PSE 2 User Guide

→

Contents

Overview

- [Overview](#)
- [Launch PSE](#)

Editors & Tools

- [Network Editor](#)
- [Synth Attacher](#)
- [FX Attacher](#)

FX Units

- [Amplifier](#)
- [Chorus](#)
- [Compressor](#)
- [Delay](#)
- [Distortion](#)
- [Equaliser](#)
- [Filter](#)
- [Overdrive](#)
- [Reverb](#)
- [FastReverb](#)

Tone Generators

- [TG Oscillator](#)
- [TG Particle](#)
- [TG DSynth](#)
- [TG Wavetable](#)

Controllers

- [Ctrl Envelope](#)
- [Ctrl LFO](#)
- [Crtl Amp](#)
- [Ctrl Random](#)
- [Junction Units](#)

Tutorials

- [Tutorial 1](#)
- [Tutorial 2](#)

Glossary & Tech

- [Glossary](#)
- [System Tech](#)
- [Wavetable Tools](#)
- [FAQ (/engine/pse/faq)](#)

[Partikl (http://intermorphic.com/engine/pse)](http://intermorphic.com/engine/pse) Sound Engine (PSE) 2 from Intermorphic Ltd.

http://intermorphic.com/engine/pse/guide/pdf/intermorphic_com_engine_pse_guide.pdf

PSE 2 Logo

Partikl Sound Engine (PSE) 2

The Intermorphic **P**artikl **S**ound **E**ngine (**PSE**) is a MIDI-driven sound engine that comprises a framework of sound synthesis and effects technologies with associated design interfaces. It goes beyond the traditional limitations of MIDI as it supports control of a number of essential effect units, which we also refer to as audio plugins, including LFOs, Reverb, Distortion, Filtering etc.

The PSE is generally driven by the Noatikl (MIDI) Music Engine (NME) (/engine/nme/guide).

Apps that include it

- Noatikl 2 (/noatikl) - See Noatikl Web Guide (/noatikl/guide)

- Mixtikl 6 (/mixtikl) - See Mixtikl Web Guide (/mixtikl/guide/index.html)

- Wotja 1 (/wotja) - See Wotja Web Guide (/wotja/guide)

- Tiklbox Relax 1 (/tiklbox/relax) - See Tiklbox Relax Web Guide (/tiklbox/relax/guide)

The PSE is also a core component in Intermorphic Tikltech (/about/#tikltech).

Note: The PSE used to be referred to like an app, which was a bit confusing to one and all, as the PSE is not an app; it is instead more like a service to apps that include it! You may still see .partikl files in some of our apps, but we will be phasing those out to instead be files with the respective apps name. Using PSE to now refer to the x-app common technology also leaves the road open for us to one day release an actual app called Partikl!

PSE Key Features

- Provides interfaces for inline live editing of sounds and effects:
  - **Network editor**: for construction of modular synth sound and FX networks comprising:
    - **Sound Generators**: for Drum Synth (DSynth), Tone Generator, Particle Generator, Wavetable Generator (Soundfont [SF2] / DLS 1).
    - **FX Units**: for Filter, Reverb, Chorus, Delay, Distortion, Compressor, Overdrive, Amp, EQ, Fast Reverb.
    - **Controller Units**: for LFO, Envelope, Amplifier.
    - **Junctions**: for Audio, Controllers.
- Includes a list of built-in preset Sound and FX Networks
- Supports a number of independently-specified synthesis modules per voice, as well as per-MIDI-line and global effects.
- Individual effect units in can be enabled, disabled, added, edited and deleted in real time, even while the music is playing.

Launch PSE

PSE interfaces are accessed from Noatikl and Mixtikl in the following ways:

Noatikl 2

Select the appropriate Synth, Track or Global FX button in the [Noatikl Mobile Design View (/noatikl/guide/#view-design)](/noatikl/guide/#view-design) or [Noatikl Desktop Network Panel (/noatikl/guide/#top)](/noatikl/guide/#top).

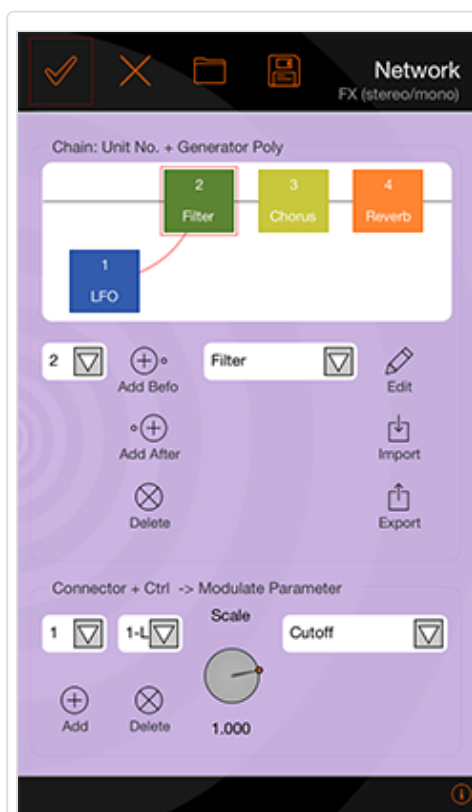## Mixtikl 5/6

Select the a Track FX or Global FX button in the [Mixer (/mixtikl/guide/index.html#mixer)](/mixtikl/guide/index.html#mixer) or select the Sound or Cell FX button in a [Content Cell (/mixtikl/guide/index.html#contentcell)](/mixtikl/guide/index.html#contentcell).

Network Editor
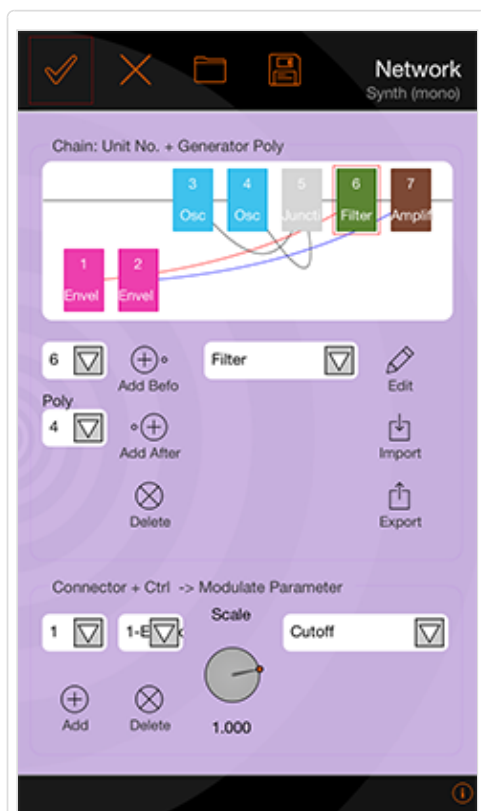
Synth Network Editor Screen

FX Network Editor Screen

The Network editor is at the heart of PSE. It is used to design either global or per-line Modular Synth Networks or FX Networks. It works the same way in both modes of operation.

It may at first sight seem complex, but it is actually quite easy to use one you understand how it works and how everything fits together!

The main controls in this unit are:

Ok - press this button to commit your changes and close the dialog.

Cancel - press this button to undo your changes and close the dialog.

Open - press this button to load either a Synth Network design XML file with extension .snd or a FX Network design XML file with extension .fxm.

Save - press this button to save the Synth Network design to an XML file with extension .snd or a FX Network design to an XML a file with extension .fxm.

Network Chain (top 1/3)

If there are no units shown then it means there is no Network defined.

The top 1/3 of the dialog shows a visual representation of the audio network design. The currently selected effect unit is highlighted.

Tap on a unit to select it and double tap (or select the Edit button) to call up the unit editor directly.

Note that signal-rate units are shown in the top row of effects units, and any control-rate units are shown in a separate row below.

Setup Area (middle 1/3)

This is where you add, delete, edit and import/export units.

"Unit" list: The left-most "Unit" list box lists the position (from left to right) of the unit in the Network. Units are always numbered from 1 upwards and they are automatically renumbered if you ever delete a unit. Select an entry in this list to highlight that unit.

"Poly" list: The left-most Synth Polyphont "Poly" list box is only displayed for Synth networks and is where you set the polyphony level used by the generators in your network. The default value is 4; set this too high, and your computer might slow down; set it too low, and you won't hear as many notes playing in all of your chords.

"Add Before" button: Tap to add a new effects unit immediately before your currently selected unit. This will usually default to being a "reverb" unit: change it to whatever you want!

"Add After" button: Tap to add a new effects unit immediately after your currently selected unit. This will usually default to being a "reverb" unit: change it to whatever you want!

"Delete" button: Tap to remove the currently selected unit from your module.

Unit Type List: This central list shows the type of the currently selected unit. It is also where you select a unit before you use the "Add Before" or "Add After" buttons. **Note** if you are designing a Modular Synth, then at least one of your units should be a tone generator! Tone generators all have a "Tone:" prefix.

"Edit" button: Tap to display the editor for the currently selected unit. Note: Some units do not have editors (Random, Junction); in which case, nothing will happen when you press this button.

The Unit editors themselves have a "cancel" button that allows you to reset parameter values back to where they were when you first popped-up the editor. If you close the dialog without pressing cancel, this will "commit" your changes.

"Export" Button: Tap to export the effect module design XML to the clipboard.

"Import" Button: Tap to import the effect module design XML from the clipboard.

Parameter Controllers Area (bottom 1/3)

This area lets you work with parameter modulation.

Connector + Controller section

The following applies to all units that are not junctions.

- The "Connector" list lets you select the effects unit (which **must** be a control-rate unit) that is modulating the parameter for your effects unit. Change the entry here to be the unit number of the control-rate unit that you want to modulate your effects unit.

- The "Controller" list shows you a count of all the effects unit inputs that modulate parameters of your currently selected effects unit. The numbers in the list are just always numbered automatically from 1 upwards. Note that if there are no entries shown in the "Param" list, then this effect unit has no available modulators, and therefore you have nothing to control on it from e.g. LFOs or envelopes (in which case, Add etc. will have no effect).

- Press the "Add" button to add a new entry to your "Controller" list.

- Press the "Delete" button to remove the currently selected Controller list item.

- The "Scale" slider lets you fine-tune a scaling factor between 0 and 2, by which the modulating input value (from the modulating unit) is multiplied, before being applied to your modulated effect unit's parameter. This allows you to carefully adjust the amount by which a parameter is modulated. Note that if more than one control-rate unit feeds a specific parameter on a target unit, they will get added together automatically (including the appropriate scaling factor).

- The "Modulate Parameter" list lets you select the effects parameter that you want to be modulated by the specified control-rate unit!

Connector + Unit section

This section displays in place of the above section when you have selected a Junction Unit.

- The "Unit" list shows you a count of all the effects unit inputs that are fed-in to your junction. The numbers in the list are just always numbered automatically from 1 upwards.

- The "Connector" list lets you select the effects unit (which **must** be a unit at the same rate as the junction!) that is to be added together to help create the output of the junction. Change the entry here to be the unit number of the unit that you want to add with other unit outputs.

- Press the "Add" button to add a new entry to your "Input" list.

- Press the "Delete" button to remove the currently selected Input list item.

- The "Scale" slider lets you fine-tune a scaling factor between 0 and 2, by which the output value of your "input" unit is

multiplied, before being added together with the other unit values that are feeding this junction. This allows you to carefully adjust the level fed-in by a particular unit. Note that if more than one control-rate unit feeds a specific parameter on a target unit, they will get added together automatically (including the appropriate controller scaling factor); which means that control-rate junctions are not often required.
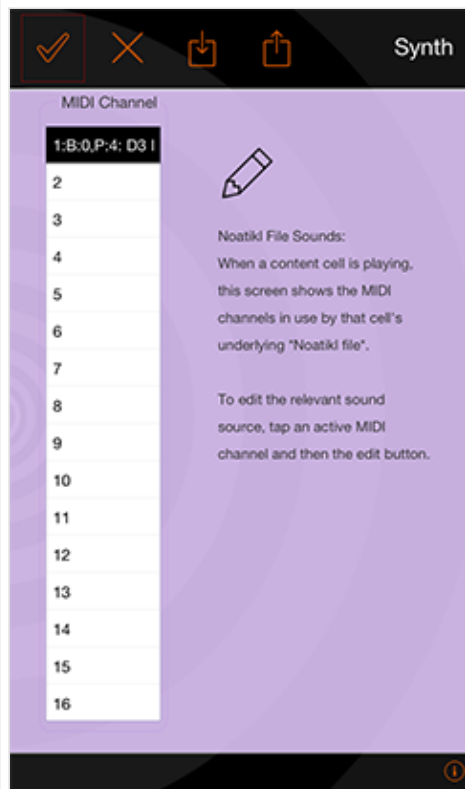
Synth Attacher (MT)

Synth Attacher Screen

Available only in Mixtikl

Mixtikl cells can contain Noatikl pieces and this dialog is what lets you get in to edit the Synth Network in use on a MIDI channel on such a piece.

The Export and Import buttons are very useful for copying Synth Networks to/from Noatikl files in other Mixtikl cells, or for exchanging with friends and colleagues.

The main controls in this unit are:

Ok - press this button to commit your changes and close the dialog.

Cancel - press this button to undo your changes and close the dialog.

Export - press this button to export the Synth Network to the clipboard.

Import - press this button to import the entire Synth Network from the clipboard.

MIDI Channels: To review per-line Modular Synth networks, select the MIDI line in question (from 1 to 16) in the MIDI Line list and then select the Edit button.

Edit button: For the currently selected MIDI line, displays the Network Editor where you can edit the Modular Synth network.

FX Attacher (MT)

FX Attacher Screen

Available only in Mixtikl

Mixtikl cells can contain Noatikl pieces and this dialog is what lets you get in to edit the FX Network in use on a MIDI channel (or at the Global level) on such a piece.

The Export and Import buttons are very useful for copying FX Networks to/from Noatikl files in other Mixtikl cells, or for exchanging with friends and colleagues.

The main controls in this unit are:

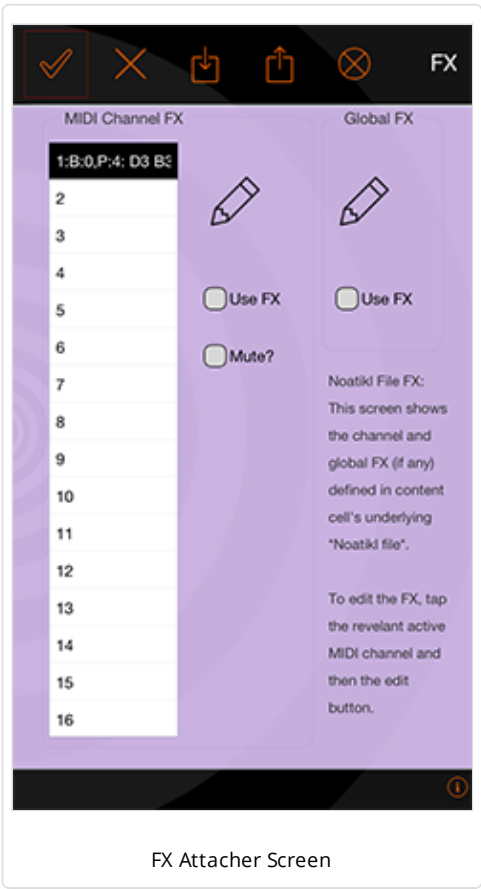**Ok** - press this button to commit your changes and close the dialog.

Cancel - press this button to undo your changes and close the dialog.

Export - press this button to export the entire FX Network to the clipboard.

Import - press this button to import the entire FX Network from the clipboard.

MIDI Channels: To review per-line FX networks, select the MIDI line in question (from 1 to 16) in the MIDI Line list and then select the Edit button.

MIDI Channel FX Edit button: For the currently selected MIDI line, displays the Network Editor where you can edit the FX network.

"Use FX" checkbox: Use to disable FX Network for the selected MIDI line.

"Mute?" checkbox: Use to mute-out individual MIDI lines. This setting is temporary; any muted line is unmuted automatically when you close the dialog. This checkbox can be useful when you want to design a specific FX, without having to listen to everything else in the mix; simply mute-out the lines you're not interested in, and un-mute them again when you're happy with the FX on for the MIDI line you are editing.

Global FX Edit button: Displays the Network Editor where you can edit the global FX network.

Global FX "Use FX" checkbox: Use to enable/disable global FX.
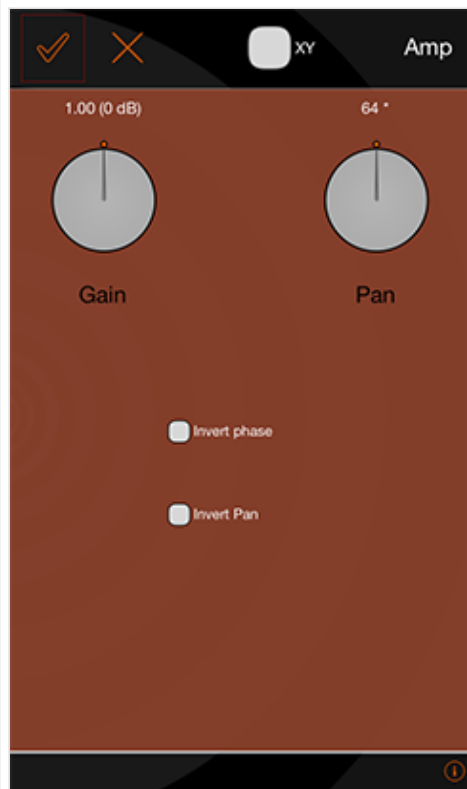
Amplifier

Amplifier Unit

If you have a weak signal, coming from a filter for example, then pump it into an amplifier. The Amplifier can multiply the input by a factor of up to 5. If you want more than this, feed it into another amplifier!

The main controls in this unit are:

XY checkbox: Checking this will display an XY grid where up/down is volume and left/right is pan position of the signal (response times may be slow and "drag memory" long depending on device).

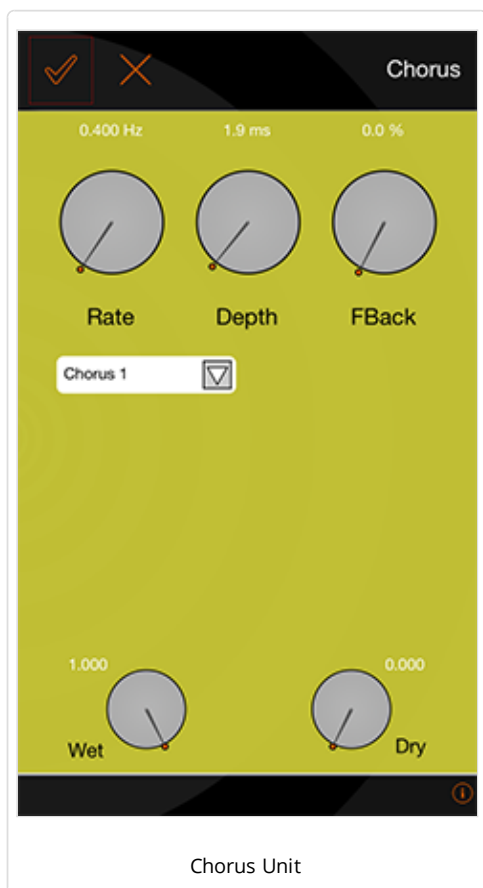**Gain**: This determines the amount of amplification to be done by this amplifier unit. The default value is 1.00 (0dB - i.e. no amplification) and values can range from 0, 0.10 (-19dB) to 10.00 (20dB)

Pan: This determines the pan position of the signal. The default value is 64 (center).

Invert phase: Checking this box will invert the phase of the signal

Invert pan: Checking this box will invert the pan of the signal

This sums the input and duplicates it at varying delay rates, with the effect of "thickening" the audio. It can beef-up a sound considerably! The delay time and frequency should be tweaked to give the right feel.

The main controls in this unit are:

Rate: A measure of the chorus modulation rate, in Hertz.

Depth: A measure of the chorus modulation depth, in milliseconds.

Feedback: The amount of Chorus Modulation Feedback to apply, as a percentage from 0 to 100 (which is the maximum).

Chorus Type: Use this to apply a "macro" setting of chorus parameters.

Dry Level: Adjust this value to alter the amount of "dry" (unprocessed) signal in the mix.

Wet Level: Adjust this value to alter the amount of "wet" (processed) signal in the mix.

Chorus Unit
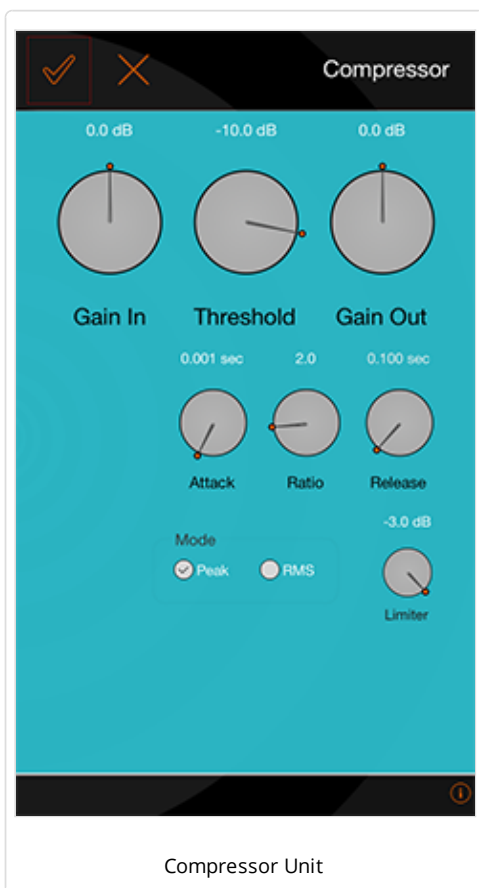
Compressor

Compressor Unit

This can be used to help compress the signal in the DSP chain.

If you view the compressor behaviour as a function of Output (in dB) against Input (in dB), the compressor displays three distinct regions of behaviour:

- **Linear** with a slope of 1 up to the Compression Threshold (CT)
- **Compressor** with a slope of 1/CR (where CR is the Compression Ratio) up to the Limit Threshold (lT)
- **Limiter** with a slope of 0 from the Limit Threshold (lT) and beyond

The main controls in this unit are:

Gain In: Input gain setting; defaults to 0 dB.

Threshold: This is the input/output level where the compressor becomes active. Below this level input = output.

Gain Out: Output gain setting; defaults to 0 dB.

Attack: This determines how quickly the envelope will respond to a positive change in the signal level. The value indicates the time it takes the envelope to rise 50% of the change.

Ratio: Determines the compression ratio. If this value is < 1 it will in effect work as an expander.

Release: This determines how quickly the envelope will respond to a negative change in the signal level. The value indicates the time it takes the envelope to fall 50% of the change.

Level Detect Mode: This has one of two values, which determines whether the signal is compressed/limited according to the peak or RMS level of the signal.

- **Peak:** Use Peak signal
- **RMS:** Use RMS signal

Limiter: The output is limited to this level. If this value is lower than dBCompThreshold, the compressor will not be in operation.

Delay

Delay Unit
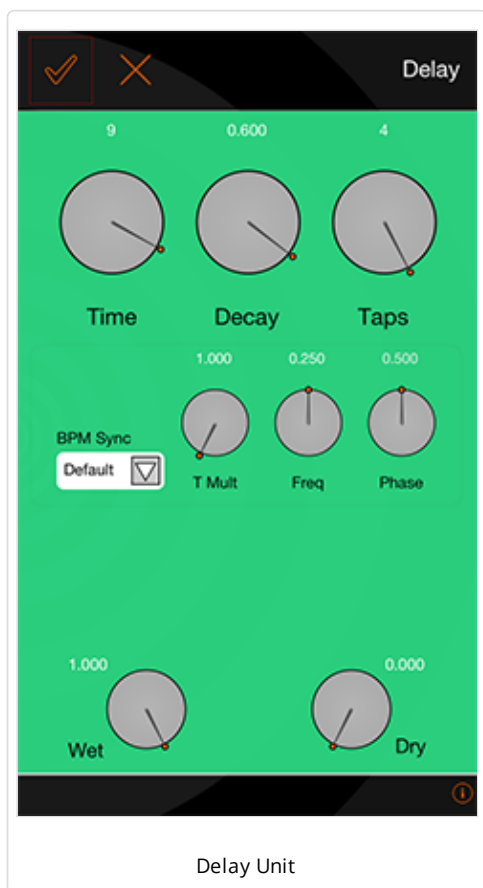
This adds interesting delay effects!

The main controls in this unit are:

Time: This parameter quickly determines the gross delay time in multiples of two. Increasing this parameter by one stop doubles the delay time, decreasing it halves it.

Decay: This determines the amplitude attenuation with each passing delay tap.

Taps: The number of "taps" in the delay system : the greater this number, the greater is the "thickness" of the delay effect.

BPM Sync: Default (non-tempo synched option) and various BPM values, e.g. BPM/4 which means generate the waveform selected in the above list at a tempo of the mix tempo (BPM)/4. This control allows for tempo-synched control of the Delay unit, for great effect.

T Mult: This parameter allows the user to fine tune the delay time in between the multiples specified by the parameter above.

Freq: Frequency (Ping Pong) is the movement across the stereo field with each passing delay tap. A high Ping-Pong frequency ensures a 'Ping-Pong' type of effect from one channel to the other. A low frequency results in a type of drift from one channel to the other. Only applies if the Delay is applied to a stereo signal pipeline.
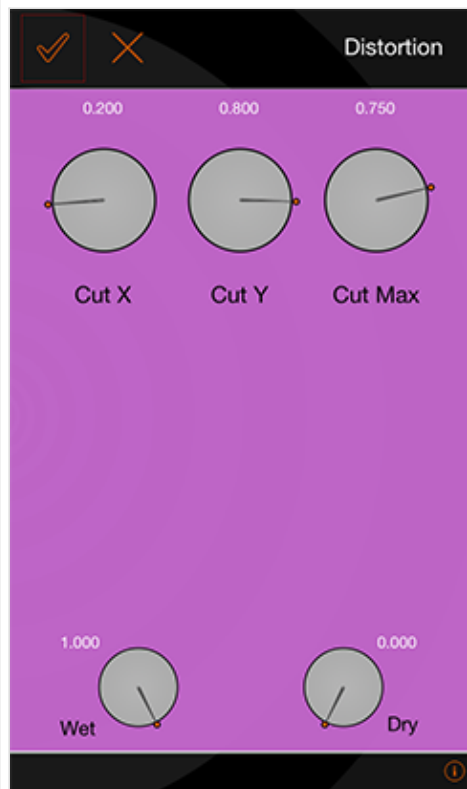
Phase: This simply determines where in the stereo field the Ping-Pong starts. Only applies if the Delay is applied to a stereo signal pipeline.

Wet Level: Adjust this value to alter the amount of "wet" (processed) signal in the mix.

Dry Level: Adjust this value to alter the amount of "dry" (unprocessed) signal in the mix.

Distortion

p>Distortion Unit

Distorts the sum of your incoming signals. Take cutoff X right down to about 0.02 to get a really distorted sound - it sounds great on a particle system! Cutoff Y acts like an amp and Cutoff Max determines the position of the output wave when the input is at 1.0 or -1.0.

Graph of the Distortion Cutoff parameters
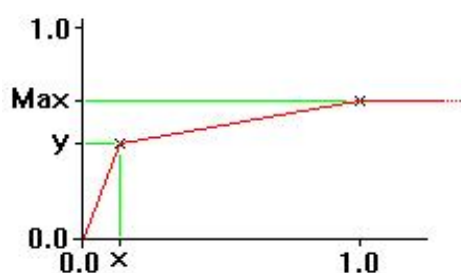
The main controls in this unit are:

Cut X: This determines the input value which will be scaled to the output value of 'y'.

Cut Y: As above, i.e. at input 'x', 'y' is the output.

Cut Max: At input value of 1.0, the output is 'Max'. Max may even be set to be lower than Cutoff Y

The combination of Cutoff X, Y and Cutoff Max contribute to reshaping the input signal.

Wet Level: Adjust this value to alter the amount of "wet" (processed) signal in the mix.

Dry Level: Adjust this value to alter the amount of "dry" (unprocessed) signal in the mix.

Equaliser

Equaliser Unit

You can use the Equaliser plugin to modify the frequency response of your audio signal. Depending on how you configure the plugin, it can operate as either a 5-band or a 10-band equaliser. Note that the 5-band equaliser takes less CPU power.

The main controls in this unit are:

On/Off checkbox: Use this to enable or disable the EQ

Lock checkbox: Use to allow all sliders to move together at the same time when you change any one of the individual slider values.

EQ bands: Specify the gain for the indvidual frequency bands. Adjust from -14 dB to +14 dB. Unused frequency bands will display "N/A" in case of 5-band EQ mode, or if band frequencies are close to or above half the sample frequency.

Amp: Specifies the amount of pre-amplification applied to the equaliser input signal. Adjusts from -20 dB to +20 dB.

Base: This slider adjusts the frequency of the lowest EQ band. All other sliders will automatically follow to maintain the frequency ratio between bands.

Filter

Filter Unit

This filter is a very flexible resonant filter. It can work in a variety of different ways depending on how you configure it with the various controls:

- Using a fixed filter frequency
- Using a built-in frequency sweep
- Using a filter frequency that is adjusted dynamically such that it tracks currently playing note pitches.

The main controls in this unit are:

XY checkbox: Checking this will display an XY grid where up/down is Q and left/right is filter cutoff (response times may be slow and "drag memory" long depending on device).

Rate: Determines the rate in Hertz at which the filter sweep moves from minimum to maximum value (and back again) [Sweep check box must be checked].

Range: This value, when added to the Filter Cutoff Min determines the upper frequency for the built-in filter frequency sweep

Cutoff: Determines the minimum frequency for the built-in filter frequency sweep

Filter Type list: Includes Low Pass, High Pass and Band Pass.

Q: the amount of accentuation (resonance) of the filter, 1.0 is the lowest value, and provides low quality filtering with little resonance, 5.0 provides high quality filtering with high resonance, and 10.0 provides the greatest resonance.

Cut: determines the cutoff or center frequency for the filter in Hertz.

Quality: The number of cascaded 12 dB/octave filter sections, defaults to 1. Use a higher value to get sharper cutoff, at the cost of greater CPU consumption.

Phase: determines the phase used by the built-in frequency sweep.

Sweep Checkbox: This checkbox determines whether or not the built-in filter frequency is in use. When not selected, this checkbox disables automatic filter sweeping.

Sync Checkbox: Only relevant when the filter is used with Noatikl part.

Freq Checkbox: When this checkbox is set, the filter frequency actually follows the currently played note pitch, where the note is generated by Noatikl. **Tip:** setting this checkbox gives a tonal quality to the Filter effect. In this case, the following parameters apply:

- **Octave Offset:** This combo-box allows you to specify the octave offset from the current note pitch
- **Semitone Offset:** This allows you to select the semitone offset from the current note pitch (taking into account the octave offset above) if Use Notes is set).
- **Micro Offset:** This allows you to specify thee microtonal offset in order to 'detune' the filter. The value supplied is in cents (1 cent = 1/100 semitone).

Wet Level: Adjust this value to alter the amount of "wet" (processed) signal in the mix.

Dry Level: Adjust this value to alter the amount of "dry" (unprocessed) signal in the mix.

Overdrive
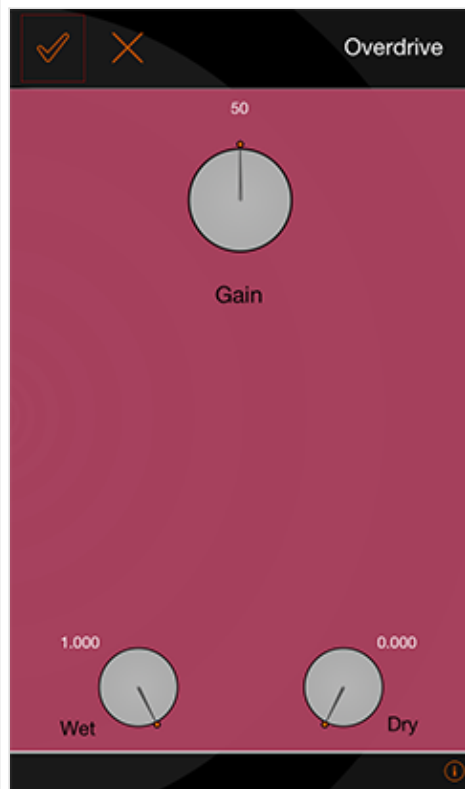
Overdrive Unit

This unit adds overdrive distortion to the sound source.

The main controls in this unit are:

Gain: Determines the amount of overdrive (distortion) applied.

Wet Level: Adjust this value to alter the amount of "wet" (processed) signal in the mix.

Dry Level: Adjust this value to alter the amount of "dry" (unprocessed) signal in the mix.

Reverb

Reverb Unit

This unit adds reverberation to the sound. This gives depth to the sound and makes it seem as though the piece is being performed in a room. You can control the 'size' and sound of this room with the parameters below.
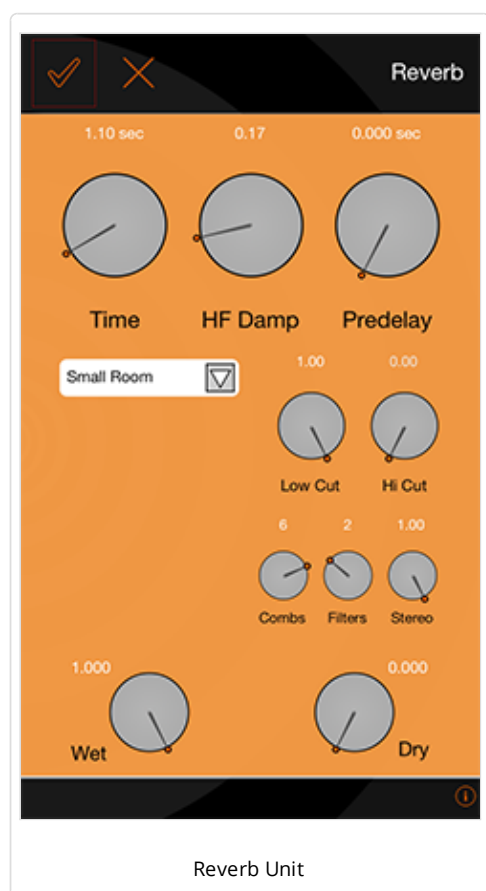
The main controls in this unit are:

Time: Determines the duration of the reverb.

HF Damp: In most environments high frequencies will attenuate more quickly than low frequencies due to damping caused by carpets, furnishings, etc. Use this control to adjust the amount of high frequency damping.

Predelay: Delay of reverb relative to the dry (unprocessed) signal.

Reverb Type List: Choose the reverb type from a number of musical and environmental presets.

Low Cut: Applies a lowpass filter to the reverberated sound. The value is the normalised cutoff frequency of the filter, with "1.00" corresponding to half the sample frequency (at 1.00 no lowpass filtering is applied).

Hi Cut: Applies a highpass filter to the reverberated sound. The value is the normalised cutoff frequency of the filter, with "1.00" corresponding to half the sample frequency (at 0.00 no highpass filtering is applied).

Combs: Determines the number of parallel comb filters used in the reverb algorithm. A higher number generates a more dense reverb tail, but is also more CPU intensive.

Filters: Determines the number of series allpass filters used in the reverb algorithm. A higher number generates a smoother reverb tail, but is also more CPU intensive.

Stereo: Adjust this value to change the perceived "width" of the sound.

Wet Level: Adjust this value to alter the amount of "wet" (processed) signal in the mix.

Dry Level: Adjust this value to alter the amount of "dry" (unprocessed) signal in the mix.

Fast Reverb
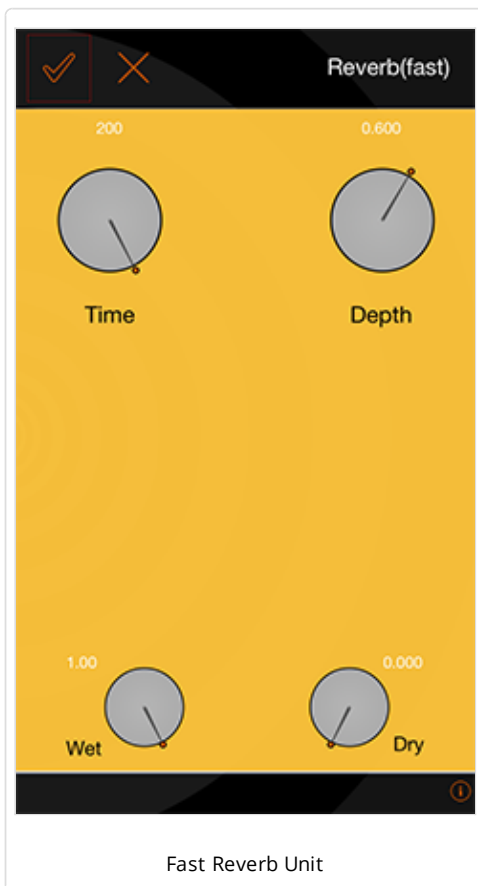
Fast Reverb Unit

With a small processing overhead this "low fidelity" unit adds a small degree of reverberation to the sound. It can sound quite ringy however, and for nice sounding reverb we recommend you use the [main reverb unit](#) instead.

The main controls in this unit are:

Time: Determines the duration of the reverb.
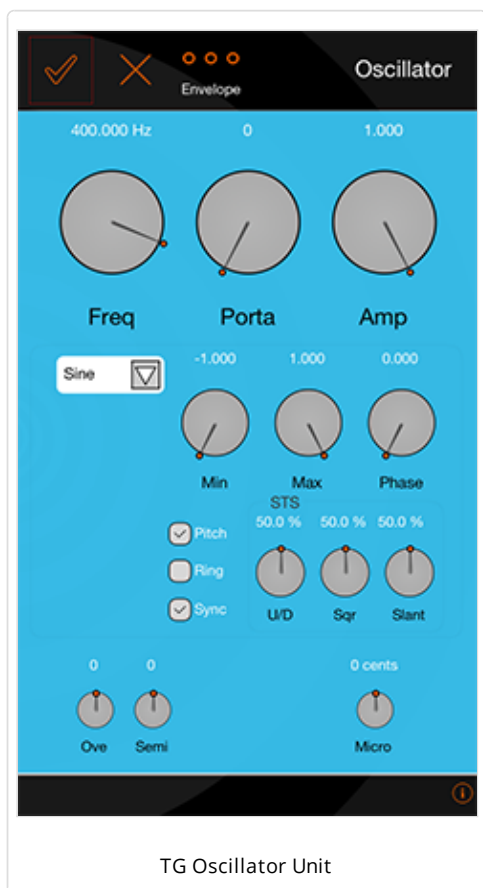
Depth: How deep the reverb is.

Wet Level: Adjust this value to alter the amount of "wet" (processed) signal in the mix.

Dry Level: Adjust this value to alter the amount of "dry" (unprocessed) signal in the mix.

TG Oscillator

TG Oscillator Unit

This dialog lets you create a range of waveforms, which are driven by MIDI pitch (via the "Pitch" checkbox); in which case, they are flexible, efficient MIDI tone generators.

Tip: You can use an audio junction unit to add together sounds from multiple tone generators to make even richer sounds!

The main controls in this unit are:

Envelope Button: Press this top toolbar button to pop-up an envelope editor that allows you to specify your own specific envelope settings for MIDI note on and note off events.

Freq: This controls the frequency of the LFO, in Hertz (Hz, cycles per second).

Porta: This defines the amount of "pitch slide" (Portamento) from one note to the next; used when "Use Midi Pitch?" is selected. Basically a measure of the exponential approach to the pitch of the next note. Measured from 0 (minimum effect, an instantaneous change to the next note pitch) to 100 (maximum exponential approach to the target pitch).

Amp: This controls the amplitude of the generated LFO waveform.

LFO Type List (below Freq): Includes Sine, Saw left, Saw right, Triangle, Square, STS (Saw, Triangle, Square) and Random.

Change the LFO Type to dramatically change the sound of your LFO! This combo-box lets you select from a number of waveform types that form the basis of the LFO waveform that is generated.

- **Sine** produces a smooth waveform with no other harmonics.
- **Saw Left** and **Saw Right** produce waves that look like a saw-tooth close up. The either slope to the left or to the right, but sound identical (they are both provided, as combining these with other waveforms provide interesting effects due to their different phasing!).
- **Triangle** sounds not quite as harsh as the saw-tooth but has more harmonics than the sine.
- **Square** produces a 'square' looking wave and sounds rougher than either the sine or saw-tooth waves. With the square wave you have the option of specifying the ratio of the up portion of the wave to the down portion. At either extreme, the square wave becomes more like a pulse wave, giving a 'pulsing' sound at lower pitches.
- **STS** is the very exciting STS wave which stands for 'Saw, Triangle, Square' because it is a hybrid between these three wave types. By varying the three parameters associated with it, the wave shape can be 'morphed' between the above three extremes. In effect, this waves covers all the preceding wave shapes, with the exception of the sine wave.
- **Random** produces white noise between Min and Max, also affected by the Amplitude. These values are produced irrespective of the pitch of the tone generator; and do not use the envelope of the tone generator. Random signals are particularly interesting when fed into filter effect units!

Min: This defines the minimum value that the generated LFO waveform can have.

Max: This defines the maximum value that the generated LFO waveform can have.

Phase: useful for slow oscillators (e.g. 0.2 Hz) - determines the start point of the oscillator.

Pitch: Select this checkbox if you want the MIDI pitch to determine the frequency of this tone generator. Use only for signal-rate units.

Ring:TBA.

Sync: Select this checkbox if you want the LFO to use the envelope to modulate the amplitude of the tone generator, or your own specific envelope settings that you can edit by pressing the "Edit Envelope" button. This only has any effect in Modular Synth networks, of course!

U/D: Only for the **Square** and **STS** wave types. Defines the ratio of "Up" to "Down" (or "On" to "Off") values (for STS waveforms, this is the ratio of u to d in the diagram below).

Sqr: Only for the **STS** wave type. Defines the "squareness" of the waveform (the ratio of s to u in the diagram below).

Slant: Only for the **STS** wave type. Defines the skew of the waveform (the ratio of a to a + b in the diagram below)

- With the Up Down Ratio at 100.0%, the Squareness Ratio at 0.0% and the Slant Ratio at 50.0%, the shape is a triangle.
- With the Up Down Ratio at 100.0%, the Squareness Ratio at 0.0% and the Slant Ratio at 0.0% or 100.0%, the shape is a Saw Right or Saw Left respectively.
- Lastly, with the Up Down Ratio at 100.0%, the Squareness Ratio at 100.0% and the Slant Ratio at anything you please, the shape is a Square wave.

Ove: This allows you to specify the the octave offset from the voice's pitch.

Semi: This allows you to select the semitone offset from the voice's pitch (taking into account the octave offset above).

Micro: This allows you to specify the microtonal offset in order to 'detune' the tone. -100 is the equivalent to a semitone down, and +100 is a semitone up.

The STS Waveform and its ratios
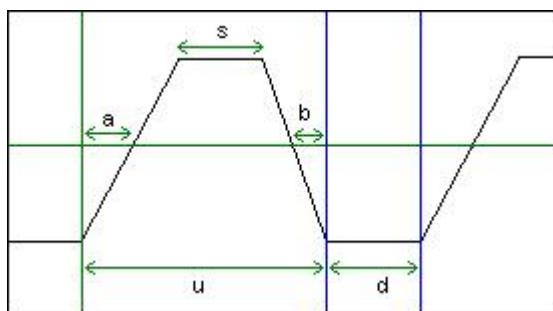
TG Particle

TG Particle Unit

This effect unit is an example of Granular Synthesis, which although is often used to do time-stretching of samples, is used for a completely different purpose here - that of creating rich layered sounds ranging from ambient forms to plopping sounds and twinkling chimes. The particle system incorporates octave, semitone and micro tonal offsets. It also has a checkbox that allows it to use the default volume envelope for amplitude shaping.

The Particle System synthesizes tones from a number of sine waves. This is really granular "additive synthesis". The unit works by streaming-out grains up to a maximum of 20 at any one time. These are randomly offset from each other to blend well. When the effect decides to generate a new grain, it takes its information from the dialog values. If the grains are reasonably large then the harmonics come in and out giving a full sound.

Tip: You can use an audio junction unit to add together sounds from multiple tone generators to make even richer sounds!

The main controls in this unit are:

Freq: This controls the frequency of the internal high-frequency LFO, which is is used to modulate the "harmonics", giving a tone that is rich with harmonic information. Only used if "Use Notes?" is not set.

Harmonics: Defines the range of multiples of the selected frequency present in the Particle System's output. What this means in plain English :) is that if you (say) have a frequency of 400Hz, and you define the Harmonic to be 3, then you may find grains with frequencies of 400, 800, and 1200 Hz in the generated waveform. The higher this value, the richer the harmonic output!

Velocity: this scrollbar controls the change of frequency **within** any one grain. Try moving this a small bit and you will find the tones start to dive or soar - good for creating horror effects!

Elements: The "number of elements" - the maximum number of grains present at any one time. The higher this value, the richer the output!

Notes?: This Checkbox determines whether the Particle System will use notes composed by the music engine, to override the Frequency control. The default is that the checkbox is set: i.e. the composed note pitches determine the frequency of the Particle System.

Amp: Defines the overall amplitude of the generated waveform.

Attack / Sustain / Decay / Pause: Changing these settings modifies an internal envelope generator, which is used to alter the shape and duration of the grains. It is easy to get twinkling effects just by making the attack small (say 1) and the decay longer (say 200). The pause determines the gap between grains. Modulating harmonic, attack, and pan position gives a nice full sound that alternates between twinkling and blending.

Ove: This combo-box allows you to specify the octave offset from the voice's pitch (only used if "Use Notes?" is set).

Semi: This allows you to select the semitone offset from the voice's pitch (taking into account the octave offset above) (only used if "Use Notes?" is set).

Micro: This allows you to specify the the microtonal offset in order to 'detune' the Particle System. -1.0 is the equivalent to one half of a semitone down, and +1.0 is one half semitone up (only used if "Use Notes?" is set).

TG DSynth

TG DSynth Unit
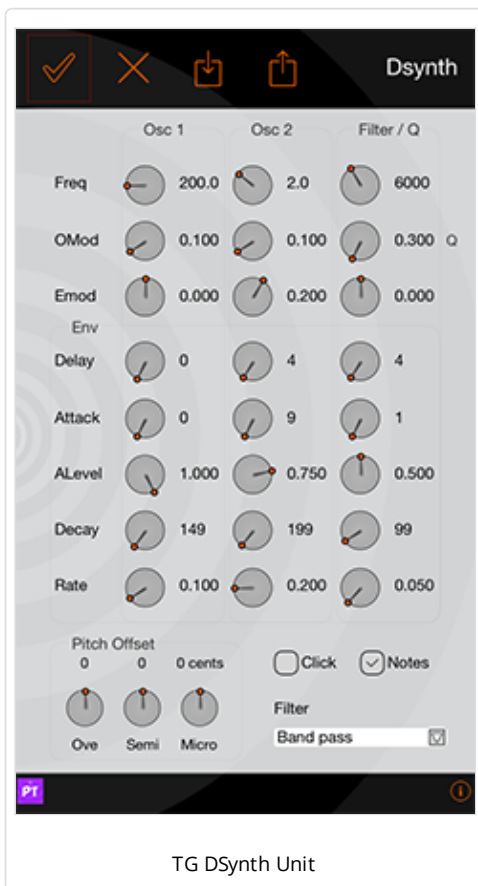
The "DSynth" effect unit is a very powerful and flexible tone generator, that can be used both as a melodic tone generator and and as a Drum/Percussion Synthesizer (which you can use to create all sorts of drum/percussion related sounds).

You can shift the pitch of the generated sound away from the notes that would otherwise be played, using the octave/semi/offset controls.

The unit works by combining two Oscillators, which cross-modulate each other's oscillation frequency. Each oscillator has its own ADSR envelope, which is used primarily to modulate the volume of the oscillator, but which can **also** be used to modulate the frequency of the oscillator!

The output of the two oscillators is merged with the output from a filtered noise generator (the bottom block of parameters), which also has its own ADSR envelope. The net effect is a unit which can create a wide array of percussive and melodic sounds. You can use the Export and Import buttons at the top of the unit to export and import DSynth settings to/ from the clipboard - have fun!

Tip: You can use an audio junction unit to add together sounds from multiple tone generators to make even richer sounds!

The main controls in this unit are:

Oscillator 1 and Oscillator 2 Parameters

Freq.: The frequency of oscillator 1 or 2.

OMod: "Oscillator Modulation" - The amount of modulation provided by the other oscillator, for oscillator 1 or 2.

EMod: "Envelope Modulation" - the amount of the envelope which should also be used to modulate the frequency of oscillator 1 or 2. If this is set to 0, then the ADSR envelope is used purely to modulate the volume of the oscillator.

Delay : The length of time it takes after a note event occurs for the voice, before the envelope starts to rise-up from zero to its Attack level.

Attack: The length of time it takes after the delay time has passed for the voice, for the envelope to rise-up from zero to its Attack level.

A Level: The target Attack Level for the attack stage of the envelope, as the envelope rises from its initial value of zero.

Decay : The length of time it takes for the envelope to decay down from the Attack Level, down to the Sustain Level of zero.

Rate: A measure of the curvature of the Decay, down to the sustain level of zero. A value of 0 gives a linear decay, and increasing values giving an increasing level of curvature - resulting in a sound that is more "plucked" in nature.

Filter/Q Parameters

Freq.: The frequency of the filter.

EMod: "Envelope Modulation" - the amount of the envelope which should also be used to modulate the filter frequency. If this is set to 0, then the ADSR envelope is used purely to modulate the volume of the filtered noise..

Q: the amount of accentuation of the filter, 1.0 is high, 0.0 is no effect at all.

Delay : The length of time it takes after a note event occurs for the voice, before the envelope starts to rise-up from zero to its Attack level.

Attack: The length of time it takes after the delay time has passed for the voice, for the envelope to rise-up from zero to its Attack level.

A Level: The target Attack Level for the attack stage of the envelope, as the envelope rises from its initial value of zero. **Tip:** A

value of zero is very good for melodic sounds; set to a higher value for more percussive sounds.

Decay : The length of time it takes for the envelope to decay down from the Attack Level, down to the Sustain Level of zero.

Rate: A measure of the curvature of the Decay, down to the sustain level of zero. A value of 0 gives a linear decay, and increasing values giving an increasing level of curvature - resulting in a sound that is more "plucked" in nature.

Click: Use this checkbox to add an initial transient when the notes are triggered; very useful for percussive sounds.

Notes: This Checkbox determines whether the drum unit will actually use the composed note frequency to override the frequency of oscillator 1. The default is that the checkbox is clear : i.e. the system will **not** use the composed note frequency for the drum unit. **Tip** : for percussive sounds, you might like to try clearing this checkbox!

Type List: The filter type to use: includes Low Pass, High Pass and Band Pass.

Ove: This allows you to specify the the octave offset from the voice's pitch.

Semi: This allows you to select the semitone offset from the voice's pitch (taking into account the octave offset above).

Micro: This allows you to specify the microtonal offset in order to 'detune' the tone. -100 is the equivalent to a semitone down, and +100 is a semitone up.
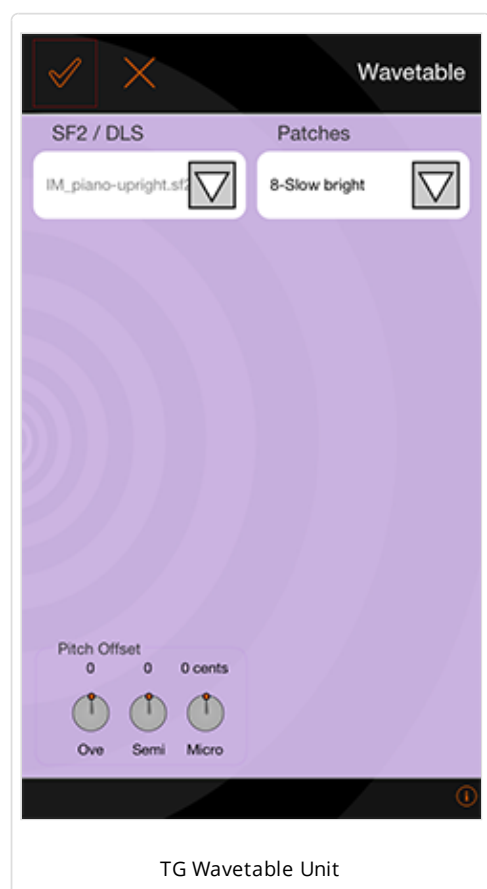
TG Wavetable

TG Wavetable Unit

General MIDI (GM) Wavetable:

When added to a Synth Network, a Wavetable Unit defaults to use the integral "General MIDI" DLS Wavetable and plays Patch "001-Acoustic Grand Piano". To change the Patch select another one in the Patch list.

Important: If a Wavetable Unit is used in a Synth Network for a MIDI line, its Wavetable / patch will override the underlying piece's Noatikl General MIDI Patch parameter (which always uses the GM Wavetable). If you DO want to use the setting in that parameter, you must DELETE all units in the Synth Network and select the OK button to accept that. If you then re-edit the Network then, although you will still see a Wavetable Unit, it is a placeholder but it is not used.

Using other SF2 / DLS Wavetables:

Important: Soundfont (SF2) Wavetable support/selection is only available in Mixtikl 6 and Noatikl 2.5+

It is easy to use custom wavetable sounds! All you need to do is to use the left list control to select the Wavetable from the list of those shown - and seven of these are included in Mixtikl 6 / Noatikl 2.5+ [Mixtikl 5 only lists the GM DLS Wavetable]. Then use the right list control to select the Patch (all patches available in the Wavetable are listed).

To use other SF2 in the relevant versions of Mixtikl or Noatikl, simple copy or transfer them to your device (/tiklpak/#copyinout).

Tip: You can use an audio junction unit to add together sounds from multiple tone generators to make even richer sounds.

The main controls in this unit are:

SF2/DLS (Wavetable) List: Used to select the Wavetable.

Patches List: Used to select the Patch in the Wavetable.

Pitch Offset:

- **Ove:** This allows you to specify the the octave offset from the voice's pitch.
- **Semi:** This allows you to select the semitone offset from the voice's pitch (taking into account the octave offset above).
- **Micro:** This allows you to specify the microtonal offset in order to 'detune' the tone. -100 is the equivalent to a semitone down, and +100 is a semitone up.
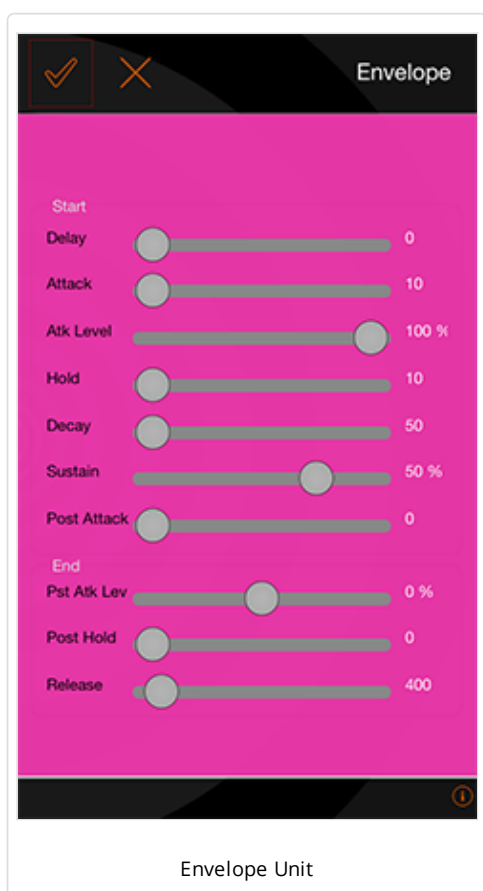
Ctrl-Envelope

Envelope Unit

The Envelope Unit is used to edit the "Ctrl-Envelope" control-rate plugin. It is only of use in Modular Synth networks, where it is triggered in response to MIDI note on events for the current MIDI line. Used in context of the TG Osc it is also triggered in response to MIDI note off events. It's only used in the context of a Synth Network and generates no sound of its own.

There are a number of values which you can adjust to modify the shape of your envelope. The shape used is shown in this diagram below.
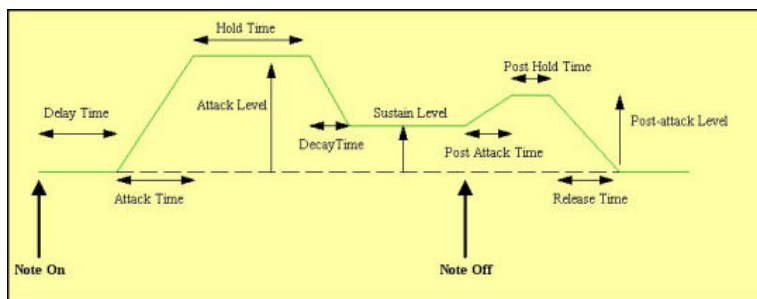
The main controls in this unit are:

Delay: The length of time it takes after a note on event occurs for the voice, before the volume envelope starts to rise-up from zero to its Attack level.

Attack: The length of time it takes after the delay time has passed for the voice, to rise-up from zero to its Attack level.

Atk Level: The target level for the attack stage of the envelope, as the envelope rises from its initial value of zero.

Hold: The length of time it takes after the attack level is reached, before the envelope starts to decay down to the Sustain Level.

Decay: The length of time it takes for the envelope to decay down from the Attack Level, down to the Sustain Level.

Sustain: The level at which the note will play, once all of the Decay, Attack, and Hold periods have completed.

Post Attack: The length of time it takes after the note stop event occurs, for the envelope to move from the sustain level to the post-attack level

Pst Atk Level: The target level for the post-attack stage of the envelope, as the envelope moves from its sustain level.

Post Hold: The length of time it takes after the post attack level is reached, before the envelope starts to decay down to zero.

Release: This time defines the time it takes for the Voice to respond to a note off event, in terms of how long it takes to decay from the Post-Attack Level to a level of zero.

Ctrl-LFO

LFO Controller Unit
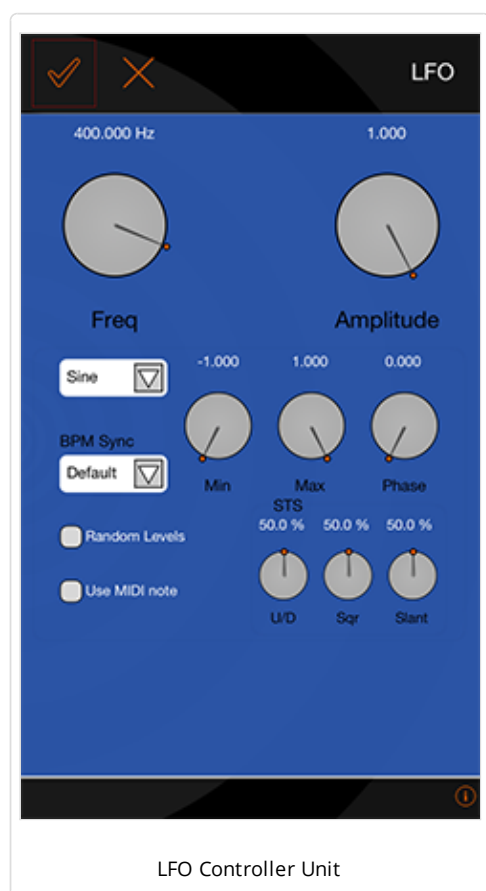
This dialog lets you create a range of control-rate waveforms. The waveforms oscillate at a pitch that you dictate yourself. They are used for modulating the parameters of other effect units; they should only be run below (say) 50Hz or you'll get some odd effects.

The main controls in this unit are:

Freq: This controls the frequency of the LFO, in Hertz (Hz, cycles per second).

Amplitude: This controls the amplitude of the generated LFO waveform.

LFO Type List: Includes Sine, Saw left, Saw right, Triangle, Square, STS (Saw, Triangle, Square) and Random.

Change the LFO Type to dramatically change the sound of your LFO! This combo-box lets you select from a number of waveform types that form the basis of the LFO waveform that is generated.

- **Sine** produces a smooth waveform with no other harmonics.
- **Saw Left** and **Saw Right** produce waves that look like a saw-tooth close up. The either slope to the left or to the right, but sound identical (they are both provided, as combining these with other waveforms provide interesting effects due to their different phasing!).
- **Triangle** sounds not quite as harsh as the saw-tooth but has more harmonics than the sine.
- **Square** produces a 'square' looking wave and sounds rougher than either the sine or saw-tooth waves. With the square wave you have the option of specifying the ratio of the up portion of the wave to the down portion. At either extreme, the square wave becomes more like a pulse wave, giving a 'pulsing' sound at lower pitches.
- **STS** is the very exciting STS wave which stands for 'Saw, Triangle, Square' because it is a hybrid between these three wave types. By varying the three parameters associated with it, the wave shape can be 'morphed' between the above three extremes. In effect, this waves covers all the preceding wave shapes, with the exception of the sine wave.
- **Random** produces random values between Min and Max, also affected by the Amplitude.

BPM Sync: Default (non-tempo synched option) and various BPM values, e.g. BPM/4 which means generate the waveform selected in the above list at a tempo of the mix tempo (BPM)/4. This control allows for tempo-synched control of e.g. the Filter unit, for great effect.

Random Levels (Checkbox): When checked, creates random control values bounded between the settings of the Min and Mix knobs. If a BPM setting is chosen in the Freq List, then these values will change at that control rate for a "sample and hold" effect.

Use MIDI note (Checkbox): Applies the waveform each time a MIDI note is played.

Min: This defines the minimum value that the generated LFO waveform can have.

Max: This defines the maximum value that the generated LFO waveform can have.

Phase: useful for slow oscillators (e.g. 0.2 Hz) - determines the start point of the oscillator.

U/D: Only for the **Square** and **STS** wave types. Defines the ratio of "Up" to "Down" (or "On" to "Off") values (for STS waveforms, this is the ratio of u to d in the diagram below).

Sqr: Only for the **STS** wave type. Defines the "squareness" of the waveform (the ratio of s to u in the diagram below).

Slant: Only for the **STS** wave type. Defines the skew of the waveform (the ratio of a to a + b in the diagram below)

- With the Up Down Ratio at 100.0%, the Squareness Ratio at 0.0% and the Slant Ratio at 50.0%, the shape is a triangle.
- With the Up Down Ratio at 100.0%, the Squareness Ratio at 0.0% and the Slant Ratio at 0.0% or 100.0%, the shape is a Saw

Right or Saw Left respectively.

- Lastly, with the Up Down Ratio at 100.0%, the Squareness Ratio at 100.0% and the Slant Ratio at anything you please, the shape is a Square wave.

The STS Waveform and its ratios

Ctrl-Amp

If you have a weak signal, coming from a filter for example, then pump it into an amplifier. The Amplifier can multiply the input by a factor of up to 5. If you want more than this, feed it into another amplifier!

The main controls in this unit are:

XY checkbox: Checking this will display an XY grid where up/down is volume and left/right is pan position of the signal (response times may be slow and "drag memory" long depending on device).

Gain: This determines the amount of amplification to be done by this amplifier unit. The default value is 1.00 (0dB - i.e. no amplification) and values can range from 0, 0.10 (-19dB) to 10.00 (20dB)

Pan: This determines the pan position of the signal. The default value is 64 (center).

Invert phase: Checking this box will invert the phase of the signal

Invert pan: Checking this box will invert the pan of the signal

Ctrl-Amp Unit

Ctrl-Random Unit

This controller simply generates a random value from -1 to +1 at the control rate (it is not an audio unit). There is no Unit interface, but you control the scale factor with the slider at the bottom.

The main controls (at the bottom):

- **Connector**: The "Connector" list lets you select the effects unit (which **must** be a control-rate unit) that is modulating the parameter for your effects unit. Change the entry here to be the unit number of the control-rate unit that you want to modulate your effects unit.

- **Controller**: The "Controller" list shows you a count of all the effects unit inputs that modulate parameters of your currently selected effects unit. The numbers in the list are just always numbered automatically from 1 upwards. Note that if there are no entries shown in the "Param" list, then this effect unit has no available modulators, and therefore you have nothing to control on it from e.g. LFOs or envelopes (in which case, Add etc. will have no effect).

- **Add**: Press the "Add" button to add a new entry to your "Controller" list.

- **Delete**: Press the "Delete" button to remove the currently selected Controller list item.

- **Scale**: The "Scale" slider lets you fine-tune a scaling factor between 0 and 2, by which the modulating input value (from the modulating unit) is multiplied, before being applied to your modulated effect unit's parameter. This allows you to carefully adjust the amount by which a parameter is modulated. Note that if more than one control-rate unit feeds a specific parameter on a target unit, they will get added together automatically (including the appropriate scaling factor); which means that control-rate junctions are not often required.

- **Modulate Parameter**: The "Modulate Parameter" list lets you select the effects parameter that you want to be modulated by the specified control-rate unit!

Junctions
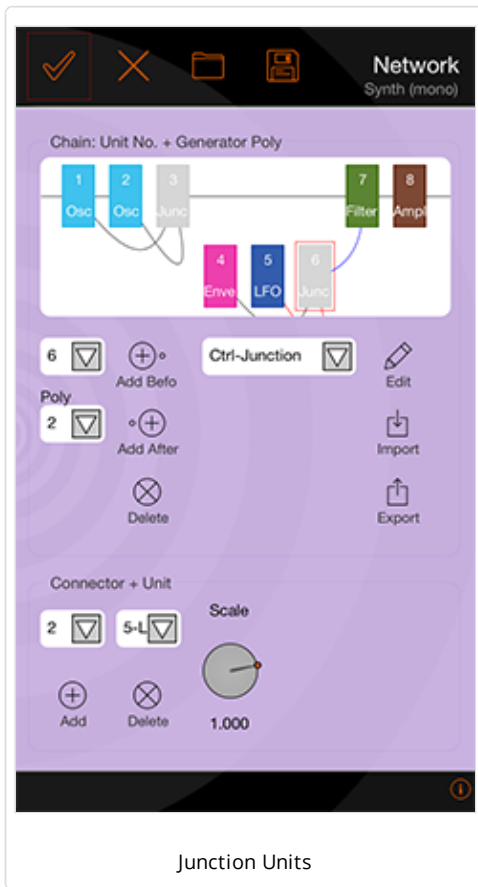
Junction Units

Junction units are what you use whenever you need to add or combine together the output from two or more units in the system. They are added from the Units drop list below the Chain display. They are shown in grey in the network and have no interface of their own.

There are two types:

- **Junction**: Audio-rate junctions for Tone Generators and FX (and available only in Synth networks). Use these when you need to add-together outputs from two or more signal-rate units. They display in the top "audio-rate" line of the network
- **Ctrl-Junction**: Control-rate junctions for Controllers (available in both Synth and FX networks). Use these when you need to add-together outputs two or more control-rate units. They display in the bottom "control-rate" line in the network

When a junction unit is selected in the top "Chain" display area, a "Connector + Unit" section displays at the bottom of the Network Editor screen and which lets you change junction inputs.

The main controls in this section are:

Connector + Unit & Scale: These only show if your currently selected unit is a signal-rate [audio] junction ("Junction") or a control-rate junction ("Ctrl-Junction").

- **Connector**: The "Connector" list shows you a count of all the effects unit inputs that are fed-in to your junction. The numbers in the list are just always numbered automatically from 1 upwards.
- **Unit**: The "Unit" list lets you select the effects unit (which must be a unit at the same rate as the junction!) that is to be added together to help create the output of the junction. Change the entry here to be the unit number of the unit that you want to add with other unit outputs.
- **Add**: Press the "Add" button to add a new entry to your "Input" list.
- **Delete**: Press the "Delete" button to remove the currently selected Input list item.
- **Scale**: The "Scale" slider lets you fine-tune a scaling factor between 0 and 2, by which the output value of your "input" unit is multiplied, before being added together with the other unit values that are feeding this junction. This allows you to carefully adjust the level fed-in by a particular unit. Note that if more than one control-rate unit feeds a specific parameter on a target unit, they will get added together automatically (including the appropriate controller scaling factor); which means that control-rate junctions are not often required.

Sine Wave
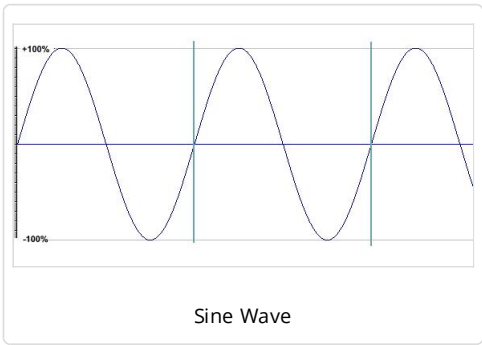
Introducing sound synthesis

PSE incorporates a very powerful and flexible software synthesizer. PSE and its pop-up plugin editors are a new audio application suite, that supports editing of effect plugin settings. The features and flexibility of PSE are such that, to make the best use of them you will find that a little understanding of the basics of sound synthesis will go a long way.

If you haven't delved too deeply into programming synthesizers before now, or if you fancy a quick refresher course, then this tutorial is for you. We will not be getting heavy with the physics and number crunching side of things here. We'll keep it practical and, hopefully useful. Let's go.

The naming of things

We have to start by establishing the meanings of the terms we will be using from here on in. Creating sound is all about finding new and interesting ways to shove air around. Here is an idealized illustration of what that might look like using a single pure tone.

Using this graphic we can identify some basic units and concepts in sound generation.

The waveform	The waveform is the wiggly line, obviously! This one is a sine wave.
A cycle	The area between the vertical blue lines in the graphic is a single cycle of this wave. In this instance it starts at zero, rises to the highest point, drop back down to zero, continues to the lowest point and rises up to zero again. You can start a cycle at any point of the waveform. It doesn't have to start from zero. The main thing to remember is, if you choose any arbitrary point on the waveform to start from, when you reach that point again, so long as you are traveling in the same direction as you were when you left it, you have completed a single cycle.
	If you start the waveform at some other point that the one shown here (say, from the highest point on the graph) you have changed the **phase** of the waveform. It doesn't make it sound any different if you listen to the tone on its own but there are good reasons for being able to change the phase of a wave if you wish. We'll get to them soon.
Frequency	Describes the number of cycles in each second. It is measured in Hertz (Hz), one hertz being one cycle per second. If the frequency is in our hearing range we perceive it as the pitch of a note. The human hearing range falls roughly between 20Hz at the low end and about 20,000 Hz at the upper end. If you can hear much higher than that then your owner needs to get you a dog licence. Human ears aren't very good at detecting pitch at the extremes of our hearing range but can be incredibly acute in the midranges.
Amplitude	Describes the range of upward and downward movement the waveform makes. We hear it as volume. Large amplitude values mean loud audio signals. Small ones mean quiet ones. If you reduce the amplitude of a wave you are said to attenuate it, if you increase it you are amplifying the signal. These amounts are typically measured in decibels (dB) which are a total pain to work with, mostly because they work to a negative logarithmic scale where minus infinity is the quietest point and zero is the point of optimal loudness. So we'll move quickly past.
	In a wholly analogue system, like a Stratocaster plus a Marshall stack, amplifying a sound beyond that optimal point gives the desirable sort of distortion sound that has been putting food on guitarists' tables for years. No such luck with a digital system. In the digital world you cannot take the amplitude of a waveform beyond 0 dB. If you try, the wave looks like it hit a brick wall and sounds ghastly!
	We mention this because putting sound into an amplifier is only one way of increasing its amplitude. Adding another sound source also increases the amplitude of the sound we hear. Which is why a full string section is louder that a solo violin.
	In sound synthesis it is not uncommon to use several sound sources to create a single sound. PSE has a built in limiter that manages sound levels internally for you to keep things out of the clipping zone. However, you will need

	to take some care of levels yourself if you want the best possible sound quality. A limiter working hard is often quite audible which might not be an effect you want to hear!
Zero crossing	If you think about it, moving air around systematically involves both pushing it and pulling it blowing and sucking. This is reflected in the graphic in that there is a line drawn through the centre of the waveform. Amplitudes above the line have a positive value, those below have a negative one. In the jargon, audio signals are usually **bipolar**.
	The point at which a waveform has an amplitude of zero is particularly important in sampling, something we will cover later. But the location of the zero crossing line has a significance in synthesis also.
	For convenience, we show the zero line in the graphic as bisecting the waveform so there are equal amounts of waveform both above and below the line. Things don't always have to be this way and there are ways for us to move this line up or down. To do so, in the jargon once again, we apply a DC offset to the waveform.
	As moving this line up or down will make no appreciable difference to the sound of the tone, why worry about it? If an audio signal is what you want, in truth it is not worth bothering with. But, if you want to use a waveform for something other than audio, like using one waveshape to control the parameters of a second then it can be very important. As we might find out later!

And that's about as much as we need to know about the components of a waveform. But there are a couple of other things we need to touch on to help understand how to synthesize sounds.

Fundamentals, harmonics and the rest.

If you play a note on any tuned musical instrument you will hear an astonishing complexity of sound. But, despite that complexity you will (hopefully) perceive a pitch. That's the **fundamental**. In all sounds that we perceive as pitched there is one frequency that stands out from all the other noises in there and this is the one the ear uses as the pitch reference. A piano, violin and oboe sound totally different to each other but, if they all play note A4 you will hear three distinctive sounds all having a common fundamental frequency of 440 Hz or thereabouts.

If you listen more closely to the sound of a single instrument playing a single note, once you get past hearing the fundamental you will hear a range of other tones. Some of these work in a musical way and sound rather like chordal notes related to the fundamental. Others have a more uneasy relationship to the fundamental or are completely atonal.

The musical ones are **harmonics**. They sound musical because they have a very precise mathematical relationship with the fundamental frequency. The frequency of the harmonics are **always a whole number (integer) ratio** of the fundamental frequency.

Just to confuse matters slightly, the fundamental is also called the first harmonic. So, a tone with a fundamental frequency of 100Hz will have the second harmonic at 200Hz (an octave), the third at 300Hz (octave plus a fifth), the fourth at 400Hz (two octaves) and so on to the limits of our hearing.

The other sounds that you can hear are called **aharmonics**. These are frequencies that have a **non integer ratio** to the fundamental. When there are a lot of these present we tend to perceive the sound as clangorous or bell like.

If these three elements were all present in a sound in similar proportions we wouldn't hear a musical note at all. We would hear noise. So, to make musical sounds we need to find some way of establishing a balance between the fundamental, the harmonics and the aharmonics for any given frequency. And that's what synthesis is all about.

There are three major routes to doing this. One is that we can start off with basic sounds that are very rich in harmonics and then use a filter to reduce or remove the ones we don't want. Oddly enough, that is called subtractive synthesis.

We can also do the exact opposite. We can take very simple tones and add them together to create complex sound. Additive synthesis, surprisingly hard to do well.

The third route is a kind of middle way between these two. If the properties of one waveform can be used to vary those of another at audio frequencies, new and complex waveforms can be generated. You could call this synthesis by modulation. The most common methods used are amplitude modulation and frequency modulation.

In reality we mix and match, using elements from all three main routes as we need them.

Blocking it out

The components of most synthesizers can be categorized into four broad groups.

The first group could be called **sources**. These are the devices that produce the raw sound you will work with. PSE allows you to use samples as sources as well as including some very well featured tone generators.

The next group we can call **modifiers.** Included here would be envelopes that shape the sound over time and filters that remove or emphasize certain frequencies.

The third group are **modulators**. They apply regular, repeated change over time to specific sound parameters. The most common of these is the low frequency oscillator, the LFO.

Finally there are the **effectors.** These are signal processing devices, reverberation modules, delays and so on. Usually they act at the end of the synthesis chain.

It is usual to define the output from modifiers and modulators as **control signals**; being as their usual job is to control the parameters of other devices in the synthesis chain. In the days of monster analogue modular synths it was considered sensible to use different coloured patch cords for control signals to distinguish them from audio and other signals. When using PSE it is also important to distinguish control signals from the audio path. We'll explain why in a moment.

For now, let's have a look at the first three device types in turn.

The right wave for the right job

Any self-respecting synthesizer will have a tone generator offering several basic waveforms as a starting point for sound creation. Why? What distinguishes one waveform from another? The answer lies in the harmonics on offer. So lets just quickly run through the more common waveforms and see what distinguishes one from the other.

Sine wave	Is a pure tone. It has no harmonics at all so there is not much point in applying a filter to one. Nothing to filter! This purity of tone make it very good for adding low end definition or kick to a bass sound. It is also good for additive synthesis where you don't want harmonics unless you put them there yourself. Put three or four of these waves together for an instant electronic organ type of sound.
Sawtooth wave	Is the most harmonically rich waveform in the box. Characteristically bright and buzzy. The starting point for thousands of classic synth sounds. A sawtooth wave has every harmonic present (theoretically) but their amplitude decreases from that of the fundamental by 1/the harmonic number. So the second harmonic is ½ as loud as the fundamental, the third harmonic is 1/3 as loud as the fundamental and so on until you hearing fails.
Triangle wave	Really just a sine wave straightened out. It doesn't have many harmonics and those that are present are quite high up. So, if you want hard and aggressive sounds this is not the wave to choose. Nice for flute sounds and soft leads though.
Square wave	A bit complex this one. A graph of a true square wave looks rather like a child's drawing of castle battlements. Square wave is a shorthand way of saying "pulse wave that spends as much time at the highest point as it does at the lowest". In this form it has a sound that is usually described as "hollow". The reason is that every second harmonic is missing, there are only odd numbered harmonics present.

But this absence of even harmonics only holds true if the time between "high" and "low" in the cycle are the same. If the wave spends half of its time at the top the ratio between the high points and the low points of the waveform is 1:2 and, as we already know, every second harmonic is missing. If we change this ratio to 1:3 (i.e. the wave spends 33.3% of its time at the top) some of the missing harmonics return. We now only loose every third harmonic instead. If the ratio was 1:4 (25%) we would only loose every fourth harmonic. And so on.

If there was a way of shifting this ratio in real-time we would be able to hear these harmonics coming and going. We could call it pulse width modulation and use it to recreate classic synth string sounds, all kinds of shimmery pad like sounds and some unforgettable bass tones. Sounds like a plan! |
| Combination waves | Unsurprisingly, combination waves can have the characteristics of most of the above. PSE includes one waveform that can be "morphed" from triangle to sawtooth to pulse wave. If you've followed us this far you should have realised that this gives you the option of tailoring the harmonic content of the waveform quite closely; a very powerful option indeed. Similarly, being able to change the shape of a wave in real-time can open the door to a whole new bag of sonic tricks. Pulse width modulation on steroids! |

Envelopes

Envelopes give us a way of shaping a sound in time. Every synthesizer will always have at least one to control the amplitude of a

signal. More sophisticated synthesizers will have several envelope generators which can be assigned to control other important parameters.

Envelopes are usually "one-shot" devices. An event triggers their start and, once underway, they transmit values that correspond to the envelope shape until they are done. They then do nothing until they are triggered again. Amplitude envelopes are usually triggered by the equivalent of a note being pressed on a keyboard.

Envelopes can be described by the stages they go through. The PSE amplitude envelope is a multi-stage envelope, having separate stages for **Attack** (how quickly the envelope moves from zero to maximum), **Hold** (how long it stays at the final attack level), **Decay** (how quickly the level falls to the..), **Sustain** (the lowest possible level during a note event) and **Release** (how long the sound will take to fall to zero from wherever it was when the note event stopped). There is a set of stages associated with when the note stops.

The control signals sent by the envelope unit in PSE can be bipolar; i.e. the control signal value can be positive or negative. This has got some implications if you want to combine control signals from more than one device.

Say you want to combine the output from an envelope and a LFO to create a LFO that fades up or down. As these could both be bipolar signals, shoving them into an adding unit (a simple mixer) won't work as you expect. Adding a minus value to a positive one is subtraction by another name so doing this will result in periodic signal cancellations and other unexpected behaviour.

If you recall your basic grade maths, you'll remember that a negative (minus) value multiplied by a positive always gives a negative result. So, if you want to combine bipolar control signals, use a negative scaling factor on one or more of your control-rate junction input scale factors!

Filters

Amongst some synth nerds, filters can acquire a mythical status, becoming objects to be worshipped or argued about into the small hours. This is rather off-putting for the rest of us and obscures the fact that, from a users point of view, they are actually rather simple devices.

There are only four things you need to know about a filter; its shape, its slope, its cutoff point and whether it is resonant.

The shape is usually what gives the filter its name. So it is a safe bet that a low pass filter will allow low frequencies through and exclude higher ones. Similarly, a high pass filter will do the opposite. A band pass will allow through frequencies that fall into a certain range and a band reject will allow everything through except frequencies in the defined range. Nothing mysterious about that.

The cutoff point is the frequency at which the filter will start to do its stuff. So a low pass with a cutoff point of 600Hz will start to attenuate anything over 600Hz but leave all the lower frequencies alone.

The slope of a filter simply determines how sharp the attenuation will be. It is sometimes expressed in dB per octave and sometimes in "poles". The famous Moog filter had a 4 pole slope which equates to a reduction of 24dB per octave. If the earlier example of a 600Hz lowpass cutoff was a 4 pole type it would mean that, by the time we got to frequencies of 1200 Hz they would be attenuated by 24dB compared to the ones at 600Hz. This is quite a steep reduction and would leave very little audible signal by the time we got beyond 1500Hz. Something more gentle, like a 2 pole slope would only attenuate frequencies by 12dB per octave. So you would hear more of the higher harmonics.

Filter resonance (sometimes called Q for reasons that don't matter) is also pretty straightforward. All this does is emphasize the frequencies around the cutoff point. It is a kind of feedback loop. The higher the Q the more pronounced are the sounds at the cutoff frequency. On some old analogue synths you could crank this up so high that the only thing you could hear was the cutoff frequency so the filter would start to behave like an oscillator. This was called self-resonance. It is not so easy to do in a digital system.

And that's filters really. If they were fix and forget devices they would be little more than glorified tone controls. But, if we can use envelopes or LFO's to change the cutoff frequency or resonance in a dynamic way, then they are the heart of a subtractive synthesis system. Hence all the attention they get.

Modulators Part 1 Slow and gentle

We've made passing reference to it before but it is now time to get into some detail about modulation. It is a huge topic because there are so many possibilities. The skillful use of modulation techniques is probably the single most important factor in getting dynamic, expressive, musical sounds out of a synthesizer.

We'll start with a definition. In synthesis, modulation is the process of using one signal to apply regular, usually cyclical change to one or more parameters of a second signal. Lets look at a simple practical example.

A violinist often gives expression to a piece by adding vibrato. When he or she waggles their finger on the fretboard the net result

is that they are making small regular changes to the fundamental frequency of the note they are currently playing. We can do exactly the same thing with our instruments.

To create vibrato on a synthesized voice we simply apply small regular, cyclical changes to the oscillator frequency. We do that by routing the output of one, low frequency (i.e. slow) oscillator to the frequency controls of the main, audio oscillator.

The change this will make depends upon the properties of the signal sent from the low frequency oscillator (LFO). And, when we use a LFO as a controller of other parameters, some things that are not important to the sound of a waveform become very relevant indeed.

There are five aspects to a LFO waveform that are important to consider; the actual shape of the wave, its frequency and amplitude, its phase and its DC offset.

Frequency and amplitude are quite easy to come to terms with. The higher the frequency of the LFO the more changes per second will be made. Amplitude is an interesting one. In our violin example, sending a 100% amplitude sine wave from the LFO to the tone generator frequency control would not give us vibrato. It would give us the sonic equivalent of seasickness. You use the amplitude controls of an LFO to determine how much change is to be applied. For vibrato, very small amplitudes around 3% will do fine.

It is useful to be able to visualise the wave shape of an LFO. If we are generating a sawtooth wave at audio frequencies then the direction of the sloping part of the wave is irrelevant to the sound. A wave that has a slope to the left sounds the same as one with a slope to the right. However, if we want to use an LFO to make gradual change to one parameter, then fall down to the beginning and start again, we would not want to use a waveform with a left facing slope. Visualise it!

Phase and DC offset are related to some degree. Lets consider phase first.

In the vibrato example, for the main oscillator to remain in tune we need the LFO to start from the zero crossing point. If it started at any other place it would automatically add something to the main oscillator and cause it to sound out of tune.

There might be other occasions when we want the effect that comes with starting the LFO at somewhere other than zero. In those circumstances we would adjust the phase to suit our purpose. Again, the best advice is to try to visualise what you want to achieve, then program it accordingly.

Finally there is the DC offset for the LFO. To go back to our violinist (for the last time, honest!) a vibrato effect that shifts the fundamental pitch up and down by equal amounts is not very natural sounding. It actually sounds far more realistic if we push the frequency in one predominant direction. The way to do this with an LFO is to shift the DC offset.

Think about it like this. The LFO is sending numerical values out to be added to the frequency of the main oscillator. As a bipolar signal with no DC offset, half of these values will be positive and half will be negative. Changing the DC offset will shift this balance. If we only want the vibrato to work up from the fundamental we would need to shift the zero crossing point right down so that the LFO sent out only positive values.

This can be hard to visualise just by applying a numerical offset value so PSE makes it very easy by giving you an option to set the ratio between positive and negative values transmitted by the LFO using two sliders. Nice!

There is a catch to doing this though. If you change the DC offset it will obviously have a non-zero value. Less obviously, if you sent this DC offset LFO to the pitch control of another oscillator you will automatically add the DC offset value to it, taking the oscillator out of tune! So, if you are shifting DC offsets in this way you need to retune the destination oscillator to compensate.

Exactly the same principles apply when using LFO to modulate parameters other than oscillator frequency. We've mentioned how the harmonics present in a pulse wave change depending upon the pulse width ratio. If you route a slow LFO to modulate the width of a pulse wave type oscillator you get a rich shimmering sort of sound as harmonics come and go that has been the basis for synth string patches since forever. It also gives some astonishing bass sounds in the lower registers. Pulse width modulation was actually hardwired into the infamous Moog Taurus bass pedals so beloved by `70's prog-rockers.

Incidentally, if you are using an LFO to modulate pulse width you will need to attend to its amplitude. Too much modulation and you can get to the stage where there are no harmonics at all - not even the first one! Silence is not always golden.

Modulation by slow, sub-audio oscillators is not too difficult to get to grips with. But there is no rule that says that modulators always have to be inaudible. However, what happens when you crank the modulator signal up into the audio range can get pretty wild. One option is to use a device called a **Ring Modulator**.

With this device you simply take two sound sources and plug 'em into it. In effect what happens after that is that the first audio signal gets spliced with other signal at the frequency of that second signal. What you end up with is a new signal that consists of the sum and the difference of the two incoming signals. Which is OK if the two are pure sine waves. But if they have harmonics attached...:)

At low modulator speeds a ring modulator just chops up the sound in quite an obvious way. This was how they made the Daleks speak! But at high modulator speeds you can get all manner of crazy, unpredictable effects, especially if the modulator frequency

is either fixed or changes in a way not related to the carrier. If you want one of those "car crash in a steel foundry" moments (and who doesn't) head for the ring modulator.

As we are talking about high speed modulation it is probably worth mentioning something that **isn't** going to work in the way you might think. At this point we need to get a bit technical about the workings of PSE. Remember what we said about distinguishing control signals from audio signals? Here's why you need to do it.

PSE is a digital synth (obviously!) so everything going on under the hood has to have a sample rate to work to. In order to save resources for the things that matter, PSE gives absolute priority to rendering the audio signal at a reasonable sample rate, typically 22Khz on a modest PC. To save CPU time, **all control signals are rendered at a very low sample rate**. The default rate is 100Hz..

In most circumstances, this low sample rate isn't an issue. You don't need to hear the actual output of an LFO or envelope unit so there is no point in rendering it at CD quality when we can do better things with the resources available. However, it does mean that you can't just stick the output of one tone generator into the frequency control input of another because the (incoming) modulating signal will be interpreted as being a control signal and so will be capped at the control signal sample rate frequency.

Going granular

Granular synthesis is a relative newcomer as a synthesis method, mainly because the tools and processing power needed to do it easily haven't been readily available until recent times. As powerful computers tended to be in academic institutions it gathered a reputation for being "something for the boffins". While the underlying maths might be the stuff of nightmares, the concept of granular synthesis is dead easy. Here goes.

If you take a very small section out of a waveform (a few milliseconds at most), add an similarly small section of silence and loop play the result at audio frequencies you get an entirely new, and rather complex waveform. The audio content of this waveform will be determined by several different things; the size and content of the original mini sample (the grain), the length of the silence, the nature of the boundary between the two (is it abrupt or does it fade and by how much?) and the playback speed.

If you think about it, this is not a million miles away in principle from good old amplitude modulation. But then we take it to another level!

If you move away from using a single grain to using multiple grains, all with different frequencies, or if we modulate the grain size and/or the grain end crossfades it all gets very strange. You very soon start to generate mutated, wholly original sounds that are not easy to achieve by other means. Which is what the buzz about granular synthesis is all about.

Now we wouldn't be telling you all this if there wasn't a way to do it with PSE. One of the modules available is a "particle generator" and mainstream this is not! What this module does is generate up to twenty little bits of simple waveforms which are then used as the grains in creating a more complex overall sound. You control a range of the more useful and interesting parameters, most of which can be modulated by other PSE modules.

What comes out of the business end of this module is not always predictable. It is very easy to get it to generate complex, untuned warbling or twinkling sounds. With a little more effort you can make some breathtaking, animated, tuned soft lead sounds. As it is hard to describe the indescribable the best way to learn about this little beastie is simply to play with it. Think about it as a hands on introduction to chaos theory!

A quick word about hybrid synthesis

The use of "real world" sounds instead of tone generators as the starting point for sound synthesis is hardly a new concept. But it wasn't until the 1980's and the development of affordable digital recording technology that the full potential of this could begin to be realised. The ability to process and manipulate recorded sounds using the familiar shaping and modulating tools is a form of hybrid synthesis.

The simplest implementation of this can be seen in most samplers which basically slap a subtractive synthesis engine onto a digital sample playback device. You can do this with PSE, no problem. And we will. Later.

But that is more than enough background for this tutorial. We can now take this knowledge forward with us and explore PSE in more practical terms by starting to build our own instruments within PSE.

Synth Tutorial 2

Working with the PSE Modular Synthesizer

Try launching PSE. This is the place where you activate and program PSE … and where MIDI sound creation for the Intermorphic Sound System (ISS) gets very interesting!

Note that the audio plugin framework that underpins PSE is an open framework, and has been designed to allow 3rd party developers to create application UIs into which they can plug in their own or other 3rd party sound processing modules, including support for the Intermorphic modules if so required.

When you start using PSE for a MIDI line, what you have is an independent, non-midi, polyphonic synthesizer The sound generation limitations of MIDI pretty much go out of the window.

PSE is no lightweight. It has a degree of flexibility that is comparable to the old, monster modular synths. It includes some features that are seldom seen, even on very expensive, "famous name" hardware synths. Inevitably, something so well featured and flexible isn't able to hide its complexity too well so there will be a bit of a learning curve to negotiate if you are going to be able to harness this power in a creative way. This tutorial aims to help you up along that curve.

We won't be creating full pieces of music in this tutorial; instead, we'll concentrate on programming sounds using PSE.

The Path

To get the best from any synthesizer you really need to have some understanding of the path the various signals take inside it. If you have had some experience of hardware synths you will probably be used to imagining a left to right signal path, with the sound generating oscillators at left hand side and the sound output stage at the far right. PSE signal path pretty much follows this convention…

- Noatikl: Click on the Synth button. This displays the Synth Network Editor Dialog.

- Mixtikl: Click on the Synth Module button, you will see that the Synth Attacher Dialog appears. This dialog lets you review the modular synth settings for each of the MIDI lines in the piece. Select an active MIDI line number, and then set the "Poly" list item to be 1, and press the edit button. This displays the Synth Network Editor Dialog.

It is here, in the Synth Network Editor Dialog dialog, that we design the sound for the selected MIDI line. As you add synth units to your sound design, they appear as boxes in the bottom part of the dialog. We sometimes refer to these units as "slots"; you can have as many "slots" as you want, where you insert the various modules and units that, together, will make your synthesizer module. The sound you eventually hear will come **only** from the signal rate unit in the rightmost slot. So, if you were to construct the simplest of synthesizers, using one tone generator and a LFO, you would have to put the tone generator in a slot somewhere to the right of the LFO. Put them the other way round and, if it was fast enough, all you would hear would be the LFO!

With the proviso that the sound comes out of the right most unit, the signal-flow is pretty strict in terms of left-right ordering. Signals from one module can only ever be passed to modules that are to the right of it in the design. In practice you must put control-rate units (your controllers and shapers, such as envelopes and LFOs) sound BEFORE the sound generating modules (i.e. before your tone generators). So, if you want to use an envelope to control the output from a LFO, you'll have to put the envelope to the left of the LFO. This does of course help you keep a handle on what your signal path is doing. With practice, you won't be confused.

Control signals are different.

Says it all really.

PSE makes an important distinction between audio signals and control signals. Here's how it works.

Most signals in the system are "audio-rate" signals; they are rendered at whatever sample rate the platform is running at, for example 22Khz.

The **only** signals that can be used to modulate parameters of a synth unit (e.g. to amplitude or wave shape) are control-rate signals, that must come from a control-rate unit. To save CPU resources for the things that matter, control signals are rendered at a very low sample rate; typically 100Hz. So, if you try to modulate a parameters between units at audio frequencies the best you are going to get is 100Hz. This is not going to be good enough for specialised applications such as FM synthesis!

The reason that we do this, is that it saves a lot of CPU horsepower; to render subsonic waveforms at near CD quality would be

pretty pointless. You don't want or need to hear the direct output of an envelope unit or a slow LFO.

On tuning

A fixed architecture synthesizer with limited flexibility can hide a lot of things from the end user. Because PSE has been designed for flexibility it is up to you to take care of some of the details to make sure that the results you get are the results you expect. This is particularly the case when making sure your MIDI line is in tune with the others in a piece.

The thing that is most likely to catch you out is when you use a LFO to modulate the frequency of a tone generator. If you use the LFO with most of its settings at the default there will not be a problem. However, if you adjust the min or max value sliders you will have a tuning problem that will need correction. Why?

If you recall in the synthesis tutorial, we said that changing these values was the same as applying a DC offset to the LFO wave, i.e. the zero crossing line has a non-zero value! So, if you route a LFO that has been shifted in this way to the frequency of another oscillator you are, in effect sending two values. One is the LFO amplitude, which will change over time and the other is the DC offset which is a constant. This DC offset will have to be allowed for by adjusting the pitch of the sound generating oscillator to get it back into tune.

So, if you have an out of tune MIDI line, check what is modulating the frequency!

Polyphony - a word about resources

All soft synths are serious processor hogs. A lot of effort has gone into minimising the hit that PSE will make on your processor but you can't cheat physics. Making complex sounds in realtime means doing hard sums very fast and there will be a limit to the strain your CPU can take.

Note: The polyphony as synth will play with is set in the Network Editor. It may not be entirely obvious (especially as the label does not always display), but the Polyphony ("Poly") drop down is to the left of the "Add Before" button.

Tips

- If your piece can use built-in wavetable (e.g. for basic drum sounds), or custom audio samples, then consider using them where possible rather than the modular synth. Remember that sample-based drum sounds can be very effective.
- **Very importantly**, use a polyphony value that is as small as your piece can get away with!
- **Wherever possible**, put your effects in the effects line (see the Network Editor) rather than directly in your synth module design.
- Be economical with units. Think about ways in which you can maybe use one unit more than once in a voice.

Finally, a word about polyphony. With PSE, each MIDI line can become an independent, polyphonic synthesizer.

But rather than always thinking of PSE as a polyphonic synthesizer, you can think also think about it as a synthesizer that has lots of multiple instances a monophonic synthesizer If you set the "Poly" parameter to 2 or more, what PSE does is create the corresponding number of **identical** monophonic synthesizers that can operate simultaneously, one for each possible note. So, setting your "Poly" parameter high is an excellent way to use up your processor resources - which you will want to try to conserve! Use it with care.

OK, enough of the preliminaries. Lets get on with making some sounds. The terminology and concepts we will be using will assume you are up to speed!

Voice 1 Simple subtractive synthesis

This is how to set up PSE as a simple two oscillator synth in the classic style. We'll go through this in some detail because, once you have got the principles of PSE established, working in more adventurous ways gets much, much easier.

Use your favourite sequencer to setup a simple MIDI piece that plays long notes with a start pitch of 30 or thereabouts to give us something to test the sound with. Take your piece, Open it with PSE and you should hear it start playing.

First up, you need to activate the Modular Synth for the voice in question. Launch the Synth Network Editor Dialog dialog to do this; set the poly to be 1 or more. Once you set a tone generator in your synth module design, this will then (and only then!) override any midi instrument settings that your piece is using.

So: we need to design the module that our MIDI line will use. As we are going to make a classic kind of synth we will need two tone generators, a filter and an envelope to control it.

Click on the Edit button for your line, and up pops the effects editor dialog. The exact content of this window will vary depending on the module you are using but they all follow the same basic conventions as noted earlier.

Click on "Type?" to change the effect type of the unit that was automatically created for you (unit 1); make this "c/envelope", which is a control-rate envelope. Then, click on the "Add After" button and change the Type of the new Unit (unit 2) to be "tg/osc" (a very flexible Tone Generator!). Select "Add After" to add another unit (unit 3) and also make this a "tg/osc" tone generator. Finally select "Add After" to add another unit (unit 4) and set this to "filter".

Using a Signal-rate Junction

Hold on - the filter is only fed from the output from the unit immediately before it; and what we really want is to feed the filter with the combined outputs from units 2 and 3! To do this, we need to route the two tone generators into the filter unit. How we do that is fundamental to getting PSE to work for you!

Select unit 4 (the filter) and press the "Add Before" button; your Filter is now moved to become unit 5! Set the type of the new unit (unit 4 - which is the *new* unit) to be "j". This is a signal-rate junction unit, which lets you add signal rates unit together. In the "Junction Inputs" area, press the Input "Add" button twice. Make sure that the first input item of the two inputs you've just added comes from Unit 2 (the first tone generator), and make sure that the second of the two inputs comes from Unit 3 (the second tone generator). You can set relative scaling factors for the two input units by playing with the scaling factors.

Here is the text that we get if we now press the "Export" button on the synth module editor window (we could subsequently reimport this if we need to by highlighting the text, copying it and pressing the "Import" button):

```
<fxm>
<unit t="c/envelope" r="c"/>
<unit t="tg/osc"/>
<unit t="tg/osc"/>
<unit t="j" i="2,1.;3,1."/>
<unit t="filter"/>
</fxm>
```

The signal-rate junction you've just added takes the the signals from the two tone generators, and adds them together. The output from your signal-rate junction automatically feeds the filter unit to its right.

OK, that's the network built. Now lets fine-tune our settings.

Select Unit 2, and click on the Edit button. This displays the Oscillator editor dialog. There is a list down box here where you can select the wave shape for this tone generator. We want a saw wave. The direction of the slope isn't relevant. Close the dialog, select unit 3, press the Edit button and get ready to edit the second tone generator!

Select a sawtooth wave for the second tone generator. Please make sure its direction is the same as its twin in unit 2 or else they will cancel each other out and you won't hear much! :)

To make a big sound, set one of the tone generators to work an octave lower than the other. You can do this with the one in Unit 2 by setting the Octave offset to 1.

Once you've done all that, select Unit 5 (the filter) and press the Edit button to display the Filter editor. Set the filter type to "low pass" and adjust the cutoff frequency and Q (resonance) to taste. The filter will automatically sweep unless you tell it otherwise! :)

Here's what the exported module now looks like:

```
<fxm> <unit t="c/envelope" r="c"/> <unit t="tg/osc"p="1280=2;1287=1;1030=1;1031=1;1026=0
;1033=0;1034=0;1028=400000;1281=1.;1282=0.;1283=50.;1284=50.;1285=50.;1288=-1.;1289=1.;1
035=0.;1040=0;1042=10;1044=10;1046=50;1048=10;1050=0;1052=400;1054=100;1056=50;1058=0;"/
><unit t="tg/osc"p="1280=2;1287=1;1030=1;1031=1;1026=0;1033=0;1034=0;1028=400000;1281=1.
;1282=0.;1283=50.;1284=50.;1285=50.;1288=-1.;1289=1.;1035=0.;1040=0;1042=10;1044=10;1046
=50;1048=10;1050=0;1052=400;1054=100;1056=50;1058=0;"/><unit t="j" i="2,1.;3,1."/> <unit
t="filter"/></fxm>
```

Next, lets use the control-rate envelope (unit 1) to modulate the frequency of our first tone generator (unit 2). Select unit 2, and and in the "Parameter Controllers" area add a controller by pressing the "Add" button. Set the source Unit to be unit 1 (our envelope), and set the Param(eter) to be Frequency. Your envelope will now be modulating the frequency of your LFO! Use the Scale slider to adjust the amount of modulation. Select unit 1 and press the Edit button to modify the shape of your envelope.

Note that the envelope editor display might look a bit complex to start with, but it it's not really! It can be used to send out negative as well as positive values (it is bipolar!).

For example, you might want an envelope shape that rises slowly to a maximum, falls down to a low level rather slowly and tails off to nothing once the note has ended. So you could set the attack time to around 5 seconds, the decay time to just less than 4, a very low sustain level and a release time of half a second.

Once you have an envelope shape you like it is then just a matter of playing around with things until you get the exact sound you want. Note that you can also use control-rate LFOs ("c/lfo") to modulate parameters; and many units have a large range of parameters that you can experiment with modulating!

Anyways, back to our example: you should hear a more familiar, if rather overused, sound. If you go back to one of the tone generators and tweak the "Micro Offset" value a bit you will find that fattens up the sound nicely.

For a final touch you might want to add some reverb. This is best done by using it as a global effect for the entire piece because you might have several voices that you want to treat similarly.

If you close the Synth Module Editor dialog, and then close the Synth Module dialog, you'll be back to the main application window. Press the "Effects" button, and once the Effects dialog opens, and press the topmost Edit button. Up pops a dialog just like the Synth Module editor you saw before, but this is now being used to edit global effects for the entire piece. Make sure that the effect type is "reverb", and edit the unit to get the sound you want. Then, close the dialog.

And there we have it. One classic, filter swept pad in the traditional subtractive style. Easy!

Now we have a sound that works it is a good idea to save it for use in other pieces. As noted above, you can export & import data directly via the clipboard and a text editor at various levels of the tool; this makes copying sounds around from piece-to-piece very easy.

If we export the settings from your Synth Module dialog; the text should look something like this (depending on exactly what you did!). We need this by the way for our next tutorial step.

```
<fxms l="0" p="1"> <fxm> <unit t="c/envelope" r="c"/><unit t="tg/osc"p="1280=2;1287=1;10
30=1;1031=1;1026=0;1033=0;1034=0;1028=400000;1281=1.;1282=0.;1283=50.;1284=50.;1285=50.;
1288=-1.;1289=1.;1035=0.;1040=0;1042=10;1044=10;1046=50;1048=10;1050=0;1052=400;1054=100
;1056=50;1058=0;"/><unit t="tg/osc"p="1280=2;1287=1;1030=1;1031=1;1026=0;1033=0;1034=0;1
028=400000;1281=1.;1282=0.;1283=50.;1284=50.;1285=50.;1288=-1.;1289=1.;1035=0.;1040=0;10
42=10;1044=10;1046=50;1048=10;1050=0;1052=400;1054=100;1056=50;1058=0;"/><unit t="j" i="
2,1.;3,1."/> <unit t="filter"/></fxm></fxms>
```

Voice 2 Waveshaping

This sounds similar in some ways to Voice 1. But this time, instead of using a filter module to remove harmonics and animate the sound we will use an envelope to change the waveform shape over the duration of each note.

If you recall some of the things we covered in the previous tutorial you will realise that changing the waveshape will change the harmonic content of the sound. And, if we get it right, we'll end up with something that sounds like a traditional filter sweep without the CPU hit that the filter involves. So, really, we are almost using additive methods to make this sound. And it is more economical. Great!

Go back to your MIDI sequencer and copy the MIDI line you prepared earlier to a new line. Alternatively, start from our example piece "tutorial/tutorial2.noatikl". Most of what we need is already set up for us and there is no point in working too hard! Now, Open this piece into PSE.

Open the Modular Synth dialog, and import the settings from the last voice:

```
<fxms l="0" p="1"> <fxm> <unit t="c/envelope" r="c"/><unit t="tg/osc"p="1280=2;1287=1;10
30=1;1031=1;1026=0;1033=0;1034=0;1028=400000;1281=1.;1282=0.;1283=50.;1284=50.;1285=50.;
1288=-1.;1289=1.;1035=0.;1040=0;1042=10;1044=10;1046=50;1048=10;1050=0;1052=400;1054=100
;1056=50;1058=0;"/><unit t="tg/osc"p="1280=2;1287=1;1030=1;1031=1;1026=0;1033=0;1034=0;1
028=400000;1281=1.;1282=0.;1283=50.;1284=50.;1285=50.;1288=-1.;1289=1.;1035=0.;1040=0;10
42=10;1044=10;1046=50;1048=10;1050=0;1052=400;1054=100;1056=50;1058=0;"/><unit t="j" i="
2,1.;3,1."/> <unit t="filter"/></fxm></fxms>
```

Open the module editor for line 1, export the settings; close module editor, select line 2, open the module editor for line 2 and set poly to 2. Press the "Import" button and both lines are now playing the same way! Now, Select the filter unit and press the Delete

button to remove it. The last unit is now the signal rate unit, which just adds the two tone generators together and is a lot easier than implementing a filter in software!

Open up the Editors for the two tone generators. Change the wave type for each of them to STS. You'll notice that you get some interesting control sliders for this wave type.

Here is the synth module definition at this stage:

```
<fxms l="0" p="1"> <fxm><unit t="c/envelope" r="c"/>  <unit t="tg/osc" p="1280=2;1287=1;
1030=1;1031=1;1026=0;1033=0;1034=0;1028=400000;1281=1.;1282=0.;1283=50.;1284=50.;1285=50
.;1288=-1.;1289=1.;1035=0.;1040=0;1042=10;1044=10;1046=50;1048=10;1050=0;1052=400;1054=1
00;1056=50;1058=0;"/><unit t="tg/osc" p="1280=2;1287=1;1030=1;1031=1;1026=0;1033=0;1034=
0;1028=400000;1281=1.;1282=0.;1283=50.;1284=50.;1285=50.;1288=-1.;1289=1.;1035=0.;1040=0
;1042=10;1044=10;1046=50;1048=10;1050=0;1052=400;1054=100;1056=50;1058=0;"/>  <unit t="j
" i="2,1.;3,1."/>  <unit t="filter"/></fxm></fxms><fxms l="1" p="1"> <fxm><unit t="c/env
elope" r="c"/>  <unit t="tg/osc" p="1280=5;1287=1;1030=1;1031=1;1026=0;1033=0;1034=0;102
8=400000;1281=1.;1282=0.;1283=50.;1284=50.;1285=50.;1288=-1.;1289=1.;1035=0.;1040=0;1042
=10;1044=10;1046=50;1048=10;1050=0;1052=400;1054=100;1056=50;1058=0;"/>  <unit t="tg/osc
" p="1280=5;1287=1;1030=1;1031=1;1026=0;1033=0;1034=0;1028=400000;1281=1.;1282=0.;1283=5
0.;1284=50.;1285=50.;1288=-1.;1289=1.;1035=0.;1040=0;1042=10;1044=10;1046=50;1048=10;105
0=0;1052=400;1054=100;1056=50;1058=0;"/>  <unit t="j" i="2,1.;3,1."/></fxm></fxms>
```

The STS wave is one of the quietly outstanding features of PSE. The three parameters, controlling the up/down ratio, squareness and slope give you the capacity to morph a waveshape.

It might be a good idea to set up a simple voice just using this waveshape in a single tone generator and play around with the sliders to get the feel for what you can do with this. Basically, if you set the up/down to 100%, squareness to zero and right/left% to either 100% or zero you will hear a sawtooth wave. If you now move the right/left slider towards the centre you will hear the sound mellow until, at 50%, you get a triangle wave. Now move the squareness slider and you hear the sound harden again as the wave become more pulse like. We already know that changing the up/down ratio of a square wave (the duty cycle) will change the harmonic character in other ways. Check it out!

So, by varying these three parameters in different ways in real time you can generate some excellent harmonic changes without resorting to a filter. Lets do it!

We don't need to change the current Envelope settings, but you can play with this if you like. But we **do** want to route the envelope controller (unit 1) to the appropriate destination!

In the Parameter Controllers area, for both units 2 and 3, add a controller from unit 1. Ensure that this new controller is used to modulate the Up/Down Ratio parameter. Now do this again, adding new controllers, but this time to use unit 1 to also modulate the squareness ratio. Then do this again for the slope ratio!!

As this is the "suck it and see" school of synthesis, the next bit is down to you and your ears. Hit play and adjust the settings levels on these parameters until you get a sound you like. You'll find that, if you overdo the levels you can flatten the waveshape out and loose the sound altogether!

Here is the synth module definition at this stage:

```
<fxms l="0" p="1"> <fxm> <unit t="c/envelope" r="c"/><unit t="tg/osc"p="1280=2;1287=1;10
30=1;1031=1;1026=0;1033=0;1034=0;1028=400000;1281=1.;1282=0.;1283=50.;1284=50.;1285=50.;
1288=-1.;1289=1.;1035=0.;1040=0;1042=10;1044=10;1046=50;1048=10;1050=0;1052=400;1054=100
;1056=50;1058=0;"/><unit t="tg/osc"p="1280=2;1287=1;1030=1;1031=1;1026=0;1033=0;1034=0;1
028=400000;1281=1.;1282=0.;1283=50.;1284=50.;1285=50.;1288=-1.;1289=1.;1035=0.;1040=0;10
42=10;1044=10;1046=50;1048=10;1050=0;1052=400;1054=100;1056=50;1058=0;"/><unit t="j" i="
2,1.;3,1."/> <unit t="filter"/></fxm></fxms><fxms l="1" p="1"> <fxm> <unit t="c/envelope
" r="c"p="1040=0;1042=307;1044=0;1046=0;1048=0;1050=0;1052=400;1054=100;1056=50;1058=0;"
/><unit t="tg/osc"p="1280=2;1287=1;1030=1;1031=1;1026=0;1033=0;1034=0;1028=400000;1281=1
.;1282=0.;1283=50.;1284=50.;1285=50.;1288=-1.;1289=1.;1035=0.;1040=0;1042=10;1044=10;104
6=50;c="1#1290,1.;1#1290,1.;1#1293,1."/> <unit t="tg/osc"p="1280=2;1287=1;1030=1;1031=1;
1026=0;1033=0;1034=0;1028=400000;1281=1.;1282=0.;1283=50.;1284=50.;1285=50.;1288=-1.;128
9=1.;1035=0.;1040=0;1042=10;1044=10;1046=50;1048=10;1050=0;1052=400;1054=100;1056=50;105
8=0;"c="1#1290,1.;1#1290,1.;1#1293,1."/> <unit t="j"i="2,1.;3,1."/> <unit t="filter"/></
fxm> </fxms>
```

It shouldn't take you too long to find settings that sound like the filter sweep we created in the first example. If you check the two sounds side by side in the example piece you'll find that this approach actually gives you a greater range of harmonics than using a simple filter. The sound will tend to be richer and brighter, sitting up in the mix more perhaps.

Voice 3 Gentle Ring Modulation

We can use the ring modulator mode of the "osc" unit to give some quite pleasant quality tones that are very similar to those you get from basic amplitude modulation. The key is to make sure that both the tone generators track the pitch of the composed notes.

A ring modulator has to have two inputs for you to hear anything. If you just load up the ring modulator on its own you'll hear nothing. The PSE oscillator use the internal oscillator as one permanent input. All you need to do is route another source into the ring modulator and off you go.

Change one of your modules such that it is one "osc" unit followed by another "osc" unit. Be sure to check the "ring-modulate?" checkbox on the second unit; verify that the "Use MIDI Notes?" checkbox is also checked.

If you play with the various oscillator parameters you'll discover that the harmonic content of the sound generated depends mostly on how far apart the two tones being modulated are.

Voice 4 Ring Modulated weirdness

Go back to your Ring Modulating unit. Uncheck the "Use MIDI notes".

What you get is a weird, metallic noise! The quality of the noise will depend upon the relationship between the frequency coming in from the tone generator and that of the oscillator in the ring modulator. Adjust this frequency until you get a cleaner, bell like tone. It should be somewhere around 700 Hz.

If you want things to get dafter, add a control-rate LFO "(c/lfo")" to modulate the ring modulator's oscillator frequency. Try this. Add a control-rate lfo ("c/lfo") as the first unit. Set the wave type of this LFO to Random and the type to Step. Set the frequency to 2 Hz. Now add this unit as a modulator controller of the ring modulator frequency. Hit play! :)

What you should be getting now is weird, randomized electronic chime noises that change every half second. Strange, isn't it?

Don't forget that you can use custom user samples (e.g. in Ogg format) as the source as well as another PSE module so you can do ring mod sound warping to your own samples in realtime. And who wouldn't want to do that?

Voice 5 Particles. Making ring modulation look ordinary

Now if you thought ring modulated noises were strange, check out this option. As an alternative to a comprehensive tone generator module, the PSE packs the utterly unique Particle generator (if you'll pardon the pun!).

The concept is based on granular synthesis. This module generates little wavelets of sound punctuated by little sections of silence. With careful control of the comprehensive set of parameters on offer you can generate the most unworldly sounds.

A little alert before we start. This module does a lot of maths! Don't overuse it in a piece or there won't be much processor time left for anything else.

We'll use it to create an Sci-fi movie background music sort of thing.

Open piece tutorial3.noatikl, which is like tutorial1.noatikl but with a higher frequency (around 55). Assign a "tg/particle" unit as your only unit for MIDI line 1. And that's it! :)

Well, not really. The difficulty now is that this all gets very subjective and rather hard to write about. You just need to hit play and then adjust the many parameters on offer in the Particle module until you get a sound you like. So instead lets just check out the key variables.

The harmonic parameter is one of the most influential factors on the end sound. Low values will keep the base frequency of wavelets close to each other. Larger values will mean much greater differences between individual wavelets and a much wider harmonic range to the sound.

Frequency velocity is like a mini pitch envelope applied to each wavelet. Small values give nice detunes and sweeps. Larger values can cause queasiness!

The attack, sustain and decay parameters are like a mini amplitude envelope for each wavelet. Small values make each wavelet distinct and audible. Larger values will cause a smearing and blending of the sounds which is not unpleasant.

Pause governs the time between each wavelet.

The "Number of Elements" value sets the number of wavelets to be generated. Set this lower and reduce the processor hit for this module.

You will hopefully have noticed that every parameter can be modulated by another unit, so there is plenty of scope to sculpt some quite unique and complex pieces with this little baby!

Voice 6 The Drum Synthesizer

This one is a little bit different because, with this beast, you can achieve so much. So some explanation is in order first.

To make decent percussive sounds using a modular synthesizer takes quite a few modules, most of which would be rather over specified for the task to hand. Such an approach would waste a lot of computational resources. So we took a different path.

The drum synth module incorporates just the tools you need to create a huge range of tuned and untuned percussive sounds with no waste. It looks a bit complex at first sight because it packs a lot into a small area of screen but once you get the hang of it you'll love it.

The module incorporates three tone generators – two generate sine waves, one generates coloured noise. The noise generator has a multi-type resonant filter. Each tone generator has its own simple attack/decay envelope that controls the amplitude. This envelope can also be routed to control pitch in the case of the sine wave generators and the filter cutoff frequency for the noise generator.

The output of the three generators is added together so you control the mix between them using the envelope level parameter. A word of warning – this module has been designed to kick - it goes loud.

One sine wave generator is designated the master and can be set to any frequency from the low 40's to about 900 Hz. The other sine wave generator slaves to this frequency, i.e. its setting is a ratio of the frequency of the first one. We do this mostly to save CPU resources (the slave oscillator's maths are slightly less taxing)

One thing worth noting is that the two sine wave generators can cross-modulate the other's frequency. What's the point of this?

Well, using the output of one sine wave generator to modulate the frequency of the other will give some metallic-like tones. It is called frequency modulation synthesis. Exactly how metallic will depend upon the difference between the frequencies of the two generators and on the depth of the modulation.

If you then take that modulated output and make a loop so it modulates the frequency of the first tone generator (modulate the modulator as it were!) then the results can get even more harsh and chaotic. At high modulation depths this is just what you need to start to create struck metal percussive effects. Lower depth values will dirty up drum sounds for you rather nicely.

You can force this module to track composed note values. There are loads of possible uses of this feature. The obvious one is to get several percussion instruments for the price of one. If you want to set up an alternating high/low cowbell figure simply load up the relevant preset, enable note tracking on this module and set up a pitched fixed voice type or something similar to take care of the tuning. You can also create a whole battery of tuned percussion instruments; marimbas, bells by using this option.

Enjoy!

And that just about concludes this quick jaunt through the practical application of PSE. We've really just scratched the surface but hopefully you have collected enough pointers from this and the previous tutorial to make your own way from here.

External Links

- [Video Synthesis Tutorial (http://vimeo.com/2206614)](http://vimeo.com/2206614)

Glossary

FX / Effect Unit or Audio Plugin : An effect unit is a audio processing stage, which can be used to perform a variety of exciting things. You can have as many effect units in your design as you want, subject to the processing power of your target platform.

Tone Generator Plugin : A special sort of effect unit that knows how to generate note data in response to MIDI note events; for example, this includes "tg/dsynth" (the "DSynth"), and "tg/particle" (the "Particle System" which is a granular synthesiser unit).

Signal Rate Plugin : An audio plugin that works at the normal audio processing rate of the system; most plugins are considered to be audio-rate plugins.

Control Rate Plugin : An audio plugin that works at a lower rate than the normal audio processing plugins; used to control parameters on other units (for example, c/lfo and c/envelope are both control-rate plugins).

FX / Effect Module or FXM network: An effect module is the grouping of all effects units used by a particular Voice, or by the global effects, or by a synthesizer module. Think of it in terms of a row of Effect Units that works from left-to-right.

LFO: Low Frequency Oscillator. Used to do a number of things; control-rate LFO (c/lfo) can be used to control other effects unit parameters such as filter cutoff frequency or perform pitch modulation. The signal rate LFO can be used as a direct frequency generator, or to apply Ring Modulation effects if run at a low enough frequency.

Envelope: A control-rate unit that creates a waveform in direct response to a note on MIDI event; used to modulate various critical parameters within synth modules. For example, envelopes are used to shape the volume of a note when it is triggered, and can also be used for all sorts of neat effects such as modulating the pitch of a note while it plays.

PSE MIDI implementation chart

- [Modular Synthesis Framework](#)
- [Audio Data Pipeline in the Synth](#)
- [MIDI implementation chart](#)
- [General MIDI 2 (GM2) feature implementation chart](#)
- [Supported RPN's](#)
- [Supported SYSEX Messages](#)
- [Modular synth SYSEX style commands in MIDI files](#)

Introduction - PSE MIDI Implementation Chart

PSE has been designed to work on both desktop and mobile platforms. It supports various sound synthesis approaches including modular synthesis, wavetables, and even mobile-targeting features including SP-MIDI compliance and 3GPP reduced wavetable sets. It uses wavetable data in various formats including DLS format and Ogg, in combination with entirely synthestic generated waveforms.

The [modular synthesis](#) approach allows for incredibly rich and open-ended MIDI sound generation with 3rd party extendibility. Click [here](#) for full information!

Support for up to 4096 MIDI channels

The PSE synth is capable of supporting more than 16 MIDI channels. In fact, it can support up to 4096 MIDI channels! This functionality is used by various MIDI Meta file players which might require more than 16 channels to render their MIDI-like data.

Modular Synthesis Framework

PSE has incredibly powerful and flexible support for modular synthesis. This allows for very rich and open-ended MIDI sound generation with a 3rd party extendible plugin set.

Synthesizer networks are also referred to as [FXMS networks](#), and are declared using XML blocks of the form <fxms>...</fxms>.

PSE is a modular, 3rd-party extendible synthesizer that allows very fine-grained control over the sound that it creates. PSE includes a number of optional tone generator classes, including but not limited to:

- [Oscillator](#) - the flexible signal-rate tone generator
- [DSynth](#) - suited to generation of both melodic and percussive sounds
- [Particle](#) a highly effective granular synthesis unit

If you want help creating content that uses modular synthesis, here are some useful resources:

- [Click here for the modular synthesis tutorial.](#)

The sounds may be created in a variety of manners; each MIDI line can have its sound generated completely independently, using one of the following techniques:

- using built-in MIDI wavetable sounds (this is the default behaviour of the PSE software synth)
- using custom audio sample data stored in either DLS wavetables or compressed audio formats such as the Ogg format.
- using arbitrary tone generator analog-style plug-in synthesis modules (such as PSE's [DSynth](#) or [Particle System](#) tone generator plugins), under management of the [FXMS](#) class. In this case, sounds are synthesized in real-time according to the algorithm implemented by the tone generator plugin in question. There is huge flexibility in this approach! The Mixtikl Player app makes use of these extensions.

Once sounds have been rendered, they may be processed using arbitrary [FXM](#) audio effects networks, using whatever audio plugins might be available on the system.

- separate internal and external audio effects pipelines may be attached to each MIDI line

- global internal and external effects pipelines may also be used, that are applied to the final mix from all MIDI lines.

Note that the difference between internal and external effects pipelines, is that an internal pipeline is in general for creating and internal ownership by a MIDI Meta file player (or application); external pipelines are for direct creation, attachment and manipulation by an external application. Because internal effects should not generally be modified by an application, this means that a relevant file can be spatialised by an program that attaches a 3D spatialiser as a global external effect, without having any adverse effect on any built-in internal global effects that might be part of the file definition.

FXM networks are the fundamental DSP pipeline class of the ISS; they are used to implement the modular synthesizer processing. They can contain both signal-rate and control-rate sub-networks

Effects pipelines are defined to the MIDI synthesizer as <FXM> networks, wrapped-up in <effects> ... </effects> XML tags, using either:

- PSE meta file data
- SYSEX commands within MIDI files (click here for more details).
- Programmatically...

Synth modules are also defined to the MIDI synthesizer as <FXM> networks, but wrapped-up in <fxms> ... </fxms> XML tags. They are supplied to a piece using:

- PSE meta file data
- SYSEX commands within MIDI files (click here for more details).
- Programmatically...
- Using a GUI that allows directly interacting to create a parameter definition for a specific audio plugin.

Audio Data Pipeline in the Synth

There are 6 stages in the audio processing for the synth:

1. Computation of basic instrument audio. This consists of the sample data for all the notes rendered during a particular instrument played on that channel (note that the sample data might be custom sample data, as supplied for example within an file). The audio data at this point is in mono. If any Modular Synthesis is defined, then it is done here as an alternative to the default MIDI wavetable rendering approach.

   - Examines all the currently playing notes for the channel and renders the wave data for that instrument according to the pitch and velocity (loudness) associated with each note. The wave data itself is mono and the result of superimposing the wave data for all the notes for this control block is likewise mono. The sample data used might be custom sample data (as supplied for example from a file).

     Important note on Modular Synthesis: if modular synthesis is in use, due to an FXMS module being defined for this MIDI line, then the appropriate FXMS module is used to synthesise the data (rather than using the default wavetable synthesis approach).

2. Adjustment of audio to account for volume attenuation on that channel.
   - Applies a volume reduction if necessary. It is used when the volume to be applied is greater or lesser than the default volume.

3. Separation into interleaved stereo data if panning required in the stereo field.
   - Applies panning when necessary. This only happens when the output is in stereo and panning is not disabled. Panning may be disabled if some other part of the system will apply a level of spatialisation, such as a 3d sound spatialiser.

4. Application of custom per-line audio-plugin effects.

   - Applies custom per-line audio-plugin effects. Some examples might be a distortion effect, a phaser, or a 3d sound spatialiser. In the latter case of a 3d spatialiser panning would be disabled in stage 3 since the 3d effect adds its own panning. There are two sub-stages in this effects processing. The first handles 'internal effects', the second handles 'external' effects.

     Internal effects are typically associated with the content itself. An example might be a distortion effect which is to be used on channel 4 for the duration of the piece. This bundling of effects in content can be achieved in a file. These effects will always be in place for that particular piece of content regardless of the external application that is playing it.

     External effects are those that an external application might create in order to apply application specific effects to

any midi content. As the application owns the effects it can also change the parameters in real time. An example might be a 3d sound spatialiser, whose parameters would be manipulated by the external application to reflect the position in space which is deemed to be the source of the sound from that MIDI channel.

Internal and external per-MIDI line effects can be set.

5. Application of inbuilt MIDI reverb and chorus.
    - Adds the audio from this channel into the inbuilt reverb and chorus effects buffers according to the current MIDI controller send levels. These reverb and chorus buffers are processed at a later stage on a global level. This stage may be disabled for the channel if desired, for example when the custom effects required do not want to make use of global effects.
6. Application of global internal/external custom audio-plugin effects (not shown in the diagram)
    - Applies global internal/external custom audio plugin effects, if any are defined. Such global effects are applied to the final mixed audio data generated from the summed output of all synthesized data from all MIDI lines, after application of any global reverb or chorus effects. Global internal effects are applied first; these are specified using MOMR_PROPERTY_FXM_INTERNAL with a MIDI line value of -1; such effects should not generally be defined by any code other than from a MIDI meta file handler. Global external effects are defined using MOMR_PROPERTY_FXP, and any application is free to set such global effects. Global external effects are used, for example, to 3d-audio spatialise the entire MIDI file.

Figure 1 below shows this pipeline. As we can see, stage 4 is not executed if there are no custom effects (either internal or external) present for that channel. In stages 2 and 3 the audio data is converted into stereo interleaved buffer (if the output format is in stereo) and volume and panning adjustments are made as part of that operation. As previously mentioned, panning may be disabled in certain circumstances such as the use of 3d sound custom effects.

Synth Audio Pipeline
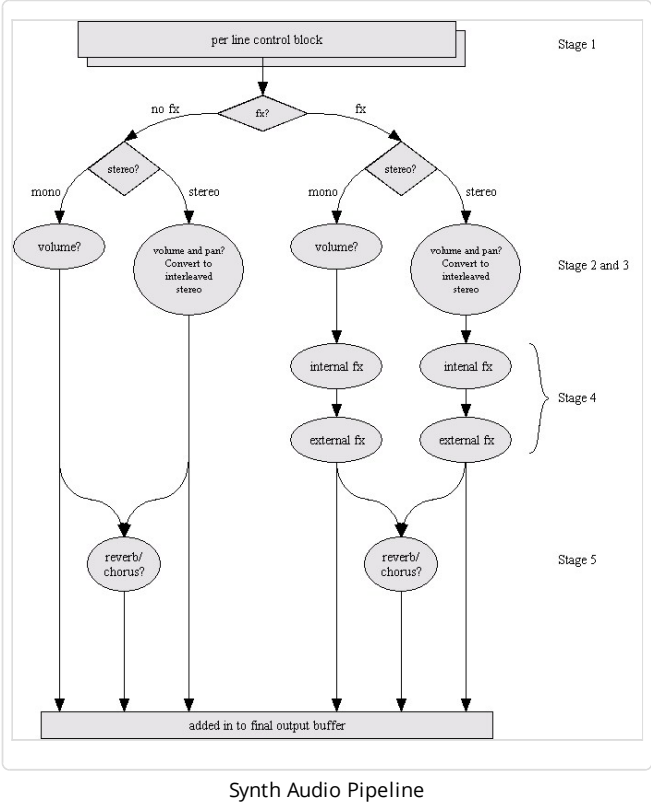
Figure 1: The PSE synth audio pipeline from per channel rendering to effects processing and final mixing.

MIDI implementation chart

Note: The PSE synthesizer is purely a software renderer, and does not support transmission of MIDI events.

For the chart below, '**O**' means supported, while '**X**' means **not** supported

	Function	Recognized	Remarks
Basic channel	Default	1 - 16	
	Changed	1 - 16	

Mode	Default	Mode 3	
	Messages	X	
	Altered	X	
Note number		0 - 127	
Velocity	Note on	O	
	Note off	X	
Aftertouch	Key	X	
	Channel	X	
Pitch bend		O	
Control change	0, 32	O	Bank select (MSB, LSB)
	1	O	Modulation Depth
	7	O	Volume
	10	O	Pan position
	11	O	Expression
	64	O	Hold Pedal
	71	O	Filter Resonance (affects filter Q)
	74	O	Filter Brightness (affects cutoff frequency)
	91	O	Send to reverb (reverb depth)
	93	O	Send to chorus (chorus depth)
	98, 99	O	NRPN (LSB, MSB) - reserved
	100, 101	O	RPN (LSB, MSB) - reserved
	120	O	All sound off
	121	O	Reset all controllers
	123	O	All notes off
Program change		0 - 127	
System exclusive		O	GM2 reverb and chorus parameters
System common	Song position	X	
	Song select	X	
	Tune request	X	
System realtime	Clock	X	
	Commands	X	
Aux. messages	All sounds off	O	
	Reset all controllers	O	
	Local on/off	X	
	All notes off	O	
	Active sensing	X	
	System reset	X	

General MIDI 2 (GM2) feature implementation chart

	Feature	Compliance	Remarks
Sound source type		wave-table	DLS level 1 and level 2 formats supported
Number of notes		32+	configurable to suit device
MIDI channels		16+	Can operate with up to 4096 channels
Channels	melody	yes	Channel 10 defaults to rhythmic (i.e. drums), all others default to melodic. Channels can be switched between melodic and rhythmic.
	rhythm	yes	
Modes		mode 3	(OMNI OFF, POLY)
Rhythm channels	note off ignored	no	
	mutual exclusivity	no	
Effects	types	reverb / chorus	Full support for both reverb and chorus subject to how you configure your platform. Note: chorus has one tap.
	channel sends	yes	
	chorus to reverb	yes	
Master volume		no	
Master tuning	fine	no	
	coarse	no	
Reverb	type	yes	
	time	yes	
Chorus	type	yes	
	mod rate	yes	
	mod depth	yes	
	feedback	yes	
	send to reverb	yes	
Scale / octave tuning adjust		no	
GM system messages	GM2 system on	no	
	GM1 system on	no	
	GM system off	no	
Active sensing		no	

Supported RPN's

RPN#	Parameter	Values and Response

0	Pitch Bend Sensitivity	a maximum range of +/-127 semitones

Currently no NRPNs are supported, but can be introduced given sufficient demand.

Supported SYSEX Messages

Message	Message bytes	Remarks
SP-Midi	0x7f 0x0b 0x01 <channel-number> <poly value (MIP)> ... <channel-number> <poly value (MIP)> 0xf7	The synth recognises SP-Midi commands and mutes/unmutes channels appropriately.
Reverb	0x7f 0x04 0x05 0x01 0x01 0x01 0x01 0x01 <reverb-parameter> <reverb-value> 0xf7	Can be used to set the reverb in accordance with General MIDI 2.
Chorus	0x7f 0x04 0x05 0x01 0x01 0x01 0x01 0x02 <chorus-parameter> <chorus-value> 0xf7	Can be used to set the chorus in accordance with General MIDI 2.

Modular synth SYSEX style commands in MIDI files

A MIDI file can have SYSEX-style commands to modify the configuration of the synthesizer in real-time! Specifically, we use a sub-category of "cue point" meta events to deliver a parameterised payload of XML configuration data. Audio effects etc. can be changed at a variety of different levels; the potential to completely or partly change the sound of your piece in real-time is very exciting!

The supported "cue point" text meta event formats are as follows.

Synth module settings:

```
Change entire synth module definition in real-time:
  <fxms ...
e.g.
  <fxms l="0"> <t="tg/dsynth"><fxms>


Change synth module unit parameter:
  <fxmsp v="nnnn,uuuu,pppppppp,vvvvvvvv"/>
    nn=line
    uu=unit ([0] ... x)
    pppppppp=parameter
    vvvvvvvv=value (float DSP or integer)
e.g.
  <fxmsp v="0,1,1024,.5"/>


Change synth module unit input scale factor:
  <fxmsi v="nnnn,uuuu,ii,ssssssss"/>
    nn=line
    uu=unit ([0] ... x)
    ii=input index#
    ssssssss=scale (float DSP or integer)
e.g.
  <fxmsi v="0,1,1,.5"/>


Change synth module unit controller scale factor:
  <fxmsc v="nnnn,uuuu,cc,vvvvvvvv"/>
    nn=line
    uu=unit ([0] ... x)
    cc=controller index#
    ssssssss=scale (float DSP or integer)
e.g.
  <fxmsc v="0,1,1,.5"/>
```

Internal effects unit settings:

```
Change FX line unit parameter:
  <fxmip v="nnnn,uuuu,pppppppp,vvvvvvvv"/>
    nn=line
    uu=unit ([0] ... x)
    pppppppp=parameter
    vvvvvvvv=value (float DSP or integer)
e.g.
  <fxmip v="0,1,1024,.5"/>


External target:

Change FX line input scale factor:
  <fxmii v="nnnn,uuuu,ii,ssssssss"/>
    nn=line
    uu=unit ([0] ... x)
    ii=input index#
    ssssssss=scale (float DSP or integer)
e.g.
  <fxmii v="0,1,1,.5"/>

Change FX line controller scale factor:
  <fxmic v="nnnn,uuuu,cc,vvvvvvvv"/>
    nn=line
    uu=unit ([0] ... x)
    cc=controller index#
    ssssssss=scale (float DSP or integer)
e.g.
  <fxmic v="0,1,1,.5"/>
```