g. 440 Hz), ratio compared to a base note number (e.g. if the base key is set to 60, or C3, then to achieve standard tunings C2 and C4 would need to have a ratios of 0.5 and 2.0, respectively). Setting tuning values for 128 individual notes could obviously get fairly tedious, so Absynth provides some shortcuts for common situations such as the ability to link octaves such that tuning a particular note for a given octave will affect the tuning of that note in all octaves. Thus for a standard tuning this would mean tuning twelve notes instead of 128. But the "octave" doesn't necessarily have to be defined in terms of 12 notes, either on the keyboard or in terms of the actual intervals between notes. Absynth provides parameters to set the number of notes per octave and the interval that defines the octave. Contemplating tweaking in this area sure makes me glad my music just uses standard equal tempered tunings, but, for those who are more adventurous, Absynth provides a great deal of power in this area, adding to the significant number of factory tunings Absynth includes, and a number of presets which take advantage of those tunings to provide more exotically flavored patches.

Time to Play

All the power Absynth, or any other softsynth for that matter, has is all well and good, but the bottom line questions I always come to are how it sounds, how useful the sounds it comes with are for the type of music I want to make at the time, and how easily the sounds can be tweaked if it turns out I find something that is close, but not quite right, for a particular use or truly get adventurous and decide to create a sound from scratch. Much of my music tends to be in country and pop/rock genres where I often know ahead of time roughly what kind of sound I need, and go directly to the best softsynth or softsampler and sample library I feel is most likely to provide the needed sound. For example, if I'm looking for an electronic organ sound for a rock or country song, I will generally turn to NI's B4, and a very large percentage of the time I'll find a preset that will work just fine, or one that is real close where tweaking a few drawbar settings gets me the rest of the way.

Absynth really isn't an instrument aimed at emulating other instruments, though, and when it comes to synth sounds, especially ones where I'm looking for something a bit more obscure, of the "I'll know it when I find it" type, things become a bit more hazy. First is the question of what softsynth to pick as a starting point since many softsynths overlap to one degree or another in terms of the types of sounds they can create. I may have some feel that suggests it would more likely be digital than analog or vice-versa, and that may invite starting with a particular synth, or a small number of synths, to explore patches, but that can involve a lot of exploring with a synth that comes with as many presets as Absynth does (somewhere in the neighborhood of 800).

Because Absynth covers so many different types of synthesis and sampling, it can overlap to some degree with almost any other softsynth, and, thus, makes a good first choice in cases where it isn't obvious which synthesis technology is the right one. Perhaps more importantly, though, whether it is trying presets to find something in the ballpark or starting from scratch with a basic idea of what needs to be achieved, basic sound programming in Absynth tends to be pretty intuitive once you get a feel for its sound architecture. Thus, tailoring things from a reasonably close starting point, or even from scratch, seems a good deal easier to bite off on than with many other synths (e.g. I would equate embarking on a DX7-style sound programming task with contemplating a visit to the dentist for a root canal). Once things are in the ballpark on the sounds needed, there is also all the power of the envelopes and real time control to take something that is good and add the finishing touches to take it over the top.

The net is that I found Absynth relatively quick for getting in the general vicinity for non-emulative sounds -- I wouldn't even consider going to it as a first call for organ, piano, guitar, or other emulative patches. Then, once I had something in the ballpark, where on other synths I might still be tempted to search through presets further to see if I could find something even closer, with Absynth I found myself more inclined to just start tweaking to take the sound the rest of the way. That really speaks for the relative intuitiveness of the programming interface, and has everything to do with Absynth's having a user interface design that really could not have existed in the hardware world.

One minor wish list item in this category is, while the user drawable waveforms are great for flexibility, it would be nice to be able to use the current waveform setting for whatever is being modified as the starting point for tweaking. As it is, it is necessary to start from a sine wave each time a waveform is edited. The Waveform window's ability to load waveforms, including factory presets and user waveforms, does allow loading whatever waveform was loaded in a particular module, as long as you remember its name (i.e. since that name will no longer be showing once you tell Absynth you want to create a new preset). Thus, you can get to where you were fairly quickly. However, this extra step does slow things down slightly versus, for example, just being able to click on an "edit" command and dive right into editing the current waveform.

A few specific observations that arise both from just playing around with Absynth and from attempting to use it on real projects might be in order. In the playing around department, I found many of the presets provided with Absynth to be sufficiently inspirational that I made a note to myself to revisit some of those sounds in the future to see what kinds of compositions might arise from playing with them further. Had I been the type who keeps a recorder running at all times, I probably would already have a few compositions in progress from my "playing around" sessions. Another observation is that, while my general philosophy on synth sounds is not to mix too many instances of the same synth in a single project because the uniformity in individual sound can limit the richness of the sound of the mix, I found Absynth very conducive to using for multiple tracks in a single project, perhaps even as the sole synth in some types of projects. I strongly suspect the reason for that is a combination of the multiple sound generation technologies that Absynth provides along with the complex programming and control capabilities that help avoid the risk of one-dimensional sounds. Finally, while I could imagine Absynth's filling the need for a variety of types of sounds, my sense is that the area where it stands out as the potential first call synth for that area would be where there is the need for complex, evolving sounds, be it for pads or for lead work.

Not all my experiences using Absynth were rosy, though. The first project I worked on with Absynth was with Absynth 2.0.1 under SONAR 3. I was using Absynth on a total of 4 tracks, two with synth pads, one with an ornamentation type instrument, and the other for a bass part. To say Absynth crashed a bit would be a bit of an understatement. For example, each time I'd insert a new instance of Absynth, assign it to MIDI and audio tracks, then start tweaking, odds of a crash rose significantly. I found myself repeating the same step multiple times, and having to remember to save early after adding an instance so I didn't have to re-add that instance later on after the crash, almost inevitably, occurred. And that is only one example.

Thankfully, within a week or two, the Absynth 2.0.4 update came out, and Absynth has become much, much more stable for me. I have still experienced a few crashes, but nowhere near as many as with 2.0.1, and there have been times when I've been using Absynth nearly all day and going days between crashes. As of the moment, based on release 2.0.4, my sense is that Absynth isn't quite as solid on the reliability front as most of the other commercial softsynths I use regularly. However, at least on my system, and for the uses I've made of it (generally song demos in a single user home studio), it is stable enough to be usable for real projects, and the sound quality and programmability benefits far outweigh any minor stability concerns.

Wrapping Up

Earlier in this review I alluded to my dilemma with respect to how to describe Absynth in a way that would demonstrate its uniqueness despite its having various areas of overlap with many other synths. I now end the review with a similar challenge with respect to how to summarize with respect to suggesting where Absynth fits in terms of potential applications. In particular, Absynth can do so many things, some of which overlap with other softsynths, but where does it really stand out?

Perhaps the best I can do is to suggest that, in terms of my own thinking of when I will go to Absynth first over the various other softsynths I have at my disposal boils down to two key circumstances:

The first case is when I know I need a synth sound of some type, but it's not obvious what specific flavor of synth is needed. Absynth's multiple synthesis and sampling technologies and easy tweaking facilitate this use. My sense is that the odds for getting a sound that will work in this scenario from Absynth are far better than they are with a synth that focuses on a single type of synthesis technology.

The second situation is when I need a sound that is clearly electronic, but has the richness and playability of an acoustic instrument. Absynth strongly suits this role due to its advanced programmability and real-time control options, which go far beyond where most other softsynths leave off.

Absynth may not be for everybody. For example, I really don't see it fitting very well into acoustic music -- a sampler or physical modeling synth will likely do a much better job of the acoustic instrument emulation in that situation. However, for anyone whose music either is purely electronic or uses significant electronic elements, I would consider Absynth a "must have" softsynth. I tend to be somewhat more toward the former end of the spectrum, incorporating electronic elements once in awhile, but much more frequently staying within the acoustic domain. However, from my experience with Absynth to date, my sense is that having Absynth available may actually encourage me to add more electronic elements because there is also a fun factor in the programming that it's been ages since I have experienced with a synth of any kind, no less a softsynth, and that aspect makes me want to find excuses to use it.

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

