



## **Review: Cakewalk SONAR 7 Producer Edition by Rick Paul - 13th December 2007 -**



Most CakewalkNet readers, and all current CakewalkNet editors, use Cakewalk's SONAR digital audio workstation for MIDI and/or audio production. For example, I've been using SONAR since version 1.0 came out, back in the year 2000, though I go back even farther using Cakewalk products as my primary DAW, having started with Cakewalk Pro Audio 9. As such, most of the time we focus CakewalkNet product reviews on products that add value to SONAR. Every once in awhile, though, we like to take an updated look at what's new with the SONAR platform itself. (See our earlier reviews of [SONAR 2 XL](#) and [SONAR 4 Producer Edition](#).) Now is one of those times.

## **Background**

In September of this year, Cakewalk announced the release of SONAR 7 Producer Edition, the latest release of their flagship DAW. We'll take closer look at many of the new features below. Just in case you haven't seen the press release or been to the [Cakewalk web site](#) recently, here's the bullet list of features from the product announcement:

- Integrated Step Sequencer view provides the most innovative step sequencer feature set available in any DAW today
- Smart MIDI Tools—intuitively use one tool for multiple editing tasks; completely customizable
- Real-time Drag-Quantize allows selected notes and events to be quickly aligned with snap settings
- Multilane controller editing in Piano Roll view; controller data can be moved/copied across lanes; multiple controls can be viewed/edited per lane to economize screen real estate
- MIDI display features including MIDI Magnifier, MIDI Meters, and Velocity Colorizer
- Enhanced MIDI editing with functions for splitting, gluing, and muting notes
- More ACT (Active Controller Technology) presets for popular hardware
- New instruments: Z3TA+ 1.5 waveshaping synthesizer, Dimension LE (featuring Garritan Pocket Orchestra), Rapture LE, and DropZone sample player/synth; Over 1000 instruments sounds
- Roland V-Vocal 1.5, now with Pitch to MIDI conversion
- LP-64 Multiband linear phase mastering compressor/limiter with AutoQ functionality
- LP-64 EQ linear phase mastering EQ with 20-point control curve
- Internal sidechaining for Sonitus:fx Compressor, Sonitus:fx Gate, VC-64 Vintage Channel, and 3rd party VST plugins with sidechain (multi-input) capability

- External Insert Plug-in lets you insert and mix with external hardware effects and instruments with automatic delay compensation
- Boost 11 Peak Limiter
- Dim Solo keeps tracks in context by reducing volume of unsoloed tracks by 6, 12, or 18 dB instead of muting
- Drag & drop EQ settings across channels in the console view
- Shortcuts for assigning a series of audio inputs, and routing selected tracks/busses to the same input/output/main
- Original SMPTE time stamps now stored with clips with easy revert to original time stamp
- Right-click at cursor to Import audio/MIDI data
- Wave-64 support provides capabilities to record large scale (2 GB+) projects
- High bandwidth multi-track recording optimizations
- New file format import/export options including Sony Wave-64, AIF, CAF, FLAC, Sound Designer II
- Integrated Audio CD ripping and burning
- Cakewalk Publisher 2.0 for uploading and presenting music online

That is a fairly long list of new features, perhaps the longest yet for a new SONAR Producer Edition release. Thus, it may help to summarize a bit. Major upgrades in MIDI functionality, including a brand new integrated step sequencer, improve capabilities and productivity for MIDI sequence creation. A number of new instruments provide new sound generation capabilities for those sequences. There are also new options for getting data in and out of SONAR including from and to audio files, from and to audio CDs, and onto web sites. SONAR's already-extensive bus routing options now include internal sidechaining for plug-ins that support that feature, and SONAR includes multiple plug-ins that support sidechaining. A new suite of mastering plug-ins adds to what can be done "in the box" without needing third party products. There are also performance optimizations and new features aimed at high-end studios and high-end audio recording projects, including the ability to integrate external audio hardware almost as easily as a software plug-in (of course, that may be useful to home studio users, too). There are also a number of general features, often aimed at improving productivity.

Of course, these new features are only a small subset of SONAR's features overall. A sampling of the high-level features brought forward from earlier versions of SONAR include:

- Pristine 64-bit mix engine; first true 64-bit end-to-end signal path
- Customizable user interface: put favorite features right at your fingertips
- AudioSnap suite of audio timing tools featuring multiple time stretching algorithms including iZotope Radius™, percussion mode, and groove clip
- Active Controller Technology (ACT): dynamically maps effect, mix, and instrument parameters to any MIDI controller or hardware control surface, automatically remaps parameters to whatever view has focus
- Roland V-Vocal VariPhrase technology for perfection of vocal tracks through exceptional audio stretching, pitch correction, and dynamics, phrasing, and vibrato—now updated with pitch to MIDI conversion
- Perfect Space Convolution Reverb powered by Voxengo
- VC-64 Vintage Channel powered by Kjaerhus—now updated with sidechaining
- SurroundBridge—easily use stereo effects in a surround environment
- BitBridge enables 32-bit VST effects and instruments to run smoothly in SONAR when running in an x64 (64-bit) environment
- POW-r Dithering
- iZotope Radius Time Stretching
- MPEX 3 Time Scaling
- Bus & Synth Waveform Preview
- Clip-Based Effects and Editing
- Crash Recovery safeguards projects against faulty plug-ins and other sources of crashes

SONAR 7 is still a Windows-only application. The 32-bit version of SONAR 7 is supported under Windows XP and Windows Vista. The 64-bit version of SONAR 7 is supported only under Windows Vista, which is a change from

SONAR 6, which was also supported on the 64-bit edition of Windows XP. Note that this doesn't necessarily mean the 64-bit version of SONAR won't run under the relevant version of Windows XP. It simply means you're on your own if you encounter problems unique to that environment. As for Mac users, they're only welcome to the SONAR party if they don't mind letting their Mac's "dress up" as Windows by running Windows under Mac OS X with Boot Camp, which Cakewalk now officially supports. I've actually heard the Mac hardware running this way makes a pretty good platform for SONAR.

The minimum system requirements Cakewalk lists for SONAR 7 include either a Pentium 4 1.3GHz or Athlon XP 1500+ or higher CPU, 256 MB of RAM, a graphics card supporting 1024x768 resolution with 16-bit color, and at least 100 MB for full program installation. Recommended system requirements include either a Pentium 4 2.8 GHz (EM64T) or Athlon 64 2800+ or higher processor, 1 GB or more RAM, and a graphics card supporting 1280x960 or better resolution with at least 24-bit color. Additional system requirements include a DVD drive for software installation, Windows-compatible MIDI interface (if you'll be connecting any external MIDI devices, including control surfaces), and a WDM- or ASIO-compatible audio interface (for low latency operation -- the minimum requirement in this area is Windows-compatibility). How far even the recommended system requirements will go in meeting any individual user's needs will depend on that user's usage patterns. For example, even 1 GB of RAM won't go very far if you're running multiple sample-based softsynths that load their samples into memory. As with many applications, a good rule of thumb is to buy as fast a processor and as much memory as you can afford. That said, if you're happily running SONAR 6 on your older, less-configured system, SONAR 7 is also likely to run happily on that system. However, if you also want to use some of the new plug-ins, such as Dimension LE or the new linear phase plug-ins, those will add fairly significant CPU demands to the picture.

The U.S. list price for SONAR 7 Producer Edition is \$619, though street price is under \$500. Upgrade price from SONAR 6 Producer Edition is \$179. Cakewalk also offers the lower-priced, SONAR 7 Studio Edition, which includes fewer bells and whistles (see below for differences). That lists at \$369, with a street price under \$300, and a SONAR 6 Studio Edition to SONAR 7 Studio Edition upgrade goes for \$99. Various other upgrade and crossgrade prices are available. Cakewalk recently introduced the new Cakewalk Pro Suite, which, in addition to SONAR 7 Producer Edition, includes Project5 Version 2.5, as well as the Dimension Pro and Rapture softsynths. A sampling of list prices for that bundle includes \$999 for first-time customers, \$479 for customers upgrading from SONAR 6 Producer Edition, and \$349 for customers who already have SONAR 7 Producer Edition, among other upgrade possibilities. Street prices won't help you on upgrade options there, but the street price for new customers is \$799. No matter which version you pick, SONAR is still a bargain compared to most of its competition.

I received the SONAR 7 Producer's Edition Upgrade from SONAR 6 Producer's Edition. That package included the SONAR 7 DVD in a nicely designed, standard format DVD case (no cheap paper envelopes, thank you!), along with the 230-page *SONAR 7 Producer & Studio Edition User's Guide*. The latter includes sections on SONAR basics, an extensive tutorial, fairly in-depth coverage of SONAR 7's new features, and a troubleshooting guide. If you need a full SONAR 7 reference manual, you can either use the context-sensitive on-line help system, grab the full PDF-format reference manual off the installation media, or purchase the printed version from Cakewalk.com. Cakewalk's choices on paper versus on-line documentation seem to make a good tradeoff between getting both new and upgrading users up and running quickly and keeping product costs, and the amount of physical bulk users need to store, down.

Installation of the software was painless, taking about 10 minutes on my system, including copying all the softsynth content -- on the order of 3 GB worth! I was glad to see the installation program provided an explicit chance to choose different content folders for the softsynths, as I like to separate out sample and other softsynth setting libraries, which grow significantly over time, from application programs, which stay relatively constant in their disk space consumption. At the end of the installation process when running SONAR 7 for the first time, you are prompted to specify your previous version's preferences file. I was surprised that, even when you are selecting a preference file, the migration facility defaults to not keeping your earlier default folder assignments. It's easy enough to override this default, but it seems like that default is inviting user error. I also noted that my toolbar layout changed, despite my having migrated my SONAR 6 preferences. Curiously, it did seem to preserve some of my layout, but not all. I keep the Views toolbar running vertically down the left-hand side of my screen, and its position was preserved after the upgrade. However, some of the toolbars I'd positioned horizontally across the top of my screen in SONAR 6 did not appear in SONAR 7, and some additional toolbars were installed in SONAR 7. Perhaps

the vertical Views toolbar's being in the same place was purely coincidence? In any event, this is a minor consideration since it only needs to be readjusted once after the SONAR 7 upgrade. (Cakewalk notes that, because toolbar changes made in SONAR 7 included new buttons and changes in functionality, they wanted to make sure customers who were upgrading were at least exposed to the new options.)

As in recent versions of SONAR, software protection is via the combination of a serial number, the e-mail address you use to register the software, and the registration code sent to you at that e-mail address. Registration is required, but you can run the software for up to thirty days before registering. This method of software protection is one of the most user-friendly methods in the industry. The initial authorization takes only a minute or so if done via the Internet. If somewhere down the line your hard disk crashes, you massively upgrade your system, or you even switch systems, all you need to do is enter the same information you'd entered the first time to reauthorize SONAR. Cakewalk even makes it easy to keep that information together by including a sticker inside the DVD case with spaces to write down the e-mail address you used for the authorization and the registration code returned. (Note: If you change e-mail addresses later, it still doesn't matter as, for authorization purposes, you still just use the one you'd originally used to register.)

This review cycle was begun using the initial release of SONAR 7 Producer Edition under Windows XP SP2 on a Core 2 Duo E6600-based system with 2 GB of RAM, and an E-MU 1820M audio and MIDI interface. About two weeks into the cycle, Cakewalk made the 7.0.1 patch available, and that was used for the remainder of the review. (NOTE: Cakewalk indicates that a 7.0.2 patch update will be available sometime later this month.)

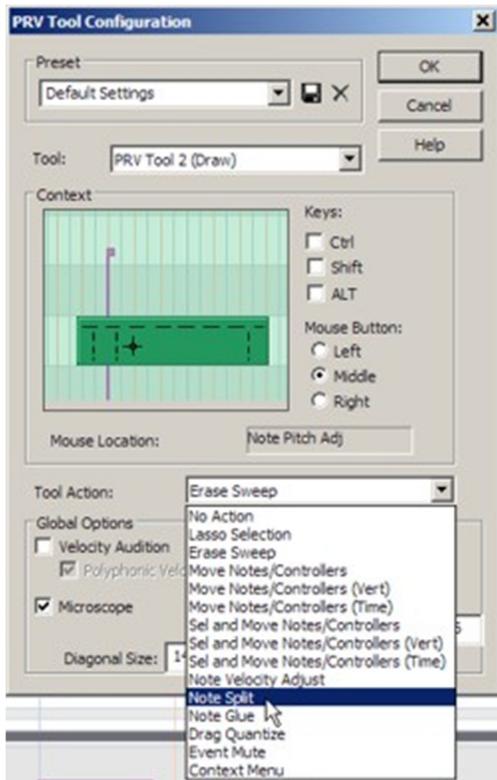
Let's dive into some new features.

## **MIDI to the Max**

Cakewalk started off in the MIDI sequencing world before adding audio functionality and migrating to the more general DAW world. While SONAR has always been highly functional in the MIDI realm, the lion's share of the new features introduced at each step of SONAR's evolution has concentrated on the audio side of the DAW functionality. It's not that MIDI users were neglected or anything. In fact, new softsynths and Synth Rack improvements, among other enhancements, are of particular interest to MIDI users. However, the general user interface and workflow in the MIDI area has stayed relatively constant while other areas got more attention. SONAR 7 aims to change that.

SONAR 7's MIDI enhancements aim squarely at improving productivity when manipulating MIDI data. Flexible MIDI tools provide the potential to use each tool for multiple tasks, making it highly efficient to switch between tasks as needed, while also letting you customize the interface to each tool to your tastes. Display features like multilane controller panels in the Piano Roll View, a MIDI data magnifier, MIDI meters, and notes shaded based on velocity make it quicker to see at a glance what is happening in your MIDI data. New MIDI editing functions allow splitting, gluing, and muting notes, thus making it quicker to get from what you've got to what you want. Real-time drag-quantizing makes it quick and easy to align selected notes and other events with snap settings. Let's take a closer look.

Historically, SONAR's Piano Roll View (PRV) has provided three tools: Select, Draw, and Erase. Each featured a fixed set of operations, some dependent on context (e.g. the position of the cursor with respect to a note or within a note). If your workflow involved using operations on different tools in close proximity to one another, the only way to do that was to switch back and forth between tools, which could be a drain on productivity. Of course, different users work in different ways, so, had Cakewalk decided to put everything on one super tool, this could have made things worse for some users. Furthermore, what might be intuitive to one user might be counter-intuitive to another, and differences in pointing devices, such as the number of buttons on your mouse, might also make any such super tool more or less acceptable. I can just picture Cakewalk's marketing and development people tearing their hair out trying to figure how to balance a bunch of conflicting requirements and preferences.



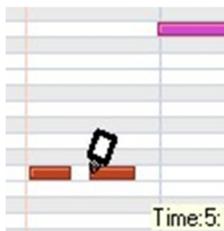
Cakewalk tackles this issue head on in SONAR 7 via their new flexible PRV tools. SONAR 7 features the same three tools as previous versions of SONAR. What is new, though, is that you can configure every operation for each of the three tools. For example, if you have a three-button mouse, and you click with the middle button in the lower center portion of a note, what would you like to happen if you're using the Draw tool? SONAR's default is to delete the note, and, if you drag from there over additional notes while holding the mouse button down, it will erase all the notes you drag over. Perhaps, though, you'd like it to split the note you're clicking on instead? No problem. Just go to SONAR 7's new "PRV Tool Configuration" facility, select "PRV Tool 2 (Draw)" from the Tool menu, then click with the middle button of your mouse on the relevant portion of the note in the context diagram. Once you've done that, you can assign the Tool action to your preferred function (see screen shot at left), and you're ready to go. You'll probably also want to save your settings under a new preset in case you want to be able to revert to the default settings later on. You can make as few or as many changes as you want, to one or all three of the tools. When you consider that you can employ up to three mouse buttons, plus modifier keys (Ctrl, Shift, Alt) and combinations of modifier keys (e.g. Ctrl-Shift) for each key, there are plenty of potential variations. You can even create that one super tool, if you like. Just keep in mind that, once you start changing tool definitions, your documentation will be your memory, along with the visual clues some, but not all, of the tools give you in the form of cursor

changes.

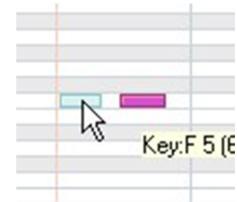
The flexible PRV tools system also provides the ability to ease the transition for users moving to SONAR from other DAWs. Presets are provided to emulate the MIDI tool behavior of Cubase, Digital Performer, and Logic Pro 7. ProTools seems curiously absent, though ProTools is much better known for its audio side than its MIDI side, and many ProTools users use another MIDI sequencer, such as DP or Logic, alongside ProTools.

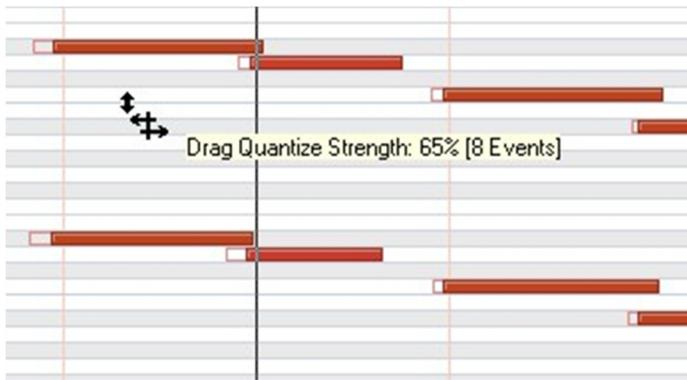


In addition to providing the potential for customizing the MIDI tools, SONAR 7 provides a number of new PRV operations that are carried out through the MIDI tools. On the editing front there are new functions for splitting one note into two shorter notes and gluing multiple notes at the same pitch together. These can aid productivity in a big way, for example when working out arrangements that will ultimately be played by multiple instruments, but which were recorded from a single keyboard performance. Of course, they can also be used for fixing mistakes, but we don't make those, do we? No, we just, er, change our minds from time to time.



SONAR 7 also provides a new, non-destructive note-level mute operation, which is also accomplished using the MIDI tools. A muted note looks like a faint outline of the original note, giving the impression of a ghost note. You won't hear it on playback, but it is easy to change your mind by simply unmuting it later, whether that is in the next operation, many operations down the line, or even in a totally separate editing session a few days or months later. It's a great tool for auditioning changes without having to worry about whether you can get back to where you came from without losing some other changes you made along the way.



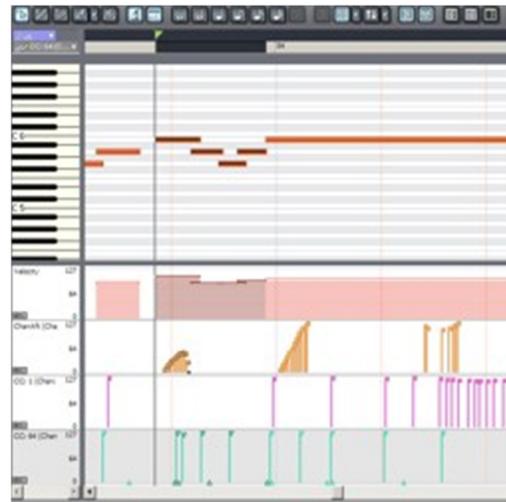


One more new MIDI editing operation, called Drag-Quantize, allows quickly quantizing one or more notes by dragging the cursor up or down to set quantization strength. Drag up and the quantization strength increases. Drag down and the quantization strength decreases. The quantization grid is based on your current snap settings. You can see the proposed changes visually in real time as the original note positions remain in outline form while the notes themselves slide based on the level of quantization reflected by your operation. This makes quantizing notes much more intuitive than picking a quantization

strength percentage out of the air, using Process/Quantize to try out your setting, seeing how it looks, then undoing and trying again if necessary.

Creating expressive MIDI tracks often involves using multiple continuous controllers (CC's), such as sustain pedal, expression pedal, pitch wheel, and mod wheel, on a single MIDI part. Prior to SONAR 7, all CC data was represented in a single pane of SONAR's PRV (or optionally overlaid on the note pane in recent versions). It was possible to turn the display of individual events off and on to filter what was displayed. However, if you really wanted to edit potentially any CC appearing in the track at any point, it could mean a lot of toggling back and forth on which CC's were visible and which CC was active for input. Depending on how you worked within those constraints, it could either make for a very cluttered display or a lot of toggling in cases with overlapping CC types.

In SONAR 7, Cakewalk has given the controller pane multilane capabilities. While you can still do things the old way if you prefer, keeping all controllers to a single pane, you now also have the option of spreading CC's out across panes to your heart's content, including having one lane per CC type if you prefer. At any point, you can add or subtract lanes in the controller pane by clicking on a plus or minus sign, respectively, in the relevant lane. When editing controller values with multiple lanes, there is no longer any confusion about which controller type you'll edit when you reach into a group of densely packed CC's, nor any having to define the "current controller" when you want to enter new controller values by hand. Simply go to the relevant lane for that controller and edit or input as you like. Your choice of lane lets SONAR know which controller type is applicable.



Also in the category of "new controller tricks":

If you'd like to copy CC values from one type of controller to another type of controller (e.g. expression pedal values to volume pedal values), SONAR 7 makes this easy. Go to the lane with the first type of controllers and select the values you want to copy. Next, right click on the lane header for the second type of controller and pick "Copy Selected Events to This Lane" from the popup menu. You can move controller events from one lane to another in a similar way.

SONAR 7 also provides the toggle button-invoked option of having controller events selected when selecting notes that overlap those controller events. For example, say you've painstakingly put in expressive controls for that saxophone emulation. Maybe there are some pitch bends on the start of the note, volume swells during the course of the note, vibrato introduced partway through the note, with its speed and depth varying as the note is held, and other expressive programming to simulate what a real sax player might do over the course of a held note. If you move that note forward or backward in time, it would sure be nice to have the expressive changes maintain their positions relative to the note. For example, you don't want the bend up to the note via pitch bend controls to move to the middle of the note, just because you moved the note earlier in time. With the "Select Controllers Along With Notes"

button engaged, you don't have to worry about that, because the relevant controllers get selected when you hit the note. When you move the note, the controllers then move along with it. That sure beats having to manually make sure you've got the right controller values selected. There's a bit of a catch, though. When notes overlap inadvertently (or even intentionally for legato phrasing), there may be cases where you want some, but not all, of the overlapping CC values to be selected. This is because some of the CC values may "belong" (i.e. functionally based on how you programmed or played things) with the note you're not selecting. That is why this feature is optional and easily toggled off and on. In that case you'll have to decide whether it is easier to toggle the switch on first and have it get most of the right stuff automatically, then manually remove the CC values you don't want, or to go back to the old way of manually selecting precisely what you do want.

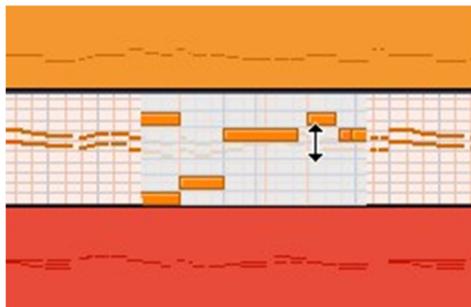
Note velocity is now treated specially compared to other controllers in some cases. This is logical because velocity is actually part of the note events themselves, not separate CC events. One difference is purely in the display of velocity information. There is now an option to show velocity values only for selected notes, leaving it hidden the rest of the time. Another difference is a new behavior. Previously, when you painted velocity values within SONAR, all notes in the range of your painting were affected. Now, if any notes are selected, only those notes will have their velocity values affected. If no notes are selected, velocity painting will work as it did prior to SONAR 7.

The bottom line for these "new tricks" and changed behaviors is better efficiency for getting what you want to do done, while making it easier to avoid inadvertently doing things you don't want to do. While most of these changes, in and of themselves, may feel small, the idea is that saving a bit of time in a lot of places adds up to a lot of time saved.

In one particular case, however, there is a seemingly small change that itself provides the potential of a good deal of saved time. It also makes it feasible to directly do things that seem like they should have been obvious, but which weren't available in previous versions of SONAR. SONAR has had multi-lane MIDI tracks for some time now. One common use of these is to record a number of takes, the "comp them" into a more polished take (i.e. create a composite take from the best of the raw takes). This allows, for example, replacing wrong notes from one take with the right notes from another one. It also allows experimenting with different possible parts by looping the recording over and over until you've got a bunch of options then figuring out which parts, from any of the takes, you want in your final part. In the process of reviewing your raw takes to see what you've got, it is reasonably easy to cut the entire parts up into smaller clips that are likely to be "units" you will be inclined to ditch or keep. These units might last an entire song section, be a musical phrase, or even just be a few problematic notes. The idea is you can mute clips, solo clips, and so on to quickly determine what the composite part would sound like without having to delete material you might change your mind and want back later. Problem, though: What if a particular clip sounds like it might end up being "the clip" you'll want, but there are a few wrong notes in it? Or perhaps there are a few notes that need to be slid forward or backward in time. Either way, you're not quite sure enough that this clip will be "the clip" to delete the other competing clips. You want to try the edits to be sure. In SONAR 6 and earlier, when you brought the track up in PRV, you'd see all the notes from all the overlapping clips, which might well be playing very similar parts. To be quite blunt, it was a bloody mess!

SONAR 7 addresses this consideration, along with other related situations where you may either want muted clips to be invisible or at least obviously differentiated from unmuted clips. There is a new capability to hide notes from muted clips in the PRV. Without this option turned on, notes from muted clips appear faint, and in a desaturated color, similar, but not identical, to the muted notes described above. With the option turned on, though, those notes aren't visible at all. The notes you see in PRV represent only the unmuted clips. Since you're likely to be muting clips you aren't auditioning at any one time (unless you prefer cacophony, of course), this essentially allows you to edit only the parts you're considering, without being confused by parts that are likely to go away, at least in this version of your comped part. Of course, you can always unmute the clips later, or make the muted clips visible. They didn't go away or anything. This capability is just letting you focus on what you're specifically trying to do at that moment. Prior to SONAR 7, you either had to do a lot more clip cutting up at the Track View level, move clips to be edited to their own track to focus on them, or otherwise work around all the notes that could conceivably be stacked in the same space as the notes you're trying to edit. This is a biggie for anyone who comps MIDI parts. While this is great news, I still have one significant complaint in this area. This capability only works in the PRV. If you prefer doing your tweaks in the Staff View, as I definitely do when there is a question as to possible wrong notes, or in the Event List View, you've still got the same confused picture as prior to SONAR 7. Perhaps Cakewalk

might include other MIDI views in SONAR 8? Still, being able to get the same view as you're hearing in the PRV is a huge improvement.



If you've ever done any extensive MIDI editing in either SONAR's PRV or the Track View's per-track PRV mode, you've likely experienced some frustration with needing to zoom out to get a view of context, while having to zoom in to check, and possibly modify, details. If you're doing a good deal of tweaking in different parts of a song, this could mean a lot of zooming action. SONAR 7 introduces the Piano Roll View Microscope Mode to address this situation. Whenever a PRV mode view is zoomed out sufficiently far to cross a certain threshold of detail visibility (or, more accurately, the lack thereof), the PRV microscope mode is active. When you hover your

cursor over an area of notes with this mode active, the notes immediately surrounding your cursor are magnified to a level that gives you a much better picture of the details. It also allows you to edit those details almost as if you were zoomed in to that level of detail in the first place. The threshold can be configured, with the default being that microscope mode becomes active when note heights are less than 6 pixels. You can set that to be anywhere from 2 to 20 pixels. The amount of magnification is also configurable, as is the decision as to whether note times/lengths should also be magnified when notes widths become sufficiently narrow, and, for that matter, whether microscope mode is active at all. If enabled, microscope mode is available in both the Track View (only for MIDI tracks displayed in PRV mode) and the Piano Roll View. This mode is great for quickly inspecting what is going on in a section of your track. It can be a bit finicky, however, for actually making changes. Even with the magnification, the size of the notes may be smaller than optimal, but, more significantly, because the scale is different than the surrounding notes, it can be a bit confusing to gauge where a note should go, though pitch previews can be of help for vertical orientation. Also, as you move the cursor to move the note, the area being magnified moves, which can be disconcerting. Of course, some changes are easier than others. For example, it is a great way to find the note you need to change, right click on it, then fill in explicit values in a dialog box. It is a bit more challenging to slide a note around in pitch, and more challenging still to accurately slide a note around in time. Perhaps the biggest challenge, though, is to be able to accurately get the note you want when that note is surrounded by dense continuous controller information in the Track View's PRV mode. Turning off the display of the other controllers helps get the note, but then I also found that moving a note that way then turned back on display of additional controllers if the mode to select controllers with notes was enabled. It seems that Cakewalk may have a bit of fine-tuning to do with this capability.

Another new visual monitoring capability SONAR 7 provides in both of its PRV modes is getting a feel for note on velocity differences based on the note's color. In particular, while the main color of the notes remains constant for a given track, the shade or intensity of that note is varied depending on how loud or soft that note is. A loud note is more intensely colored than a soft note. Though the differences between the colors of very loud and very soft notes can be quite pronounced, for example almost making very soft notes look like a somewhat different color (perhaps due to running out of headroom for one color in the RGB color mix before running out of headroom in other colors?), most of the color differences are subtle enough that it can be tough to distinguish them, at least for my eyes with my monitor. As such, this capability may be best suited to picking out gross problems, such as notes played way too loudly or softly, which could potentially mean mistakes, as opposed to for seeing finer nuances. Then again, we already have velocity tails or bar graphs, depending on your preference settings, for seeing finer details of velocity patterns.



As such, this capability may be best suited to picking out gross problems, such as notes played way too loudly or softly, which could potentially mean mistakes, as opposed to for seeing finer nuances. Then again, we already have velocity tails or bar graphs, depending on your preference settings, for seeing finer details of velocity patterns.



The last MIDI monitoring feature I'll mention before moving on doesn't appear in the PRV at all, but rather in the Track View and the Console View. SONAR now provides MIDI meters. There are two portions of these. The largest, most visible portion looks somewhat like a normal SONAR audio meter in that it rises and

falls over the course of a part's being played. Rather than reacting to volume information, though, it is reacting to MIDI velocity. While velocity is very commonly mapped to volume in a MIDI instrument, or at least bears some relationship to volume, all other things being equal, not all instruments use velocity for volume. For example, a number of the Garritan software instruments use velocity for the type of note attack while using another controller, such as mod wheel or expression pedal, for volume. The other portion of these new meters is at the far end (i.e. at the top or right, depending on whether the meter is oriented vertically or horizontally, respectively) of the meter, where the overload indicator would go on audio meters. It is not, however, an overload indicator, but rather a MIDI activity (or MIDI data present) indicator. Anytime any MIDI data is being played, be it notes, continuous controllers, system exclusive information, or whatever, this will light up. Together, these meters can help in sanity checking that what you think should be going on is actually going on, including whether velocity levels are in line with your expectations. If everything looks good here, but you still can't hear the sound you thought you should hear, especially, though not exclusively, with an external MIDI module, you know the problem lies elsewhere, such as in the routing assignments or further down the line in the routing of MIDI and audio signals. I have to be brutally honest here. This is probably my least favorite change in SONAR 7, and my biggest wish for it is that I could disable these meters. I can appreciate why someone using external synth hardware might find these invaluable, but I haven't used external synth hardware since somewhere around SONAR 2. I can also see why people doing General MIDI arranging, or other similar work where instrumental volume may be largely controlled via MIDI (i.e. as opposed to audio settings), might appreciate the velocity meters. For me, though, they are simply a distraction. I do all my softsynth mixing at the audio level, with velocity being used mainly in the feel of a performance, not the volume of that performance. Additionally, each time I see the MIDI activity indicator portion of the meter light it feels like I'm getting an overload. (To be fair, the MIDI activity indicator is not red like the audio meter overload indicator, nor does it stick in the "on" position. It is simply the placement of the indicator, and its background being the same color as the overload portion of the audio meters, that gives this impression at a quick glance.) While I can hide MIDI tracks once I'm far enough along in my arrangement process, and that is when I'll be doing most of my mixing-oriented tweaks, I have to keep those tracks visible until such point as the tracks are finalized. It would sure be nice to be able to toggle the MIDI meters off without also toggling the audio meters off.

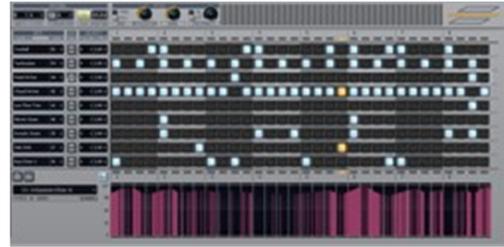
That one complaint, and other niggles, notwithstanding, I hope the information and opinions I've provided on these MIDI enhancements will adequately convey that SONAR 7 has made a major leap forward, albeit in large part due to the combination of lots of smaller steps, on the MIDI front. These capabilities in and of themselves would make SONAR 7 a must have upgrade for anyone who spends a significant amount of time doing MIDI editing. That's not all SONAR 7 provides, though, so let's move on.

## One MIDI Step at a Time

There is one more significant new MIDI feature in SONAR 7 that I intentionally omitted from the above discussion. I did this for two reasons. First and foremost, this feature is big enough that it warrants a section of its own. Secondly, as big as this feature is, it isn't something everyone, even all MIDI users, will care about, though I daresay some people who might initially be inclined to skip over this feature may find it more useful than its basic nature might suggest.

The feature I'm talking about is SONAR 7's new Integrated Step Sequencer. I have to admit the notion of a step sequencer doesn't excite me. While I do a lot of MIDI work, I either use third party MIDI loops (e.g. for drum parts) or manually play and record my own parts using my keyboard controllers. The last time I used a step sequencer, other than just to explore a new feature here or there (e.g. Cakewalk's Project5 also has a step sequencer), is back when I was using a Roland TR-909 for drum parts. In that case there was no choice but to program things in a step-wise fashion. Nevertheless, I vowed to look at SONAR's new step sequencer with an open mind, and it turns out that this isn't just your run-of-the-mill everyday step sequencer. While it certainly can be used that way, the way Cakewalk has implemented this, especially in how it has been integrated tightly into SONAR's more general MIDI loops capabilities, also lends it to some different, or at least nuanced, applications. The best quick summary I can think of is to suggest it is to MIDI loops what SONAR's Loop Construction view is to audio loops. Let's take a closer look.

You create a new instance of the step sequencer by setting your focus on a MIDI track, then calling up the Step Sequencer View. This takes a little getting used to as you also have to be careful no other step sequencer clip (see below) is selected when you do that or you'll just get that existing instance of the step sequencer opened. Once you're inside the step sequencer it looks much like you might expect if you've used drum machines sequencers or other computer-based step sequencers before. The main portion of the interface is dominated by a note matrix, with individual note values running vertically and time running horizontally. You can also open up a continuous controller pane, which shows one CC type at a time, across the bottom of the interface. The controller pane can also be closed to save space when you're not using it.



Looking at the details of the note matrix, just to its left you will find note names and MIDI note numbers for the corresponding rows of the note matrix. One thing that is important to note is that the notes may not be in order of note pitch. I'll mention one way they can get out of order below, but suffice it to say that, if that is disconcerting (and it can be if working with pitched material), you can manually drag the rows up and down to reorder them, insert rows, and delete rows. Note that it is also possible to have more than one row mapping to the same note. By default, the note names will be their MIDI pitch name, such as C1, C#1, etc. However, one key feature is that the step sequencer integrates tightly with SONAR's drum maps. Thus, if you are using a drum map, the note names will come from the drum map note names. This is one attribute that helps make the step sequencer an extremely nice interface for programming drum parts. (There are other attributes that add to that, as well.)

The main body of the note matrix is dominated by the rows of pitches and columns of "steps" (see below). To insert a note at any given pitch, you simply left-click on the step button that occurs at the point where the row of its pitch and column of its step occur. To delete the note, right-click on the step button. If you want to create note lengths that are longer than a step, you can tie multiple steps together. If you only want to tie two steps together, you can simply click in the space between those two steps. If you want to tie more than two steps together, first insert notes at all the relevant steps (a quick way to do this is to click-drag from the first step to the last note), then Ctrl-click in the space between the first two steps you want to connect, drag through the space between the last two steps you want to connect, then release the mouse button. Note velocity defaults to 100, but that can be changed on a row basis or on an individual note basis. To change it on a row basis, there are two velocity-oriented controls to the left of each row. One is Offset, which adds or subtracts from the note velocity indicated in the individual steps within that row. The other is Multiplier, which, as implied by its name, is used to multiply the step velocity to get the actual note velocity sent by the step sequencer. Of course, you have to be careful that you don't go outside the valid range of MIDI notes or you'll end up getting all values of 0 or 127, but the multiplier can have two decimal digits, to allow fractional multipliers such as 1.02 or 0.65. The note velocity can also be tweaked at the individual note level. To do this, either double click on the note and use the spinner control to increase or decrease the velocity or directly type in a new number, or left-click on the note to set focus on that note, then use your mouse wheel to spin the value up and down.

Directly above the note matrix are a number of global controls that affect the operation and interpretation of the note matrix. A Mono/Poly mode switch determines whether entering a new note in a column of the matrix turns off all other notes in that column. It does not, however, affect any previously entered polyphonic parts unless you make new entries into the corresponding columns. An Articulation knob controls how long a note is held, from zero to one hundred percent of its value. The default is to hold it out for its full duration, but you can make a part more staccato by turning the value down to taste. A Swing knob controls the amount of swing applied to the notes, with 50%, or "no swing", being the default. Given how rigid step-sequenced parts can be, I find that a little bit of swing, say somewhere in the 55% to 60% range, can be very helpful for making the sequences sound a bit more human. Finally, a Portamento toggle and knob control whether SONAR uses MIDI controller #65 to send portamento (slide between notes) times to your MIDI module. Whether portamento actually results will depend on whether the specific software or hardware synth you're using does indeed support portamento on CC#65.

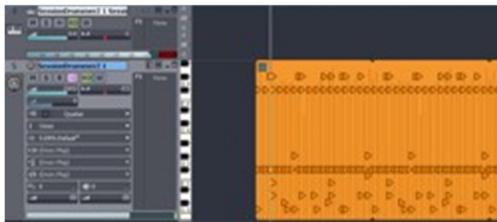
Down below the note matrix is the controller pane. It is hidden to save space by default, but there is a toggle switch to expose or hide it. When it is visible, a control on the left-hand side picks the type of controller to be programmed. You can only program one controller at a time, because there is only one controller pane, but, once you've done that,

you can program another controller, and the first controller's values are still stored in the step sequencer's programming and sent to the synth module. You can always bring them back for later editing as needed. The main part of the controller pane is reminiscent of SONAR's PRV controller pane. Simply draw the control values in using the left button of your mouse or erase them using the right button of your mouse. You don't get helpful numerical status information like you do in SONAR's PRV, nor is there any fine control over positioning. Still, the ability to simply draw or redraw controller curve shapes or isolated CC events should suffice for most uses. Using these controller curves to control softsynth parameters is another great way to humanize otherwise mechanical-sounding sequences.

So, what exactly is a step? It is a subdivision of a beat, and you can have anywhere from 1 to 16 beat subdivisions. If a quarter note gets a beat, this means a step could be anything from a quarter note down to a sixty-fourth note. Sixty-first notes anyone? In a typical case, such as a two bar loop of sixteenth notes, you might configure the step sequencer for 8 beats and 4 steps (i.e. per beat). That would give you a total of 32 columns, each corresponding to a sixteenth note, with subdivisions marked every four steps (i.e. at quarter note boundaries). There's nothing saying you have to go traditional, though. Let's see, if I'm calculating things correctly, if you use 7 steps per beat, with a quarter note getting a beat in your current time signature, that would make each step a twenty-eighth note. I don't even want to think about wanting to notate that one! However, this flexibility could be useful if you need to deal with tuplets that go beyond the garden variety, and being able to specify any number of beats you like (from 1 to 64) also provides flexibility for dealing with unusual time signatures. 7/8 with septuplets anyone? All you'd need to do for a two bar loop there is have your time signature for your project set to 7/8, then set the step sequencer for 14 beats of 7 steps.

For the progressive- and avant-garde-leaning types who still aren't satisfied, there's more. The step sequencer also provides a "fit to quarters" field and toggle that will allow you to fit however many beats and steps you've configured inside it to a specified number of quarter notes. For a simple case, say you initially create the two bars of 4/4 scenario with sixteenth note steps. Now, if you use the fit to quarters capability to fit that sequence to three quarter notes, your entire step sequence loop will be compressed to three beats. I don't even want to think about what note denominations that will yield -- feel free to do the math if you like. The notion of combining several looping instances of these things, all set to different fit to quarter values, brings up visions of forms of musique concrète that use tape loops of varying lengths, all playing against each other, to create an evolving set of patterns.

Getting back to the somewhat more mundane, it is worth noting that you can be programming the step sequencer in real time, while the music is playing. If you're programming a beat, or another type of sequence, where you don't need to reference the other tracks in your project, the step sequencer's own Play button will loop through the sequence and nothing else, and you can build up the patterns over time. If you want to hear the pattern you're building in the context of your project, you can use SONAR's looping to set the relevant range of measures. I'd recommend using either just the length of your step sequencer loop or the length of the entire section of music over which you'll be looping that sequence as your SONAR loop range.



If you do need to use more than one iteration of the loop for checking things in context, you'll be happy to know SONAR 7 makes this easy. Each step sequencer sequence appears in SONAR's track view as a special form of MIDI loop. You stretch it out just like any other MIDI loop to extend it to the range of time you need. Double clicking on the MIDI clip that represents the loop also brings up the step sequencer for editing. Copying and pasting the loop to get another instance links the two loops together, such that changes made to one instance are immediately effective in all

other instances. However, if you want to separate the instances so you can make changes to one of them that aren't reflected in the others, SONAR also lets you unlink the instances through a right-click menu command. That new, unlinked copy of the loop is still a step sequencer clip, but it is a unique instance that can be edited without affecting the original linked instances. This makes it easy to create variations on your pattern, and that second instance can also serve as the basis for additional linked instances, such as to use for another section of the song which gets repeated a few times throughout the song.

The integration of the step sequencer with SONAR goes beyond being able to roll out step sequencer loops and double click on a step sequencer clip to bring up the Step Sequencer View. Let's say you're using the step sequencer to program parts, especially initial parts you'd like to touch up later on, for example using SONAR's groove quantization capabilities. Once you're done programming the part, you can right-click on the step sequencer clip and do a Bounce to Clip to convert it into a normal MIDI clip. After that, you can edit the clip's contents manually in any of SONAR's normal MIDI views, process the data with any of SONAR's MIDI commands, and otherwise treat it like any other MIDI data, which it actually is at that point. If you change your mind and want to go back to the step sequencer for further editing, Undo will get you back there as long as you don't mind losing any other changes you might have made since the bounce when you roll back. Of course, this is also assuming you haven't saved and reloaded the file such that Undo is no longer applicable. Even if you have saved and reloaded the file, SONAR allows converting normal MIDI clips to step sequencer clips. When you do that, SONAR will prompt you for the type of note to be considered a step (e.g. a sixteenth note). When you do, it should be noted that it is as if you were quantizing the MIDI part to the step size you've picked, and continuous controller information may be lost altogether. I noticed some of it coming across one time, but none coming across another time, and, in neither case did the step sequencer version of that data accurately reflect what had been in the MIDI clip. I suspect the controller data was being quantized similarly to the note data, only reflecting the nearest value to the step size in the end. Whether this is a bug or by design, I'm not sure. Additionally, note order in the new step sequencer version of the data seems to be in the order the notes appear in the sequence, rather than in any logical pitch order. It probably also goes without saying that any swing, articulation, and other nuanced settings will be lost due to the strict beat-based quantization. Obviously you're not likely to want to use this capability as a pseudo cross-session Undo, but that isn't really the point. The idea really is to allow you to take a part you've played manually, or brought in from a third party MIDI loop, and read it into the step sequencer for further programming.

From an architectural standpoint, the step sequencer is simply spitting out MIDI notes, which are then fed through the rest of SONAR's MIDI processing signal flow as any other MIDI clip data on a MIDI track would be. Why is this important? It provides flexibility. For example, perhaps you want to program the melodic or rhythmic notes you want in a sequence into the step sequencer. However, you may also want to add keyswitches for a sampler, or additional continuous controller automation that does not loop with the sequencer clip, for example varying over the length of a song section instead. Or maybe you'd like the notes output from the step sequencer to be run through one or more MIDI plug-ins. For example, I made a sequenced bass part feel a lot more human by running it through the Style Enhancer Micro plug-in that goes back even earlier than SONAR 1.0 in Cakewalk's DAW days. The preset I was using there added pitch bends and other MIDI controller information to try and make it more closely emulate a human-played electric bass part. That made for yet another way to humanize the step sequencer's output.

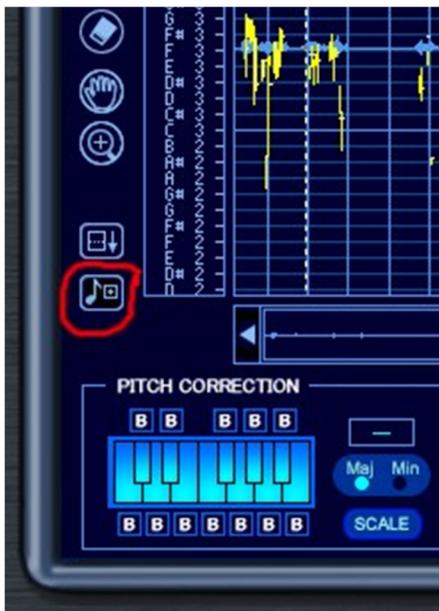
When I think of step sequencers in general, I think of mechanical-sounding, loop-based music. On the other hand, a step sequencer can be a highly efficient way to program rhythmic parts, such as drum beats, trying things out easily while auditioning the part solo or in context of a song section. SONAR 7's integrated step sequencer is designed in a way that gives it the benefits of the easy programming world, while also providing possibilities for minimizing the robotic nature of step-sequenced parts. The key lies both in the flexibility within the step sequencer itself, and in the integration with the rest of SONAR. Whether you start playing a part first, then bring it into the step sequencer for programming, or you program a part from scratch in the step sequencer and enhance it later using other SONAR facilities, all the facilities of SONAR are at your disposal. This could allow, for example, sequencing a Latin percussion part in the step sequencer, then bouncing it to a clip and groove quantizing it against human-played drum loops. For that matter, it could even allow groove quantizing your part against a recording of a live drummer, where SONAR's AudioSnap facility had been used to detect the transients from the audio clips from the live performance, thus allowing the programmed percussion parts to be tightened against the live playing. The result would be almost as if the step-sequenced percussion "players" were following the drummer's lead. In my book, the key word in "integrated step sequencer" is "integrated". That is what truly sets this step sequencer apart as a tool that may even be useful to SONAR users who wouldn't normally consider using a step sequencer. Of course, any polyrhythm and musique concrète fans may be more interested in mixing up unusual loop lengths with note denominations that would cost me too many gray cells to name. I think we can probably all get along.

# "Just Sing the Part"

I don't know how others make up the parts that eventually get into their arrangements, but I often find myself starting out by just singing along with what is already there in the project. Because I'm a keyboard player, quite a bit of the time, my sung parts have a lot more expression in them than an instrument with 12 discrete steps per octave can possibly have. When I then try to replicate the part by playing it on a keyboard, somehow it just doesn't have the same feel as it is missing slides between notes and other little ornaments I may have been singing. Wouldn't it be great if I could just record my singing the part to get the pitches and inflections, then have something figure out what I was actually doing and map it to the relevant MIDI information to drive my softsynth or softsampler of choice?

A related scenario is where I've played and sung a song for a rough piano/vocal demo, and now I want to try and document the song in the form of a lead sheet to get the melody and chord changes down. SONAR will help me out on the piano side, because I can record that as a MIDI part, export it to my notation program, and play around further from there. Sure, it doesn't do a sheet music-worthy job of translating what I play into usable notation, but it at least gets things somewhere in the ballpark, and down in some form to help me remember what I played. I can then either use the original track for reference or as a starting point for creating a cleaned up part. The vocal is just an audio track, though. Even if I were to AutoTune it (or V-Vocalize it?) to robotic precision, I'm out of luck for turning that into notation. Or am I?

SONAR 7 now provides version 1.5 of V-Vocal, which features the byline, "now with pitch to MIDI conversion." That sounds like it could be just what the doctor ordered. In some ways it is, though it's got a long way to go before it ends up being a "cure". Let's take a look.



The basic process couldn't be any simpler. You sing your part in as you normally would, yielding an audio clip. Next you make that clip into a V-Vocal clip and tighten up timing, tuning, and any other aspects as you might if purely using V-Vocal as a corrective tool or for enhancing your part. Once you've got the vocal (or other monophonic pitched) part ready and inside V-Vocal, you need to decide if you are looking for only the notes to be reflected in the extracted MIDI part, or also pitch bend information to get nuances down (i.e. as for the first of my scenarios above). If you do want the pitch bend data included, simply right-click in the main portion of the V-Vocal interface, and pick the Pitch to MIDI Settings dialog from the menu. There you can turn the collection of pitch bend data on, set the pitch bend range (in semitones) and set the timing resolution (in ticks). Once you've accepted your settings, go to the Pitch to MIDI button, drag it, and drop it onto the MIDI track where you want your the detected MIDI clip to go. It's as simple as that!

Well, that would be a dream come true, but, unfortunately, this dream isn't quite coming true just yet. The first problem I encountered was that the MIDI clip neither goes where I dropped it in the target track nor gets positioned in the same location as the source audio data from V-Vocal.

I'm not quite sure what the logic inside this capability is doing, but the part was consistently placed quite a bit further along in the song than where it came from or where I dropped it (which I was targeting to be right at the start of the audio clip). This is with no snap settings active, too. That isn't the end of the world, as it is easy enough to slide the clip around in time. I'd much prefer, however, if the MIDI data were automatically aligned with the position of its source data, at least by default. (NOTE: Cakewalk indicates that this drag and drop positioning issue is fixed in the upcoming 7.0.2 patch update.)

The second issue I encountered is that the length of the MIDI data was considerably shorter than the length of the equivalent data. It is as if it had been somewhat compressed in time. In particular, if I aligned early parts of the clip with the audio data, the later parts of the clip would be rushing or, after awhile, way early.

Placement and time conversion issues aside, how did the pitch detection work? Well, let's put it this way, I'm not going to suggest V-Vocal has anywhere near perfect pitch. It was generally in the ballpark, within a half step or a step. However, the vocal itself was way more in tune than the MIDI part, with or without pitch bend information added to the picture. This was true even in cases where I'd painstakingly, and robotically, tuned the vocal prior to doing the pitch detection. The MIDI notes were also considerably staccato compared to the vocal part. I think that may be a natural side effect of having tuned a vocal with consonant sounds, as any tuning plug-ins I've ever used ignore the consonants, as they are un pitched by nature, thus these may not have been included in the MIDI note durations.

The bottom line? As a starting point for editing, this facility could be useful if what I believe must be bugs with respect to placement and time compression get resolved. I didn't try this capability with other acoustic instruments as I don't actually have any around that I play worth beans, and there really isn't much practical point to trying it with softsynths as I'd be recording the MIDI directly in that case. I'm somewhat surprised in how inaccurate the pitch detection portion of the picture is, even considering I was using a vocal, but I've never had much luck with V-Vocal for automatically tuning vocal parts, either, where, by comparison, I have very good luck with AutoTune in most cases. (While I do use V-Vocal at times, I always do manual correction.) The presence of this capability is a start in the right direction, but, unless it works a lot better with monophonic acoustic or electronic instruments other than MIDI keyboards, the practical use for it at this point is likely to be limited. Hopefully this will be the target of significant improvements in a patch release.

## Rack 'Em Up

Before we leave the MIDI-related topics altogether, it is worth mentioning a few small, but useful, changes to SONAR 7's Synth Rack, a change that users with a hardware synth rack may appreciate, and, last, but certainly not least, several new toys SONAR 7 provides for filling out the Synth Rack.

This may seem like a small thing, but one of my favorite changes is that SONAR 7 allows renaming modules in the Synth Rack. Why am I so jazzed about this feature? Consider that, in a heavily softsynth-based production, which pretty much all of my recordings are, it can get confusing as to which softsynth is playing which role. For example, is a particular string section part being played by Dimension LE, Garritan Personal Orchestra, KONTAKT 2 with its Vienna Strings Library samples, or ...? Or perhaps I'm doubling the part, and may be using more than one module to play that part. More importantly, though, a multi-functional softsynth may be being used for more than one role. In any of these cases, renaming the synth module to something more meaningful in context can help keep things straight.



SONAR 6 added Synth Rack icons, which could be helpful in identifying the role of a synth module quickly when looking at the Synth Rack. Still, the module names themselves would take forms like "Dimension LE 3" (i.e. for the third instance of Dimension LE). That wasn't very helpful when trying to assign a new MIDI track to send its output to the specific instance being used for, say, a cello patch. With SONAR 7, though, I can simply name the relevant instance of Dimension "Cello", and, when I'm trying to select a MIDI port, the device name will show up as "Cello 3" (i.e. the name I gave it, plus the instance number of the softsynth in the rack). Personally, I still like to add my own shorthand codes for the type of synth module (e.g. "DM" for Dimension) to help me know what type of module is being used, especially in cases where I may be using multiple modules to play the same part. However, the functional names make things much quicker when changing input/output assignments or just trying to make sense of routing when signal flow problems arise. Also, even when just looking at the Synth Rack, sometimes the icon itself may not be meaningful enough to let me know quickly what is what. This could be because I don't have an appropriate icon for the part (e.g. I don't have one for a pedal steel, so might substitute a dobro icon to get in the general vicinity of a guitar played with a slide), or it could be because the nature of the part is something that defies unique iconization (e.g. "high synth pad" versus "low synth pad"). The bottom line is, having the flexibility to name

things however best suits your workflow can make things more efficient by making it quicker to figure out what is what, instead of wasting time trying to sort out signal flow.



Have you ever accidentally deleted a softsynth from the Synth Rack, or deleted the wrong synth by mistake? If so, you'll appreciate another seemingly minor enhancement in SONAR 7. Now, when you try to delete a synth from the rack, SONAR asks you to confirm deletion of that specific synth. I know that will save me a bit of "user error" from time to time.

While many users are exclusively or primarily using softsynths nowadays, Cakewalk has also included a feature long requested by users who are still using hardware synthesizers along with MIDI library programs. SONAR now provides the ability to reload instrument definition files in real time, without having to close and restart SONAR. (Users who haven't been around long enough to have used hardware synths may be interested to know that instrument definition files describe a hardware synth to SONAR. This allows users to do things like find a patch by name, rather than just putting in program numbers, make sure bank change handshaking protocols are correct for the synth, map continuous controller numbers to meaningful names and so on. Many softsynths make this kind of information exchange automatic, though enough don't that I've sometimes wished for instrument definition files for softsynths to try and bridge the gap when the softsynth vendors don't go far enough.) To force a reload of the instrument definition files (Master.ins and INSTRMAP.INI), simply press SONAR's Reset (Panic) button.

Even prior to SONAR 7, SONAR provided a fairly rich slate of included virtual instruments, including the Pentagon I analog synth (Producer Edition only), PSYN II subtractive synth (Producer Edition only), RXP REX loop player (Producer Edition only), Session Drummer 2 drum module, Roland TTS-1 General MIDI 2 module, Roland GrooveSynth classic drum machine and synth sound module, SFZ SoundFont sample player, and Cyclone groove sample player. SONAR 7 adds:

- Dimension LE (with Garritan Pocket Orchestra) -- a limited edition version of Cakewalk's Dimension Pro sample-based synthesizer.
- Rapture LE -- a limited edition version of Cakewalk's Rapture electronica-oriented wavetable synthesizer.
- Z3TA+ -- an analog-style synthesizer based on band-limited waveshaping technology.
- DropZone -- a drag and drop sample and REX loop player.

We won't be looking at these instruments in depth, as that could itself take a full review per instrument. However, let's look at some of the highlights of each of the instruments.

## Dimension LE

I first encountered Dimension with Cakewalk's Project5 V2, which included the initial release of Dimension (i.e. before Dimension Pro came out and extended both the product and the library). It is a very powerful product with multiple personalities. In particular, looking at the range of sounds that it can make, which range from realistic acoustic instrument emulation to fairly far out electronic sounds, it begs the question as to whether it is a sample player or a synthesizer. The truth is that it is both.

A powerful sample playback engine lies at the core of Dimension, and that aspect of it makes it possible to use Dimension as you would any sample player. However, Dimension also includes advanced synthesis capabilities, including wavetable synthesis and physical modeling (waveguide) synthesis. This combination of capabilities makes it possible for Dimension to run the gamut between acoustic emulation and analog-style synthesis, while exhibiting particular strength in the territory that lies in between the two ends of the spectrum. At the sound architecture level,

each Dimension sound is comprised of from one to four "elements". An element is a sound generation component that has its own complete engine with a player and per-voice effects.

While Dimension LE is, as its name suggests, a limited edition product, it is capable of playing back any sounds created for its fuller-featured siblings, including Dimension Pro. Some sound tweaking capabilities, and the ability to save your own sound programs, are also provided. While those fall considerably short of full Dimension Pro capabilities, they do nevertheless provide a fair amount of flexibility. At the individual element level, there are lo-fi effects (bit reduction and "decimation" or sample rate reduction), a multimode filter (multiple types of low pass/high pass/band pass filters as well as a comb filter and a "pink" filter, which is a multiple knee filter that darkens the tone of a sound without affecting its sound's character), and a multimode drive control (which includes tube saturation emulation among its five modes). There are also two global effects. The first is a modulation effect, which can be one of two types of choruses, a phaser, or a combined chorus and phaser. The second is a simple reverb. Next up there are mix controls to turn each of the elements on or off, adjust the amount of signal from each element that gets sent into each of the global effects, and control the panning and volume of the element. Finally, there is a master limiter on Dimension LE's output. If that sounds like a fair amount of control, you may be wondering what isn't there. The answer lies in both more advanced sound shaping capabilities (e.g. per-element EQ and modulators), as well as in some more basic sound-generation-specific capabilities (e.g. the ability to transpose each element, define key ranges, etc.). Then there is also the difference in the size of the included sound library.



Speaking of the sound library, Dimension LE includes a sample library of just over 800 MB in size. One of the major features of this library, responsible for over 650 MB of it, is the [Garritan Pocket Orchestra](#), which is a subset of [Garritan Personal Orchestra](#) programmed specifically for Dimension. The subset included here includes solo and section strings, solo brass, single and multi-instrument woodwind ensembles, harps, harpsichord, and orchestral percussion. Beyond the Garritan library, Dimension also includes "dimensions" (see below), pads, real and synth basses, organs, electric pianos, bells, leads, and techno and trance sounds, and a multi-functional collection of electronica-oriented sounds. The collection of bells is particularly nice. Dimensions (i.e. the group of sound programs) are another highlight, and perhaps best exemplify Dimension's nature. These patches tend to be of the highly creative variety, often starting with sampled sounds or a mixture of sampled sounds and analog waves, then using creative programming to create something totally unique. The names are often similarly creative (e.g. "Flourescent Windchimes", "Dimensional Brain 1", "Phantasmal Thought"), and may give little or no clue to the nature of the sounds they make. Then again, playing around with these sounds, which can be inspirational, is one of the true pleasures of playing with Dimension LE.

## Rapture LE



Bringing up Rapture LE after Dimension LE may make it tempting to think of Rapture LE as a superset of Dimension LE. After all, it has a similar element-based architecture, with six elements instead of four, a somewhat similar set of per element DSP processors (but with one extra filter), and an identical program selection interface. The two most obvious differences at the plug-in interface level are the darker color of the graphics, and a middle section called Modulators (see below).

While it is true that Dimension and Rapture do have some commonality in their architecture, there is a very basic low level difference at the sound generation level, in the form of an unassuming-looking parameter called "Multi", which is visible, but not editable, in the LE version of Rapture. This parameter turns the currently loaded wavetable into multiple oscillators, with different detunings for each voice. Besides just having the one voice, like a Dimension element would have, Rapture elements can have 3, 5, 7, or 9 voices in a single element. Multiply that times up to six elements, and we're talking PHAT!!! While you can't edit this parameter in Rapture LE, you can take advantage of

the sound design capabilities it affords since Rapture LE can load any sound programs designed in the full version of Rapture.

Beyond the sound generation difference, Rapture also offers more complex modulation matrix capabilities, all with time signature synchronization possibilities, at the individual element level. Most of these capabilities are also not accessible for editing in Rapture LE. Nevertheless, these underlying features provide particular aptitude for creating evolving sounds that synchronize with the beat of the music, such as would be common in electronic dance music, and many of the included patches take advantage of these capabilities. Rapture is also capable of making more straightforward analog synth sounds, but what sets it apart is its modulation matrix capabilities.

While most of Rapture's modulation matrix capabilities are not available for editing in Rapture LE, Rapture LE does include one important subset of these in the form of seven step generators per element. These step generators work similar to an LFO to modulate pitch, the cutoff frequency and resonance parameters of both filters, panning of the element, and amplitude of the element. Unlike an LFO, you define the shape of the modulation by drawing the steps -- from 2 to 128 of them for each of the step generators. The steps can either be free running, at a fixed frequency, or synchronized to the beat at various note values. For example, a step generator to modulate pitch could be used to generate sample and hold effects or, with less depth, something more akin to vibrato or repeating pitch envelopes.

Rapture LE also makes the per-element DSP controls accessible, including the control that allows determining the order of the processors in the DSP chain. The per-element DSP processors are the same as offered in Dimension LE except that there is one extra filter. Rapture LE does not make Rapture's global effects modules available, but I find this to be a minor limitation since it is easy enough to follow Rapture LE up with whatever plug-in processors you might have available.

On the sound library front, Rapture LE provides approximately 200 sound programs, including a variety of sequenced sounds, basses, leads, pads, keys, and "simulations", along with teaser collections from two Rapture sound library expansion packs. To my ears, even when Rapture is nominally simulating another instrument, it still feels like a synth. That is not a dig on Rapture, but rather more a note on its basic characteristics, which are much more geared toward electronic sounds than emulating acoustic instruments. Rapture's nature also makes Rapture LE a bit more of a tease than Dimension LE. In particular, with an acoustic instrument emulation, we tend to get used to playing technique's being one of the biggest differences from song to song and part to part. With synth sounds, though, especially motion-based sounds like those at which Rapture excels, the programming tends to take center stage, and there is more likely to be a temptation to tweak the programming to fit the song. While Rapture LE does still provide a fair potential for tweaking, I imagine most users who make the kind of music Rapture ideally suits will be wanting to upgrade to the full version when they can.

## Z3TA+

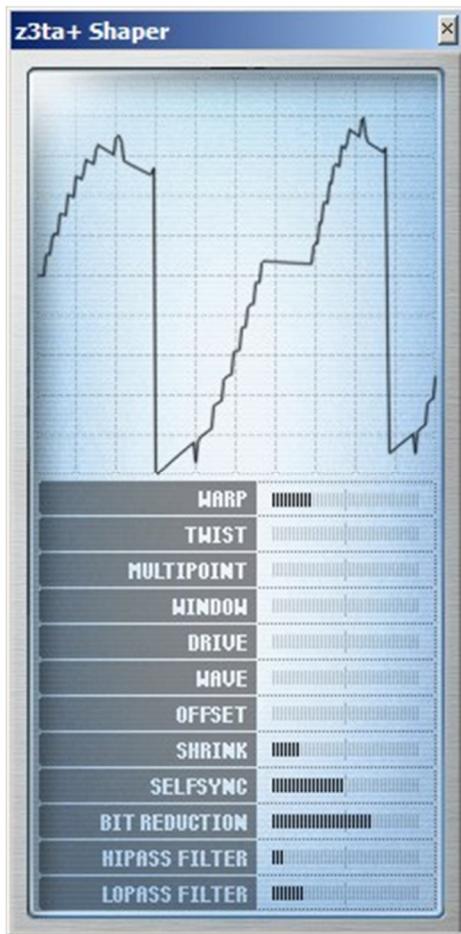
Unlike Dimension LE and Rapture LE, Z3TA+ (pronounced "zeta"), which is included in SONAR Producer Edition only (i.e. not in SONAR Studio Edition), is a full version. Designed and developed by Cakewalk's Rene Ceballos prior to his Cakewalk tenure, and thus still bearing the rgc:audio branding, Z3TA+ could rightly be considered an ancestor of both Dimension and Rapture. Though it is a different animal than either, and its user interface looks considerably different than Dimension and Rapture do from each other, there are also some architectural similarities such as the multiple sound generators and sophisticated modulation matrix capabilities.



Z3TA+ starts out with up to six "oscillators". However, we aren't talking just sawtooth waves, square waves, sine waves, and triangle waves. There are plenty of varieties of those (e.g. 11 varieties of sawtooth waves including 4 vintage saws, 2 octaved saws, a saw mixed with a sine octave, and other saw flavors with names like "2-bit saw", "noisy saw", "multisaw", and "half saw"). These oscillators are based on wavetables, though, and there are also

additional waveforms that you wouldn't be likely to find in an analog synthesizer. Some sample category names include "Baz It" (huh?), "Disto" (distortion), "Additive", "Instruments" (e.g. organs, piano, fret, voice), and "Others" (e.g. "FMish", "Noisy", "Very Noisy", "Who Knows"). Keep in mind, each of the six oscillators can have different wave types. There are on the order of 60 raw waveforms in all, and PWM (pulse width modulation) is available with any of those.

Each oscillator also has its own independent waveshaper, which essentially takes the basic waveform and messes with it in various programmatic ways. Transformations available include warp, twist, multipoint, window, drive, wave, offset, shrink, selfsync, bit reduction, hi-pass filter, and low-pass filter, and all can be used individually or in combination with any or all of the others. Check out the screenshot at left to get a feel for the potential in this area. This mangled waveform started life as a vintage sawtooth wave. On caveat: There is no "good taste" constraint, so proceed at your own risk.



Like Rapture, Z3TA+ has a Multi mode that allows stacking oscillators, 8 individual oscillators (four stereo oscillators) in this case, with adjustable detuning, for going into phatville. In addition to being able to mix the oscillators together, Z3TA+ allows its oscillators to inter-modulate each other using a circular inter-modulation scheme where oscillator 1 can modulate oscillator 2, which can modulate oscillator 3, and so on down the line through oscillator 6, which can modulate oscillator 1. Any oscillator can modulate the next oscillator in line in one of four ways: ring modulation, hard sync modulation (next oscillator in line restarts its phase every time the current oscillator restarts), phase modulation, and frequency modulation. Alternately, an oscillator's output can be sent to the mixing bus to be added with the output of other oscillators.

Ultimately, the oscillator outputs go to two mixing buses, which then go through a filter stage. There are two filters, which can be used in parallel, one off each of the mixing buses, or used serially such that the second filter filters the output of the first filter. Filter types include various slopes of low-pass, high-pass, band-pass, and band-reject filters, along with a formant filter, which is used to emulate vowel sounds.

Z3TA+'s also includes an effects processing section, which is controlled from a separate page of its user interface. Effects modules available include distortion, modulation (e.g. flanger, phaser, chorus), three separate delay modules, reverb, an equalizer, and a amp/speaker/special equalizer device simulator.

Alongside all this, there is an up to sixteen-element modulation matrix, which pretty much allows modulating anything from anything. For example, you can modulate oscillator pitch and level, oscillator pulse width, filter cutoff and resonance, and various effects parameters, among other parameters. You can modulate these via oscillators, envelope generators, LFOs, an X-Y pad, and more.

We're just touching on Z3TA+'s capabilities, but I hope it is clear that there is a huge amount of power in this synth. To get you started Z3TA+ includes 768 sound programs (6 banks of 128 programs apiece). These cross many different functions (e.g. basses, pads, leads, etc.), but most all tend toward the synthetic-sounding, with many taking advantage of Z3TA+'s rich modulation capabilities to create evolving sounds. It's tough to characterize a whole batch of disparate sounds, but the best comparison I could make between Z3TA+ and Rapture is that Rapture may sound a bit more analog, in the sense of analog warmth, and is likely to take up more space in a mix, all other things being equal. By contrast, Z3TA+ sounds more digital, but may be easier to fit into a mix in a supporting role to more

prominent sounds. Neither is likely to be mistaken for an acoustic instrument emulator. (Cakewalk reminds me that Z3TA+ can also be dropped in any FX bin to use its internal effects engine for processing audio.)

## DropZone

DropZone is a simple drag and drop sampler that supports WAV, AIFF, Ogg, REX, and SFZ formats. Samples can be looped or played as one shots. REX loops can either be played as loops or played a slice at a time.

Like Dimension and Rapture, DropZone's architecture mixes multiple elements together to get a composite sound. In this case there are only two elements, each of which can be a wavetable, sample, multisample, or REX groove. Keyboard and velocity range parameters make it possible for velocity switching and keyboard mapping to determine when each of the elements is played. A multimode filter is available for each element, as are envelope generators and LFOs to modulate pitch, amplitude, and the cutoff frequency of the filter. There are also coarse and fine tuning controls to control each element's pitch in increments of half steps or cents, respectively. There is also a keyboard tracking parameter to set the pitch difference between MIDI notes in cents, thus providing the potential for fixed interval microtuning.



DropZone also allows getting into simple sample looping, for example if you've dragged a pitched sample into one of the elements, and would like to create a sustaining section of the sound if you hold the note long enough. Simply drag the left and right endpoints of the loop visually against the wave diagram. Loop direction can also be varied between left to right, right to left, and back and forth.

DropZone's capabilities may not give high-end samplers a run for their money, but DropZone does provide a basic sample manipulation workbench that can be fun to use, and utilitarian for basic sample manipulation needs. It also provides a basic REX loop player, along with some sound manipulation capabilities to mangle those loops. I can imagine it could be a fun tool for use in dance music or other contexts in which rhythmic sound effects would be useful, though its ability to play SFZ multisamples also allows it to do duty as a more general purpose sample player. DropZone also includes a just under 300 MB library of sound programs, REX2 loops, wavetables, and multisamples.

## Sidechaining

Moving over to the audio front, the addition of internal sidechaining is big news. In the hardware world, compressors, gates, and some other types of processors often support something called a key input. What the key input allows is basing the processing of the hardware module on something other than the signal it is processing. For example, in the case of voiceover work, it may be useful to duck the music tracks when the voiceover artist is speaking, while bringing them back up to level in between phrases. In this case, using the voiceover artist's voice to tell the compressor when to bring down the music track will do the trick, so the voice becomes the key input. Another classic application for a sidechain compressor is gating a synth pad based on a highly rhythmic track, such as a hi-hat, to get that rhythmic gated synth effect that is so popular in dance music.

In the hardware world, routing signals for the voiceover application is fairly straightforward. Simply patch a cable from a send on the mixer's voiceover channel to the key input on the compressor. In the DAW software world, however, where plug-in processors are the rule, you need two things to do this. First, you need a plug-in that supports an extra key input, instead of only the input for the audio it will be processing. Once you have that, you need a way to route audio to the plug-in's key input (which is also often called a sidechain input). SONAR 7 covers both bases, via three newly sidechainable plug-ins and basic routing enhancements to send audio to plug-in sidechain inputs. Various third party plug-in vendors also provide plug-ins with sidechain inputs.



When you insert a plug-in that supports sidechaining in SONAR 7, SONAR automatically creates a new bus that is available for assignment almost anywhere in SONAR. I say "almost anywhere" because SONAR's normal rules with respect to where specific buses can be assigned apply. In particular, the bus won't be available anywhere a signal routing loop could be created. Beyond that, though, it is available for use in track outputs, bus outputs, and for insertion as a send bus from tracks and buses. The name of the sidechain bus reflected at the point where it can be assigned reflects both the name of the plug-in and the name of the track or bus in which it is inserted. In most cases you will want to use a plug-in's sidechain input as the target for a send so that the signal driving it still goes to its normal destination. In many cases, you will also want to configure that send to be pre-fader to keep a relatively constant level feeding the plug-in's key input despite any fader

automation in the recording. Just as with any other bus, you can assign multiple source signals to the sidechain bus, for example if you want multiple narrators to be able to duck the music bed behind their narration. Once you've got that routing set up, the rest of the action happens on the plug-in's end, and what happens beyond that point is up to the specific plug-in receiving the sidechain input.

SONAR 7 includes three plug-ins that support sidechain inputs: Sonitus:fx Compressor, Sonitus:fx Gate, and VC-64 Vintage Channel (Producer Edition only). In the case of the two Sonitus:fx plug-ins, you won't see anything different than you've ever seen inside the plug-in itself, at least with respect to the plug-ins' controls. If a sidechain signal is present, it is automatically used as the key input for the applicable plug-in. The only visible difference inside these plug-ins is the readings you'll get on the input meter. In particular, the signal you see there is from the sidechain, not the signal actually being compressed by the plug-in. Beyond that, the rest of the action happens under the hood.

With VC-64, and also many third party plug-ins, the plug-in interface itself provides an interface to explicitly select between the key, or sidechain, input and the plug-in's source audio input. VC-64 allows directing the sidechain input to either or both of the compressors and/or to either or both of its equalizers. There is also a Listen button, located in the compressor section, to allow listening to the sidechain input, rather than the output of VC-64. This may be useful for setting the controls that do their processing based on the sidechain signal. Applications for sidechaining compressors with VC-64 are similar to those for Sonitus:fx Compressor, except that there are two compressors available, and you have the flexibility of sidechaining either or both. Sidechaining equalizers, on the other hand, seems a bit odd. Unlike with compressors or gates, where the sidechain is simply used to control the clamping down of the gain reduction circuitry, an equalizer doesn't actually have a key-type function. Rather it is processing the signal fed to it, or the sidechain signal when it's Key switch is enabled. Depending on the configuration of VC-64's signal flow, and which of the equalizer circuits is being fed a sidechain input, this may mean hearing either only the sidechain signal (processed with any EQ settings) or a blend of the audio signal being fed directly into VC-64 and the sidechain signal. There is one configuration, however, "Selective w. SC-Filter", where one of the equalizers doesn't feed VC-64's output, but, rather, can be used to process the signal that is being used to as a key in one of the compressors. Thus, that specific EQ's being fed from the sidechain could be used to exaggerate aspects of the sidechain input signal.

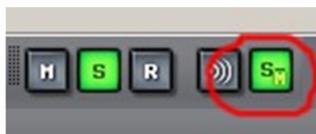


I found the SONAR's sidechaining worked as I would expect in most cases. However, I occasionally had problems after setting up a sidechain bus for the first time in a given project, where a static-like noise would be produced and mixed with the signal, or where a loud burst of noise that would send the meters up to some high number like +9dB would be produced then the audio engine would drop out and be unwilling to restart. When these issues occurred, saving the project, then closing and reopening the project, would generally be sufficient to get things working. This

happened both with third party plug-ins, such as PSP MasterComp, and the Sonitus:fx plug-ins. I could not peg down any specific pattern of when such issues did and did not happen, though it is possible the issues may have been more frequent when remixing a project that had initially been started in a version of SONAR prior to SONAR 7. (Most of the real life projects I worked on during the course of this review were remixes.) It is also important to note that there are some non-obvious considerations that need to be taken into account when bouncing or freezing tracks using sidechain inputs, and there are also ramifications on what appears when soloing tracks within projects using sidechaining. Those are beyond the scope of the review, but worth keeping in mind when beginning to use sidechaining in your projects. All in all, sidechaining is a powerful new feature, and Cakewalk has made configuring sidechain connections quite straightforward.

## Mixing and Playing on the Bus

SONAR 7 includes a number of enhancements that fall under the general category of improving productivity in, or making other enhancements to, the general experience of playback, mixing, and just general use of the audio side of SONAR. If you don't mind, I'm going to start with my personal favorite from this category



A new Dim Solo feature allows highlighting tracks, like Solo does, but without losing sight of the big picture. When Dim Solo mode is engaged and you solo one or more tracks, instead of having all other tracks be muted, those other tracks are attenuated, or "dimmed", by either 6, 12, or 18 dB. The specific attenuation level is a user preference set in SONAR's Audio Options. This is a wonderful feature for doing

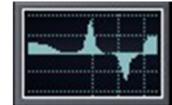
the kind of detailed work that we often do using Solo. For example when comping a part or tuning a vocal, it helps avoid losing touch with how the track we're focusing on fits into the mix, especially in terms of any subtleties of how the track may feel against the rhythms of the mix. I do have two "wishes" with respect to this feature, though. First, I would have preferred if the amount were completely variable instead of just having the three options. There were times when I would have liked even more than 18 dB of attenuation, as well as times I would have liked the attenuation amount to fall somewhere in between these three values (which correspond to one half, one quarter, and one eighth, respectively, of the perceived volume of the original signal). Secondly, I would have preferred if the level settings were available on a toolbar instead of having to go into Audio Options every time I want to change the attenuation amount. These are settings that are likely to get changed fairly frequently, due to working in different contexts at different points in a project, not "set and forget" settings like most of the values in the Audio Options.

SONAR 7 now provides right-click menu shortcuts for assigning multiple track inputs or outputs, or all bus outputs, in a single operation. For example, let's say you've just created a number of new audio tracks for overdubbing background vocal parts. You'll be tracking each harmony part to a single track, with multiple takes per part for later comping, then moving on to the next part on a second track, and so on through four harmony parts. Of course, you could use a track template to set this up if you've planned ahead, but let's say you didn't, and you just created four audio tracks for this use. Now you'll want to assign the same inputs and outputs to all four parts. In SONAR 6 and earlier, you could select the four tracks, go up to the Track menu, then pick the Input command, assign the relevant inputs, then go back to the Track menu, pick the Output command, and assign the relevant outputs. That's not too horrible, mind you, but having to go up to a menu way up at the top of the screen seems, well, a bit too Mac-like for us Windows users. With SONAR 7, you can just pull down the input box for one of the tracks, select "Selected Track Inputs" off the pop-up menu, then pick the port you want to assign to all the tracks. You can do a similar thing with pulling down the output control for one of the tracks and picking the "Selected Track Outputs" command from the pop-up menu, where you can assign the tracks to either an output port or a bus. A related scenario is if you don't want to assign all inputs to a single port, but, rather, a series of ports, for example as might happen if you were assigning all the tracks for a multi-miked set of drums or a group of background singers who each had individual mics. In that case, instead of picking "Selected Track Inputs" of the pop-up menu for the input control on one of the tracks, you would pick "Selected Track Input Series", then pick the first port in the series of ports you want to assign. SONAR would then assign one track per port with port order corresponding to track order. That represents an even a bigger time savings than the other scenarios in that you'd previously have had to assign each port individually, going through the initial pop-up menu and one sub-menu to assign each port. A similar shortcut is available for assigning buses to a single output port. That one doesn't seem quite as useful because the odds of wanting to assign all buses to a single output port seem relatively slim. The reason for the limitation there is that you

can't actually select buses. If that limitation is ever removed, it would definitely be useful to be able to select several buses then assign them all to the same bus or output port (e.g. assigning all submix buses to the main mix bus).

Here's one that may sound a bit strange to relatively new SONAR users, but I'm guessing that most long-time users will be wondering why it took so long to arrive. SONAR can now play back with no data. I expect those who haven't yet encountered the need for this are scratching their heads, thinking those of us who are standing up and applauding are out of our minds. Here's just one simple scenario, though, that will demonstrate the need for this feature: It's time to set tempo for a project, where you'll be tracking the project to a metronome (i.e. no ACID loops, MIDI loops, etc.). You want to hear the tempo so you can see if it feels right, but you don't want to start recording yet. The obvious solution might seem to be to turn SONAR's audio metronome on, with it set to sound on playback, but the problem is that playback wouldn't engage in SONAR 6 and earlier versions of SONAR, because there was nothing to play. SONAR 7 solves this, and other related needs (e.g. triggering playback on external devices, continuing playback beyond the end of a project to allow all MIDI notes and their sustains to decay naturally, etc.), by providing a global option as to whether to stop playback at the end of the project. If you leave the option unchecked (it is checked by default to mimic the behavior of earlier versions of SONAR), hitting the end of the project will not stop playback, and the project would go on playing, possibly nothing, forever. You can also start playback when there is absolutely no data in the project, thus allowing the metronome to sound without a recording and making it easy for you to hear the tempo and adjust settings before starting to record. This may seem like a minor thing, but the need for it tends to strike on most every project for people who start recording to a click.

Here's another simple, but common scenario. You're using SONAR's built-in Sonitus:fx channel strip EQ on multiple background vocal tracks, and you'd like the same settings on each of those tracks. Prior to SONAR 7 you would have to recreate the settings on each track, unless you'd known ahead of time what settings you were going to use so the tracks could have been created from track templates. Or, alternately, you could have used the Sonitus:fx Equalizer as a plug-in, then Ctrl-dragged it from FX bin to FX bin to copy it. In SONAR 7 you can simply go to the Console View, pick the track whose EQ settings you want to copy, then Ctrl-drag its EQ graph from one track to another. This is very quick to do, even when repeating it for a number of tracks, and no plug-in interface gets opened up like in the case with dragging plug-ins between FX bins.

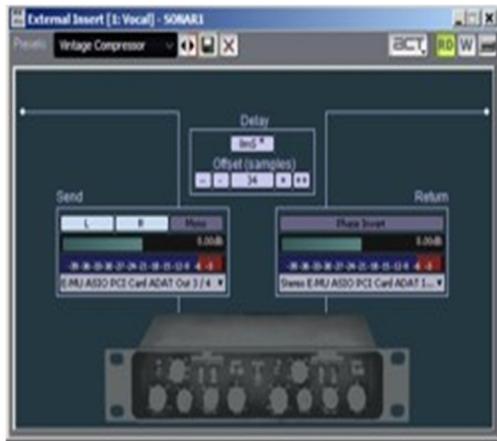


Anyone who's experienced glitches or slowdowns while playing back projects that update the screen rapidly will be glad to hear that Cakewalk has implemented a new CPU conservation mode. This mode dramatically reduces the number of updates to SONAR's graphical user interface, leaving more power to be dedicated to playback performance. To invoke this new mode, simply press the Pause key on your computer keyboard. This key works as a toggle to turn CPU conservation mode on and off. When conservation mode is turned on, SONAR's user interface is only updated once per second. This will obviously affect the visual accuracy of meters, the Now Time pointer, and other real time indicators. However, in practice, if you need the extra power, this is a fair tradeoff, and once-a-second updates are still helpful for keeping a general pulse on your progress through the project. There is no visual indicator to indicate the mode is active, but if your screen is only updating once a second, that's a pretty good clue, and simply toggling the Pause key will let you know for sure. This mode can be very useful when working with GUI-intensive plug-ins, as long as regular meter updates aren't critical at that point. It could be particularly useful in cases where you need to use real time mixing (see below), where there is no option for slower-than real-time rendering of a part.

## Hardware Plug-ins?

I used to have a rack full of hardware synthesizers and effects doing the sort of things I accomplish with plug-ins. It didn't take me very long after getting my feet wet in the software world to switch from being a "gear slut" to being a "plug-in junkie". It also didn't take me long to decide to get rid of pretty much all my hardware, as I much preferred working in my small studio without all the heat those pieces of gear generated. Not everyone prefers the software-only world to the hardware world, though. Even among those who do prefer the software world, there may be one or two hardware processors they can't live without, be it a vintage compressor, a stomp box with a unique character, or even just running through an 8-track tape loop to use "tape as an effect".

While SONAR's bus structure has long made it possible to use hardware processors in conjunction with SONAR projects, doing so was not always straightforward. For example, say what you really wanted to do was incorporate a hardware compressor somewhere in the middle of your vocal signal chain, with the rest of the elements of the signal chain being plug-ins. You would have had to artificially split things up, sending the first part of the signal chain out to a hardware output routed to the hardware compressor. You'd then route the output of the hardware compressor into a new SONAR track's input, where you could input monitor the signal to get settings. Eventually you'd need to record the output of the hardware compressor to a track so you could mix it down with the rest of the SONAR project. That new track would then get the rest of the plug-in signal chain put in its FX bin, and finally the track would be routed as necessary for the rest of the mix.



SONAR 7 has made a major enhancement in this area, making it possible to treat hardware effects as plug-ins, at least to a degree. First, you can insert the module right at the point in a track or bus FX bin where you want it to be. When you do that, you'll get an External Insert plug-in header. From that plug-in header's user interface (see the screen shot at left), you can pick the send and return hardware outputs you are using for connecting your piece of gear. You can send left channel only, right channel only, stereo, or mono (i.e. sum of left and right channels) signals as needed. The facility also provides the ability to calculate the amount of delay compensation needed, though you can offset that manually if you find that the calculation doesn't quite get the right result in practice. You can also offset the send and return levels, if necessary, and flip polarity on the external return.

Once you've got things configured, you can audition sounds in context just by playing back your project as you normally would while manipulating the controls on your piece of gear. When it comes time to freeze the track or do a mixdown, you will need to take advantage of the new Real-Time Bounce feature (see below) so as to account for the real-time nature of the hardware (just one of its limitations versus software plug-ins). If you'll be using the same setup frequently, you can save a preset with all the applicable parameters configured.

I mentioned it was possible to treat the hardware processor as a plug-in "to a degree". There are some limitations, of course. First and foremost, you only get one instance at a time if you've only got one piece of hardware (perhaps the biggest reason those of us on budgets tend to go down the software route). Secondly, I've already mentioned the need for a real-time bounce. The third key one that comes to mind is that you won't be able to save the knob settings of your hardware to restore next time you come to work on your project. If you happen to have a piece of gear that has MIDI control, you may be able to manage its settings in your project that way, but, face it, if you've got a vintage compressor, the odds of its having MIDI control are zilch. Thus, it is probably a good idea to freeze your track as soon as possible, thus freeing up the hardware processor for its next use and saving you the agony of having to restore settings from a piece of paper or your memory.

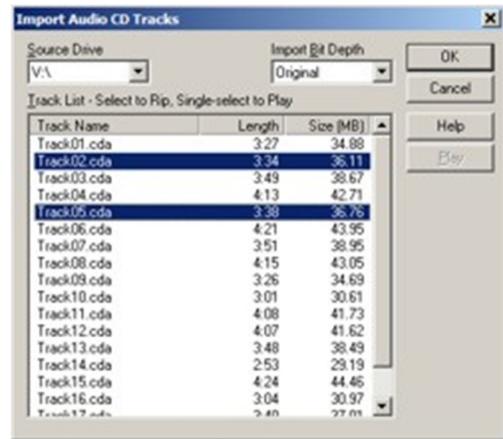
Unfortunately, I did not get a chance to try out this feature with real hardware, since I don't actually use hardware processors in my studio these days. The setup seems quite elegant, though, and a big improvement over the way things would have needed to be done in the past.

## Are You Coming or Going?

SONAR 7 also introduces a number of enhancements I might loosely categorize as being related to getting audio in and out of SONAR or, in some cases, removing former limitations on SONAR's ability to record a lot of audio. Let's take a look at these.

If you've ever needed to import sound effects from audio CDs, been asked to record a singer armed with karaoke tracks, or gotten involved in making soundalike recordings, you have probably wished for this first enhancement. SONAR now lets you rip one or more tracks from audio CDs directly from its File Import menu. The Import dialog

box allows provides preview capability, but note that this capability uses your computer's default sound card as configured in Windows settings. You have the choice of importing the tracks at their original bit depth or any other bit-depth supported by SONAR, and multiple tracks can be imported in one go. If you do import multiple tracks, they'll be placed in separate, successive tracks in your project. Placement of imported tracks can be a bit quirky. SONAR tries to use any available audio tracks first, inserting new tracks only if it runs into a MIDI track, or off the end of a project, before it finishes with all the tracks it needs to import. If you happen to have a softsynth's audio track located after the track position where you want to import one of the tracks -- for instance if you have one audio track and then a softsynth audio track, and you are importing two or three audio tracks -- the softsynth audio track will become the target of one of the tracks imported from the CD. It seems to me that it would have been more desirable for softsynth audio tracks to be treated as special for these purposes. By contrast, if a MIDI track appears before all audio tracks are imported, SONAR inserts additional audio tracks prior to the MIDI track in order to finish the import operation. Any new tracks inserted begin at the Now Time in your project, which is highly desirable when placing sound effects.



SONAR 7 adds a right click menu item for importing audio or MIDI data, depending on the type of track at the cursor position. This is another one of those little convenience features that will save a bit of time here and there.

When you are importing audio data, the Import dialog box now provides the ability to select a preview bus from among any of the buses existing in SONAR at that point. This is helpful since it means you can hear the audio you are auditioning through the same speakers you'd be using with your project. It's too bad that same capability wasn't extended to the CD import preview. It also would have been nice to do something similar for MIDI, though that would obviously be a little trickier in that a softsynth would have to be specified rather than a bus. You can add both to my wish list for the next version of SONAR.

SONAR 7 supports a number of new file formats for importing an exporting audio data. New import formats include Sony Wave-64 (.w64), Apple AIFF (.aif, .aifc, .aiff), NeXT/Sun (.au, .snd), FLAC (.flac), Sound Designer II (.sd2), and Apple Core Audio Format (.caf). New export file formats include Sony Wave-64 (.w64), Apple AIFF (.aif), NeXT/Sun (.au), FLAC (.flac), Sound Designer II (.sd2), Apple Core Audio Format (.caf), and RAW (.raw).

Speaking of the Sony Wave-64 format, SONAR can now also use that format for its internal files, which it will do automatically if it detects file size will need to exceed the 2 GB limitation of RIFF WAVE (.wav) format files. This is especially important at high sample rates where, for example, a 64-bit, 192 kHz file would be limited to 11.65 minutes in the RIFF WAVE format. The new limitation for this sample rate in the Sony Wave-64 format, which can be as large as 8,388,608 terabytes (good luck finding a disk big enough to hold one of those!), is 833,999,931 days. If I've calculated correctly, that is significantly upwards of 2,000 millennia! Somehow I think that is more than sufficient for any project any human being is ever likely to produce. Do we have any immortal beings in the house interested in testing that limitation? Of course, these extremes may be beyond the bounds of practical comprehension, but it's not so difficult to imagine a live concert recording or film sound project's exceeding the limitations of the RIFF WAVE format, especially once you get into using high sample rates.

The notion of supporting longer projects also required changes to the SONAR timeline, which has now been upgraded to 64, rather than 32, bits. SONAR 6's limitations on maximum project time ranged from 13.52 hours at 44.1 kHz down to 3.10 hours at 192 kHz due to the 32-bit timeline. There was also a limitation of no more than 9,999 measures. With SONAR 7, these limits are expanded to upwards of 2 billion days at 44.1 kHz and almost 556 million days at 192 kHz, and up to 559,240 measures.

When it comes time to bounce a track in SONAR, be it for exporting to an external audio file or rendering a submix for further processing, we generally want the process to go as fast as SONAR and our computer hardware will let it. However, there are a few circumstances in which SONAR's "Fast Bounce" option will not yield the sonic results we

desire. One key one is in the case of using external inserts, such as that vintage compressor I mentioned when we were talking about using hardware like plug-ins. Some other common ones are when rendering certain disk streaming sample players, where the samples streaming from disk may not be able to keep up with SONAR's bounce speed and something is amiss in the handshaking between SONAR and the plug-in that should tell SONAR to wait until the plug-in is ready. Similarly, there are also some plug-ins, even ones that don't do disk streaming, that simply don't render properly, for whatever reason, with a Fast Bounce.

SONAR has long provided the ability to turn Fast Bounce off, which resulted in real time bouncing. In SONAR 7, though, when Fast Bounce is turned off, a new option called Audible Bounce becomes available. In an Audible Bounce, you will actually hear the relevant tracks playing back as the real-time bounce occurs. This allows, for example, handling of external inserts, since those hardware processors need the audio information they are intended to process to yield their desired results. These enhancements appear not only in the Export Audio mixing dialog, but also in the dialogs for bouncing to tracks, applying audio effects, exporting a video mix, and specifying options for freezing tracks.

SONAR 7 also redefines one particular aspect of exporting tracks when nothing is selected to specify what is to be exported. In earlier versions of SONAR, SONAR assumed it should bounce the entire duration of the project, which would generally be a good choice if exporting the entire mix. However, it isn't as helpful a choice when exporting tracks since it would waste a lot of space making all tracks last the entire length of the project. SONAR 7 changes this behavior such that, when nothing is selected, and the export source category is set to Tracks, each audio file will have the same duration as its original source track. In some cases, this might cause complications in aligning files, for example if you are creating those track exports specifically to send to someone using a system other than SONAR (or even an earlier version of SONAR that can't handle a SONAR 7 project file) for mixing. In particular, if the files do not begin at the beginning of the project, the recipient may not know how they should be aligned. There are two options here. One is to make sure all tracks do begin at the beginning of the project, or at least at the same place. The other is to export the files in Broadcast Wave format to ensure their timestamps are saved with the files. In the latter case, as long as the recipient's system supports placing Broadcast Wave files at their original timestamps, the files will get aligned properly without wasting extra space to extend tracks on the front end.

## Mastering Plug-ins

Softsynth users aren't the only ones who get some new toys in SONAR 7. SONAR 7 also provides some new audio plug-ins, aimed squarely at people who are mastering their own material, or perhaps doing mastering work for others. All SONAR users get the new Boost 11 Peak Limiter, while SONAR Producer Edition users also get two new linear phase processors in the form of the LP-64 EQ and the LP-64 Multiband Compressor.



Cakewalk's new Boost 11 Peak Limiter is designed for transparent peak limiting and volume maximization at the final stage of mastering. The idea is to try and make the track sound as loud as you can, preferably within the bounds of good taste, while still keeping the signal from exceeding the maximum output level you specify (e.g. for purposes of preparing a file for inclusion on an audio CD). The plug-in is about as easy to use as any plug-in could be. Simply set the Output control to the maximum level you want to allow, then adjust the Boost control to achieve the loudness you desire. Scrolling displays of the waveform coming into and going out of Boost 11 provide helpful visual feedback, especially with respect to the degree of peak limiting that is going on with respect to the input waveform. Behind the scenes, Boost 11 is using look-ahead peak detection in order to reduce the gain just before a peak. It is also using program-dependent release times in order to smoothly restore the gain just after a peak while trying to avoid distortion or pumping that could occur from use of released times that are too fast or too slow, respectively.

That all sounds great in theory, but how does it work? Basically, it just does. Like most volume maximizers of this sort, you have to be careful not to push it too hard, or you get into square wave territory, with harsh-sounding mixes.

One niggle here is that Boost 11's Boost control is very easy to turn too far by default. It goes from no boost to 24 dB of boost. Though you can use the shift key in conjunction with turning the dial with your mouse to get resolution down to 0.01 dB, it seems like it would have been even more helpful to just reduce the maximum amount of boost allowed and provide finer resolution without needing a modifier key. Does anyone really need 24 dB of boost? That is making your mix 16 times its original apparent loudness! I'm thinking if you're going much over 6 dB either your mix was way too soft to begin with, in which case it would make sense to boost it somewhat prior to hitting Boost 11, or you're creating a hypercompressed mix that is going to be painful to listen to for more than a short while. Similarly, the gain reduction meter could do with more resolution for getting a better picture of what is going on when doing more reasonable degrees of peak limiting and volume boosting. I also wouldn't mind having some additional scale markings on the input and output meters, or, for that matter, if the waveform displays were significantly larger. Those are niggles, though, and the bottom line is Boost 11 does what it's supposed to do, and what it's supposed to do is something most everyone needs at one point or another.

Cakewalk's new LP-64 Equalizer and LP-64 Multiband Compressor are both of the linear phase variety. That is, they are designed to avoid the per-band, frequency-dependent phase shift that occurs in traditional processors when the audio signal is split out into frequency bands, processed, then recombined. These phase shifts can color the sound, introducing a subtle loss of clarity or focus due to the smearing of the signal that occurs. By contrast, linear phase processors maintain consistent phase shifts throughout the spectrum, avoiding this smearing, and then SONAR's automatic plug-in latency compensation eliminates the phase shift altogether in the final result. The result is applying the processes the specific processor is intended to apply without the side effects of coloration caused by frequency-dependent phase shifting. Both plug-ins also provide look-ahead peak detection to prevent clipping and Program Dependent Release (PDR) to minimize pumping and distortion.

LP-64 EQ is an up-to-20-band equalizer featuring a graph-based user interface, which Cakewalk calls the Heads-Up Display (HUD). EQ settings are entered primarily by manipulating points on the graphical curve of the HUD. Creating a new band of parametric EQ is as simple as double clicking on the curve at the point you want to create the band, then dragging the new control point to the location where you want it to be in terms of frequency and gain/attenuation.



Hovering over any control point displays its numerical values if they aren't already displayed based on Show/Hide button settings for the relevant parameters. Depending on the mode of operation, you can either manipulate Q directly in the Normal mode of the plug-in, or indirectly based on spacing and position of control points in the Curve mode of the plug-in.

The user interface for the Normal mode of LP-64 EQ operates very similarly to most other equalizers that provide a graphic interface, where Q is an independent parameter that can be entered numerically or adjusted graphically. In the latter case, this involves displaying Q via a toggle button, which then shows both the numerical value and a separate node for adjusting Q on the HUD. That special node can then just be dragged up or down to adjust Q. In LP-64 EQ's Curve mode, there is no independent control over Q. If you want to make a steeper curve around a particular area of the graph, you simply put additional control points very close to it on one or both sides. The idea here is that you are directly shaping the curve, rather than adjusting a mathematical value that shapes the curve. In practice, shaping the EQ curve in Curve mode tends to be more intuitive, but, if you need to replicate specific values, for example to compare to settings on another EQ plug-in, it is easier to get precise Q settings in the Normal mode.

It is worth noting that, though the HUD defaults to a vertical range of -24 dB to 24 dB of gain, clicking on the dB scale and dragging it allows zooming in, with the narrowest possible range of display being -6 dB to 6 dB of gain -- a much more useful scale for mastering purposes. Even when the scale is set to its narrowest range, however, it is possible to create a deep notch cut by dragging a node down to a negative infinity "trough" area at the extreme lower end of the gain scale.

Beyond the HUD, there are also output meters with numerical peak hold values, and an Output knob for boosting or attenuating the signal as needed. Styling of the meters and knob is consistent with the styling of similar elements in Boost 11 and LP-64 Multiband.

In general, LP-64 works well for its intended purpose -- i.e. providing frequency-dependent boosts and cuts without otherwise coloring the signal. My biggest complaint, and I consider this to be a serious one, is that dragging the control points on the HUD while playback is going makes the signal cut out for each little move, to the point where it sounds like it is stuttering if you are trying a bunch of small adjustments. Wondering if this might just be the nature of the beast with linear phase processing, I tried a similar thing with [PSP Neon](#) (PSP Audioware's linear phase EQ). Audio continued to operate smoothly in Neon. Since adjusting EQ is a process best done by listening, this stuttering sound is quite distracting. Working around it really means making an adjustment, listening, adjusting your adjustment, doing some more listening, and so on, whereas it would ideally be possible to be listening in parallel with making adjustments. The issue has further implications, though. LP-64 EQ's parameters are nominally automatable, and dragging the nodes while recording automation does result in writing automation information to track or bus envelopes. However, that automation is totally unusable because the same stuttering effects that occur during playback also occur in the rendered result. While I don't often automate EQ myself, I know some people do, so this needs to be kept in mind until such point as Cakewalk fixes this issue, which one would hope is categorized as a bug (i.e. otherwise there would be no point in supporting automation).



LP-64 Multiband does for multiband compressor/limiters what LP-64 EQ does for equalizers. That is, it provides linear phase coherency, and the transparent processing that goes with that. LP-64 Multiband features up to 5 bands of dynamic processing. Somewhat unusually, instead of having you specify the crossover frequency between bands, LP-64 has you specify the center frequency for each band. It then determines the crossover frequencies based on the setting of the band you are adjusting and the settings of adjacent bands. This is actually reasonably intuitive as you can set the frequency by soloing the band and dragging the band's control point node around to try to hone in on the area you

need to process. Unfortunately, LP-64 Multiband suffers from the same stuttering-type audio artifacts LP-64 EQ has when dragging its control points around. The control point can also be used to directly set the gain for its band. Bands can be enabled or disabled, but note that disabling a band doesn't really turn LP-64 into, for example, a 4-band compressor. While there will be only four bands of dynamic processing active at that point, there will still be five bands of frequency ranges. One band may not have any dynamic processing applied, but the effect of its frequency band is still felt in limiting the frequency band of any contiguous frequency bands. That said, this shouldn't be a major issue since you could simply set the band you wanted to merge with a contiguous band to have identical settings to that contiguous band to get a similar effect.

Center frequency and gain for the individual bands can be set directly on the frequency/gain graph by dragging the control nodes around as in LP-64 EQ. To set Attack, Release, Threshold, or Ratio controls, however, you must select the node for the bands whose settings are to be adjusted, then set the values for that band via a knob or direct numeric entry of the value to be used. LP-64 Multiband also provides an "All" button to allow setting the value for all bands identically by setting the value for any band.

LP-64 Multiband's graph control can be zoomed to cover a narrower vertical range, just as noted above for LP-64 EQ. LP-64 Multiband has both Input and Output level control knobs and peak hold indicators, and these are again similar to the style of the Output control in Boost 11.

Aside from the issue mentioned above with respect to audio artifacts when dragging control nodes around to set frequency and/or gain, which I found somewhat less bothersome on a multiband compressor than on an equalizer, LP-64 Multiband also works nicely. It delivers gain reduction without coloring the sound, which is what it's supposed to do.

Users who have been using SONAR for a long time may already have favorite mastering plug-ins. For anyone who doesn't, though, Boost 11 and the two LP-64 plug-ins will be nice tools to have. Even users who do have mastering plug-ins may find these three new plug-ins of interest in some cases. In particular, all three of these plug-ins support 64-bit host-to-plug-in and plug-in-to-host interfaces, so won't result in your audio data's being ratcheted down to 32-

bits on either end as would happen in the case of plug-ins that have not yet updated their interfaces for 64-bit operation. Unfortunately, there are still far too many vendors who have not done this.

## Sharing with the World

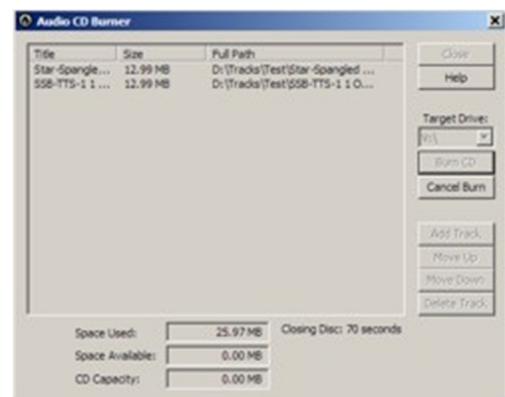
Once you've got your finished master recording in hand, it's time to share it with the world. For SONAR users doing music production, this generally means either burning an audio CD, be it for use as a master for manufacturing or just to have something for a band to listen to overnight to decide if they like the current mixes, or putting the music up on the web. Cakewalk has added tools in SONAR 7 to cover both areas.

In case your results need to end up on CD, SONAR now includes integrated CD burning. Let's be clear right up front that this is not a true CD mastering-oriented solution ala Sony's CD Architect. You won't find the kind of detailed controls needed for production CD mastering. There is no sub-code burning, CD-TEXT, control over inter-song spacing, etc. The facility is using the Windows IMAPI service, which burns in Track-At-Once format, not the Disc-At-Once format required for CD masters for replication. You do need to make sure the Windows IMAPI service is available, but that is likely to be the case by default if you have a Windows-compatible CD or DVD writer in your Windows XP or Vista system.

If you're still with me, you've no doubt realized that there is still value in an integrated CD burning facility without being able to burn replication masters. This is exactly the sort of solution that could work for making those rough mix CDs for the band to listen to overnight, or even just to hand out to your friends as a party mix or possibly even to take to the club for your night as a DJ. The basic routine goes like this: You mix your song to 16-bit 44.1 kHz Wave, and SONAR's Export Audio dialog box provides a checkbox to allow you to add your mix to the Audio CD Track List. If you choose any other format than Wave (or Broadcast Wave) format, you won't get the checkbox. If you choose any other bit-depth than 16 bits, or any other sample rate than 44.1 kHz, you won't get the check box. You make sure the checkbox is checked and do your mixdown, and SONAR adds a path to that file into a list of tracks to be burned to CD. Then you repeat the process for your next mixdown, and so on until you've got all the tracks you want for your CD. Now that you're ready to burn a CD, you call up the Burn Audio CD function from either SONAR's Tools menu or its separate Start Menu icon in Windows (in the same program group where SONAR 7 is installed). You will then see the dialog box at right, with your tracks listed in the order you added them. If the order you want is the order they're in, you can just put a blank CD in your burner and press the Burn CD button, and you're on your way. Otherwise, you can move tracks up and down, add tracks, and/or delete tracks until you've got your track list right, then burn the CD. If all goes well, you'll have your CD in a few minutes.

I found there are at least a few non-obvious things that could go wrong. In one case, after I'd exported a mix file, asking that it be included in my CD track list, I decided the file was messed up (I'd forgotten to include enough extra space at the end of the mix to let sounds and reverbs tail off on MIDI instruments). Thus I went and deleted the file in Windows. Of course, SONAR's CD burning facility knew nothing of this and the file name was still there. In the meantime, I'd forgotten about this (temporarily anyway), so when I hit "Burn CD", I got an error message regarding the file, but the burning program kept going, burning the other track I did have in the list to the CD. I would have preferred if it had required me to make the track list right before going ahead with the burn, since odds may be good that, in a scenario like this, there may be a file that should have been there instead of the one that had been deleted.

In another case, I'd put a CD-RW in the drive that already had content on it. When I do that with Nero it prompts me as to whether I want to erase the contents of the disk, and, if I choose to do so, it then erases the disk for me and burns my CD. Here, though, not only did it not give me that option, but there is no function to erase a rewritable CD. Thus, I had to go out to Nero to erase the CD-RW before I could return to SONAR's burning program to burn the CD. Given SONAR's CD program isn't fully featured enough for making master CDs for replication, it would seem there might be a good chance users would want to use CD-RW media, at least occasionally, for handling rough



mixes, given that they can then be rewritten the next day with the updated mix. While it is true that CD-RW media is not as compatible with many CD players as CD-R media is, nowadays, with many people using DVD players as CD players in their home stereo systems, or playing CDs on their computers, there are many situations where using rewriteable media make sense.

The bottom line for this facility is that it does provide a convenience in some contexts. However, if your contexts veer from the relatively limited contexts this supports, you're likely to need another CD burning program, too. In that case, I have to wonder how frequently SONAR users would use this facility rather than just turning directly to the program they are using for the contexts SONAR doesn't support. Where I see SONAR's facility providing the biggest convenience is in the case of someone who is pumping out a lot of one-time mixes in a relatively short space of time, and needs to be able to compile the track list for a CD of the results quickly to either take the CD and go somewhere with it, give it to client to listen to, or send it out to a client. In that case, this facility can be a timesaver for getting the track list together, and it does do the job well enough.

For those SONAR users who want to put their music out on the web, SONAR 7 now includes Cakewalk Publisher 2.0. In a nutshell, Publisher is designed to help you create a customized streaming music player for use in embedding your music on a web site, then upload the player and tracks to the web. More specifically, it creates an Adobe Flash-based MP3 file player, to play a playlist of your MP3 files while also displaying optional information such as artist name, song name, album name, a web URL, and album art.



You can create a Publisher playlist either by adding MP3 files directly to Publisher or by telling SONAR to add tracks to a Publisher playlist each time you mix a file down to MP3 format. Note that you can't have SONAR add tracks to publisher when mixing to WAV files or to any other non-MP3 format. One minor idiosyncrasy I noted was that, after adding a track to Publisher via SONAR's Export Audio process, the Publisher interface was brought up automatically, but SONAR was hung in the background until Publisher was closed. Thus, my process for exporting an album's worth of tracks to Publisher was to mix the file and have it added to Publisher, close the Publisher interface, then move on to the next file and repeat the process. However, there was no problem working in Publisher and SONAR in parallel at other times, only when files were exported directly from SONAR to Publisher.

Once you have your playlist together, the Publisher interface allows filling in the informational fields, adding album art for each track, and reordering the tracks by moving tracks up and down in the playlist. Playlists can be named and saved, so you can work with multiple playlists more or less in parallel, though SONAR only exports to the most recently accessed playlist.

Publisher also allows you to customize the look and behavior of the three players it provides. For example, you can change size and colors, and also determine whether the player starts playing automatically when the web page loads and if it keeps repeating the playlist until the user hits a Stop control.

At some point before the first time you upload files, you'll need to configure settings for your web server so Publisher knows how to login and where to put the files. Publisher uses FTP to upload its players, the playlist, and your MP3 files, so it is necessary that the web server on which you store the players and MP3 files has FTP capabilities. However, no FTP capability is required on the web sites hosting pages in which your customized player is embedded. (Of course, you do have to have some way of adding the HTML code to embed the player.) You can configure multiple hosts in Publisher.

Once you've got your playlist and host information configured, you use Publisher's "Publish" button to have Publisher upload everything needed for your custom player to your content-hosting site. If all goes well, you should see a running status message as to what is happening while Publisher does its work, then at some point Publisher

will present you with the player dialog and the HTML code you'll need to embed the player on your web site. For some reason I can't explain, the first several times I tried publishing my files with Publisher, it did not upload my files, nor did it give me any error messages. I tried this a number of times in two bootup sessions. Cut to a few days later, and I tried again, and everything got uploaded on the first try, and on every try thereafter. The only thing I can think of that may have changed was that I rebooted my router in between, so perhaps something was strange in my network in the first few sessions. Still, it would have been helpful if Publisher had given me some kind of meaningful error message, or even an error message at all.

Once I did get the files uploading properly, then embedded the codes in a web page, I only got a white square where the player control should have been. At that point I'd been trying to use Publisher's full player, which includes both album art and a track list (there is also a player that includes just album art and another that includes only the transport controls). I tried several times uploading this way, and still only got the white square. After that I decided to try uploading the player with just the album art, and that one worked fine. I then decided to try uploading all three players, after which I could get any of the three players to work. This seems to suggest that Publisher may need the player with just the album art to be uploaded for any of the players to work (or at least to get the full one to work -- I didn't try uploading just the slim player with only transport controls). There is no harm done in that the players aren't very large, but it was confusing, as there were no error messages of any kind to suggest something was missing. With all the uploading of changes I was doing, one thing I also noted is that Publisher had to upload everything, including all the MP3 files in the playlist, every single time it uploaded anything. There was no facility to have it only upload just what needed to be changed (e.g. just an extra player).



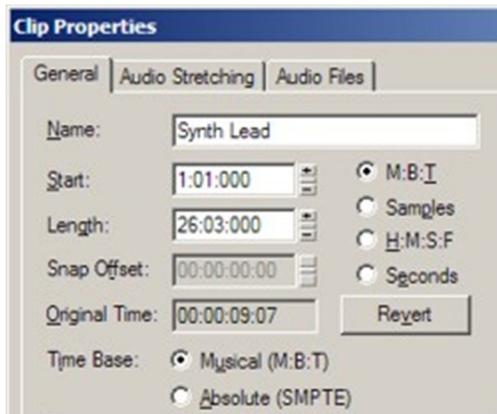
Now that I had my custom player up and working, the next hurdle I encountered was that all my tracks in the playlist sounded like "The Chipmunks Sing the Songs of Rick Paul". The story there is I'd uploaded 96 kbps MP3 files, created by SONAR's MP3 encoder, to keep size down and also not provide full quality streams since I am trying to sell downloads of the tracks I'd uploaded. Well, the 96 kbps MP3 files have a 32 kHz sample rate, whereas 128 kbps MP3 files have a 44.1 kHz sample rate. It appears that the Publisher players expect 44.1 kHz MP3 files, and play the MP3 file at that rate whether the MP3 file is meant to play at that rate or not. In troubleshooting this issue, I also noted that, if you right click on the player, there is a Download command available (and it does indeed work), thus seemingly defeating one purpose of creating a streaming player.

The bottom line on Publisher is that, if you are happy to work within its assumptions, meaning using only 128 kbps or higher MP3 files, uploading extra players if needed to get the one you want to work, and you are okay with the potential for users to download the MP3 files you are streaming through Publisher's custom players, Publisher can streamline the process of getting your songs streaming, in a relatively user-friendly fashion, on a web site. At present, though, it has a number of quirks, and at least one potentially fatal flaw for those who want their files available for streaming only, not downloading.

## We've Got Your Back

Of course, we always want things to go well, free from the intervention of Murphy. Sometimes, though, things just don't quite go right. The troubles that ensue may be of our own making, for example if we've been moving clips around for some artistic purpose then ultimately realize we've made a mess of things. Alternately, system troubles, such as file corruption, may cause one of our valuable projects to become unreadable. While SONAR 7 can't work miracles, it does include a pair of new features that just might help in these two scenarios.

SONAR audio and MIDI clips now have a new Original Time property. This property stores the original SMPTE time stamp associated with the clip. Depending on how the clip was initially created in SONAR, the original time stamp will be created either when the clip is recorded, when the audio or MIDI is imported, or when a submix is created by bouncing one or more clips to a new clip. From the point of creation on, though, the Original Time is never changed, even if you move the clip around in time. If things get too far out of hand, you can go into the Clip



Properties for the clip and use the Revert button to make the clip revert to its original SMPTE time stamp. This may or may not also send the clip back to its original M:B:T (Measure:Beat:Tick) time stamp, depending on whether the project's tempo map has changed in a way which changes where the clip would have been positioned in M:B:T time. Specifically, the clip is going back to the same absolute time position in the project as when it was created, independent of whether tempos may have changed its M:B:T time stamp. Given that this original time stamp reversion capability could be considered something of a panic button, I find it amusing, though wholly appropriate, that the reversion can be undone. That's sort of like having a panic button to undo having pressed the panic button.

Back in SONAR 5, Cakewalk introduced "Safe Mode", a special way of opening a project file geared at diagnosing and working around the decidedly distasteful experience of opening a project that causes SONAR to immediately crash. Invoked by pressing the Shift key along with the Open button when opening a project, this mode allows you to choose which plug-ins get loaded with the project. The basic idea is that most of these crashes were caused by plug-in issues, and controlling which ones got loaded could help narrow down which one was the offender while also providing a way to access most of your project data. (The major exception would be settings for the offending plug-in.)

While SONAR's Safe Mode is a big help in getting back to life in the face of one of the most common causes of problems opening a project file, it doesn't address all possible reasons for problems in opening a project file. For example, the file may have been legitimately corrupted due to hard drive problems or a system crash that occurred at the wrong time. (While there's never a good time for a system crash, one that occurs before a SONAR project file has been flushed to disk from the Windows write through cache would be a particularly bad timing with respect to project file integrity. A crash that occurred during a file saving operation would be similarly ill-timed.) Another scenario, back in the court of plug-ins, might be if a badly behaved plug-in corrupts SONAR's memory, thus causing bad data to be written to the project file when it is saved. Yet another on the hardware and/or media side would be the need to load a file from archives when either the optical drive used to write the archive file was going bad at the time of writing or the media has since been corrupted, but where there is still a way to physically grab the file off the disk in its corrupted state. These kinds of situations are really bad news, and the question isn't so much whether SONAR can save your file or not -- the odds aren't good -- but whether SONAR can help you save enough of your data to be an improvement over going back to square zero.

In these scenarios, you may be at the point where SONAR doesn't really even try very hard to open the file, instead presenting an error message such as, "the file you are trying to open is not compatible with this version of SONAR." SONAR 7 provides a new File Recovery Mode to attempt to allow you to recover data from a project in this scenario. It is also invoked by holding down the Shift key when opening a file. This mode only opens up the Track View in case the corruption may be in other view-only-oriented aspects of the project. It also prompts for whether plug-ins should be opened or not, just as in the older Safe Mode. Additionally, it ignores various strict error checks that would normally cause SONAR to abort loading a project file, and detects and skips over corrupt chunks of data, if needed. Note that this is somewhat akin to SONAR's ignoring its own common sense, in terms of its internal protocols that would generally guarantee its not making a mess of things or allowing other elements (e.g. plug-ins) to make a mess of things, all in the spirit of trying as hard as possible to let you recover something better than nothing. As a result, any data you do recover could be suspect, and SONAR's operation may be compromised, even to the point of its crashing or hanging. Thus, it is important to save any recovered files as a new project, then shut SONAR down and restart it before attempting to go beyond the file recovery, especially before working with any projects that were not previously corrupted.

I hope no one reading this will ever have the occasion to have to attempt to use SONAR's File Recovery Mode to open a file. Alas, I was not so fortunate. I recently decided to remix one of my archived projects from 2001. It had been recorded and mixed in SONAR 1 -- talk about ancient! Unfortunately, the CD-R it had been archived to suffered some integrity problems, and this was back in the days before per-project audio, where getting the full

project archived meant using CWB files. I could read the CWP file off the CD just fine, but I'd need the CWB file to get the audio, and just doing a Windows-level copy wouldn't do the trick. I ended up using a third party utility to extract the file, albeit in its corrupted form, but SONAR wouldn't read it, giving the dreaded error message I mentioned above. For purposes of this review, I suppose you could call this a fortuitous opportunity to try out File Recovery Mode. The good news is that SONAR at least tried to process the CWB file, naming a couple of the softsynths used in the project and asking me if I wanted to open them -- I passed on that since I was mainly interested in resurrecting the audio data, mainly the vocal, given I had a valid CWP file for the MIDI side of things. It then also asked me if I wanted to open some other plug-ins, but the names presented to me were blank. Again, I declined. Ultimately, it told me it could not open the project, so apparently the data was too corrupted for even File Recovery Mode to save it, which I suppose isn't so surprising given that Windows couldn't even read the file sufficiently to copy it. While it would have been nice to tell you of a scenario where this new feature helped rescue a corrupted project, it seems to me that the nature of the beast in this area is that there can be no guarantees. To be perfectly honest, I'd prefer not to have another project that makes me need to try this feature again.

## Producer or Studio?

We've been focusing on Cakewalk's flagship DAW, SONAR 7 Producer Edition. While I believe that is the version of SONAR that will be best suited to most professional and advanced amateur uses, our readers on a tight budget may be interested to know that there is a less expensive alternative, called SONAR 7 Studio Edition, available. Studio Edition provides most of the basic features of Producer Edition, but sacrifices surround mixing, a few advanced features, and a number of software instruments and audio plug-ins in tradeoff for a savings of about \$200 at street price.



The following features of SONAR 7 Producer Edition are not included in SONAR 7 Studio Edition:

- LP-64 Linear Phase Equalizer
- LP-64 Linear Phase Multiband Compressor
- VC-64 Vintage Channel
- Roland V-Vocal VariPhrase technology
- Z3TA+
- PSYN II subtractive synth
- Pentagon I analog synth
- RXP groove instrument
- Perfect Space convolution reverb
- surround mixing
- Sonitus:fx Surround Compressor
- POW-r dithering
- MPEX-3 time stretching

In addition, AudioSnap in SONAR 7 Studio Edition is limited to processing one track at a time.

## Playing Around

Trying out new product features is always fun, and those new features are the reason longtime SONAR users, like me, choose to upgrade from whatever version of the product we've been using to the latest release since there's no sense in monkeying around with something that's working. (Of course, if it weren't working, we'd probably be checking out other DAWs for crossgrade possibilities.) Our overall experience with any DAW, however, is rarely dominated by a set of new features. Rather, it is a mix of features that have been in the product from day one,

features added to the product over the many past "latest greatest" versions, and occasionally even a new feature that instantly works its way into our day-to-day usage of the product. Just as significantly, the product's stability greatly affects our experiences as DAW users. How did SONAR 7 fare overall?

I received SONAR 7 Producer edition in early October. At that point I was in the final stages of a couple of production projects, with another remixing project just starting up. Within a day of receiving the product, and satisfying my curiosity as to the new features I was most excited about, I put SONAR 7 right to work, taking up from where I'd left off a day or so earlier with SONAR 6.2.1. My first project was a mastering project at 64-bit/96 kHz, using primarily the [PSP MasterPack](#) plug-ins in the mastering chain. However, I needed a multi-band compressor in the chain, so the arrival of LP-64 Multiband was fortuitous, and it slotted right into the project, filling the bill nicely over the Sonitus:fx Multiband I'd previously been using, yielding more transparent processing. Not a bad start! The project went off without a hitch.

Over the next week I finished another mastering project, and began work on the remix project I alluded to above in the section on the new File Recovery Mode. The latter was forcibly taken back to the MIDI stage due to the CWB file corruption on the archive disk, but came back together nicely as far as I've taken it to date (it's now sitting in a queue of projects waiting for vocal tracking). The next few weeks saw additional remixing work, often incorporating newer, better softsynths, use of AudioSnap to tighten loop-based parts taken from different sources (e.g. ACID loops for drums and Virtual Guitarist parts for rhythm guitars) against each other, and use of lots of different audio plug-ins, mix automation, and just general tweaking -- basically the things I might do on any random day when working a remix project. (NOTE: When I say "remix project" I generally don't mean a dance remix, but rather trying to upgrade the quality of an older recording to reflect the better quality tools and improvements in my production quality over the years. It generally involves at least upgrading softsynth and audio plug-in effects quality, and typically also redoing a few parts in the arrangement, though sometimes making more extensive arrangement modifications. Vocals may or may not be redone from scratch, depending on the recording.) In total, I did some work on at least seven different remix projects, some of which I've since finished, others of which had their arrangements upgraded pending new vocal tracking, and at least one of which got a brand new vocal, including complete comping and tuning. I also did one "slam dunk" piano/vocal/drum loop work demo for a brand new song, going from soup to nuts on that one. I work at various paces over time, but I have to say this particular period of time saw a lot more intense use of SONAR than my norm, in large part because I am trying to get an album together, and have been experimenting with older recordings that may be candidates for inclusion, trying to narrow down my list of suspects while upgrading the quality of the recordings that will ultimately appear on the album.

Through all of this, SONAR 7.0, and later 7.0.1, which I installed exactly two weeks after my 7.0 upgrade, largely performed as expected, or at least on par with SONAR 6.2.1 (e.g. softsynths that tended to cause crashes in SONAR 6 also tended to cause crashes in SONAR 7). At no point did I give serious consideration to reverting, and a number of the new features -- multiple controller lanes in PRV, dim solo, sidechaining, and LP-64 Multiband to name a few key ones --worked their way into my list of SONAR's "couldn't live without" features I now use on a regular basis.

That is not to say I didn't encounter any problems. I definitely did. I've mentioned some of those above, such as the occasional noise spikes and audio engine stops that occurred when initially making sidechain connections. For the most part, though, the issues seemed largely confined to aspects of the new features, so could not be considered a step backward from SONAR 6. Also, at least initially, SONAR 7 seemed, empirically, to be a bit more stable than SONAR 6.2.1 was.

If that last sentence seemed a bit ominous, that was intended. Whether it was coincidence or not, I am not sure, but awhile after installing the SONAR 7.0.1 patch I began having an increased number of reliability problems, specifically in conjunction with projects begun in versions of SONAR prior to SONAR 7. I should note that, when I am talking of older SONAR versions, I am generally talking quite a bit older as I have no reason to remix recent projects. Projects I've worked on during this period go back as far as SONAR 1, and include projects worked on in at least SONAR 2 and 3. The general nature of the problems I am talking about in this area begins with reading in an older project, needing to make a few softsynth and/or other plug-in changes (e.g. due to a missing old plug-in or simply wanting to replace the part with better sounds), then having audio stop working in the project. At that point, I would try a variety of approaches to getting sound back, sometimes succeeding and sometimes not. When I didn't succeed, I might take the approach of starting a fresh SONAR 7 project, then copying the MIDI and/or audio data

from the old project to the new project to get things going again. When I did succeed, sometimes the course of events enroute to getting sound back seemed like voodoo -- e.g. disconnecting and reconnecting multiple synths in the Synth Rack, where I could have sworn I'd tried the same sequence of actions previously without success, but it worked the one time and things were fine afterward. Sometimes the "trick" might even be as simple as saving the file, closing the project and/or SONAR, and reloading the project.

In one particularly egregious case, with a project originally saved in SONAR 3.1.1, I wasn't having any luck getting sound back, so I tried saving the project to a new file, then closed the project and SONAR, then reopened SONAR and the project. SONAR "poofed" on me. (Popular use of "poofed" in this context means that SONAR just disappeared, with no crash message or Dr. Watson log. When this happens, the SONARPDR.EXE process tends to remain hung, as do the ASIO drivers for my E-MU 1820M sound card, and a reboot is necessary to get SONAR working again. While a number of other applications have crashed on me, SONAR is the only application that I use today that has ever poofed on me. In the past Corel PhotoPaint 6 also did it on occasion, but I've since upgraded to PhotoPaint 12, and the newer version hasn't ever poofed on me.) After rebooting and reloading SONAR, I tried reloading the project, and it also poofed on me, and has done so consistently thereafter. I am able to open it up with SONAR's Safe Mode by not loading one or more of the plug-ins, but I haven't had the time to determine which specific plug-in or combination of plug-ins is the culprit, and saving the file after loading it this way with just one softsynth (NI Kompakt) disabled, where I was able to load the project in Safe Mode, then reloading that project without Kompakt in it, also resulted in SONAR's poofing. In the interim, I'd simply gone back to the original SONAR 3.1.1 project, reloaded that, stumbled across some "voodoo" to get the audio working, and succeeded in saving that version of the project to a new project file that doesn't cause SONAR to poof. (I might note that the version of the project that makes SONAR 7.0.1 poof does not make SONAR 6.2.1 poof. However, simply saving a new version of the project in SONAR 6, then loading that in SONAR 7, still causes SONAR 7 to poof.)

In general, I have seen more crashes and poofs, not just on this one project, since updating to SONAR 7.0.1. However, I can't definitively say that the 7.0.1 patch was the cause of those issues. It could simply be that the nature of the specific files I've been working with, or what I've been doing in SONAR since the update, has ended up touching on more sensitive aspects of SONAR. For example, much of my in-depth new feature testing, aside from use of those new features in real life projects, was done in 7.0.1. It certainly wouldn't be unusual if newer features were more prone to issues given they haven't had the runtime out in the field that older features would have had. I also use an extremely wide selection of softsynths and audio plug-ins, and, especially when remixing, the plug-ins that appear on any given project could vary widely from project to project. Thus, it could also be that the specific combinations of ingredients being used before and after the 7.0.1 update resulted in some riskier combinations after the update.

I don't want to give the impression that SONAR 7.0.1 is unstable. Overall, it seems quite stable. The issues I've encountered tend to largely be concentrated around an initial period of bringing an older project into SONAR 7 and replacing older plug-in assignments with new ones. Once the project is gotten to working status, it generally doesn't have further problems. Most of the problems I've had outside of these scenarios or with idiosyncrasies of SONAR 7's new features tend to be consistent with issues I've had in similar contexts in SONAR 6, and some of the issues I've had in SONAR 6 seem less prevalent in SONAR 7.

The bottom line, in terms of my real life use of SONAR 7, is that I consider SONAR 7 to be an improvement over SONAR 6. Its stability, aside from the one specific area, is at least on par with SONAR 6, if not an improvement. Most of the time I am working in SONAR all day without any glitches. The new features can also significantly improve productivity in certain phrases of my projects, for example with Dim Solo being a major help when comping vocals and needing to both focus in on the vocal itself while also considering it in context. I wouldn't want to go back, and, despite having left SONAR 6 on my system as a "security blanket", I haven't actually gone back to it to do any real work.

## Closing Notes

There's a popular saying that goes, "if it ain't broke, don't fix it." This advice is often used as a rallying cry for software users to avoid upgrading to new versions of a product, especially in cases where some may view the updates are more about "bells and whistles" than true product improvements.

While I'd be hard-pressed to suggest SONAR 6 was "broke", I think even some of the most diehard upgrade holdouts may find something to compel them toward a SONAR 7 upgrade. For anyone who does any significant amount of MIDI editing, it's pretty much a no brainer. In fact, if it weren't for such great new audio enhancements as sidechaining and dim solo, I might even be inclined to characterize SONAR 7 as "the MIDI editor's release". Speaking of which, sidechaining and dim solo will be pretty compelling features for pretty much everyone. For those who are doing their own home mastering, and don't already have a package of third party mastering-oriented plug-ins, the new mastering plug-ins will be a big help. Anyone still using hardware processors in their studio will likely find the new external inserts a major improvement in convenience and productivity. The post production crowd will likely find the new features for handling long projects mandatory -- in fact, that may be one context in which SONAR 6 was "broke by omission".

I will admit to one key disappointment with SONAR 7, and that is the lack of improvement in the area of notation capabilities. I've mentioned in the past feeling that this was perhaps one of the main areas where SONAR lags behind some important competitors, most notably Cubase, LogicPro, and Digital Performer, and this limitation hurts SONAR in notation-heavy markets such as orchestral film scoring. I was encouraged a few months back when Cakewalk put forth a user survey almost entirely dedicated to sussing out users' notation requirements. Maybe for SONAR 8...

That one area aside, there is a lot to like in SONAR 7. In fact, all in all, this may well be the most compelling SONAR upgrade to date.

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

