Pulsaret

granular synthesis

user manual v.2.0.0

apeSoft

www.densitygs.com

Overview

Starting

• system requirement

Macintosh

Pulsaret requires a Mac PPC or Intel machine running OS X 10.4 or later, and 1 GB RAM.

Windows

Pulsaret requires a Windows XP/Vista/7 machine and 1 GB RAM.

Pulsaret uses QuickTime in order to read correctly media files (including MP3), therefore QuickTime must be installed on your system. On Windows we recommend a complete installation of QuickTime choosing all optional components.

• installation

Macintosh

1. Download Pulsaret.dmg from www.densitygs.com
2. Double click to open the disk image

Windows

1. Download Pulsaret_setup.msi from www.densitygs.com
2. Double click to install

• copyright

This program is copyright shareware and it is not freeware.
You can download the unregistered version of the program and give it to your friends or to any other person as long as for no charge. This program cannot be distributed in shareware compilations CDs without prior written approval from the author.
No responsibility is taken for any damage or losses caused by this package.
All program trademarks belongs to its respective author.

• demo limitation

Demo version it is fully functional but the application will quit after ten minutes. One every minute Pulsaret emit a white-noise. Buying a license you will disable this hateful behavior.

• purchase a license and authorizes

- Download the software from the web page: www.densitygs.com or somewhere else:
- Purchase the license software from Kagi: http://store.kagi.com/?6FHML_LIVE&lang=en
- in five/ten minutes you will receive an email with your activation code;
- Launch the software, open the “about” then click **Authorize** button. Now you need enter activation code, sent you from Kagi (**Thanks for your purchase e-mail**) and your e-mail employed for the kagi transaction. Click on **Authorize** to register you license.

• update to v.2

You can UPDATE to Pulsaret v.2 from Pulsaret v.1 (Mac/Win). N.B. you must already own a Pulsaret v.1 registration code! Buy Pulsaret v.2 from Kagì by inserting **UPDATE COUPON DISCOUNT** sent you from apeSoft. If you are a Pulsaret v.1 user but you have not Coupon update discount or you have lost it, please contact apeSoft (ape@kagi.com).

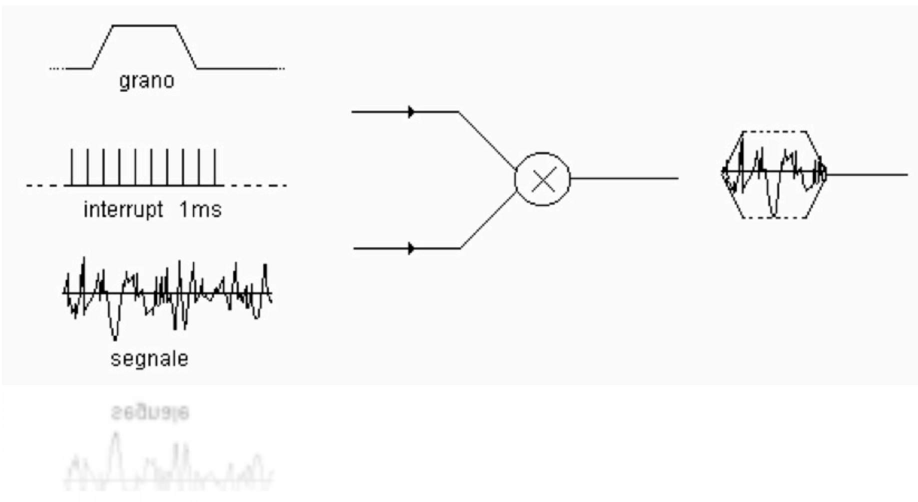
See purchase a license and authorizes.

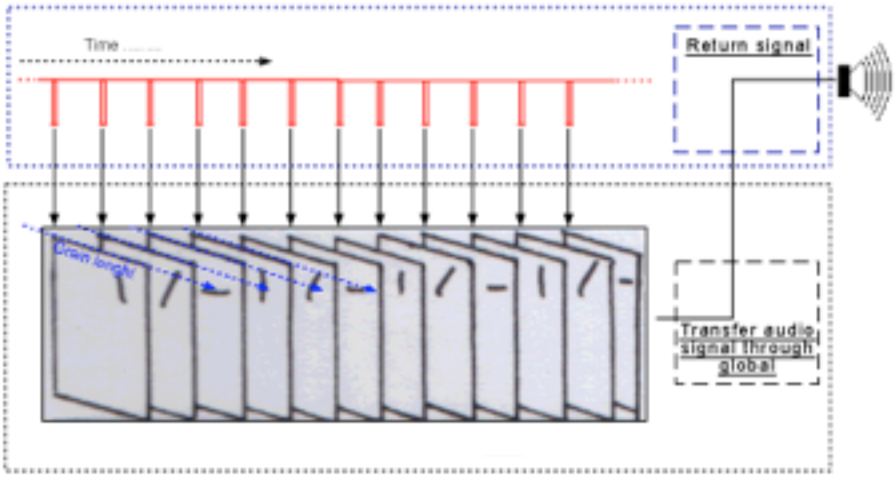
• background

The first official release of Density (2001), developed in Csound language and based on Eugenio Giordani 's GSC4 (Granular Synthesis for Csound). GSC4 was the first patch for granular synthesis on Csound implementing Barry Truax model.

Density can generate thousands of grains dynamically, I preferred this way, rather than a fixed number of "voices" (oscillators). The overlapping factor grains, depends only on the actual CPU power. Thus you have not limits in grains number for second (density).

Below an easy granulation model.



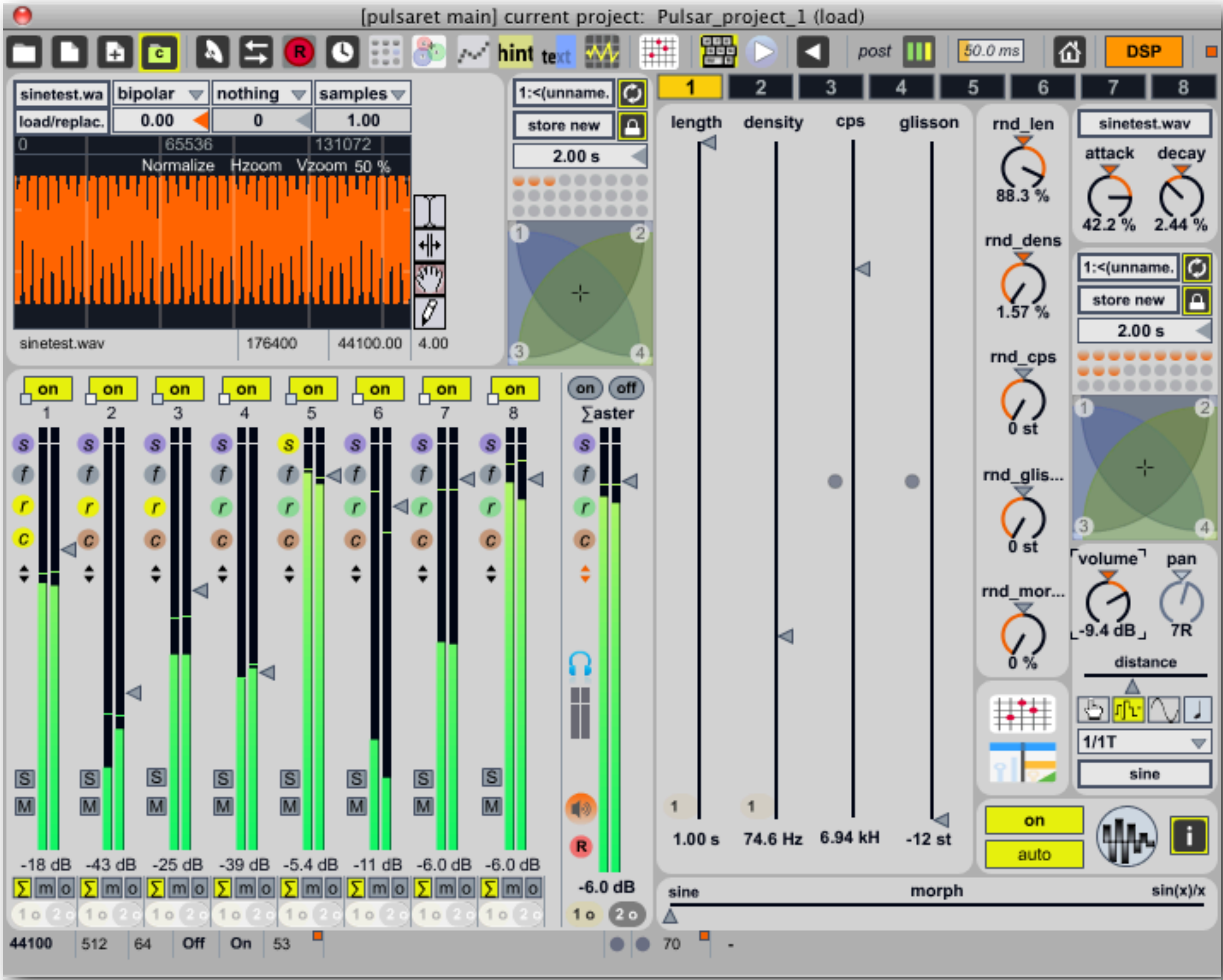


DensityGSC (Csound version) is still available for free at: www.alessandro-petrolati.it/densitygsc.html it works on Windows XP but not in Windows Vista, it seems magically resurrected with Windows7. DensityGSC is a discontinued product.

New Pulsaret is completely rewritten in [Max/Msp 5](), available for both Macintosh and Windows, more stable, flexible, improved audio quality, restyling GUI look (Graphic User Interface) with native effects Hv_pads, FiltersEQ, Snapshots Sequencers improviser unit etc...

- GUI layout

The most important Pulsaret parameters are placed inside of only one window. Pulsaret GUI (Graphic User Interface) is divided in three parts, WINDOWING MODULE (top left), MAIN MIXER (below) and GRANULAR STREAMS (on the right). Many controls are accessible via pop-up menù. WINDOWING generate and edit waves prototypes shapes, employed for envelope granulation, the MIXER, as a classical mixer, host granular STREAMS (eight) gain faders, solo/mute switches, headphones monitor channel, native effects, VST slots, I/O routing signals, MASTER fader (∑) and master/multichannel recorders.
Tools bar on the top, sets all Pulsaret preferences and open many Pulsaret functions like DSP settings, MIDI I/O, Master/Multichannel Recorders HV_pads etc... Tools bar show graphically (icons) the menùs bar items, tools bar it will be explained in the course of this handbook.



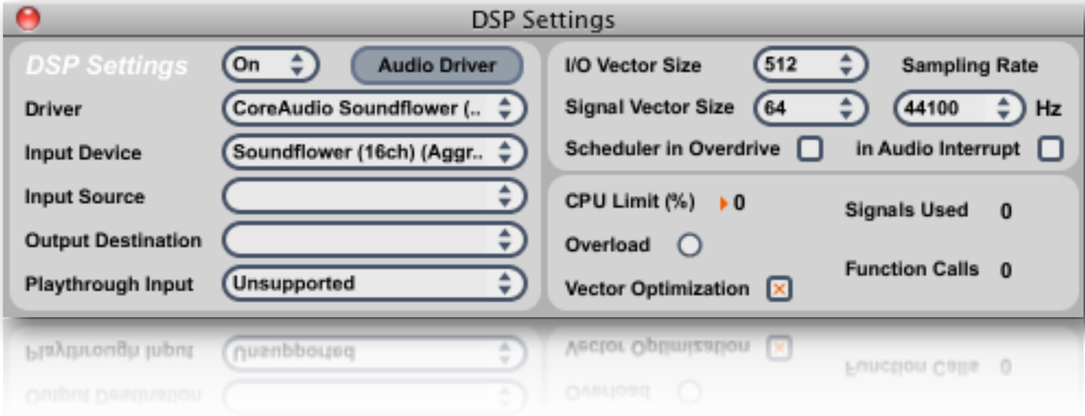
At the bottom of window, we have the status bar showing some informations like DSP settings, CPU % usage; on the right - sound file informations and grains overlapping factor - for the granular stream selected.

You can select one of eight granular streams, clicking on the “stream selection tab”, also you can use keyboard keys (shift+1,2...8) in case of shortcuts mode enabled.

The **Hint clue window** (toolbar or menù bars), explains Pulsaret GUI widget, when the mouse is over it. Pulsaret it's easy to learn, thought for live performances. The main functions will soon be assimilated and memorized, so, you can close clue hint window definitively, freeing the GUI from annoying hint.

- DSP settings

The engine DSP (Digital Signal Processing) can be activated clicking DSP toolbar (loudspeaker icon) or loudspeaker icon in the MASTER channel. When DSP is turned on, both became orange color.
The audio DSP parameters are displayed in the DSP Status window. You can open **DSP** settings from application menù View/DSP Cmd+D (Mac) or Ctrl+D (Win) or through the toolbar (loudspeaker icon).
The DSP Status window is arranged as a group of menùs and checkboxes that set all DSP parameters. Status window serves as a monitor for your current audio settings as well.



At the left of the DSP Status window is a pop-up menù for turning the audio on/off and a set of pop-up menùs that let you select an audio driver and configure its input source and output destination.

The second pop-up menù allows you to view and select an audio **driver**.
A brief summary will suffice for now:

- None: This setting shuts off audio processing.
- Core Audio: This is the default audio driver for Macintosh. It interfaces with the system's built-in Core Audio system and can be used with the built-in audio of the computer, or, with the proper software support, a third-party hardware interface, such as ASIO.
- MME or DirectSound: (Windows only) On Windows, Pulsaret loads the MME driver by default. If you have correctly installed external hardware and it also supports DirectSound, it should also appear as an option on the pop-up menù.
- ad_rewire: This driver supports a standard developed by Propellerhead Software that allows sound generating applications (ReWire Devices) to send multiple channels of audio and midi to other applications (ReWire Mixers) that process and output it. Selecting the ad_rewire driver enables Pulsaret to function as a ReWire Device to route audio from Pulsaret into applications that support ReWire (such as Live, Digital Performer or Cubase).
- ASIO: (Windows only) If you have a third-party audio interface which supports ASIO (a cross-platform audio hardware standard developed by Steinberg), and it is installed correctly, it will be found by the Pulsaret ASIO driver. You may have as many ASIO devices as you wish; they will all be found by the driver and will appear in the Driver pull-down menù in the DSP Status Window preceded by the word ASIO.
- ad_nonreal: This driver enables Pulsaret to work in non real-time mode, allowing you to synthesize and process audio without any real-time processor performance limitations. Real-time audio input and output are disabled under this driver.

Only one audio driver can be selected at any given time. Pulsaret saves the settings for each audio driver separately and will recall the last used audio driver when you restart Pulsaret.
The next two pop-up menùs are active only when using the Core Audio driver on Macintosh or ASIO drivers. When the Core Audio driver or either the MME or DirectSound drivers on Windows are selected, the pop-up menùs allow you to change the audio input source. These settings can also be changed using the Audio MIDI Setup application on Macintosh or the Sounds and Audio Devices Properties window (Start > Settings > Control Panel > Sounds and Audio Devices) on Windows, but only with these menùs while Pulsaret is running.
When ASIO is in use, the pop-up menùs allow you to set the clock source for your audio hardware and whether or not to prioritize MIDI input and output over audio I/O.

The DSP Status Window lets you control the size of the blocks of samples (called signal vectors) that Pulsaret uses. There are two vector sizes you can control.

- The I/O Vector Size (I/O stands for input/output) controls the number of samples that are transferred to and from the audio interface at one time.
- The Signal Vector Size sets the number of samples that are calculated by Pulsaret objects at one time. This can be less than or equal to the I/O Vector Size, but not more. If the Signal Vector Size is less than the I/O Vector Size, Pulsaret calculates two or more signal vectors in succession for each I/O vector that needs to be calculated.
  With an I/O vector size of 256, and a sampling rate of 44.1 kHz, Pulsaret calculates about 5.8 milliseconds of audio data at a time.

The I/O Vector Size may have an effect on latency and overall performance. A smaller vector size may reduce the inherent delay between audio input and audio output, because Pulsaret has to perform calculations for a smaller chunk of time. On the other hand, there is an additional computational burden each time Pulsaret prepares to calculate another vector (the next chunk of audio), so it is easier over-all for the processor to compute a larger vector. However, there is another side to this story. When Pulsaret calculates a vector of audio, it does so in what is known as an interrupt. If Pulsaret is running on your computer, whatever you happen to be doing (word processing, for example) is interrupted and an I/O vector's worth of audio is calculated and played. Then the computer returns to its normally scheduled program. If the vector size is large enough, the computer may get a bit behind and the audio output may start to click because the processing took longer than the computer expected. Reducing the I/O Vector Size may solve this problem, or it may not. On the other hand, if you try to generate too many interrupts, the computer will slow down trying to process them (saving what you are doing and starting another task is hard work). Therefore, you'll typically find the smaller I/O Vector Sizes consume a greater percentage of the computer's resources. Optimizing the performance of any particular signal network when you are close to the limit of your CPU's capability is a trial-and-error process. That's why Pulsaret provides you with a choice of vector sizes.

Technical Detail: Some audio interface cards do not provide a choice of I/O Vector Sizes. There are also some ASIO drivers whose selection of I/O Vector Sizes may not conform to the multiple- of-a-power-of-2 limitation currently imposed by Pulsaret's ASIO support. In some cases, this limitation can be remedied by using the ASIO driver at a different sampling rate.
Changing the vector sizes does not affect the actual quality of the audio itself, unlike changing the sampling rate, which affects the high frequency response. Changing the signal vector size won't have any effect on latency, and will have only a slight effect on overall performance (the larger the size, the more performance

you can expect). A signal vector size of 1024, it is 23.22 milliseconds. The Signal Vector size in Pulsaret can be set as low as 2 samples, and in most cases can go as high as the largest available I/O Vector Size for your audio driver. However, if the I/O Vector Size is not a power of 2, the maximum signal vector size is the largest power of 2 that divides evenly into the I/O vector size.

You can set the audio sampling rate with the Sampling Rate pop-up menù. For full-range audio, the recommended sampling rate is 44.1 kHz. Using a lower rate will reduce the number of samples that Pulsaret has to calculate, thus lightening your computer's burden, but it will also reduce the frequency range. If your computer is struggling at 44.1 kHz, you should try a lower rate.

N.B. Every Pulsaret project save current Sampling Rate value, thus every project can have differents Sampling Rates.

The Scheduler in Overdrive option enables you to turn Pulsaret's Overdrive setting on and off from within the DSP Status window. When Overdrive is enabled, the Pulsaret event scheduler runs at interrupt level. When overdrive is not enabled, the event scheduler runs inside a lower-priority event handling loop that can be interrupted by doing things like pulling down a menù. You can also enable and disable Overdrive using the Options menù. Overdrive generally improves timing accuracy, but there may be exceptions, and some third-party software may not works properly when Overdrive is enabled.

The Scheduler in Audio Interrupt feature is available when Overdrive is enabled. It runs the Pulsaret event scheduler immediately before processing a signal vector's worth of audio. Enabling Scheduler in Audio Interrupt can greatly improve the timing of audio events that are triggered from control processes or external MIDI input. However, the improvement in timing can be directly related to your choice of I/O Vector Size, since this determines the interval at which events outside the scheduler (such as MIDI input and output) affect Pulsaret. When the Signal Vector Size is 512, the scheduler will run every 512 samples. At 44.1 kHz, this is every 11.61 milliseconds, which is just at the outer limits of timing acceptability. With smaller Signal Vector Sizes (256, 128, 64), the timing will sound 'tighter.' Since you can change all of these parameters as the music is playing, you can experiment to find acceptable combination of precision and performance.

If you are not doing anything where precise synchronization between the control and audio is important, leave Scheduler in Audio Interrupt unchecked. You'll get a bit more overall CPU performance for signal processing The next portion of the DSP Status helps you monitor your system's performance.

The CPU Limit option allows you to set a limit (expressed in terms of a percentage of your computer's CPU) to how much signal processing Pulsaret is allowed to do. Pulsaret will not go above the set CPU limit for a sustained period, allowing your computer to perform other tasks without Pulsaret locking them out. The trade-off, however, is that you'll hear clicks in the audio output when the CPU goes over the specified limit. Setting this value to either 0 or 100 will disable CPU limiting.

The number next to Signals Used shows the number of internal buffers that were needed by Pulsaret to connect the signal objects used in the current signal network. The number of Function Calls gives an approximate idea of how many calculations are being required for each sample of audio. Both of these fields will update whenever you change the number of audio objects or how they are patched together.

Vector Optimization only applies to PowerPC computers. Vector optimization allows four samples to be processed within the space of a single instruction. However, not all audio signal processing algorithms can be optimized in this way (for example, recursive filter algorithms are substantially immune from vector optimization). Pulsaret itself no longer uses vector optimization, but third-party audio objects may still use it. In other words, unless you are using a vector-enabled third-party audio object on a PowerPC computer, this setting will have no effect.

Mixer

• faders and solo/mute

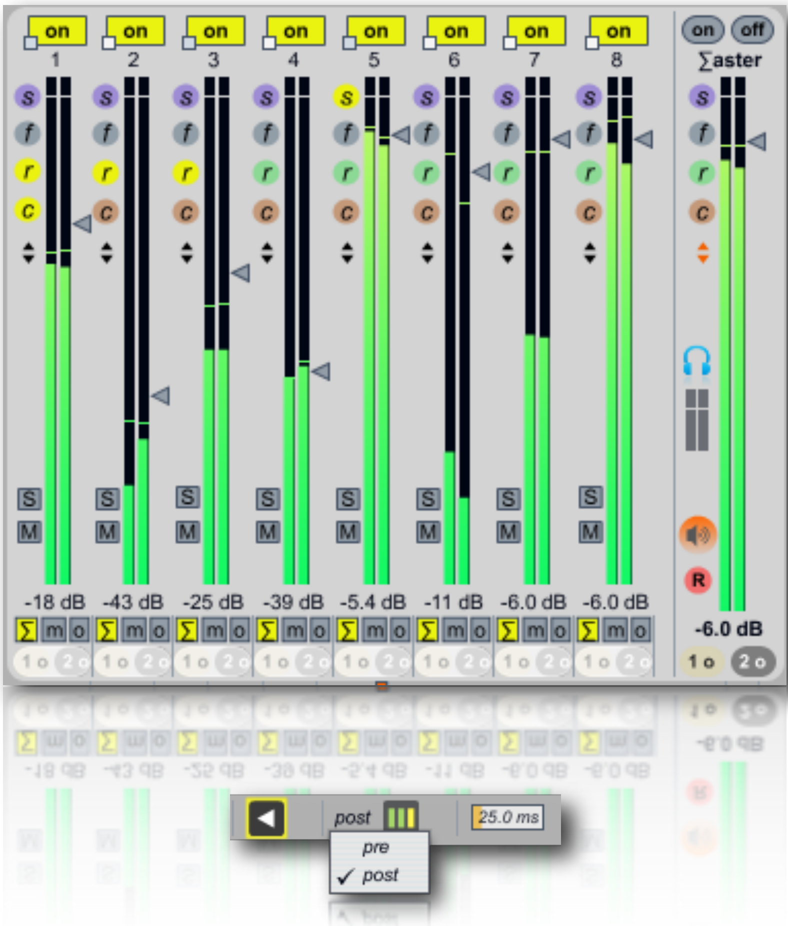
Vertical faders (1,2...8) receive stereo granular stream audio signal. Master (∑) channel (on the mixer right) sum all stereo faders and output audio data on the physical hardware audio device.

The 'mute' and 'solo' buttons, labelled 'M' and 'S' respectively.

- Activating the mute button mutes the channel, i.e. the affected channel can't be heard in the final mix.

- Activating the solo button mutes **every other** channel, i.e. **only** the affected channel can be heard in the final mix.

N.B. When you select one or more channels in solo, all the others channels automatically became muted.



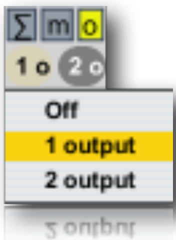
Toolbar, (left to right)
The first toggle (**back arrow**) is sliders **mouse** (*mousing*) behave (no knobs and other widgets but onli vertical sliders) with the follow options: **relative** and **absolute**. In relative mode (default) keeps it relative position when you click on it. Moving the mouse outputs higher or lower values in relation to that relative position. In absolute mode, will automatically jump directly to the clicked location.
N.B. **mousing** acts in the Mixer and in the granular streams parameters (vertical sliders).

With the second tool, **meters** (central), you can specifies whether the faders displays and outputs the signal level pre-fader or post-fader;
Most right **fade-time** set solo/mute fade in/out time in milliseconds, when you activating solo/un-solo or mute/un-muted, the signal amplitude is smoothed with a linear ramp.

- I/O mapping

The Pulsaret mixer is multichannel, real number of inputs ADC (Ana-logic to Digital Converter) and outputs DAC (Digital to Ana-logic Converter) depending to the hardware audio device. Each channel can be routed on the MASTER channel, on the MONITOR channel (logical) or directly on a **Physical** hardware channel. By default all streams goes in the MASTER.
Each stream also is routed on logical channel with predefined stereo number channels. Starting from the first stream mapped on 17/18, second 19/20.... eight 31/32; instead the MASTER channel send audio on **1/2 logical channels** while **MONITOR** send audio to **3/4 logical channels**

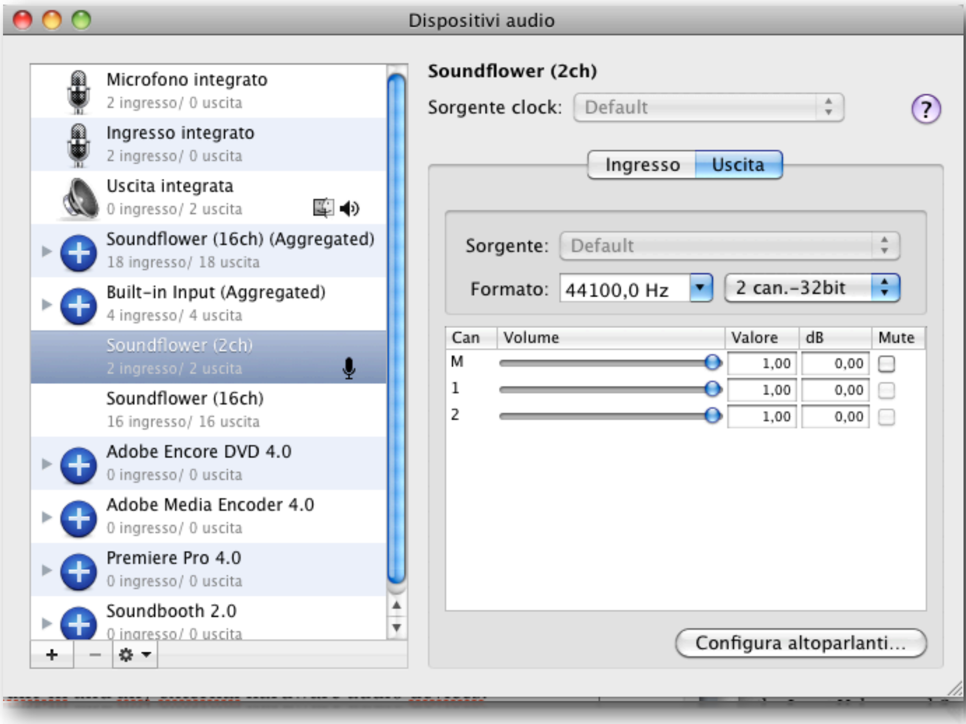
Output signal is routed in three different ways: ∑ (on the left) address signal on MASTER channel (default); '**m**' routing signal on MONITOR channel, explained below and labelled '**o**' routing the signal in physicals channels used by your hardware device, when selected you will see lighting the below menùs, from where you can chose physicals hardware output ports (red highlight).



In the above example only two channels (stereo) are displayed, the outputs channels numbers is depending by hardware device.

Using Core Audio
Core Audio provides audio I/O to Mac applications from both the computer's built-in audio hardware as well as any external audio hardware you may have.
If you have external audio hardware, it should come the drivers to interface with Core Audio. When these drivers are installed and the hardware is present, Core Audio will include the external device as a Core Audio choice in the Driver menu in the DSP Status window.

The Sound part of the System Preferences application can be used to set basic sound settings for the system, such as the Output volume, left/right balance, and sound output device, as well as the Input volume and sound input device. You can also use the Audio MIDI Setup application (located in /Applications/Utilities) for more detailed control of the sound I/O settings. Note that modifications you make to the Sound section of the System Preferences application, such as changing the output volume or balance, are reflected in the audio MIDI Setup (and vice versa). You can open the Audio MIDI Setup application by clicking on the Open Audio Control Panel button in the lower left corner of the DSP Status Window.



The Audio part of the Audio MIDI Setup application shows Input settings on the left side, and Output settings on the right.

The *System Settings* let you choose which audio device is used for system audio input and output, while the *Selected Audio Device* menu allows you to control the various settings for the built-in and any external hardware audio devices.

When using external audio devices, the *Input Volume* and *Output Volume* sliders can be used to set the overall input and output volumes of the selected device (they are not available when using the built-in audio controller). The *Device Mute* checkboxes allow you to mute the input and output devices, if applicable.

Play Through is available on PowerPC Macs only. *Play Through* checkbox just under the Input Volume slider lets you choose whether or not the input device is 'monitored' directly through to the output. When playthrough is enabled, the dry signal from the input source will play through to the output mixed in with any processed signal you may be sending to the output in Pulsaret. Disabling playthrough will enable you to control how much (if any) dry signal from the audio input is routed to the output. The *Input Section* allows you to select the Input Source (for example Line or Mic input for the selected device) as well as the sampling rate and bit depth in the *Current Format* pop-up menu. Similarly, the Output Section also allows you to select the sampling rate and bit-depth in its *Current Format* pop-up menu. The available selections will vary, depending on your audio hardware.

You can set the volume levels for the individual audio input and output channels, mute individual channels, and/or select them for playthrough using the controls located below the Current Format menus. The lower part of the window is used to display the current input and output settings.

Using MME Audio and DirectSound on Windows

Three types of sound card drivers are supported in Windows: MME, DirectSound and ASIO. Your choice of driver will have a significant impact on the performance and latency you will experience with Pulsaret.

The MME driver (ad_mme) is the default used for output of Windows system sounds, and are provided for almost any sound card and built-in audio system. While compatibility with your hardware is almost guaranteed, the poor latency values you get from an MME driver make this the least desirable option for real-time media operation.

DirectSound drivers, built on Microsoft's DirectX technology, have become commonplace for most sound cards, and provide much better latency and performance than MME drivers. Whenever possible, a DirectSound driver (ad_directsound) should be used in preference to an MME driver. Occasionally, (and especially in the case of motherboard-based audio systems) you will find the DirectSound driver performs more poorly than the MME driver. This can happen when a hardware-specific DirectSound driver is not available, and the system is emulating DirectSound while using the MME driver. In these cases, it is best to use MME directly, or find an ASIO driver for your system.

The best performance and lowest latency will typically be achieved using ASIO drivers. The ASIO standard, developed by Steinberg and supported by many media-oriented sound cards, is optimized for very low latency and high performance. As with the DirectSound driver, you need to verify that performance is actually better than other options; occasionally, an ASIO driver will be a simple ‘wrapper’ around the MME or DirectSound driver, and will perform more poorly than expected.

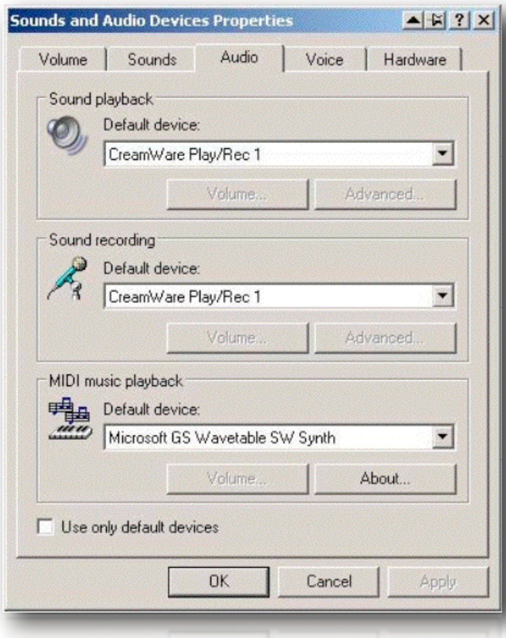
Using MME and DirectSound Drivers with Pulsaret on Windows

On Windows, Pulsaret loads the MME driver by default. If you have correctly installed external hardware, it should support playback and recording with the MME driver and the Direct Sound driver in the Driver Menu of the DSP Status Window.

If an audio device only supports MME or DirectSound, the Windows OS does an automatic mapping of one to the other. Since many audio devices initially did not support DirectSound, Microsoft emulated DirectSound with a layer that bridged from DirectSound to MME. Currently, there is greater support for native DirectSound drivers, and sometimes when you use MME drivers Windows is actually running a layer to convert from MME to DirectSound.

Note: Some devices such as the Digidesign mBox only support the ASIO driver standard. In such cases, you will need to select the proper ASIO driver in the DSP Status Window. See the section ‘Using ASIO Drivers on Windows’ for more information.

You can make overall changes to the basic operation of your default audio driver by accessing the Sounds and Audio Devices Properties window (Start > Settings > Control Panel > Sounds and Audio Devices). Here you can select Audio devices, and create settings for balance and output volume.



Pulsaret supports the use of different input and output devices with MME and DirectSound drivers. Use the DSP Status Window to choose input and output devices.

Input and Output Devices

When using MME or Directsound drivers, you may choose input and output devices from the pull-down menus in the DSP Status window, which will be automatically populated with the drivers for your audio hardware. When using the MME and Directsound drivers, it is possible to use different audio devices for input and output simultaneously. However, this is not recommended or supported and unless there is some external (from Pulsaret) provision for synchronizing the devices dropouts will likely occur over time.

Thread Priority and Latency Settings

Both MME and Directsound drivers include settings for Thread Priority and Latency. These are both set by default to settings which we hope will work on your computer in the majority of situations. However, you may find that when you are working with a patch that you have problems which you may be able to resolve by changing some of these settings. If your audio is crackling or there are glitches in it, you may want to try increasing the latency setting. This has the disadvantage of making your audio feel less responsive in real time, but it will allow the audio driver more time to work on the extra audio demands you have placed on it.
If your system is slow in other areas -- such as screen redrawing or general timing accuracy -- you may wish to decrease the thread priority of the audio driver. This will give your other tasks more room to get done, but may also result in you needing to increase latency in order to give your audio driver room to breathe at the new lower priority.
Timing between the control scheduler and Pulsaret is best when the I/O vector size is on the order of 1ms. We recommend setting the IO vector size to 128 samples. Having a setting of the latency separate from the I/O vector size allows this to work without audio glitches on most hardware.

Using ReWire with Pulsaret

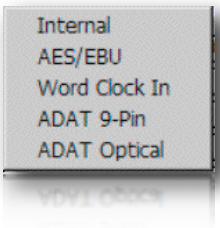
The ad_rewire driver allows you to use Pulsaret as a ReWire Device, where Pulsaret audio will be routed into a ReWire Mixer application such as Cubase. Both Pulsaret and the mixer application must be running at the same time in order to take advantage of ReWire's services. The mixer application should be also compatible with ReWire 2 or later for best results.
When the ad_rewire driver is selected, audio from Pulsaret can be routed to any of 16 inter- application ReWire channels which will appear as inputs in ReWire mixer host applications. The first time ad_rewire is

selected it will register itself with the ReWire system. Subsequent launches of ReWire Mixer applications will then offer Pulsaret as a ReWire device.
For example, after the Pulsaret ReWire Device is registered, Cubase will have a Pulsaret menu item in the Devices menu. When you choose it you will see a list of the audio outputs from Pulsaret. They will default to the off state. Click on any of the buttons to activate that channel. Once the channel is activated it will show up in the Cubase Track Mixer.

Using ASIO on Windows

Selecting an ASIO driver from the DSP Status window allows Pulsaret to talk directly to an audio interface. To use ASIO soundcards your device needs to be correctly installed and connected; The Pulsaret ASIO driver will find it at startup.
All correctly installed ASIO devices should be available to you for selection in the DSP Status window. However, Pulsaret does not check to see if the relevant audio interface hardware is installed correctly on your system until you explicitly switch to the ASIO driver for that interface card. If an ASIO driver fails to load when you try to use it, the menus in the rest of the DSP status window will blank out. Switching to the MME and/or DirectSound driver will re-enable Pulsaret audio.

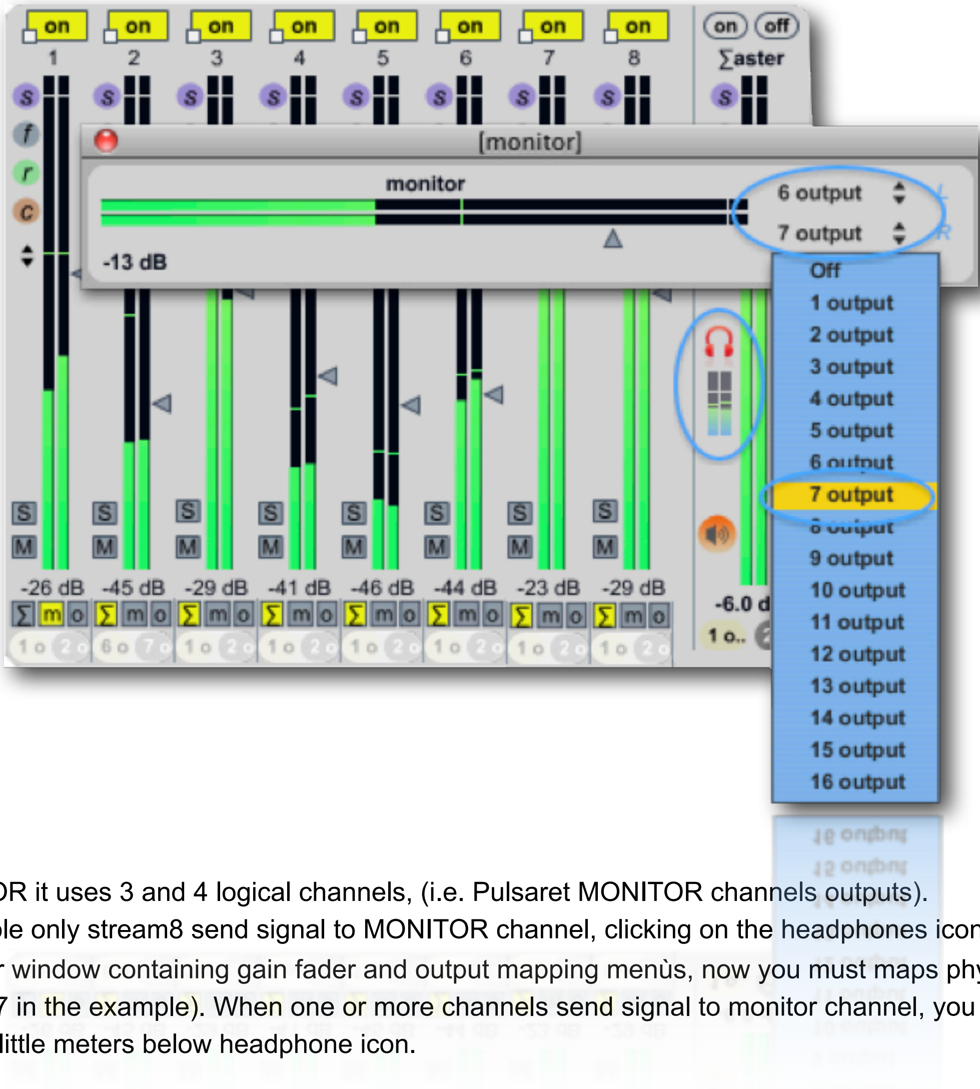
The *Clock Source* pop-up menu lets you to set the clock source for your audio hardware. Some ASIO drivers do not support an external clock; if this is the case there will only be one option in the menu, typically labeled Internal.



The *Prioritize MIDI* pop-up menu allows you to set the clock source for your audio hardware and whether or not to prioritize MIDI input and output over audio I/O.
Many ASIO drivers have other settings you can edit in a separate window. Click the Open ASIO Control Panel button at the bottom of the DSP Status window to access these settings. If your interface card has a control panel in its ASIO driver, the documentation for the interface should cover its operation.

- monitoring

Activating the monitor button (labelled '**m**') on the channel/s, the affected channel/s can't be heard in the final mix, for instance you can hear sound in headphones.



The MONITOR it uses 3 and 4 logical channels, (i.e. Pulsaret MONITOR channels outputs).
In the example only stream8 send signal to MONITOR channel, clicking on the headphones icon you will open Monitor window containing gain fader and output mapping menùs, now you must maps physical channels (6/7 in the example). When one or more channels send signal to monitor channel, you will see signal in the little meters below headphone icon.

N.B.
For monitoring signals you need a multichannel audio hardware device (four or more outputs channels).

- on/off stream trigger

Above Mixer channels we have streams triggers (*on/off* toggles). The same thing is achievable with *on/off* toggle placed on the granular stream GUI, (see Granular Streams for more details).

On/off buttons at the top of the master channel are shortcuts to turn *on/off* all granular strems together. They work regardless of the of trigger mode selected (see below).

N.B. If ***DSP*** engine is *off* they will be disabled ignoring mouse clicks.

There are many ways to trigger (i.e. turn-on/off) granular streams:
- mouse click on/off button in the mixer or in the streams GUI;
- by sending a MIDI CC (Control Change);
- pressing spacebar keyboard, according to selected mode;

In order to use spacebar (computer keyboard) for triggering granular streams, you must enable toolbar **shortcuts** toggle.

(from the left) enable/disable keyboard **shortcuts**, if enabled triggers menù mode will be actived:
•**none** no spacebar trigger; **current** trigger only selected stream;
•**subscribed** streams (switch button in the mixer);
•**all** on/off eight streams together.
•

When **subscribe** is selected, all eight buttons (little orange) for subscription will be enabled. These are located close to the *on/off* toggles.
Selecting one or more, you will be able to trigger them with spacebar.

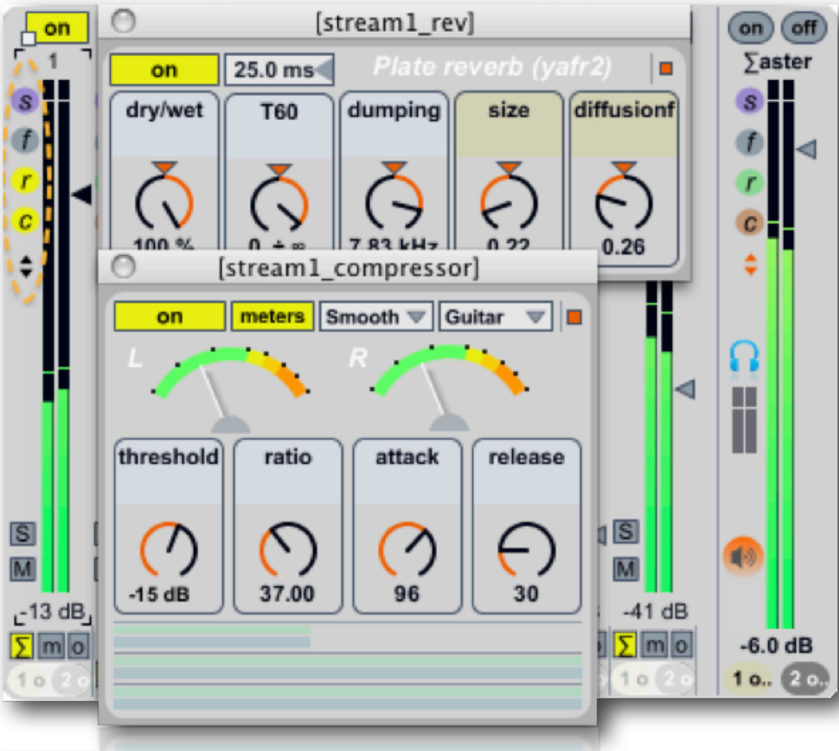
Shortcuts enable also following keys:

- **Cmd** (Mac) or **Ctrl** (Win) **+ 1, 2, … 8** to switch among granular streams;
- **escape** to hide chromatic keyboard in the streams GUI (if shown) and/or close dialogs messages box;
- **enter** key to accept;
- WavePad: **select**(F1); **loop**(F2); **move**(F3); **draw**(F4).

(see Granular Streams for more details).

- native effects

Every stream channels have own native effects, the four colored buttons display the window effect.



Topdown we have:

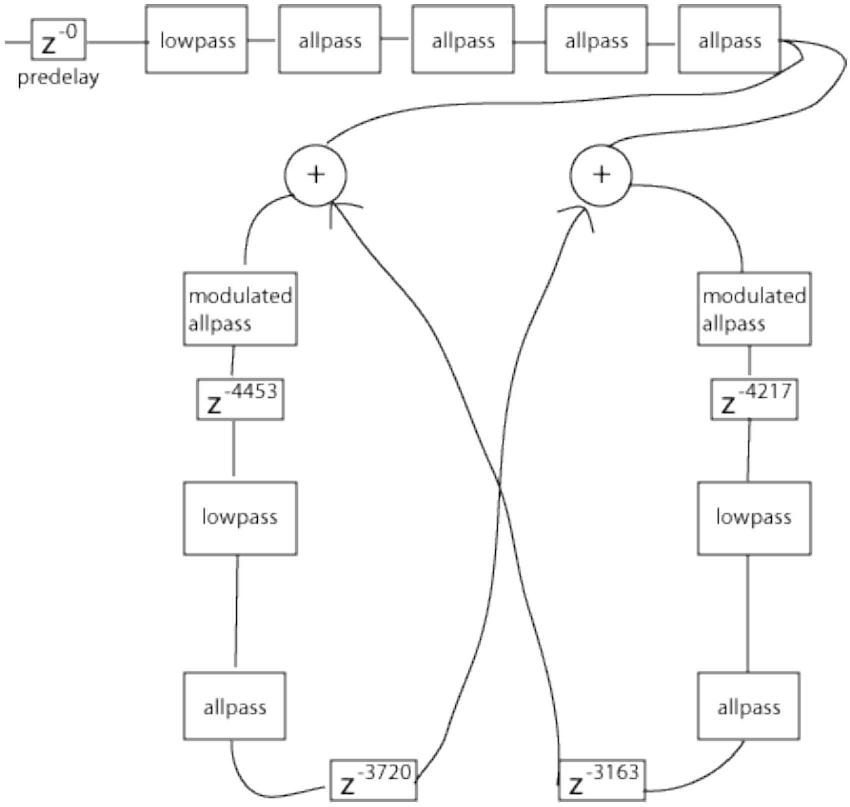
- snapshots-sequencer (labelled *s*);
- Filter multiband Equalizer (labelled *f*);
- Reverberation module (labelled *r*);
- Dynamic compressor (labelled *c*);

audio chain is processed as follow:

stream signal >>> sequencer >>> filterEQ >>> reverber >>> compressor >>> *VST_slots*

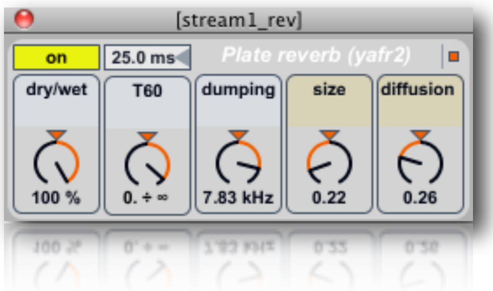
See **snapshots sequencer improviser unit**, **FilterEQ** and ***VST_slots*** for more explanations.

Reverberation module, a plate reverb in the style of Griesinger by Randy Jones.



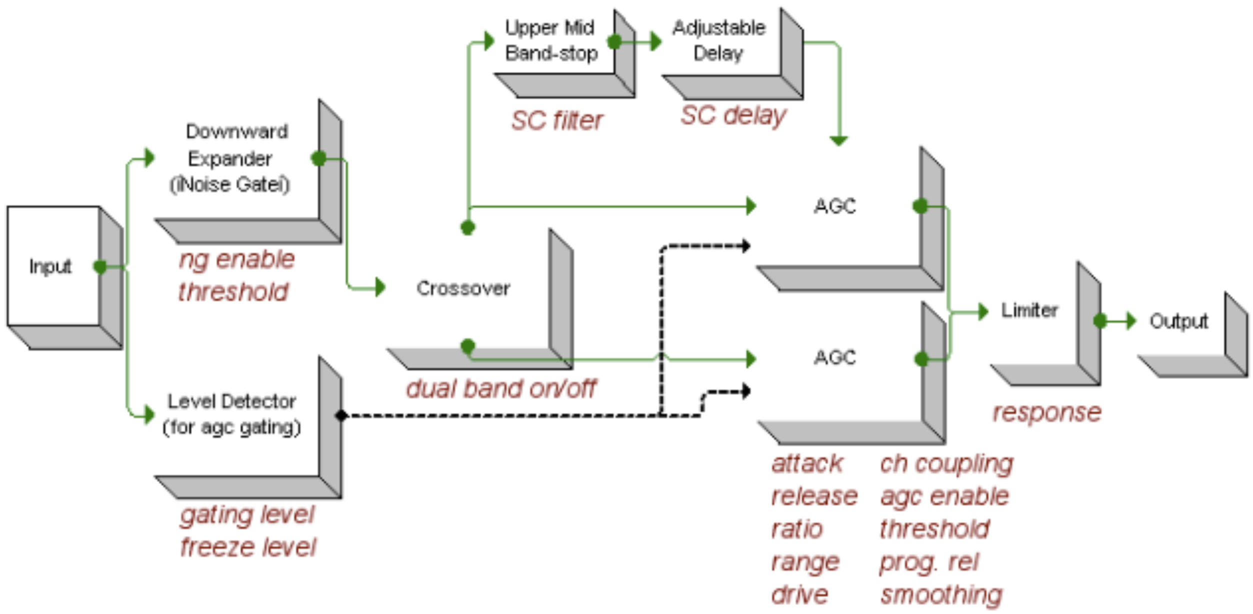
Parameters (from left to right):

- dry/wet balance (%)
- T60 reverberation time (0÷∞)
- high frequency damping (Hz)
- room size (filters coeff.)
- diffusion (filters coeff.)



N.B. the last two right knobs (brown colored) are general settings reverber parameters, you should not move it during performance, anyway values are mouse-filtered, their values pass only when the mouse button is up.

Dynamic compressor is a dual-band fully-featured signal compressor with limiting, gating, side-chain.



on/off bypass dynamic compressor;
enable/disable meters, these values describe the current state of various internal gain levels of the
compressor:

- compressor gain (left, right);
- noise gate gain (left, right);
- limiter gain (left, right)

The last two right menùs on the right sets

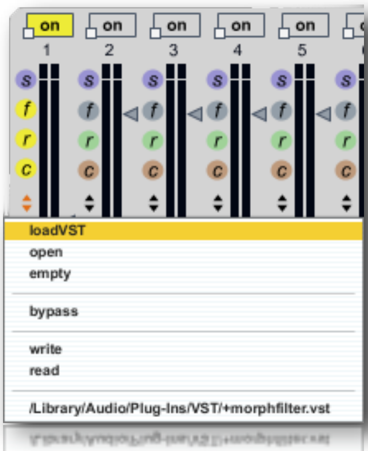
- limiter response mode: **punchy** or **smooth**. **Punchy** response yields extremely short attack and release times, useful for transparent limiting, or to create loudness. However, if over-used, intermodulation distortion may result. Smooth response uses longer attack and release times. The result is still a fast look-ahead limiter, but with less intermodulation distortion and less punch.

- the most right chose one of build presets for the dynamic processor: **Guitar, Bass, Vocal, Drums** and **Program Material** - An attempt at smooth "gain riding" of mixed program material as well as can be done with a non-multi-band processor.

threshold sets the compressor threshold (in dB below full scale). This is the main compression threshold. Any signal above the threshold will be reduced, and any signal below the threshold will be amplified, according to the range and ratio parameters;
ratio sets the numerator of the compressor gain reduction ratio, from 1:1 to Infinite:1;
attack sets the rate at which the compressor is engaged when the signal level exceeds the Threshold. The value range is 0-150 on a logarithmic scale, with larger values indicating faster attack;
release sets the rate at which the compressor releases its gain adjustment when the signal level no longer exceeds the Threshold. The value range is 0-150 on a logarithmic scale, with larger values indicating faster release. This rate can be modified by the release gate and freeze thresholds.

- VST slots

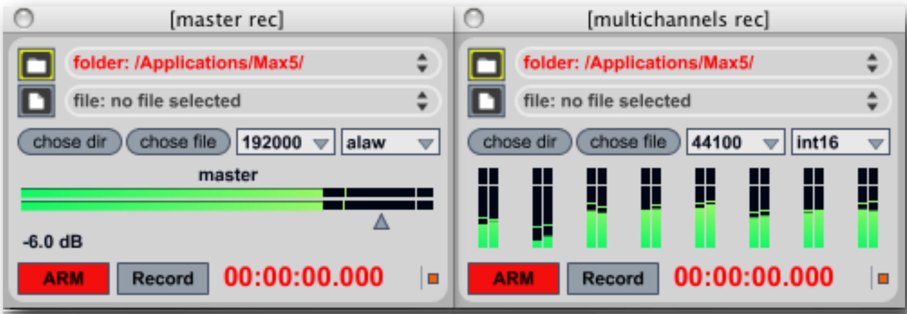
The last effect of the audio chain is a VST[1] slot.



There are nine VST slots in Pulsaret, for the eight granular streams and the MASTER channel. Load and manage VST plugin is easy. Clicking on menù (double arrows), **loadVST** item allows you to choose a VST plugin file (***.vst*** on Mac or ***.dll*** on Windows); **open,** show the VST effect window; **empty** cleans current VST slot which means no plug-in (default); **bypass** enable or disable VST effect; **write/read** export/import VST bank file; the lowest item show the path and name **of VST loaded**. Clicking on pathname you will also **open** VST GUI (as the **open** item).
All VST settings can be exported inside of Pulsaret project folder.
Some VST are **not fully compatible** with Pulsaret therefore you may not successfully export their contents.

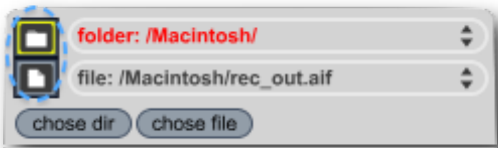
- Recorder

You can capture all Pulsaret audio signals in two different ways: "master rec" and/or "multichannel rec". From menùbar, toolbar or 'R' button (on the MSTER channel) you can open **Recorder** windows.



Before to start recording, you need toggle on **ARM** recorder unit. You can change output file *Type* (*quantization bit deep*) and *Sampling Rate.* Now you will see stereo signal through the V-meters. You are ready to record (click on Record toggle).

- **Master recorder** have only an horizontal fader witch you can controls the signal **gain** (post-master). **Multichannel recorder** show simply the 16 **V-meters** for the granular streams;
- **Sampling Rate** (default 44100) for the output file is taken by the current DSP setting. If Pulsaret SR is different from File out SR, the file output it will be resampled;
- **File Type** (quantization bit deep) default 16 bits;



folder/file targets switch: in **folder** mode (default) a new audio file is created each time on starts a recording. The file name is generated by using the current date and time in the machine, in order to avoid name conflicts, and to reflect the order of recording. The default location where the file is recorded is the Pulsaret Application folder. You can change location by clicking into the "**choose dir**" button, and select a

[1] *Steinberg's **Virtual Studio Technology** (**VST**) is an interface for integrating software audio synthesizer and effect plugins with audio editors and hard-disk recording systems. VST and similar technologies use Digital Signal Processing to simulate traditional recording studio hardware with software. Thousands of plugins exist, both commercial and freeware, and VST is supported by a large number of audio applications. The technology can be licensed from its creator, Steinberg. (Wikipedia)*

specific place. If one wishes to set a different name and/or location for each new file, it is possible to use the "**chose file**" button. Once a file is created, each new recording will be written into it. When one doesn't want to overwrite its previous content, one should create a new file again.
You can switch anytime between folder/file and you can also use pop-up menùs where is displayed file/folder path. The mode which is currently active has a red color.

Multichannel recorder write a multichannel audio file containing all the eight granular streams signals (i.e. stereo channels 2 x 8 streams = 16 channels).
Unlike the Master recorder, in Multichannel you cannot rescale channels record gains.

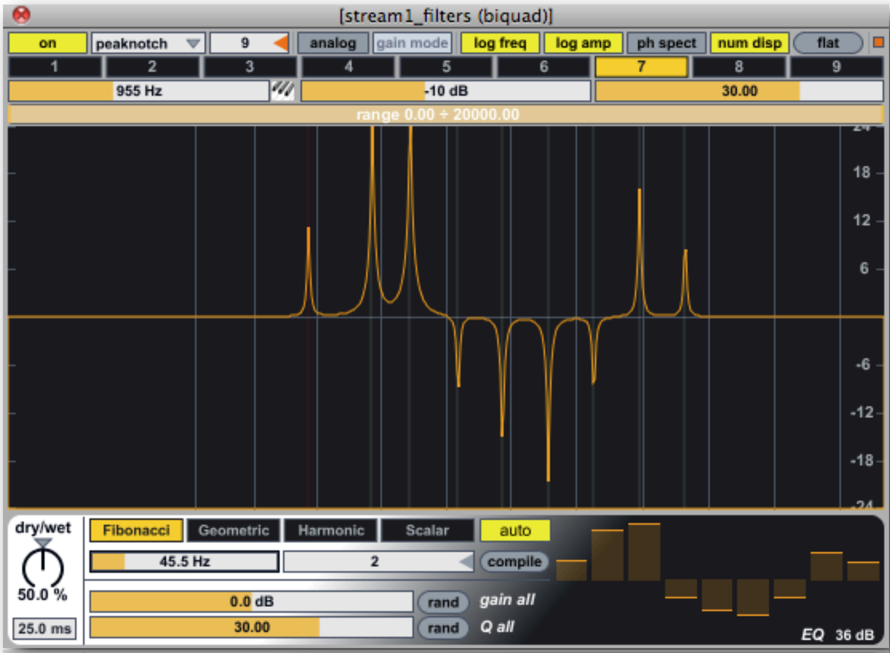
N.B. master/multichannel recorders, they receive audio signals before to being sent to a physical or logical channels, so even if the output is disabled (off) or muted ‘M’, they will continue to receive and recording the signal normally.

N.B.
- You can recording with master and channels recorders simultaneously;
- Record start/stop it will be not saved in the Pulsaret project;
- If Master or Multichannel Recorders are at work, you will see ‘R’ button (on the Mixer Master channel) to flash.

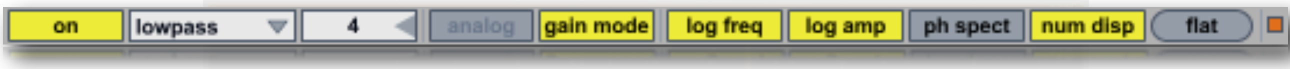
Filter Equalizer

• general

One of the most important and powerful Pulsaret feature is the multi-band filter equalizer. You can dynamically employ up to 24 two-pole filters (i.e. biquad), often referred to as "second order sections". In the Graphical filter editor you can set higher-level parameters such as **frequency**, **amplitude** and **resonance** (Q or S) through the mouse or selecting a filter band and using fine sliders.



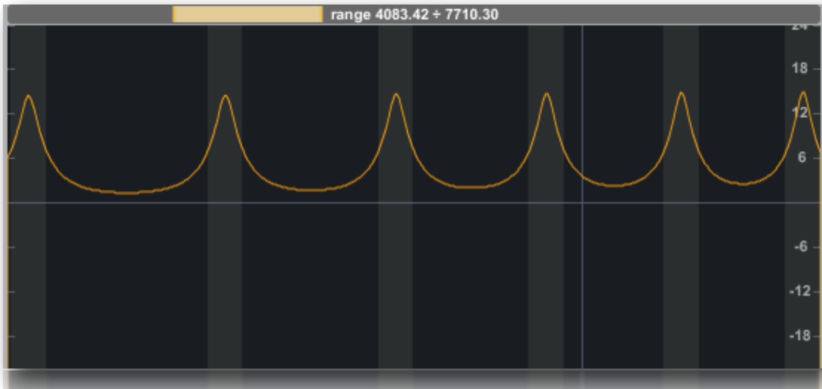
Tools bar is divided in three rows. Follow explanation top-down and left-right:



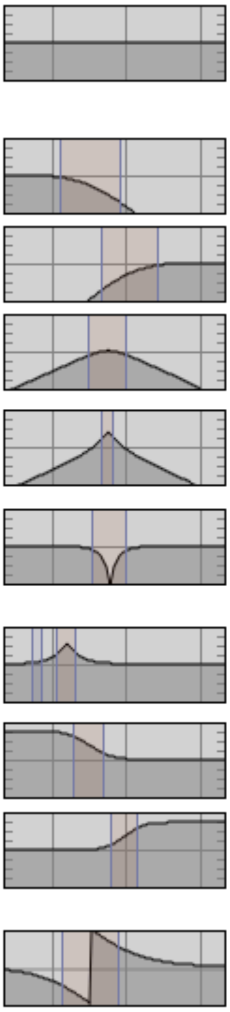
on/off switch to enable/bypass filter;
peacknotch (default) menù sets a filter kind: 0 - display only 1 - lowpass 2 - highpass 3 - bandpass 4 - bandstop 5 - peaknotch 6 - lowshelf 7 - highshelf 8 - resonant (see filter table at the end of this section);
numbox (showing 4 in the above image) sets the number of cascaded biquad filters displayed in the filtergraph, the range is between 1 and 24 (default 4).
analog toggles the analog filter prototype parameter for the bandpass, and peaknotch filters. Unlike the standard digital versions, these "imitation analog" filters do not have a notch at the Nyquist frequency, and thus imitate the response of a analog filter. The imitation analog filters are slightly more computationally expensive, so this option disabled by default;
gain toggles the gain parameter for the lowpass, highpass, bandpass, and bandstop filters. The traditional definitions of these filters have a fixed gain of 1.0, but by gain-enabling them, their amplitude response can be scaled without the additional use of a signal multiply at the filters output.

log freq, **log amp** and **ph spect** sets editor view range:

- log/linear frequency display;
- log/linear amplitude display;
- view phase spectrum -pi to pi;
- show parameters when mousing over the editor;
- resets all filters gains;



select a region of the spectrum for frequency zoom, Shift-clicking to extends the range; Command-clicking (Mac) or Control-double-clicking (Win) and dragging shifts the current range up or down. Option-clicking (Mac) or Alt-clicking (Win) and dragging up or down expands or shrinks the range. Command-double-clicking (Mac) or Control-double-clicking (Win) selects the entire range.



Display
filter coefficients only

Lowpass
cutoff frequency, gain and Q

Highpass
cutoff frequency, gain and Q

Bandpass
center frequency, gain and Q (determining bandwidth at -3dB from center freq)

Resonant (bandpass with constant skitrtwidth)
center frequency, gain and Q (determining bandwidth at -3dB from center freq)

Bandstop
center frequency, gain and Q (determining bandwidth at -3dB from center freq)

the lowpass, highpass, bandpass and bandstop filters take just cutoff/center freq and Q. (The filter gain parameter will always be set to 1. if the filter is not gain-enabled)

for all filters, shift-click constrains gain, cmd-click sets gain to unit gain (1.0)

Peak/Notch
center frequency, gain and Q (determining bandwidth at -3dB/+3dB from center freq)

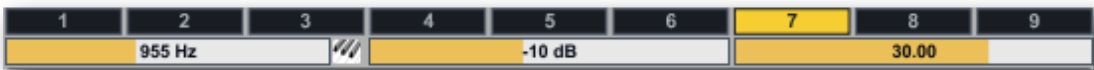
Low Shelf
center frequency, shelf gain and S (determining transition width at -3dB/+3dB from c.freq)

High Shelf
center frequency, shelf gain and S (determining transition width at -3dB/+3dB from c.freq)

the peaknotch, and shelving filters take cutoff/center freq, gain and Q/S. (Q = damping coefficient, S = slope coefficient)

Allpass
center frequency, gain and Q (roughly determining 90 degree transition in phase on either side of center frequency)

(displaying phase response)



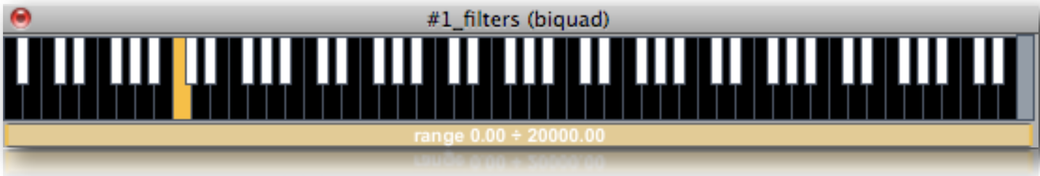
Tab numbers (in the example is selected 7), select a detail filter band.

- cutoff/gain

N.B. filters band could be not progressive in frequency, therefore you could have, for instance, the second cut-off frequency upper of the follows bands and vice versa.

Three sliders below sets respectively:

- cutoff (center freq) in Hz of the currently selected filter;
- gain in db of the currently selected filter;
- Q (resonance) or S (slope - used for the shelving filters) of the currently selected filter;



you can also set the cut-off frequency, opening the musical keyboard and clicking on the keys. Keyboard control also the gain of selected band filter, mouse clicking in different heights on each key.
If shortcuts is enabled, pressing escape key you will hide keyboard. If toggled the most right keyboard (most right key) switch, the keyboard will be always in foreground (disabled auto exit when click on a key).

The lowest sections is considered improvisation unit. You can experiment settings filters band with one of the COMPILERS (red frame); manage randomly or together gains/Q (purple frame) and make manually an equalization for each band (blu frame).

dry/weet balance between filtered/un-filtered signal, you can also set dry/wet fade-time in ms (25 ms default);

- compilers

*...inspired to Eugenio Giordani **Stria** Multilevel Interactive Sound Synthesizer.*

They increase complexity and coherence between the filters band space. You can create your pitch space by the way of four different criteria of construction by determining a pitches grid. The frequency range is depending by fundamental frequency and numbers of band filters. When a frequency exceed Nyquist[1] frequency (i.e. SR/2) the filter band generation is inhibited.

- Fibonacci COMPILER

---

[1] *The **Nyquist frequency**, named after the Swedish-American engineer Harry Nyquist or the Nyquist–Shannon sampling theorem, is half the sampling frequency of a discrete signal processing system. It is sometimes known as the folding frequency of a sampling system. The sampling theorem shows that aliasing can be avoided if the Nyquist frequency is greater than the bandwidth, or maximum component frequency, of the signal being sampled. (Wikipedia)*

Fibonacci[1] **COMPILER:** the *ratio* of two consecutive Fibonacci numbers converges on the Golden Mean Ratio (approximately 1:1.618) as its limit.
The computer music composition Stria by John Chowning[2] was based in almost every structural aspect on this r*atio*. Fibonacci compiler uses this sequence of numbers for compile the grid pitches.

You can set two parameters for the purpose to create a certain Fibonacci pitch grid. You have to specify:

- base frequency in Hz i.e. fundamental frequency (left slider 35.1 Hz) for the Fibonacci spectrum;
- first member of the series (value of 2 in the example).

To complete the operation and generate the pitch grid you must press the **compile** button.
When **auto** is enabled, changing value of one of parameters, (for each compilers methods) grid pitch it will be automatically compiled.

- Geometric COMPILER

Geometric COMPILER create “geometric pitch space” based on a *ratio number*. For example if you specify 1.5 (i.e. 3/2), you will create a pitch space grid based on the natural interval of 5th. Since you can choose any relationship, you can virtually have infinite pitch spaces.

I*nterval ratio* defines a coefficient of multiplication, this serves to build up the whole pitch grid while the *fundamental frequency* is the starting point from which grid take origin.

- base frequency in Hz (fundamental frequency) for the geometric spectrum;
- ratio for the geometric spectrum (i.e. 1.6 also represented by the ratio 8/5)

In the below example, a geometric pitch grid is based on the interval 1.6 and fundamental frequency 55 Hz. The grid will consist of a total of 13 bands starting from 55 Hz up to 15.481,1279 Hz as sketched in the next table:

---

[1] ***Leonardo Pisano Bigollo*** *(c. 1170 – c. 1250)also known as* ***Leonardo of Pisa****,* ***Leonardo Pisano****,* ***Leonardo Bonacci****,* ***Leonardo Fibonacci****, or, most commonly, simply* ***Fibonacci****, was an Italian mathematician, considered by some "the most talented western mathematician of the Middle Ages. (Wikipedia)*

[2] *John M. Chowning is known for having discovered the FM synthesis algorithm in 1967. In FM (frequency modulation) synthesis, both the carrier frequency and the modulation frequency are within the audio band. In essence, the amplitude and frequency of one waveform modulates the frequency of another waveform producing a resultant waveform that can be periodic or non-periodic depending upon the ratio of the two frequencies. (Wikipedia)*

#of_FREQS	Hz	operation
13	15539.276	$55*1.6^{12}$
12	9675.702	$55*1.6^{11}$
11	6047.313	$55*1.6^{10}$
10	3779.571	$55*1.6^{9}$
9	2362.232	$55*1.6^{8}$
8	1476.395	$55*1.6^{7}$
7	922.746	$55*1.6^{6}$
6	576.716	$55*1.6^{5}$
5	360.488	$55*1.6^{4}$
4	225.280	$55*1.6^{3}$
3	140.800	$55*1.6^{2}$
2	88.000	$55*1.6^{1}$
1	55.000	$55*1.6^{0}$

You can obtain the same result using the next recursive formula:

$$f_i = f_{(i-1)} * \text{iterval\_ratio}$$

where f_i is the i-th frequency (i.e. $f_2 = f_1 * 1.6 = 55.00 * 1.6 = 88.00$).

- Harmonic COMPILER

Harmonic COMPILER creates a bands grid according to the natural series of integer numbers (1,2,3...etc), starting from a fundamental frequency. There are two different ways to do it:

- base frequency in Hz (fundamental frequency) for the harmonic spectrum;
- INHARM factor for spectral expansion/compression. You can stretched or compressed harmonic spectra putting the INHARM parameter at a non zero value;

In addition you can create *stretched* or *compressed* harmonic spectra putting the **inharm** parameter at a non zero value. Stretched spectrum is obtained from values greater than zero, while for compressed spectrum you have to select values less than zero.

$$F_n = F_0 * n^{(1+exp)}$$

The stretched or compressed spectra are created using the following formula:
where:
F_n = n-th frequency
F_0 = fundamental frequency
n= index of F_n frequency
exp = factor of expansion/compression (inharm)

The inharm factor can vary from -0.2 +0.2 When inharm = 0 the generated frequency grid is pure harmonic.

- Scalar COMPILER

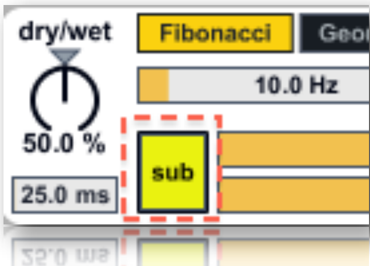
Scalar COMPILER method distributes linearly all bands from *freq min* to *freq max* range.

- minimum frequency for the scalar spectrum:
- maximum frequency for the scalar spectrum;

Alternatively you can impose the **gain** (top slider 19 dB in the example) and **Q** (below) at all filter bands in the spectrum. Also randomize all values.

An hidden function in the equalizer multi-slider, is Equalizer gain (0 ÷ 56 dB).
Mouse scrolling up/down on the value, you can change filters gains. This value must be considered bi-polar, so in the above example 38 dB it means -38 +38 equalizer range.

- transitions subscribe



You can subscribe/unsubscribe the Filters for snapshots transitions. The transitions on the Filters parameters work fine when you make interpolation between 4 (or more) stored snapshots. Default is un-subscribed. For instance you can take a snapshot of current filters (bands) configuration and move them toward a second snapshot making a linear interpolation between all filters bands.

Granular Streams

• parameters

In this section we will examine the Granular Synthesis parameters. Please refer to web for more explanations about granular synthesis.
You can interact with Pulsaret parameters on the GUI (Graphic User Interface) in the follows ways:

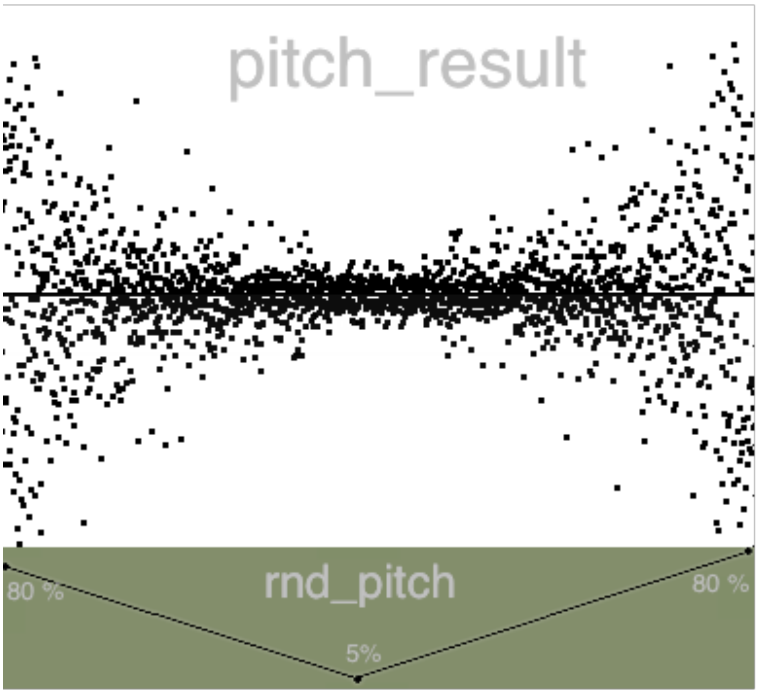
1) Click and drag (scroll) on vertical sliders (or dials) to change the value, (see *mousing* mode for more details);
2) Holding down the Command key (Macintosh) or the Control key (Windows) while mouse scroll, for precise value control;
3) Select a widget clicking on the name and use keyboard up/down arrows;
4) Drag up/down the display value (the number box below each parameter);
5) Click on the number box value, enter a numeric value and press enter key;
6) Send a MIDI CC (Control Change), (see MIDI/OSC I/O chapter in this manual);
7) Send an OSC (Open Sound Controller) message (see MIDI/OSC I/O in this manual);

The granulation engine is updated every new generated grain, the frequency of grains scattering depending from **Pulsaret** parameter, expressed in Hz (grains for seconds).
The vertical sliders provides a **deterministic values,** while the dials (knobs) provides **random** values. Summing the two values (deterministic-random) will be produced a value that is passed to the granulator.
In the follow example, the **pitch** value (deterministic) is rescaled with **rnd_pitch** (random):

pitch_result = pitch + (birnd(rnd_pitch * pitch/2))

Where **birand** is bipolar random generator. The random range is expressed in %, therefore 0 % it means only deterministic value, 100 % pitch_result can vary in the range:

pitch_result = pitch ± pitch/2

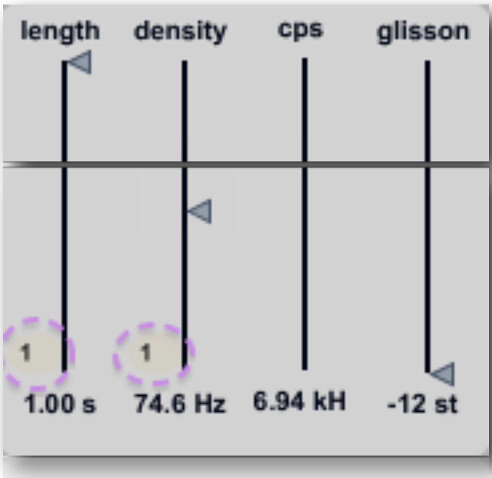


In the example **rnd_pitch** (green frame) starting from 80%, is reduced to 5 % in the middle.
With a great **rnd_pitch**, each grain can have an high random pitch mask range. Carrying **rnd_pitch** toward lowest values, the random pitch mask is reduced until 0 % which means no random pitch grain scattering.

length *slider* = grain duration in milliseconds;
rnd_length *dial* = random range deviation, in % of length;
density *slider* = grain for seconds in Hz;
rnd_density *dial* = random range deviation, in % of density;
cps = grain frequency in Hz;
freeze *button* = (small button in the middle of scanning control) switch between the value 0 (freezes) and the last control value;
rnd_cps = random range deviation, in % (12 semitone);
glisson = glissando semitones inside of the grain;
rnd_glisson = randomly moves semitones glisson (-/+ 12 semitones);
morph = morphing between two waveshapes (left=wave_1, right=wave_2, middle=mix 50%);
rnd_morph = morphing randomply between two waveshapes, in %;

volume *slider* = grains amplitude in decibel;
pan *dial* = left/right grains distributions on the stereo front (only pan = manually);
distance *slider* = stereo field width;
panningmode *tab* = select between: pan (manually), jittering (random), lfo (low frequency modulation);
beat/Hz *toggle* (only lfo) = toggle between beat/Hz; sets the panning duration (left right azimuth) expressed in note-ratios or frequency (Hz). Beat mode is synced with the global transport time, you can change global time (BPM[1]) from the global transport.
beat *menù* (only lfo) = each item switch represents the Relative-Tempo expressed in note-ratios; you can change BPM (Beat per minute) from the global transport. N.B. you need activate the global transport master clock.
panning rotate frequency *slider* = when selected lfo, set the panning azimuth in Hz, you can chose a **shape** (sine) among those available (see Windowing section for more details).

[1] ***Beats per minute*** *(**BPM**) is a unit typically used as a measure of tempo in music.*
The BPM tempo of a piece of music is conventionally shown in its score as a metronome mark, as illustrated to the right. This indicates that there should be 120 crotchet beats (quarter notes) per minute. In simple time signatures it is conventional to show the tempo in terms of the note duration on the bottom. So a 4/4 would show a crotchet (or quarter note), as above, while a 2/2 would show a minim (or half note). In compound time signatures the beat consists of three note durations (so there are 3 quavers (eighth notes) per beat in a 6/8 time signature), so a dotted form of the next note duration up is used. The most common compound signatures: 6/8, 9/8, and 12/8, therefore use a dotted crotchet (dotted quarter note) to indicate their BPM. (Wikipedia)



All the granular parameters have a prefixed values range. Some of them have an additional feature, you can divide or multiply the parameters output in order to extend range.
The rescaled value it will be passed to granulator engine, but on display you will see old prefixed widget range.

- settings

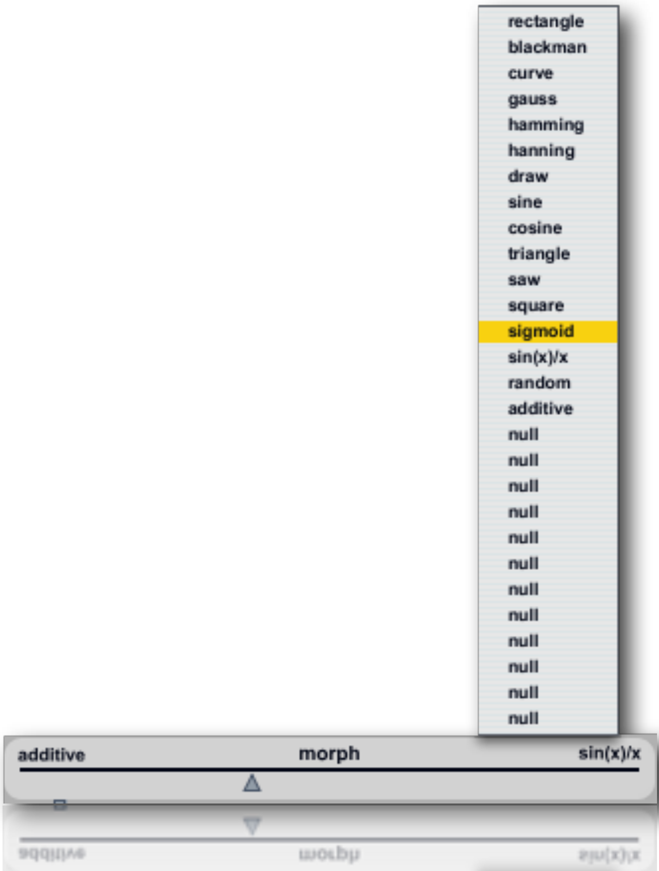
Streams setup:

-on/off granular stream works just like the play/stop buttons on the mixer, you can either use the space bar (see pop-up menù spacebar stream trigger).
-auto/manual enable disable auto grain scheduling. If disabled you can trigger manually a new grain, default value is auto then grains trainlet depending by density/rnd_density parameters.
- envelope/wave interpolation mode N.B. low interp is faster than high, but has more interpolation artifacts and cannot play buffer~ of arbitrary length.

- grain envelope

From the windowing envelope pop-up menù, you can choose a shape for granulation or loading until 12 sound files. Each stream can have a different envelope, the default shape is “draw”. when you load an audiofile (or mp3) in windowing, you will see file name in the streams envelope select popup menùs.
see Windowing in this handbook.

- wave select



Pulsaret provide a powerfull way to mix two waveshapes, every grain it will be initialized with own % mix of twi shapes. You can reach edge shapes by selecting differents wave shapes from menùs

Windowing

• general

The windowing module generate some classics envelopes and prototypes used for smoothing the grain amplitude. You can deforming some envelopes shapes like **gauss, curve or additive**, draw a new shape freehand or through **draw** prototype. Also you can load an audio file from disk (until 12 audio files, Aiff Wav or Mp3 supported). Almost all Windowing WavePad functions are identical to Granular Streams WavePad. See WavePad, WavePad display, WavePad Snap, WavePad mouse for more details.

Unlike Granular Streams WavePad, the Windowing WavePad buffer management have two additional functions:

- **reset shape**
- **reset all shapes**

When you select from menù the first of them, current envelope or shape will be restored at the original shape.
When you select reset all shapes, all prototypes and envelopes they will be restored at the original shape.

resetshape and **resetallshapes**, they work on the sixteen envelopes/prototypes restoring originals shape. While **clear** and **clearall** they work either on the sixteen envelopes/prototypes either on the twelve sound files slots, writing 0 in the all buffers.

Windowing have an important parameter that allow you to chose a **size** in samples for prototypes or envelope shape. You can also resize buffer dynamically, entering a new value expressed in samples (512 default).

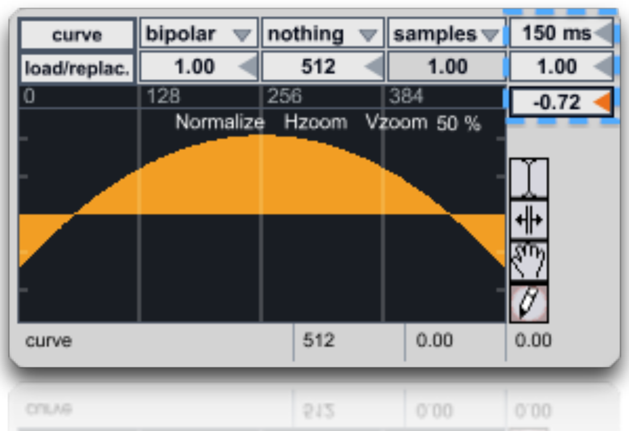
N.B. when you resize a shape, you will reset original default shape.

- shape select

There are 16 pre-generated shapes:

1. rectangle,
2. blackman,
3. **curve**,
4. **gauss**,
5. hamming,
6. hanning,
7. ***draw***,
8. sine,
9. cosine,
10. triangle,
11. saw,
12. square,
13. sigmoid,
14. sin(x)/x,
15. random,
16. **additive**

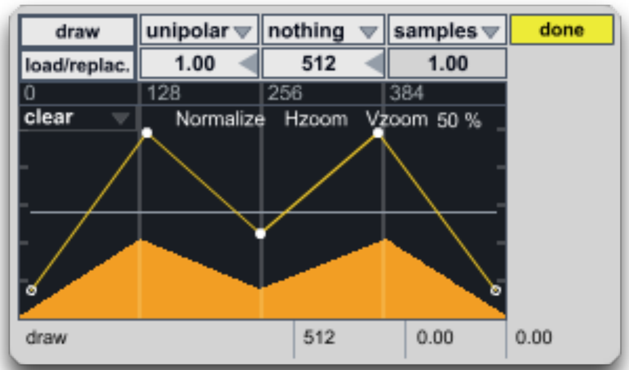
Some of them have additional controls parameters (bold). Selecting one of them, their additional parameters will be shown.



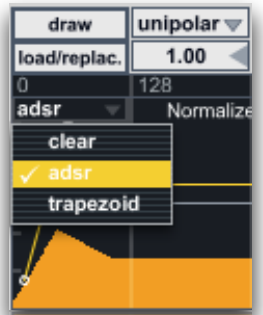
curve show three box-number, on the top (top right 150 ms) we have deformation update rate in milliseconds. Since these parameters work in real time, you can set a speed limit time to avoid cpu overhead. The other two parameters (1.00 and -0.72) sets middle/border curve deformation.

Linke curve, **gauss** show on the top the speed limit update rate and only a "Gaussian standard deviation" parameter.

• draw

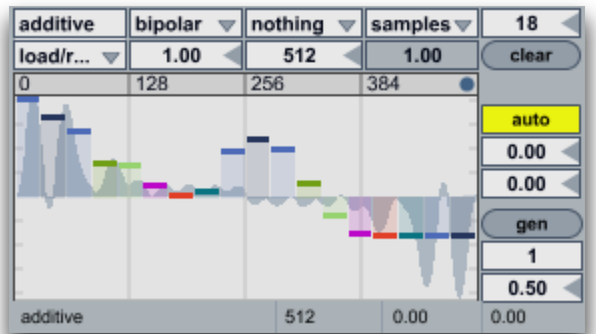


In Pulsaret, the grain envelope shape default is ***draw*** (bold italic), when selected appear yellow button (draw/done). This allows you to superimpose at the WavePad a function **break-points** where you can track the envelope segments. You can add new break-points by clicking on the function superimposed pad or shift + click on a specific break-point to remove it.



Two defaults shapes are available, **adsr** (Attack Decay Sustain Release) and a **trapezoid** (default grain envelope). You can switch between **unipolar/bipolar** automatically when you change view perspective in windowing WavePad, (see [Granular Streams](#) [WavePad display](#) for more explanations).

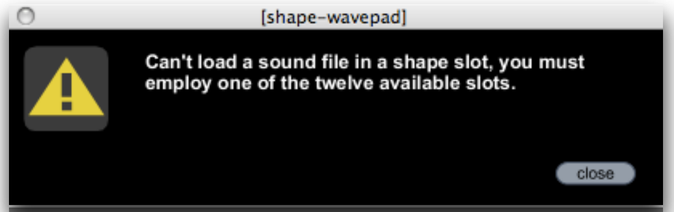
• additive



additive generate a composite waveform through weighted sums of sinusoids. It works similarly to Csound Gen19. Each slider in the additive pad, controls a specific partial amplitude. You can change the total harmonics number for the spectrum (top right **numbox 18**), **clear** all amplitude contributes of spectrum (bringing to 0). Toggle on **auto**, you will enable amplitude auto-rescale, thus the resulting shape it will be always normalized to one. N.B. **phase** and **offset** are normalized (0 ÷ 1). The two below num-box of **gen** button, sets the fine amplitude of a specific partial. You need selects a **partial number** (1 in the example), enter the **amplitude value** (0.5 in the example) at last click on **gen** to apply change.

• audio slot

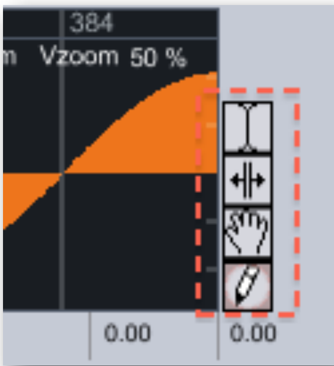
In the shape select menù, after the sixteen envelopes/prototypes you will see twelve empty slots named: **null**. You can employ them to load until twelve audio files form disk, just like **Granular streams** WavePad**.** See for further details. If you try to load, by dragging or from the menù, a sound file on a shape envelope or prototype, you will see a message box:



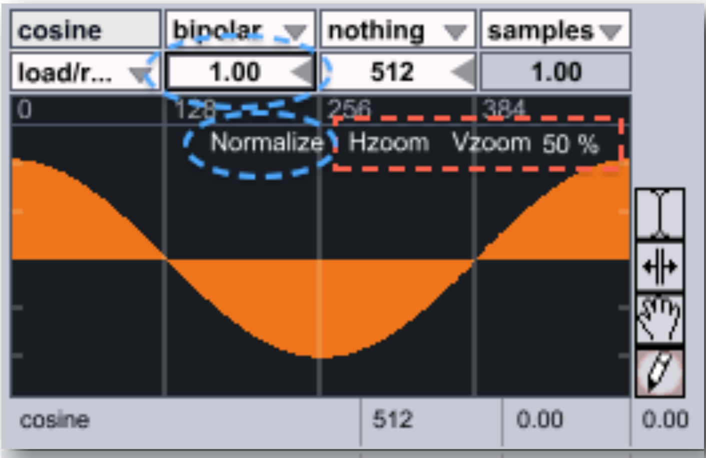
while if you drag on the Windowing WavePad a folder contains audio files, the slots they will be automatically filled. See WavePad, WavePad display, WavePad Snap, WavePad mouse for more details.

• WavePad

On the right side of the WavePad, you can choose the type of action: select, loop, move and draw.
The tools **select** (from top to bottom) and **loop** allows you to select a buffer length
move (hand) allows you to select a part (zoom) of prototypes or loaded audio file.



• WavePad display



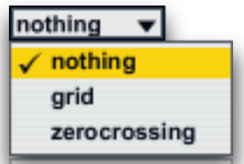
Clicking **Normalize** to scale the sample values in the WavePad buffer (0 dB), so that the highest peak matches the value of 1.
Clicking **Hzoom** to reset horizontal zoom (entire time display of file);
Clicking on **Vzoom** you resets vertical amplitude zoom (default 50 %). Mouse scrolling on right number (50 %) to set an amplitude zoom amount. N.B. unlike **Normalize**, **Vzoom** is not a destructive amplitude rescaling but only a graphics rescale.

In the image below, the last right menù (**bipolar**) switch to unipolar/bipolar view mode, unipolar mode shows values between 0 ÷ 1 while in bipolar shows values between -1 ÷1;

- WavePad Snap

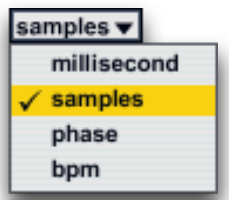
You can set the WavePad **snap mode** selection range:
bipolar switch wiev mode unipolar/bipolar;

Snap causes the start and end points of the selection to automatically move to specific points in the buffer, defined by the snap mode:



•**nothing** disables snap to allow free selection. This is the default;
•**grid** specifies that the selection start and end points (grain length) should snap to the vertical grid lines;
•**zerocrossing** instead of snapping the selection to a uniform grid, this mode searches for zero-crossings of the buffer data;

The seconds menù from the top (millisec...), sets the unit of time measurement used by the display:



•**milliseconds** sets the display unit to milliseconds;
•**samples** causes time values to be shown as sample positions in the target buffer. The first sample is numbered 0, unless the display has been shifted by the offset message;
•**phase** causes time to be displayed according to phase within the buffer, normalized so that the 0 refers to the first sample, and 1 refers to the last; N.B. grid values for phase, must be normalized (0. ÷ 1.), (see later);

- **bpm** specifies beats per minute as the time reference unit, relative to a master tempo and number of beats per bar, both of which you can set with the bpm message.

The numbox (1.00) specifies spacing of the grid lines for current unit of time measurement used by the display. N.B. grid values for **phase**, must be normalized (0. ÷ 1.)

- WavePad management

The top-left pop-up menù (Canto.wav), select one of the twelve waveforms to display on the pad. Current "granular stream" will granulate selected waveform. You can load/replace a sound file from menù or dragging file (or folder files) on the WavePad. The menù "null" item, it will be replaced with the file name.

load/replace soundfile = open a file browser to load an audio file on the selected slot (aiff, wave mp3 supported). Also you can drop a valid file on the wave pad, if it's unsupported, a message will be shown;
load/replace folder (soundfiles populate) = open a file browser to chose a folder to fill sound buffers audio files (aiff, wave mp3 supported). Also you can drop on the WavePad a sound files folder, the unsupported files will be not loaded and a message will be shown;
save buffer to file = export the contents of the buffer as Wav or Aif audio file;
clear buffer = erases the contents of buffer;
clear all buffers = erases the contents of all twelve buffers;
trim selection = will trim the audio data to the current selection. It resizes the buffer to the selection length, copies the selected samples into it, and displays the result at default settings. The buffer is erased, except for the selected range. This is a "destructive edit," and cannot be undone;
undo last selection = it causes the selection start and end points to revert to their immediately previous values. This is helpful when you are making fine editing adjustments with the mouse and accidentally click in the wrong place, or otherwise cause the selection to change unintentionally. Repeated undo commands will toggle between the last two selection states;
zoom out = resets the entire Hzoom (horizontal zoom) time display;
edit files infos = show a text-edit with information about loaded files;
open buffer window = opens the WavePad buffer window, or brings it to the front if it is already open. The windows is resizable but not editable, you can **scrolling the mouse over the buffer window;**

normalize (1.00) = scale the sample values in the buffer so the highest peak matches the value given by the argument. This can cause either amplification or attenuation of the audio, but in either case, every value is scaled. When a sound file is loaded, default value is 0, it means original wave amplitude, value greater then 0 are saved and loaded in the project. N.B. Normalized value of 0 can be interpreted like:

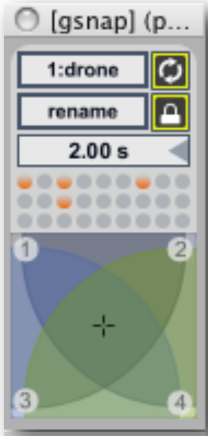
1. original file amplitude if you have not change it before,
2. normalizing at 0 value (i.e. 0.001 amplitude gain, -60 db).

Snapshots

• store/recall

A snapshot is a photo of graphical interface (GUI parameters) in the current state. Each snapshots module can store up to 100 snapshots. In Pulsaret there are ten modules "snapshots":

- **main** manage main mixer parameters;
- **streams 1 ... 8** manage granular stream parameters;
- **gsnap** (below image) manage all Pulsaret parameters (main mixer and streams);



You can make transitions between two or more snapshots in more **streams** and/or **main mixer** simultaneously. You can subscribe or unsubscribe determinate Pulsaret parameters.
N.B. **gsnap** transitions works on all Pulsaret parameters, therefore no local transitions should be at works. This in order to avoid more transitions controlling the same parameter.

When you find an Interesting "sound" you can store new snapshot just shift + click on a button in the **snap-pad**, thus the preset button will light orange. You can recall a snapshot by simply clicking on the **snap-pad** buttons; all the widgets subscribed (see later) will be restored at current snapshot value, immediately.
Alternatively a snap can be stored from pop-up menù, a message box will ask you to digit a snap number in (see below for more explanations).

• micro pad

The micro-pad's goal is to obtain intermediate values between four snapshots (interpolation). You can configure four nodes (snapshots).

locked (lowest in the image): nodes: the mouse can only edit the nodes position and size. unlocked: slider; the mouse can only changes the slider location.
refresh (highest in the image) enable/disable **auto-recall** snapshots, see below for more details.

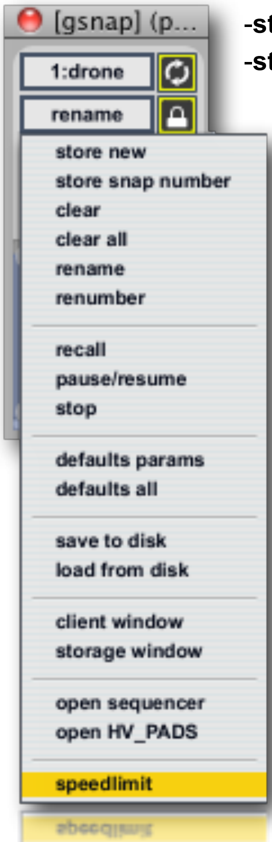
• transitions

Transitions are a linear interpolation between two snapshots, with one time duration.
You can start a transition only from the menù **recall** item.
Is important to understand that the transition occurs from the **current parameters positions,** toward the selected snapshot.

actual GUI widgets positions >>> (toward) **selected snapshot** (time duration)

You can change the **transition-time** (2.00 sec in above image). The transitions can occur simultaneously on the streams and/or on the main. If **auto-recall** (see above) is enabled, selecting from the menu a snapshot slot to start the transition.

- snapshots manage



-**store new** snapshot using the next empty preset slot;
-**store snap number**, enter a number or a list separated by space, to store;
-**clear** current selected snapshot;
-**clear all snapshot**, a message box will ask you to confirm;
-**rename**, open a window list where you can name or re-name your snapshots;
-**renumber** sort all snapshots stored into consecutive, beginning with 1;
-**recall**[1] start a transition toward current snap number;
-**pause/resume** current transition (if occur);
-**stop transition** cancel current transition (if occur);
-**default params**, resets subscribed **parameters** at default value, only for the current streams or main[2];
-**defaults all** resets subscribed **parameters** and **general settings** (not subscribed) at default value, only for the current streams or main[3];
-**save to disk** current snapshots-bank;
-**load from disk** current snapshots- bank;
-**client window,** open clients list to subscribe, unsubscribe, setting interpolation type and more;
-**storage window** open storage window displays any stored presets;
-**open sequencer** improviser unit (see Snapshots Sequencer);
-**open HV_PADS** (Hyper Vectorial Pads, see HV_Pad);
-**speedlimit** set update time limit for transitions and micropad nodes interpolations.

N.B. all the presets are saved in the project (see save/load project further), however, you can manage individually save/load snap-bank, for example useful for exchanging presets between the streams.

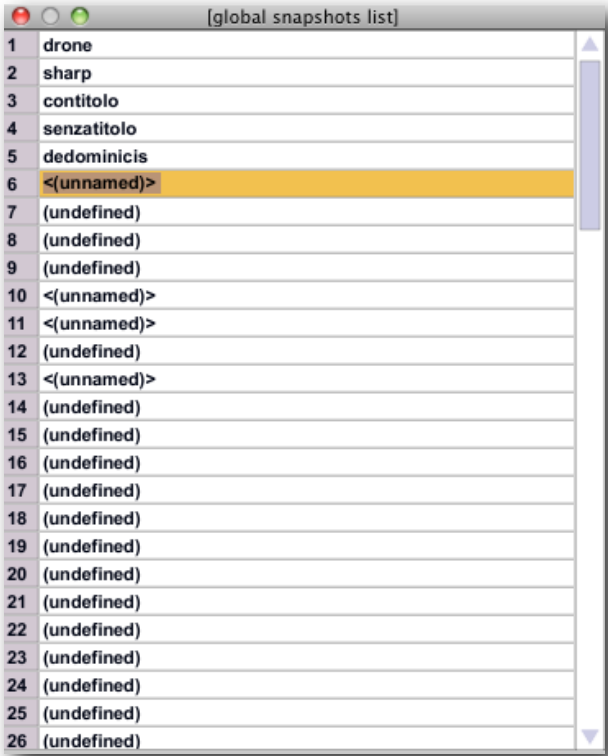
- rename window

Opening rename window to fast renaming snapshots:

[1] *you can enable or disable auto-recall, if enabled selecting a snap from the menù, it will start the transition, if disabled it will be* ***only selected****.*

[2] *gsnaps will restore all Pulsaret subscribed parameters at default values (eight streams and main mixer).*

[3] *gsnaps will restore all Pulsaret subscribed parameters and general settings (not subscribed) at default values (eight streams and main mixer).*



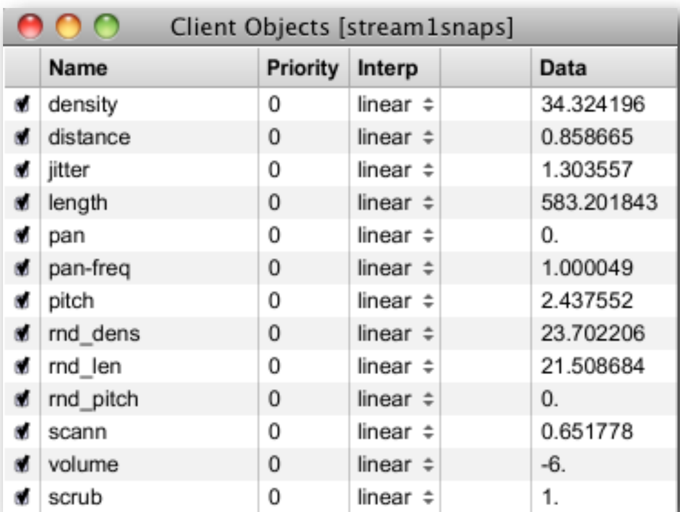
double clicks on the **<(unnamed)>** to edit text, then rename it.

You have an additional feature, scrubbing on the left column numbers, you will able to switch between snapshots. Thus this secondary unit can be considered important when you experiment your sound or during performance.

N.B.
Pulsaret store defaults values every time you open the application in the **slot 99.** Slot 99 contains either **default parameters** value, either **defaults all values** (parameters and general settings).
Slot 100 is where Pulsaret store the last GUI state (this occur when you save/load a project).

• subscribe/un-subscribe transition clients

The two windows **client** and **storage**, are both non-interactive:

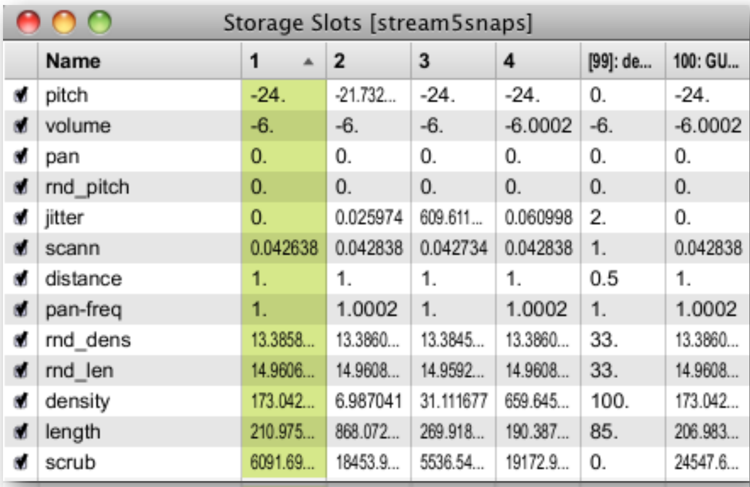


	Name	Priority	Interp	Data
✓	density	0	linear	34.324196
✓	distance	0	linear	0.858665
✓	jitter	0	linear	1.303557
✓	length	0	linear	583.201843
✓	pan	0	linear	0.
✓	pan-freq	0	linear	1.000049
✓	pitch	0	linear	2.437552
✓	rnd_dens	0	linear	23.702206
✓	rnd_len	0	linear	21.508684
✓	rnd_pitch	0	linear	0.
✓	scann	0	linear	0.651778
✓	volume	0	linear	-6.
✓	scrub	0	linear	1.

client window (accessible from the **client window** menù item, see above) shows the current **subscriptions widgets list** (ever header name), **priority**, **interpolation** and **data** belonging to subscribed widgets.

A few types of interpolation can be changed:

- **off**: no interpolation;
- **linear**: Linear interpolation. Presets recalled will be interpolated using a standard linear algorithm.
- **threshold**: Threshold. Takes optional argument (float), which sets the threshold. Presets recalled will recall data from the first preset specified when the fade amount is below the threshold, and will recall data from the second preset specified when the fade amount is greater than or equal to the threshold. e.g. threshold: '**fade**' < thresh = value a; **fade** >= thresh = value b;
- **inverted threshold**: Inverse threshold. Takes optional argument (float), which sets the threshold. Presets recalled will recall data from the first preset specified when the fade amount is greater than or equal to the threshold, and will recall data from the second preset specified when the fade amount is less than the threshold. e.g. inverted thresh: '**fade**' < thresh = value b; **fade** >= thresh = value a;
- **exponential curve**: Power curve. Takes an additional argument (float), which sets the exponent to which the fade amount will be raised. Presets recalled will recall data between the two specified presets, along the curve described. Power curves can be used to create faster or slower "attacks" and "decays" for the fade envelope;
- **table**: Table-specified curve. Takes optional additional argument (see transition curve below), which specifies the name of a table to use for curve lookup (**tab1, tab2, tab3 or tab4**). Presets recalled will recall data between the two specified presets, along the curve described in the table. Tables are assumed to contain values between 0 and 100, representing the new fade amount * 100. If the lookup fade amount does not fall exactly onto a table-specified value, linear interpolation is used to determine the new fade amount. In Pulsaret you have four draw functions to draw your curve, see transition curve below.


Storage Slots [stream5snaps]

Name	1	2	3	4	[99]: de...	100: GU...
pitch	-24.	-21.732...	-24.	-24.	0.	-24.
volume	-6.	-6.	-6.	-6.0002	-6.	-6.0002
pan	0.	0.	0.	0.	0.	0.
rnd_pitch	0.	0.	0.	0.	0.	0.
jitter	0.	0.025974	609.611...	0.060998	2.	0.
scann	0.042638	0.042838	0.042734	0.042838	1.	0.042838
distance	1.	1.	1.	1.	0.5	1.
pan-freq	1.	1.0002	1.	1.0002	1.	1.0002
rnd_dens	13.3858...	13.3860...	13.3845...	13.3860...	33.	13.3860...
rnd_len	14.9606...	14.9608...	14.9592...	14.9608...	33.	14.9608...
density	173.042...	6.987041	31.111677	659.645...	100.	173.042...
length	210.975...	868.072...	269.918...	190.387...	85.	206.983...
scrub	6091.69...	18453.9...	5536.54...	19172.9...	0.	24547.6...

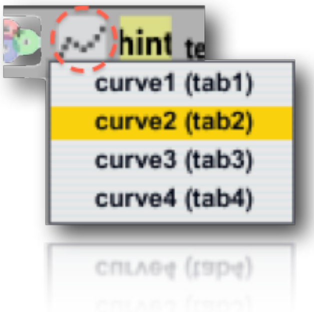
The **storage window** displays any stored presets. The active (recalled) preset is displayed in highlight green. If any client value is changed, are displayed in italics. Eventually, both of these windows will be configurable and editable, so that they can provide display and editing control for clients and storage sets.

N.B. When you open storage window, a message box will be displayed:

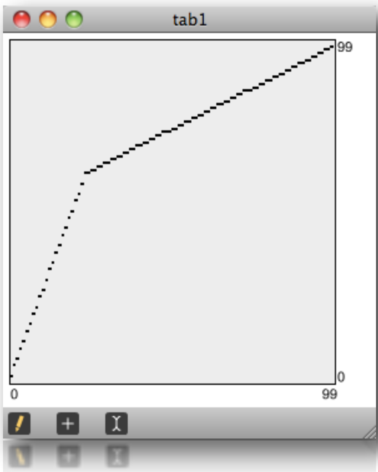
this is to avoid an overhead of CPU when displaying storage window.

• transition curve

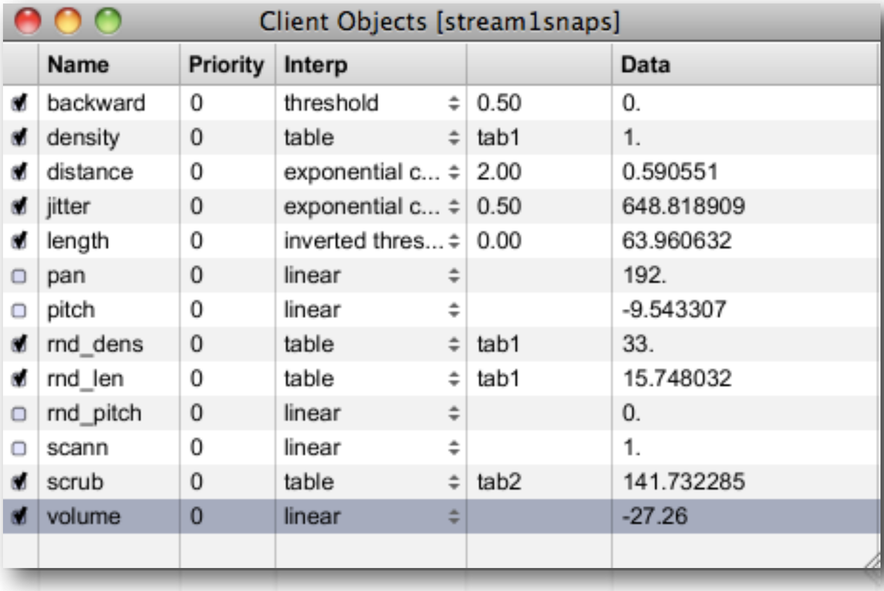
All transition by defaults are linearly interpolation. You can superimpose your interpolation curve at one or more subscribed widgets. Four draw functions are available, from toolbar or menù bar. Open one of them and draw your curve. This feature is **saved** on Pulsaret **project**.



Draw a shape on the function window



In order to maps one or more subscribed clients with one of four functions draw, you need open **clients window** and select **table** from pop-up menù (over **interp** header). Now you must digit the table name, that can be: **tab1**, **tab2**, **tab3** or **tab4.**



Client Objects [stream1snaps]

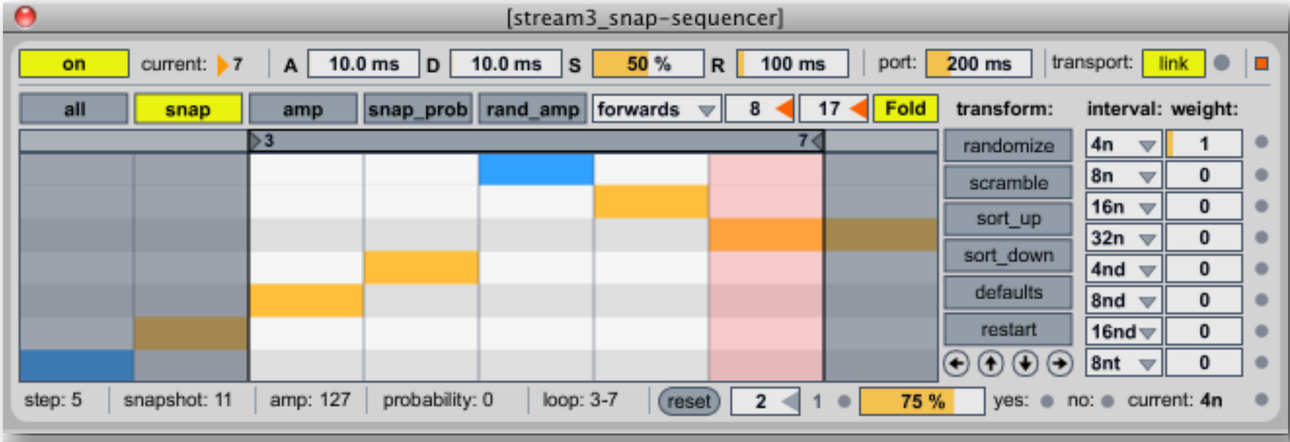
	Name	Priority	Interp		Data
✓	backward	0	threshold	0.50	0.
✓	density	0	table	tab1	1.
✓	distance	0	exponential c...	2.00	0.590551
✓	jitter	0	exponential c...	0.50	648.818909
✓	length	0	inverted thres...	0.00	63.960632
☐	pan	0	linear		192.
☐	pitch	0	linear		-9.543307
✓	rnd_dens	0	table	tab1	33.
✓	rnd_len	0	table	tab1	15.748032
☐	rnd_pitch	0	linear		0.
☐	scann	0	linear		1.
✓	scrub	0	table	tab2	141.732285
✓	volume	0	linear		-27.26

in the above example, we employ:

- **tab1** like transition curve for **density**, **rnd_density**, **rnd_length;**
- **tab2** like transition curve for **scrubb**;
- **pan**, **pitch**, **rnd_pitch** and **scann** are unsubscribed from the snapshots transitions system.

See **client window** above for more explanations.

Snapshots sequencer improviser unit



Snapshots sequencer rhythm improviser unit i.e. the extended name, the aim of this device is to control the snapshot sequence88 by improvising with different rhythmic pulses. The sequencer shown on vertical the snapshots numbers while in horizontal steps sequence.

- beats cycle

A metronome is synced to global transport and outputs a trigger on each beat. The beats are counted in what we're defining as a a cycle of <n> beats (Cycle sub-patcher). Each time the cycle restarts, we're choosing whether or not we want to change the current time interval. This is set by using a probability factor (ChangeProb sub-patcher). If the program decides to change the time interval, a trig is sent.

- time intervals

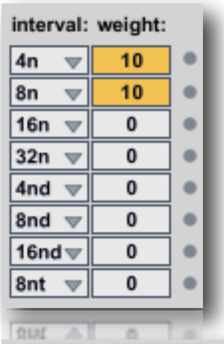
Since eight time intervals can be selected. The choice is performed with a random function, according to a weight probability for each possible interval - an interval with a weight of 2 will have twice as many chances to be selected by the random procedure that an interval that has a weight of 1.

- step sequencer

When a given time interval is chosen, it is used as a step rate for driving the step-sequencer. The step-sequencer is synced to global transport, so the steps are played in accordance with transport tempo. The snap-sequencer has many editing modes that allow for modifying the snapshots content (snap, amp, etc..).

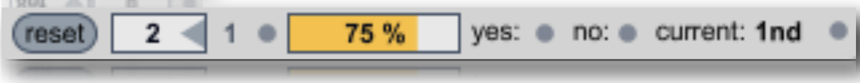
- output

The snap-sequencer outputs a list of data each time a step is triggered. Each data type (snap, amp, amp _rand...) is interpreted in order to recall a snapshot, with additional possibilities such as: random amplitude of envelope and snap recall probability.



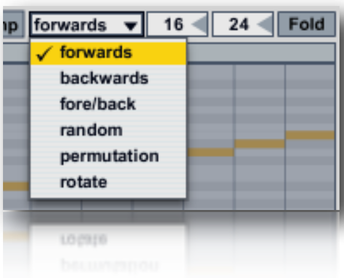
Up to eight time intervals can be selected. The choice is performed with a random function, according to a weight probability for each possible interval - an interval with a weight of 2 will have twice as many chances to be selected by the random procedure that an interval that has a weight of 1.

You can set "general changes probability" (75 % in the example) , 0 % no interval change occur, as all intervals probability weight to 0. Change and no change led (yes/no), shows if the random parts it's to change or not. Random number generator fall every "change interval" time (2 in the example), beat sequence is displayed on the right.



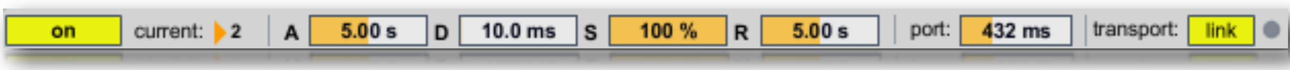
The most left **reset** button, resets sequencer to first step, in permutation mode reinit sequence, no action in random mode.

Edit mode selection is done using this set of tabs, view all displays all of the available data (and allows you to edit the snapshots). **Snap** mode displays and allows editing of the snap only. **Amp** mode allows you to edit the volume of snapshot envelope. **Duration** mode is used to change the step size of each value, while **Probability** mode allows you to manipulate the likelihood of a snapshot.



Sequencer play mode menù read steps in the explained way. You can chose the number of the step (**columns**) and snapshots range (**row**), then **fold/ unfold** lets you chose to display all possible snapshots, or only a specific set of snapshots.

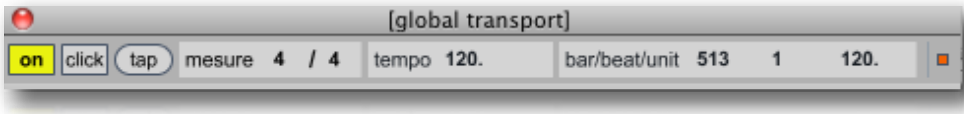
Some kinds of transformations are available their names are auto-explicative.
Through the four arrow keys at the bottom, up and down (snap number) or left and right (in time) you can shift the entire sequence.



Snapshots Sequencer is an **ADSR** (Attack Decay Sustain Release) every step envelops the streams (or main) amplitude **audio signal**.You can define a **Attack Decay** time and **Sustain** percentage, at last **Release** time. When Sequencer is turned off, audio signals (streams or main) pass through normally.

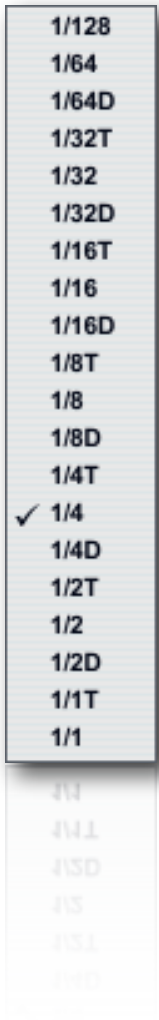
- global transport

The global transport it's designed to access quickly all what is needed to control the clock of timing Pulsaret objects.



Here is a listing of the note and tick values associated with common note durations. Note value abbreviations that can be used in Pulsaret to specify time are in bold.

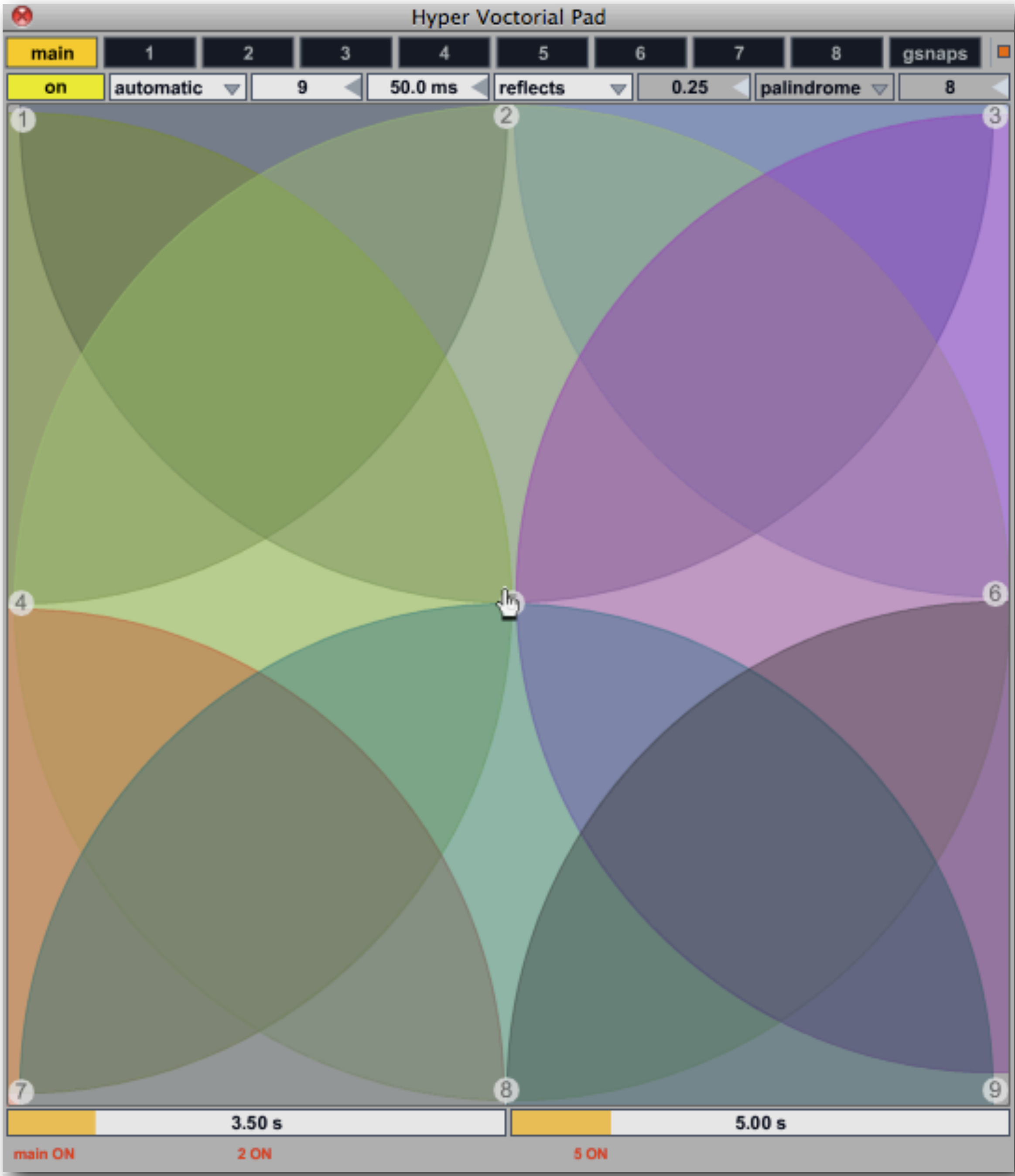
128n - One-hundred-twenty-eighth note - 15 ticks
64n - Sixty-fourth note - 30 ticks
64nd - Dotted sixty-fourth note - 45 ticks
32nt - thirty-second-note triplet - 40 ticks
32n - thirty-second note - 60 ticks
32nd - Dotted thirty-second note - 90 ticks
16nt - Sixteenth note triplet - 80 ticks
16n - Sixteenth note - 120 ticks
16nd - Dotted sixteenth note - 180 ticks
8nt - Eighth note triplet - 160 ticks
8n - Eighth note - 240 ticks
8nd - Dotted eighth note - 360 ticks
4nt - Quarter note triplet - 320 ticks
4n - Quarter note - 480 ticks
4nd - Dotted quarter note - 720 ticks
2nt - Half note triplet - 640 ticks
2n - Half note - 960 ticks
2nd - Dotted half note - 1440 ticks
1nt - Whole note triplet - 1280 ticks
1n - Whole note - 1920 ticks



In Pulsaret you can find also this formula of value.

Hyper Vectorial Pad[1]

- general explanation



HV_pad (i.e. Hyper Vector Pad) is 2D parameter interpolation user interface. Displays nodes in a 2-dimensional space, and calculates the distance from a pointThe distance factor determine the weight between snapshots interpolation. Like for snapshots-module, we have ten **HV_Pad** grouped in a unique window.

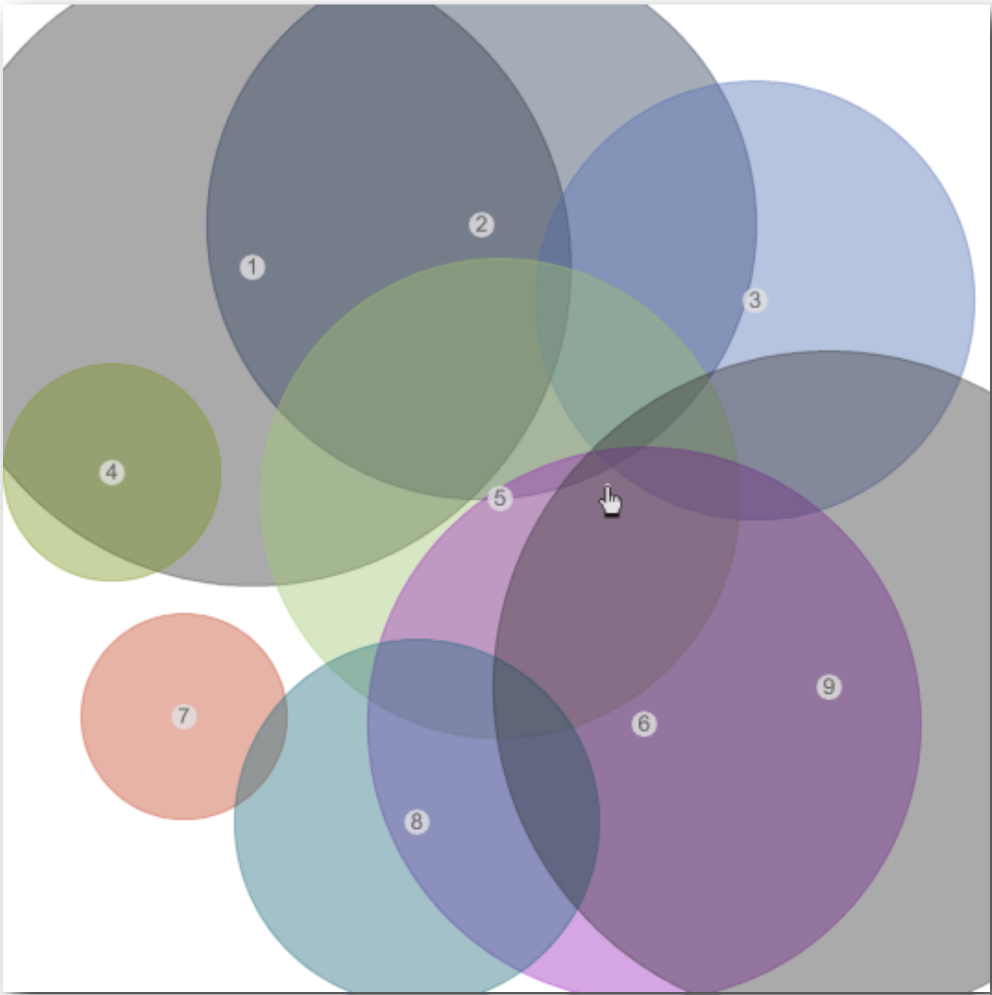
- **main;**
- **streams 1÷8;**
- **gsnap;**

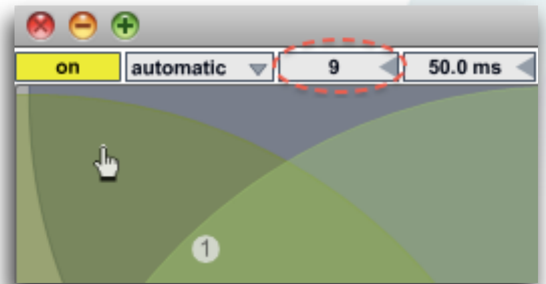
[1] *See VMCI Virtual Midi Control Interface (http://www.csounds.com/maldonado/), thanks Gabriel Maldonado.*

You can open HV_Pad window from application menù bar (cmd + P on Mac or ctrl + P on Windows) or through the tool bar button.

The vectorial pads (10) are contained inside HV_Pad: **main, streams** (8) and **global snapshots,** clicking on tab you change among them.

HV_Pad extends the functions of micro-pad, interpolating until 24 snapshots together, thus you can modify hundreds of parameters with a single mouse motion, according to a structure configured by the user. This control method is called *Hyper Vectorial Synthesis Control*. Normally the user interacts with these areas with the mouse or through self-scanning exploration, however a remote MIDI control is possible, by assigning the X Y pointers (see MIDI/OSC I/0).





In the image below, from number box (red circle) you can sets the number of nodes, you can set a specific configuration (system nodes) by **clicking** and **dragging** center of node to move it or **Option Key** (**alt**) to change node size.

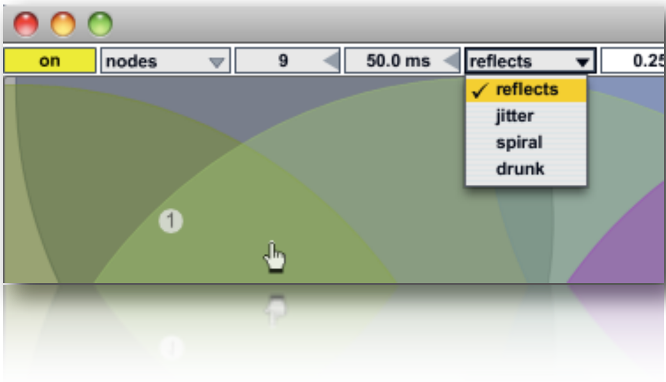
N.B. The nodes are numbered progressively, therefore each node (on the pad) must be corresponds by a stored preset snapshot. If you

employ an empty snapshot, the interpolation between them will not work properly. Also you cannot disabling completely the nodes (node area = 0), to avoid fall in the same problem.

• pad mouse behavior

The pop-up menù allow you to chose mouse behavior:
•**Automatic**: allows the mouse to change the nodes position, size and slider (pad scrub);
•**Nodes**: the mouse can only edit the nodes position and size, **alt + mouse scroll** to change node **size**; **click and drags** on the node number, to **move** it;
•**Slider**; the mouse can only changes the slider location (pad scrub).

You can automatize HV_pads scrub (slider) choosing a scanning mode:



Scanning method:

reflects causes a reflection when the cursor reaches the edge;
jitter outputting random numbers within a moving specified range around current scrub mouse position;
drunk will perform a "drunk" walk, creating unpredictable paths;
spiral centripetal, centrifugal or palindrome, with phase deformation parameter;

The **jitter step size** (number box 8) it will be showed only when jitter is selected. This parameter sets the random amount of jittering around of the mouse position during scrub pad.

Automatic exploration in **reflects**, **jitter** and **drunk**, you can interact with the pad. The values of mouse and direction, will be the new coordinates of scanning. Instead in **spiral** you cannot interact with pad

• X/Y time

The X Y indexes scanning ratio, depends by the "frequency" X/Y axes. These are controlled by the two sliders showing value in seconds (i.e. time = 1/frequency).

You can set scanning time for the two axis (2D XY). For example 3.50 sec. for the **X** axe it means that horizontal slider scrub it take 3.50 sec to pass from left edge to right edge of the pad, **Y** = 5.00 sec.

X Y parameters, they assumes different means according to selected scanning mode:

in **reflects** is expressed in **time,** is the time it takes to move from **X** = left/right and **Y** = up/down;
in **drunk** is expressed in **time**, is the deviation amount rnd(**X**) and rnd(**Y**);
in **spiral**: **X** is the time for each sub-spiral, **Y** is the total duration until reach all pad area;

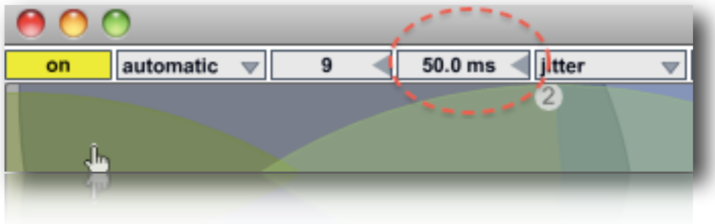
The **spiral** mode, will enable two related parameters:

centripetal moves from outside to inside, when it reaches the limit again, repeat;
centrifugal does the opposite;
palindrome is a combination of both;

The **spiral phase deformation** (number box 0.25) 0: no spiral; 0.25 circular; > 0.25 crushed; > 0.5 clockwise rotation

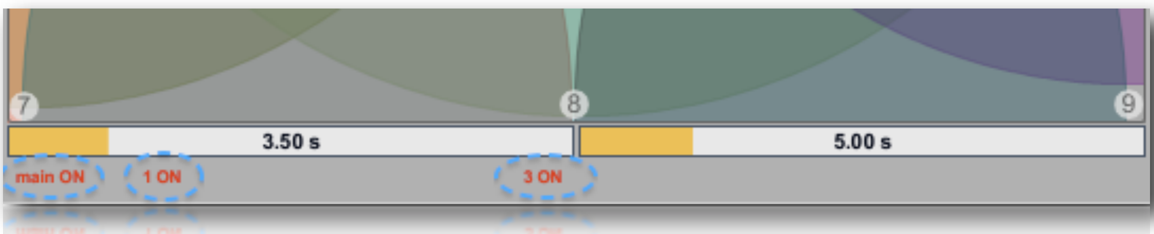
- miscellaneous

Auto-exploration is possible only when the DSP is on (see DSP_settings),



on/off toggle auto-exploration according to selected mode.

The red circled number box, sets the interpolation update time in milliseconds, this is a speed limit to avoid an overhead of cpu with complex interpolations, be careful with short times!

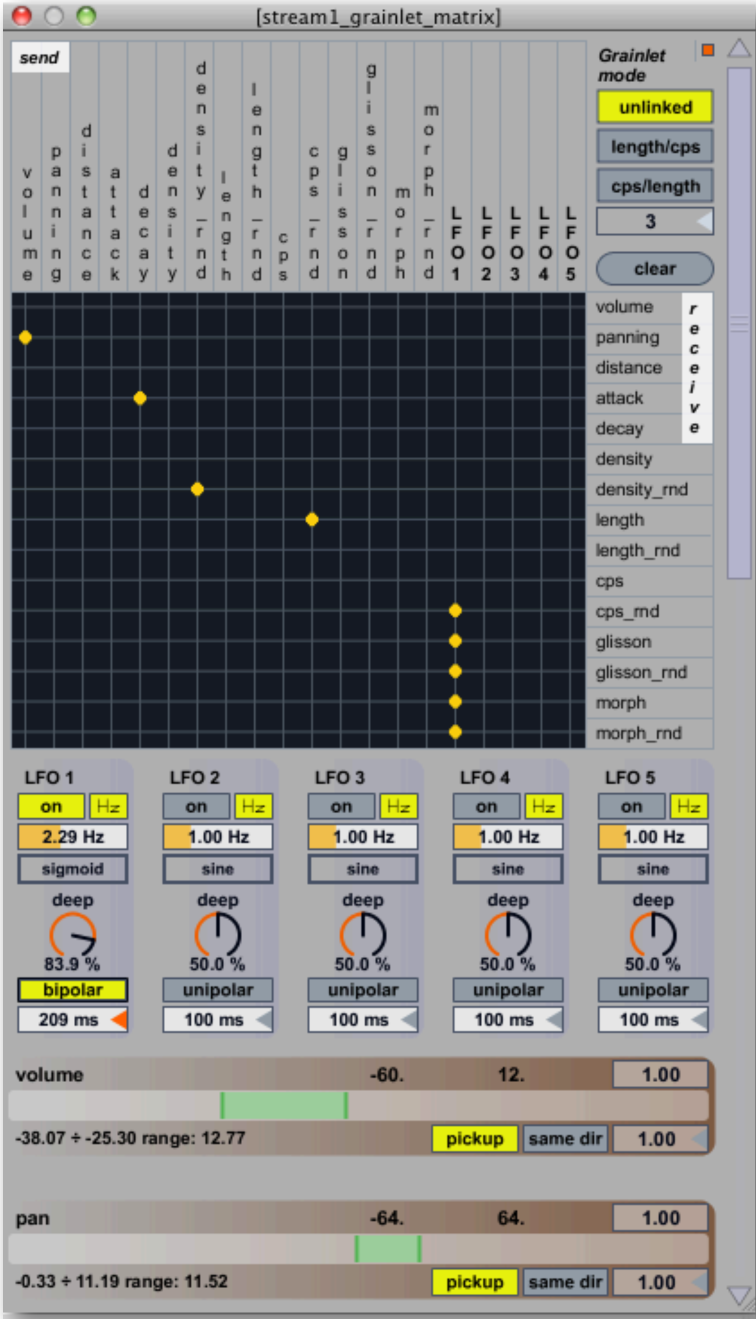


A little status bar at the bottom of HV_Pad, show the currents pads enabled for auto scanning. In fact passing among ten available pads, we need a monitor remembering us pads at works.

Matrix

• general explanation

Pulsare.m4l matrix linkage is employed to create a dependency between parameters. Moreover you can rescale, reverse controlled parameter range and/or move it by LFO (Low Frequency Oscillator). This technique is named: Grainlet Synthesis.

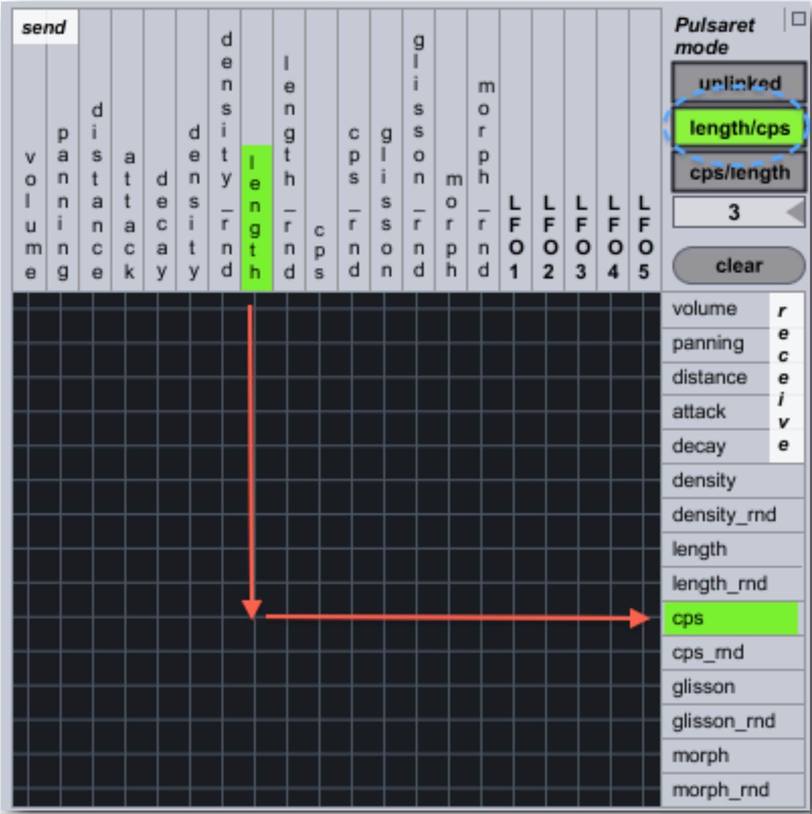


• parameter linkage

On the top of the Matrix (horizontal) we have sending parameters, They will be linked with right ones (vertical). moving a send parameter, you will move controlled (linked) parameter.
Every controlled parameter can be mapped on a own range and/or reversed, it means it will be moved on the contrary regarding the motion of the parameter that controls it.

• grainlet/pulsaret mode

In grainlet mode you you can interconnects all granular parameters freely, you can also toggle in Pulsaret mode.



A Pulsaret consists in a brief burst of energy. The grain duration contains only one cycle, varying the grano length you change "duty cycle" of pulsaret, while density remain fix and indipendent. You can link the two parameters in two ways: **length -> cps** or **cps -> length**.

cps = (1/length) * cycle*N*
length = (1/cps) * cycle*N*

e.g. 100 Hz = 1/100 = 0.01 ossia 10 ms.
Optionally you can specify the number (numberbox) of cycles for the pulsaret.

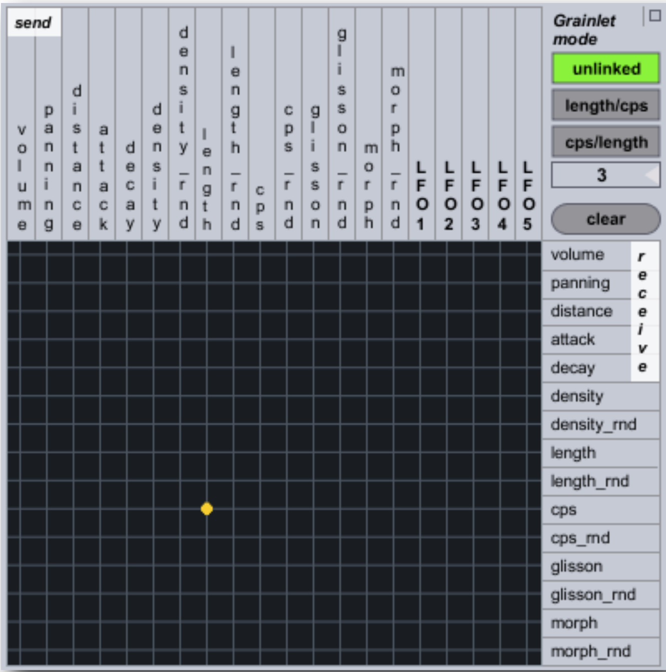
e.g.
cycleN = 2
cps = 100 Hz
length = 1/100 = 0.01 * 2 = **0.02 Sec** (ossia 20 Ms)

for the cps/length linkage:

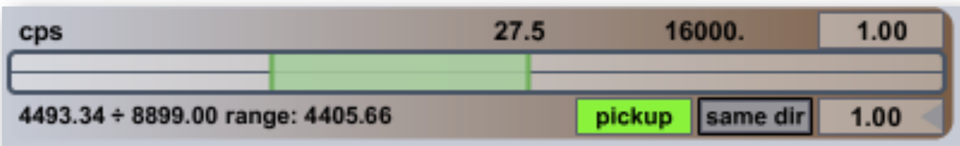
cycleN = 3
length = 45 Ms = 0.045 sec
cps = 1/0.045 * 3 = **66.66 Hz**

• parameters rescale

In the below example, **length** and **cps** are linked:



all length range is employ to move cps parameter.
If can maps entire length range on a little part of cps parameters:
You can also enter your values on the numbox (27.5 - 16.000), for changing global range and change exponential curve mapping (numbox):
- min value for the selection (click and digit a new value, then press enter)
- max value for the selection (click and digit a new value, then press enter)
- exponential base value (default 1. = linear)

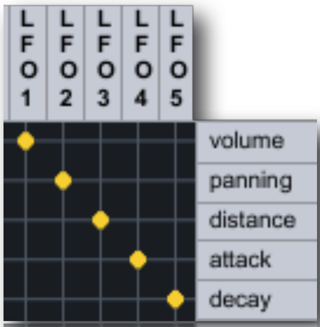


If pickup is enable, when controlled parameter is out of range, it will no moved until it will be bring inside of the range. it wock in analogous way to MIDI/OSC pickup, see MIDI/OSC for more details.

same dir/inverse, make an inverse maps of the controlled parameter.
the last parameter, controlling general rescale output, 1. it means no rescale.

- LFO modulation

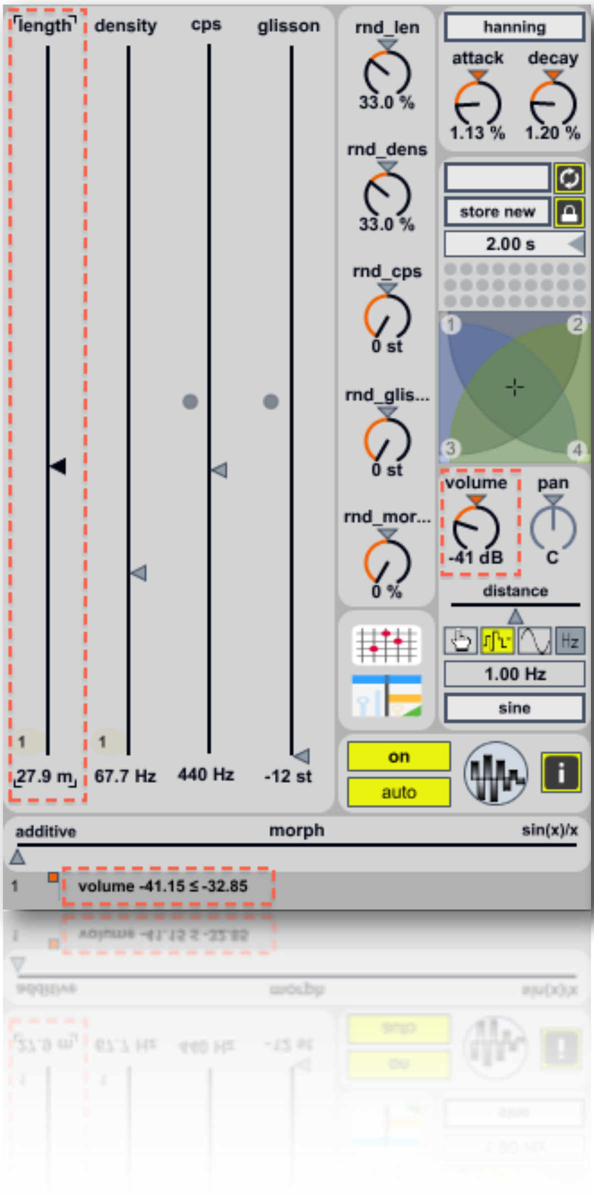
Low Frequency Oscillator (LFO) is usefull to automate one or more parameters, they will be constantly moved around one middle value depending by shape, frequency and deep of modulation of LFO.



in the above example, LFO1 (of the 5 availables) modulate volume parameter, LFO2 panning etc.... After you have assign matrix linkage, you need enable LFO clicking on the on button.

The LFO params are (from top to bottom):
-on/off LFO (Low Frequency Modulation)
-toggle between beat synced and hz; in sync mode, the beat division switches specify the delay time in 16th notes; in time mode, the pan rotation is expressed in hz.
-panning rotate frequency in Hz
-16 note: each numbered switch represents the time in 16th notes, note that these switches are only active if the time is set to sync; you can change global time by the transport unit.
-lfo shape, you can select one of the windowing shapes availables
-modulation deep in %
-uni/bipolar range, unipolar = 0 to 1 modulation range (with 100% deep); bipolar= -1 +1
-change the speedlimiting time parameter update.

- status bar



In the Pulsaret.m4l status bar, report current matrix modulation state.

The receiving parameter wait until sending parameter reach the same value i.e. **pickup** feature, (Ableton[1] Live inspired).

Pickup system wait until sending parameter reaches receive widget position. On the status bar you will see a message shown the gap between send/receive parameters.

On the most left status bar we have *grain overlaps factor,* you need enable tiny orange button.

---

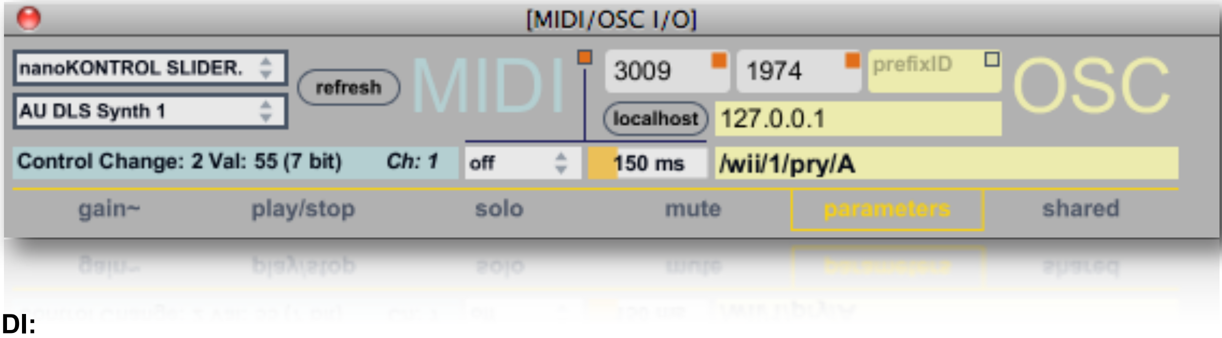
[1] *Ableton Live is a loop-based software music sequencer and DAW for Mac OS and Windows by Ableton. The latest major release of Live, Version 8, was released in April 2009. In contrast to many other software sequencers, Live is designed to be an instrument for live performances as well as a tool for composing and arranging. It is also used for mixing of tracks by DJs, as it offers a suite of controls for beatmatching, crossfading, and other effects used by turntablists, and was one of the first music applications to automatically beat match songs. It does not support traditional musical notation. (*Wikipedia*)*

MIDI/OSC I/O

• header

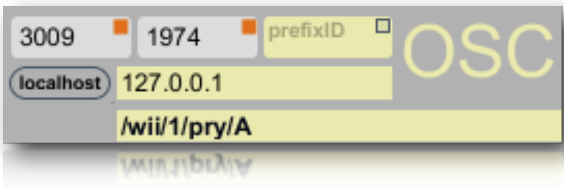
Open **MIDI/OSC I/O** from application menù bar CMD + M (Mac) or CTRL + M (Win) or from tool bar, the header is divided in two parts: MIDI (Music Instrument Digital Interface) and OSC (Open Sound Control). You can also browse quickly the parameters-families through the below menù:

- **gain~:** mixer granular streams faders;
- **play/stop:** mixer granular streams triggers;
- **solo:** mixer solo toggles;
- **mute:** mixer mute toggles;
- **parameters:** granular streams parameters;
- **shared:** common controllers;



MIDI:
Pulsaret routing only **Cc** (Control Change) MIDI messages, no *Pc* (Program Change), no *Note On/Off,* no *Poly Pressure,* no *After Touch* and other MIDI messages are supported. The Control Change messages, generally send continuous data between 0 ÷ 127 (i.e. 7 bit) steps resolution. You can enable **14 bit supports** to improve resolution (see below).
For motorized devices, Pulsaret can synchronize software parameters with the hardware controllers. Thus when you change, for example, **number of stream** from GUI, Pulsaret send value through MIDI and OSC output ports. You will have software and hardware parameters synchronized.
I/O port selecting:
The two pop-up menùs shows a list of available ports according to hardware device detected, click **refresh** to update list, useful when hot connecting a device(s).
Below the MIDI I/o ports, we have **"MIDI monitor"** where you can see raw coming MIDI data. Midi monitor interpret raw MIDI data parse, providing a short description about generic messages kind and MIDI monitor is able to understand if incoming message are 7 or 14 bit resolution.

OSC:[1]
Dispatch messages through an OpenSound Control address hierarchy with pattern matching. Values are serialized and sent over the network as OSC compatible UDP;



Input port (3009) set input port to send messages to at host;
Output port (1974) set output port to send messages;

[1] *OSC is a CNMAT Max objects, can be found at:* *http://www.cnmat.berkeley.edu/MAX*

Tiny oranges buttons enable/disable respective OSC I/O.

Very important!: default values for I/O ports: 30-09-1974 is the author birthday date.

We need also configure **host** (127.0.0.1) output , entering an **IP** address or **hostname**, clicking on **localhost** to assign default ip value (i.e. local host 127.0.0.1).

Like MIDI monitor, **OSC monitor** show OSC I/O data, you can select with mouse the text string address (OSC path) and paste it on widget controller (see below).

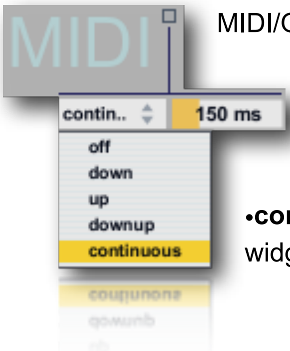
You can enter a prefix ID (can be numbers and/or text) for the the OSC output address. When an incoming OSC message is synced on a widget parameter, this read and write values employing the same path address. Some time is useful append an id prefix to differentiate path in order (for instance) to send message to another device .

N.B. OSCulator [1] can be employed like MIDI/OSC monitor beyond supplying advanced features.

N.B. All settings will be saved on the project (see overview for more explanations)

• sync

Sync system allow you synchronize software parameters with hardware devices. Thus Pulsaret send out through MIDI and OSC ports parameters data. Default **sync** is disabled for all widgets, then no MIDI/OSC data are sent. You can select enable **sync** out for each widget controller or superimpose a common feature at all widgets (see below).
For example, you can enable sync for all widgets controllers, by selecting the item from header menù.
Sync have five outputs mode:



MIDI/OSC out mode for all widgets:

- •**off** no MIDI/OSC outs data;
- •**down** send MIDI/OSC data when click-down left mouse button;
- •**up** send MIDI/OSC data when click-up left mouse button;
- •**downup** send MIDI/OSC data when click-down and click-up;
- •**continuous** send MIDI/OSC data continuously. N.B. this overwrite all settings of the widgets.

Every time you move a Pulsaret parameter on the GUI or when a transition/sequencer/HV_pad is at work, **sync** values are sent.

To avoid an overhead of outputs data, through the **number box** (150 ms) you can change MIDI/OSC output update rate in milliseconds. For instance during the snapshots transitions, you should limiting sync data in order to preserve a normal cpu overhead: **MIDI out update rate**, expressed in milliseconds set the speed limit MIDI output data.

• widget mapping

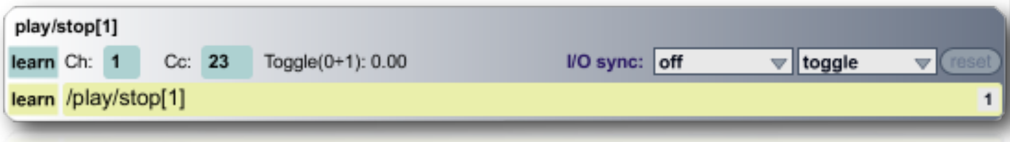
---

[1] *OSCulator is a software that can be used with many different hardware devices and software. For example, with OSCulator, your Nintendo Wiimote can talk to major MIDI sequencers or your favorite console emulator or even the Kyma sound design workstation. And your iPad, iPhone, iPod Touch or Lemur can do just the same… with great ease of use. http://www.osculator.net*

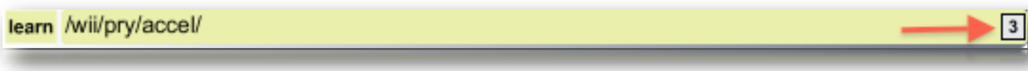
We have two widgets mapping kinds:

1. **Cc_raw**
2. **Cc_scale**

Cc_raw is addressed to mapping Pulsaret buttons/toggle parameters, Cc_raw read/write MIDI/OSC returning integer values (between 0 and 127 for MIDI), **Learn** function waits to feel a MIDI CC or OSC address, when it happens the the control is set up on the incoming CC number and MIDI channel or OSC path address.
Now you can controlling Pulsaret parameter via MIDI and/or OSC.
Current MIDI Cc, Channel number and raw value are shown. Of course you can manually change MIDI Cc or channel. N.B. sets Cc to -1 for disabling controller.



Both MIDI and OSC have **learn feature**. Toggle learn and send a MIDI or OSC data in order to assign incoming Cc or OSC address. In the example, Cc 23 on first MIDI channel and OSC */play/stop[1]* address are assigned to **play/stop[1]** Pulsaret parameter, if **sync** is enabled Pulsaret also is able to send MIDI/OSC data. It is possible change OSC path value entering a new one.

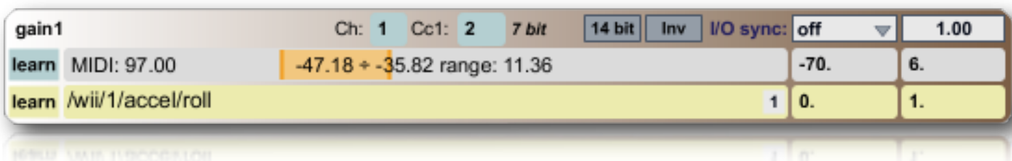


An OSC message can be a list, if you want read an element of the list, you can select an OSC input item from the list, (default first element). In the example, Nintendo WII device send the accelerometer values like a three list items, selecting 3 to read only third incoming.

I/O sync pop-up menù, set output mode for current widget, see sync for better understand.

The most right menù (toggle), set the **controller kind**:
- **toggle** switch from 1 and 0 every time MIDI/OSC I/O == 0;
- **button** send 1 every time MIDI/OSC input != 0;
- **incr** and **decr** increment or decrement (regardless of direction) by one step, every time MIDI/OSC input == 0. when values 127, or 0 are reached, the next value is wrapped;
- **incr/decr** it counts upward until it reaches 127, then counts down until it reaches 0, then up, then down, and so on;
- **controller** mode read raw data from MIDI/OSC input, raw data MIDI between 0 and 127.

Cc_scale extends Cc_raw functions, you can choose a certain range on which rescale the parameter, for example control **gain1**, which is associated, covers a range *-70 ÷ +6 (76 units)*. You know that MIDI has 7 bit resolution covering 128 steps range units (0 ÷127).



In the example, the Cc 2 is scaled on a part of the entire parameter range (11.36) by the orange horizontal selection. You can also change min/max values for the parameter (box numbers -70. and 6.), current range will be reported.

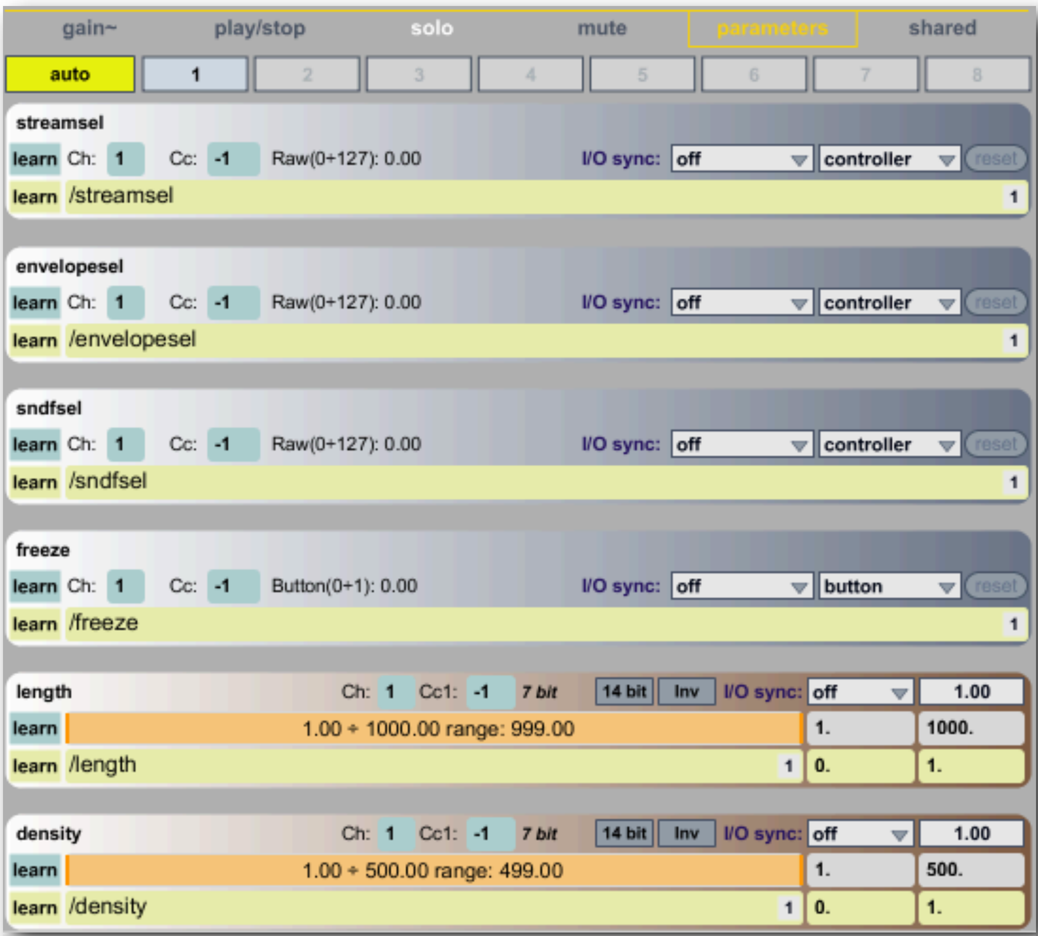
OSC assumes default values between 0 and 1. You can also change OSC input range. For example we control **gain1** through **/wii/1/accel/roll** OSC data incoming from OSCulator and Nintendo WII, we rescale the range for controlling only a middle part of parameter.

Pulsaret allow 14 bit precise MIDI input controller. This feature is achieved by union of two controls change on the gap by 32. For instance if midi learn finds Cc1: 7, the second Cc2 it will be 39 (i.e. 7 + 32 = 39). In other words, Cc1 is the most-significant byte controller number (1-127), while Cc2 is the least-significant byte controller number (1-127). Activating 14 bit, the Cc2 number it will be shown.
N.B. when you send 14 bit precise controller, Cc1 it should not be greater than to 95, this is because the Cc2 it should be not greater than to 127, i.e. 95 + 32 = 127.

Inv, invert the I/O range for the widget.
Most right menù, **I/O sync** (off) set the output mode for current widget, see sync for better understand and numbox (1.00) on top right set the exponential base value (default 1. = linear). The number is converted according to the following expression: ***y = b e-a log c ex log c.***

N.B. sets Cc to -1 for disabling controller.

- stream select



Having Pulsaret eight granular streams, we have an huge number of parameters to be controlled. Therefore it would be difficult to manage all parameters independently. An simple idea could be to assign each stream on own MIDI channel, for example stream1 -> MIDI channel 1, stream 2 -> MIDI channel 2 and so on... but each parameters of the eighth streams should have the own MIDI/OSC widget. To avoid this, Pulsaret can

be controlled following a layout strategy. MIDI Cc or OSC messages controlling only **selected granular stream**
(i.e. stream GUI in foreground).
The **streamsel** widget (first on the top), is crucial because it allows to bring the stream in foreground to be controlled via MIDI/OSC.

MIDI/OSC I/O works only on **current selected granular stream** (foreground). You need bring in **foreground** a granular streams, in order to controlling via MIDI/OSC their parameters.

Disabling **auto mode** (manual top left toggle), a menù tab will be shown allow you to chose a granular stream. Thus when you change granular stream (from main GUI), MIDI/OSC controls always the same stream. When **auto is enabled,** the granular streams in foreground(selected from GUI tab), will be controlled via MIDI/OSC.

- shared

Through this section you can controlling some shared parameters like: **windowing gauss/curve deformation, snapshots recall, snapshots time transition and HV_pad X/Y.**

Like for the **parameters** section we need selects a granular stream where send MIDI/OSC data, also this Pulsaret parameters follows same *modus operandi.* For example every **snapshots** module: **main**, **streams** (8) and **gsnaps,** (total 10) can be controlled by **main-stream-gsnap-switch** widget.
Idem for **HV_ Pads**.

The first three widgets are assigned to **Windowing** shapes: **gauss** and **curve**, see Windowing module for more details, they refers to *gauss* and *curve* shapes deformations.

The central switch tab or **main-stream-gsnap-switch** widget**,** pass the control to relative section.
When **stream** is selected, you will see a number according to selected stream.
Like **streamsel**, you can controlling **main-stream-gsnap-switch** via MIDI/OSC.

• pick-up

When **sync** is disabled, the **pickup** feature, (Ableton[1] Live inspired) is automatically enabled.
If the incoming MIDI/OSC value is different from Pulsaret GUI parameter value, pickup system wait that MIDI/OSC value reaches the widget. On the status bar you will see widget and incoming MIDI/OSC values until one of them does not match.

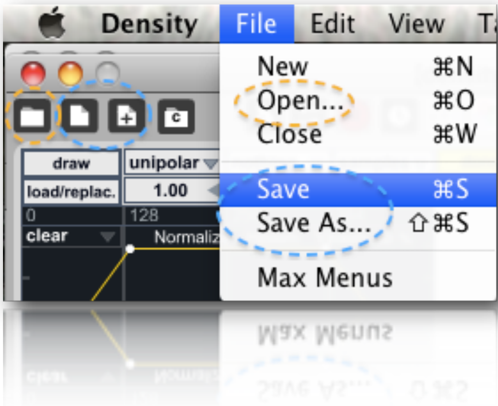
Enabling **sync** mode, will disable pickup system.

N.B.
In order does not disable **pickup** system, you should disable **sync** out when you have not a motorized MIDI controller, (like Behringer BCF2000).

[1] *Ableton Live is a loop-based software music sequencer and DAW for Mac OS and Windows by Ableton. The latest major release of Live, Version 8, was released in April 2009. In contrast to many other software sequencers, Live is designed to be an instrument for live performances as well as a tool for composing and arranging. It is also used for mixing of tracks by DJs, as it offers a suite of controls for beatmatching, crossfading, and other effects used by turntablists, and was one of the first music applications to automatically beat match songs. It does not support traditional musical notation. (Wikipedia)*

Overview

• save/load project



Project management is maybe, the most important Pulsaret feature. The idea is to capture and save all Pulsaret settings like: snapshots presets, MIDI/OSC mapping, setup configurations, sound files and sound files informations etc... in a single project i.e. **a folder** containing everything.

Open... allow you to browse filesystem to chose the project folder, N.B. you need select entire folder. You cannot select subfolders or any files contained in the project folder but from filesystem browser you need simply select the project folder.
N.B. Alternatively you can drag and drop project folder on the main mixer.

Save/Save As... create new project if not saved yet or save current project with name.

When you save a project without consolidate audio files, you will not able to open project correctly on the others computers. This because stored sampled files paths, refers to local path. Thus also if you change sampled files path on your machine, you will not able read correctly project. Alternatively **consolidate** feature, follow explained, solve the problem.

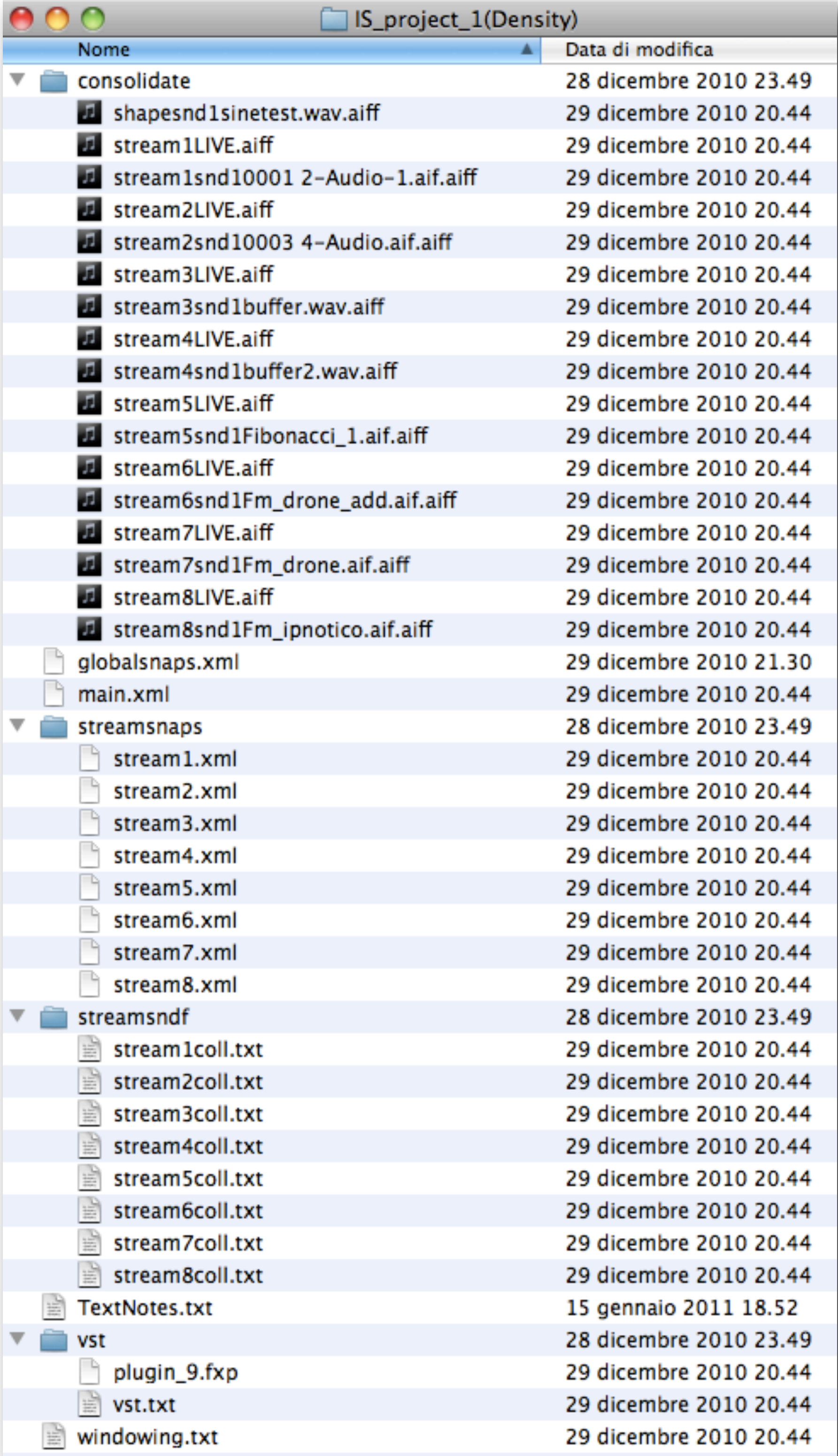
At each files exported is attached a prefix **ID,** the ID is ***shape*** for Windowing module and ***stramN*** for granular streams, where ***N*** is the slot number (12). All exported sampled files are in AIFF format, so **.aiff** extension will be attached eventually.
e.g
stream3snd2*filname*.mp3.aiff for streams or
shapesnd12*filname*.wav.aiff for windowing module

Consolidate, if enabled when you **Save or Save AS...** project, all sampled sound files loaded on the **streams slots** (12 available), **LIVE buffer** contents and/or **Windowing sampled shape slots** (12 available), they will be exported inside project folder. e.g. if enabled the files will be exported on the *consolidate folder* inside project folder, if disabled the files will be loaded from filesystem path.
Useful If you intend open projects everywhere.

N.B.

• **You cannot use \ or / symbol in the project name folder;**
• When you **Save** newly the project with **consolidate** flag off, the originals sampled files path will be restored but files stored in consolidate folder they will not deleted.

- project tree



In **main.xml** are stored all **general configuration** and the **snapshots presets** for **main** section;
In **globalsnaps.xml** are stored the **gsnap** global presets;

stream*N*.xml contains the streams snapshots presets, where ***N*** is the granular stream number;
stream*N*.txt contains granular streams data collector like path of sound files loaded, general information such: zoom, normalize, length selection, file sample length, sampling rate, time duration etc...

windowing.txt is collector file for the **Windowing** module: path of sound files loaded, general information such: zoom, normalize, length selection, file sample length, sampling rate, time duration etc...
TextNotes.txt is where Pulsaret save your note.In fact Pulsaret provide a text editor where, for instance you can write “the live set performance score” or what you want. When you save project also notes are included;

vst folder contains .fxp bank files for each allocated vst slot. In the example only the ninth (MASTER) slot, hosts a VST plugin. Instead **vst.txt** contains general information about loaded plugins: *name* and *path*;

consolidate folder contains all consolidated audio files, exported when you save (if consolidate is enabled) your project. See above for more details;

- status bar

Pulsaret status bar is subdivided in three parts, from left (frame blu) we have **main status bar** where are displayed general DSP infos like *sampling rate*, *I/O vector size*, *signal vector size, cpu utilization.*
In order to show *cpu utilization* value you need toggle tiny orange button.

N.B. You can change from status bar only Sampling Rate, see [DSP Settings] for explanations.
The current sampling rate will be saved on the Pulsaret project, this because if you have consolidated sound files in the project, it’s important load the project with own SR. i.e. The project Sampling Rate and exported consolidated files must have a coherent sampling rate.

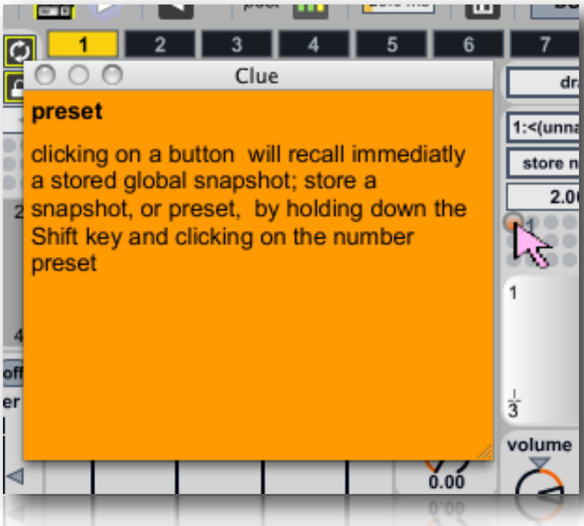
On the right (purple frame) we have the *pick-up display.* See pick-up for details
Two led green circled, displays the global transport beat and MIDI/OSC activity. You can also open their window by clicking.

On the right part of above image, yellow framed, we have granular streams information, these values refer to current selected granular stream.

Mainly, Granular stream status bar, *grain overlaps factor* (on the left) and *WavePad sampled file informations* like: current *file name*, current *file length in sample*, current file *sampling rate*, current *file length in milliseconds.* In order to enable *grains overlap report*, you need enable tiny orange button.

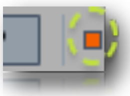
- clue window

To avoid annoying hint messages when you are working on the GUI, Pulsaret sends all hints in the **clue window**. In the clue window you can read some informations about Pulsaret GUI widget, when the mouse is over it.



The clue window CMD + I (Mac) CTRL + I (Win) is a floating window that displays documentation when you bring mouse pointer on a widget. You can use it to obtain reference information for objects then close window during performance.

- float window



Have you noticed almost all Pulsaret windows contains in a corner (generally top right) a tiny orange button? toggle it to set float/no float window. When toggle float, current window will be always in front. This feature is saved on project like general setup.

- history

ver 1.0 (november 2010) official release built with Max/MSP version 5.1.5

- eight granular streams simultaneously
- envelope/windowing menage up to 16 pre-generated shape (prototypes)
- envelope/windowing loading and menage up to 12 sound-files (aiff,wav,mp3)
- dynamic envelope buffer load/save, normalize, trim, resize length etc...
- main mixer 8 channels + 1 master, solo/mute and VST slots, Master cascade~ filter
- multichannel I/O mapping
- quick-record export master channel and/or multichannel file streams, progressive file autoname, select directory and re-sampling/quantize out file
- snapshots (presets) memory: up to 100 for each stream and 10 for the main mixer and global snapshots
- simultaneously (streams and main) transition (interpolation) between snapshots (in a given time)
- clients menage: include/exclude widgets from transitions
- four draw table for transitions curves
- micro-pad interpolating between four snapshots
- HV_pad (i.e. hyper vectorial pad), 9 snapshots pad (4 pad near), and auto-explorer (spiral, dunk, reflects) engine
- MIDI/OSC input mapping: learn/manual, rescale range and exponential curve
- MIDI/OSC output sync: enable/disable, continuously or mouse up send
- fully managing the project (as a folder), save/save as and load
- drag and drop on windowing wave-pads (audio files or folder) to fill menù soundfiles or project folder in main windows to load project
- stereo panning rotation: pan, jitter, lfo (shape/hz)
- panning time, note based sync.keyboard frequency select (streams)
- windowing sound files consolidate (copy buffers in the prj folder)
- every stream can deforming own windowing envelope (attack/decay)
- global presets, menage all Pulsaret widgets together
- snapshots list, consecutive rename
- global transport time
- snapshots sequencer rhythms improviser unit
- windowing deformation shape MIDI/OSC
- clue windows, report widgets information under the mouse
- matrix parameter linkage, rescale and LFO modulation range in Grainlet Synthesis
- Pulsaret length/cps and viceversa dependency
- cascaded series of biquad filters for each stream
- simply signal-oscilloscope, that allows you to monitor the visual progression of master waveform

ver 2.0.0 (april 2011)

- new GUI lock
- new message box does not stop audio
- TOOLS BAR quick functions access
- HV_pads 2D nodes parameters interpolation
- HV_pads add jitter exploration mode
- HV_pads "reflects", "jitter" and "drunk", now you can interact with the pad. The values of mouse and direction, will be the new coordinates of scanning
- HV_pads on/off monitor in main window
- MIDI/OSC 14 bit precise Control Change Midi I/O

- MIDI removed pich bend in favour of 14 bit MIDI I/O
- MIDI/OSC new midi monitor, parse 7 and 14 bit control change and show detailed messages
- MIDI/OSC OSC learn function
- MIDI/OSC you can append a prefix ID on the OSC Output adress
- MIDI/OSC OSC I/O send and receive automatically in the same adress
- MIDI/OSC improved rescale range (old project are't compatible, sorry)
- MIDI/OSC MidiRaw send Midi/Osc data out for sync
- MIDI/OSC MidiToggle integrated in MidiRaw OK
- MIDI/OSC new MIDI/OSC sync mode: mouse down, up, downup, continuous
- MIDI/OSC add MidiRaw input data mapping incr, decr, incr/decr modes
- MIDI/OSC MIDI CC = -1 (default) disable all MIDI I/O widgets
- MIDI/OSC toggle switch now show correctly own state
- MIDI/OSC you can chose an item by a list of four elements like OSC input
- MIDI/OSC inverse mapping for MIDI/OSC I/O
- MIDI/OSC OSC void string bug fixed
- GSNAPS defaults all turnoff DSP
- NOTES add you can save your notes in the project
- STREAMS PARAMETERS moltiply/divide (/4 /2 x2 x3 x4) feature
- GUI faders relative/absolute mousing mode switch
- panning now update correctly value when you select manual
- WAVEPAD shortcuts (F1, F2, F3, F4) enabled
- LIVE/Sample button switch
- snap-seq add global transport trigger link enable/disable
- MIXER native FILTER 24 band equalizer channels and Master
- MIXER native reverb channels and Master
- MIXER native compressor fchannels and Master
- MIXER rebuilded solo/mute
- MIXER add solo/mute fade in/out
- FILTER add filters types select
- FILTER add cutoff, gain and Q/S fine controller
- FILTER add dry/wet
- FILTER dynamic filter allocation, until 24
- FILTER dry/wet controller
- FILTER auto generate filter bands: Harmonic, Geometric, Scalar and Fibonacci
- FILTER frequency zoom in/out
- WINDOWING/WAVEPAD unsupported files does't overwrite path bug fixed
- Record Master/Multichannels save file/folder patch in project
- Record Master/Multichannels new GUI lock
- New detailed user manual online
- SNAP-SEQ step jump bug fixed
- SNAP-SEQ add portamento transition
- SNAP-SEQ some gui adjustments
- MATRIX improved rescale range (old project are't compatible, sorry)
- OSCILLOSCOPE stereo
-
- WINDOWING additive jump hrm amp select, fixed
- HV_Pad optimized auto scanning (drunk)
- HV_Pad save on the project, the last pointer position
- clue windows show now the widgets name correctly
- HV_Pad drunk/spiral/reflect/jitter cpu improved performance
- MIDI/OSC correct pan range rescale

- FILTER improved and refined
- FILTER enabled for transitions
- GSNAPS micropad became 2D nodes
- GSNAPS speedlimit transition update
- new micropad 2D-nodes

• acknowledgments

I would like to thanks Eugenio Giordani and Nicola Casetta, a special thanks to: Renato Alberti, Felix Petrescu and Pasquale Ascione.

Pulsaret.m4l Max externals and abstractions:

yafr2, plate reverb, in the style of Griesinger. Randy Jones rej@2uptech.com

Pulsaret was created with **Max5**
http://www.cycling74.com

The Author:
Alessandro Petrolati
apeSoft

ape@kagi.com
www.densitygs.com
www.alessandro-petrolati.it