



Pulsaret

Prototypes Granular Synthesis
{glisson, grainlet, trainlet,
pulsar}

Alessandro Petrolati
apeSoft

www.densitygs.com

user manual v.1.0.5

Starting

- _system requirement
- _installation
- _demo limitation
- _purchase a license and authorizes
- _background
- _GUI layout
- _DSP settings

Main mixer

- _faders
- _I/O mapping
- _on/off stream triggering
- _VST slots
- _master-rec
- _multichannels-rec

Granular streams

- _params explanations
- _general settings
- _envelope select
- _wave select

Windowing

- _general
- _select shape
- _draw
- _additive
- _buffer manage
- _wave pad
- _wave pad display
- _resize buffer

Snapshots

- _store/recall
- _micro pad
- _transitions
- _manage
- _exclude/include a client from the transition
- _draw a transition curve
- _snapshots sequencer improviser unit

Snapshots sequencer improviser unit

- _beats cycle
- _time intervals
- _step sequencer
- _output
- _transport

Hyper Vectorial Pad

- _general explanation

- _chase pad and exploration mode
- _X/Y time
- _settings
- _status bar

Matrix

- _general explanation
- _parameter linkage
- _pulsaret linkage
- _parameters rescale
- _LFO modulation

Filter

- _general explanation

MIDI/OSC I/O

- _settings
- _raw, scale, trigger
- _streams controls switch
- _common MIDI params
- _sync/pick-up

General

- _settings
- _fast icons
- _oscilloscope
- _save/load project
- _clue window
- _history
- _acknowledgments

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 to convert a media file (including MP3 files) into the sample memory of a wavePad and requires that QuickTime be installed on your system. If you are using Pulsaret on Windows, we recommend that you install QuickTime and choose a complete install of all optional components.

_installation

Macintosh

- Download Pulsaret.dmg
- Double click to open the disk image

Windows

- Download Pulsaret_setup.msi
- Double click to install

_copyright

This program is copyright shareware, 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

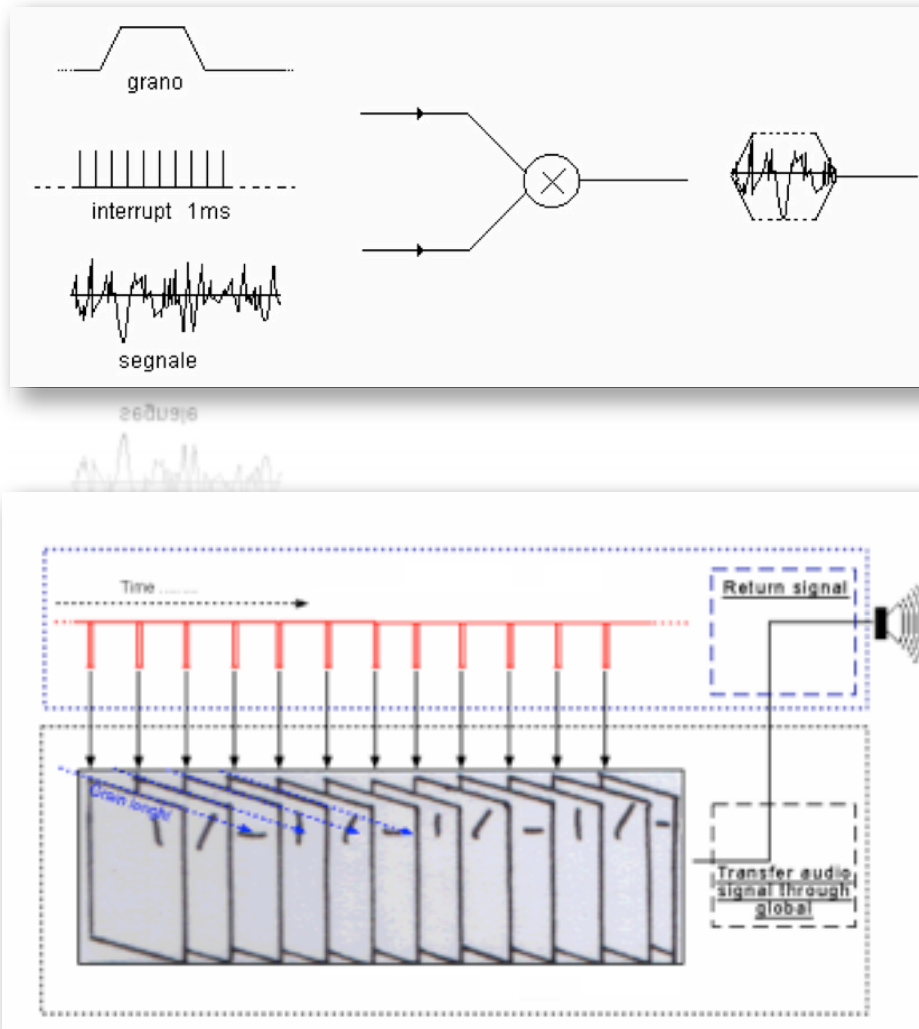
Demo version it is fully functional but will quit after 10 minutes and every minute makes a noise, buying a license you will disable this hateful behavior.

_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 window and click on **Authorize**, a dialog window will open: enter activation received by e-mail (**Thanks for your purchase**) and email used for kagi transaction then, click on **Authorize**.

_background

The first Pulsaret (DensityGSC) release, dates back to 2001, was developed for [Csound](#) language, to follow the GSC release was based on GSC4 of [Eugenio Giordani](#). GSC4 (Granular Synthesis for Csound) was the first patch for granular synthesis on Csound to implement the model proposed by [Barry Truax](#). The current Pulsaret implementation, follow a dynamic generation of the grains, rather than a fixed number of "voices" (static). The maximum density achievable, depends only on the actual CPU power.

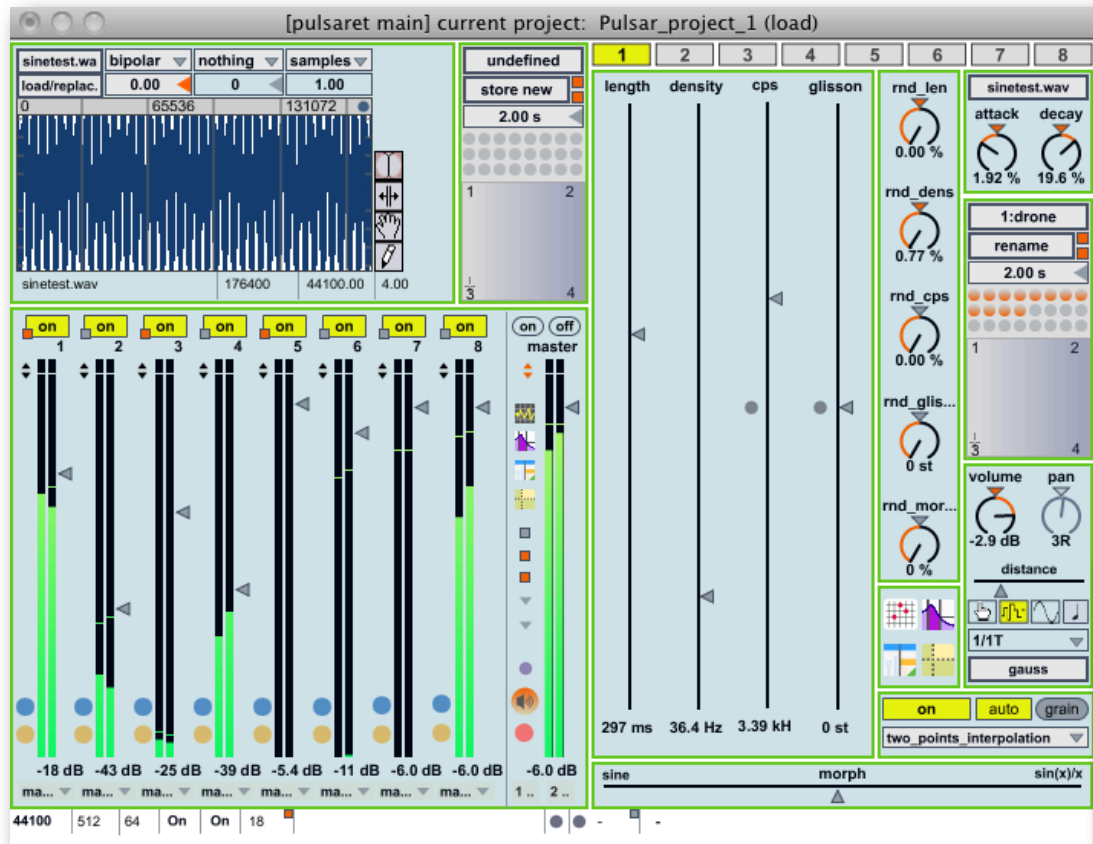


The old DensityGSC for Windows is available free: www.alessandro-petrolati.it/densitygsc.html it work on Windows XP but not Vista, it seems magically resurrected with Windows7. It's a discontinued product. The current new version, called Pulsaret, is completely re-written using [Max/Msp 5](#), runs under Macintosh and Windows, more stable more flexible and more efficient, improved audio quality, restyling look etc...

_GUI layout

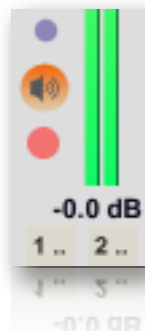
All the main GUI (Graphins User Interface) functions are contained within a single window. Almost all commands are accessible via the pop-up menù, at the top left we have "envelope menage" in the part below the "main mixer and to the right we have the "granular streams" (eight). Some menù items open windows as: HV_pad, MIDI I/O, master/multichannels Record and About. The selection of streams is done through the numbers (tab) above them (keyshort: shift+1,2...). Carrying the mouse over the widgets, will receive a hint, Pulsaret becomes so easy to learn, and is intended for live use, the main functions will soon be assimilated and memorized.

At the bottom of the main window, there the status bar, where displays infos about the DSP settings (in left) and specific granular stream informations.

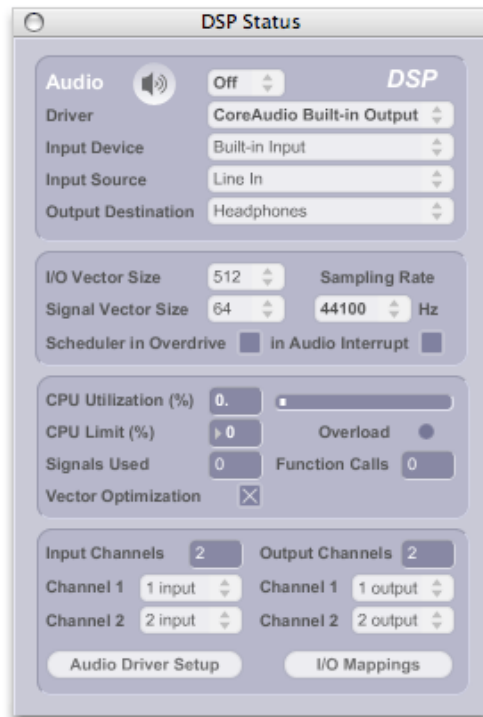


_DSP settings

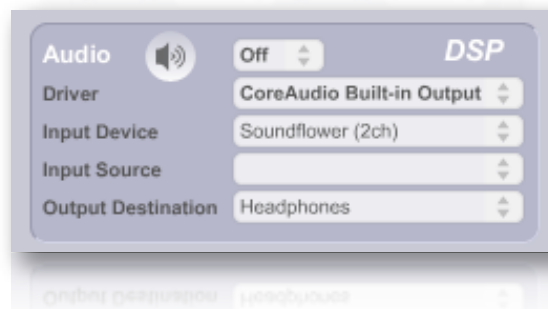
All global audio parameters are displayed in the DSP Status window. You can open **DSP** settings from application menù View/DSP (Cmd + D on OSX; Ctrl + D on Win) or clicking the violet button on the master channel.



The DSP Status window is arranged as a group of menüs and checkboxes that set all of the parameters of the audio input and output, the DSP Status window serves as a monitor for your current audio settings as



well. At the very top of the DSP Status window is a pop-up menu for turning the audio on and off and a set of pop-up menus that let you select an audio driver and configure its input source and output destination.



The second pop-up menu 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 menu.
- 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 menu 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.

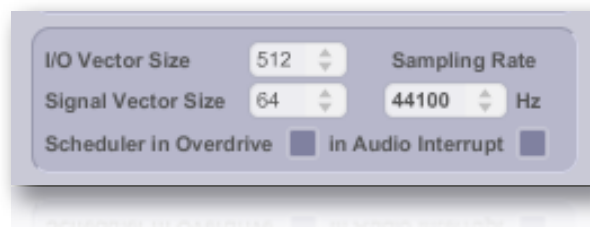
The next two pop-up menus are active only when using the Core Audio driver on Macintosh or ASIO drivers. When the Core Audio driver or either the MME or DirectSound drivers on Windows are selected, the pop-up menus allow you to change the audio input source. These settings can also be changed using the 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 menus while Pulsaret is running.

When ASIO is in use, the pop-up menus 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.

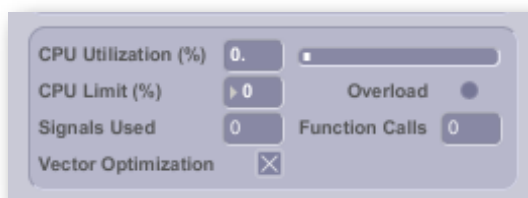
You can set the audio sampling rate with the Sampling Rate pop-up menu. For full-range audio, the recommended sampling rate is 44.1 kHz. Using a lower rate will reduce the number of samples that Pulsaret has to calculate, thus lightening your computer's burden, but it will also reduce the frequency range.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

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 menu. You can also enable and disable Overdrive using the Options menu. Overdrive generally improves timing accuracy, but there may be exceptions, and some third-party software may not work 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 Utilization field displays a rough estimate of how much of your computer's CPU is being allocated for crunching audio in Pulsaret.

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.

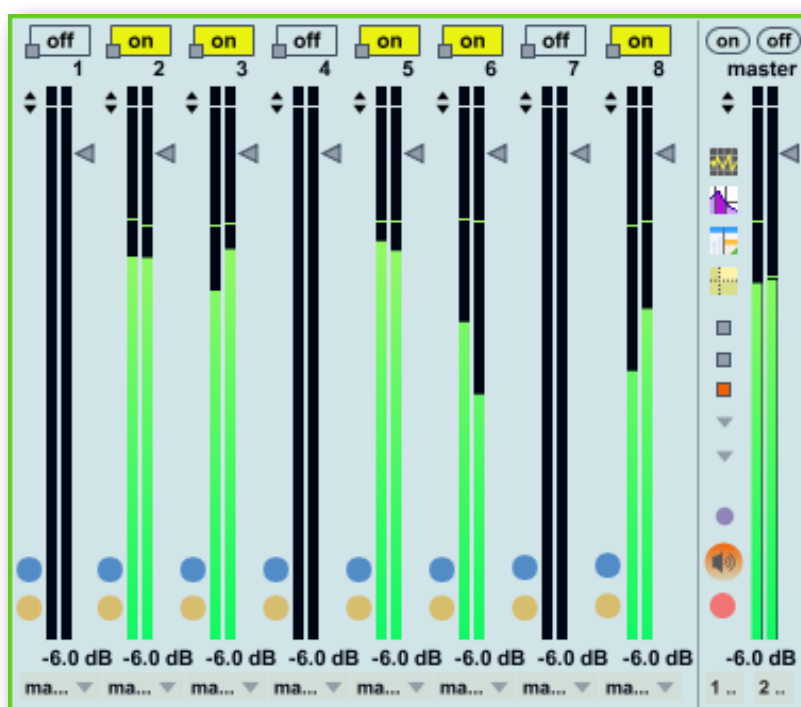
Vector Optimization only applies to PowerPC computers. Vector optimization allows four samples to be processed within the space of a single instruction. However, not all audio signal processing algorithms can be optimized in this way (for example, recursive filter algorithms are substantially immune 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.

Main mixer

_faders

The mixer has a classic look, each granular stream have a channel (stereo fader) on the mixer. On the far right is the master control, all channels are summed and scaled from the master, as a normal mixer.

The solo/mute (blue, yellow), behave classically, small arrows in the upper left of each slider are the VST plug-ins slots.



The buttons (on/off) above the channels, turn on/off its granular stream, on/off buttons above the master channel, turn on/off all granular streams.

N.B. When the DSP is off they will be disabled and ignore mouse clicks.

The record red button (under DSP) on the master channel, opens master-rec and multichannels-rec windows, master/multichannels-rec will be explained later.



From the menu preferences, you can specify whether the faders display and output the signal level pre-fader or post-fader

I/O mapping

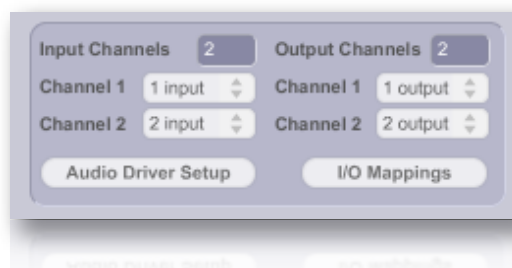
Pulsaret 1.0 supports multichannel outputs, each stream can be routed on MASTER fader or on a logical stereo channels. By default all eight streams are routed to the master channel, but you can choose a logical (they will be explained below), the first stream is mapped on 1/2 logical channels, the second on 3/4 etc...

From DSP_settings window, you can also I/O mapping.

The Master channel is able to write directly on the physical hardware channels (i.e DAC digital analogic converter). This is very important to understand because one behaves unlike from streams channels.

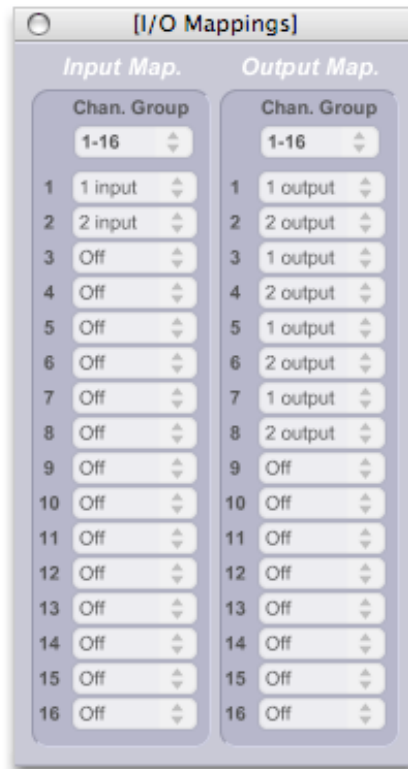
The same function is possible from I/O mapping, on the **logical channels 17/18**.

From **DSP_settings** you can map logical I/O channels.



The pop-up menu labeled Input Channel 1, Input Channel 2, Output Channel 1, Output Channel 2 allow you to map the first two logical channels (first granular stream) of I/O in Pulsaret (i.e. Pulsaret master output) to physical channels used by your audio driver. Different audio drivers give you different options, for example, the MME driver on Windows only supports two channels, so you will normally use the default options. To map additional logical channels, use the I/O Mappings window, which you can view by clicking the I/O Mappings button at the bottom of the DSP Status window.

To assign multiple logical channels to one physical channel of an output device, use the I/O Mapping window. Click on the I/O Mappings button at the bottom of the DSP Status window.



The configuration shows that logical channels 1, 3, 5, and 7 have been mapped to the left output channel of the current audio device, and logical channels 2, 4, 6, and 8 have been mapped to the right output channel of the current audio device. Therefore, four granular streams (from 1 to 4), are mapped on the physical 1/2 stereo output. If you have a multichannels soundcart, you can chose a different channel for each granular stream.

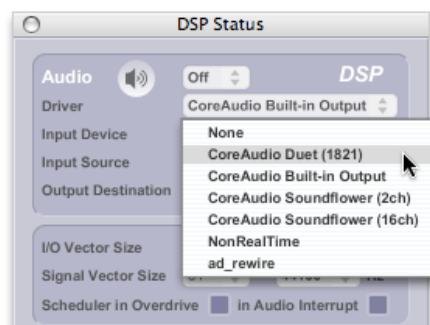
N.B.

I/O Mappings are saved for each audio driver.

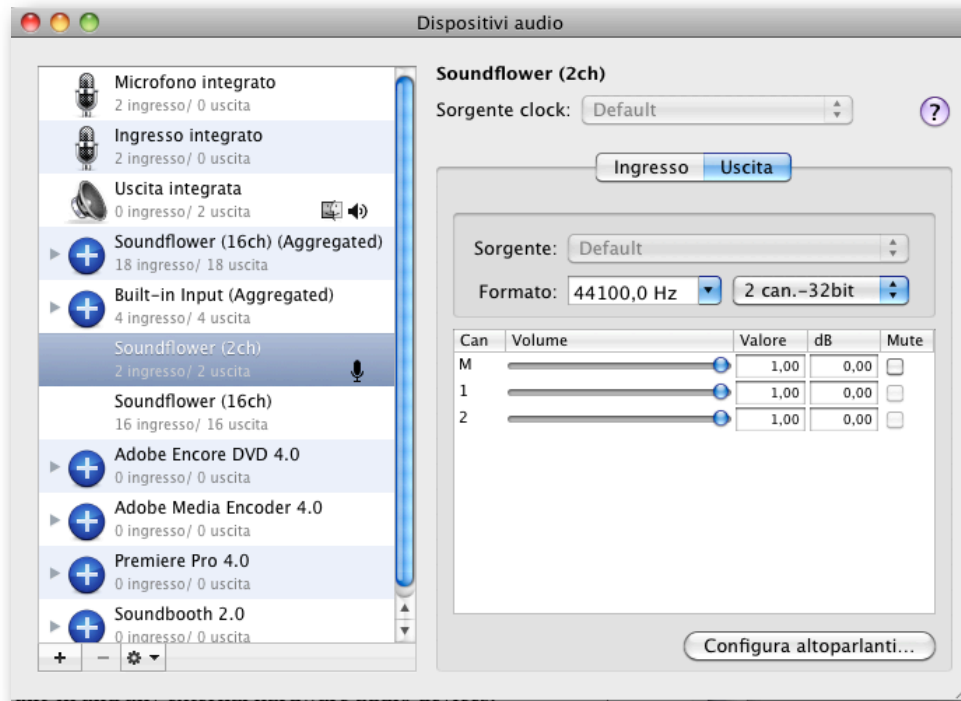
Using Core Audio

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.

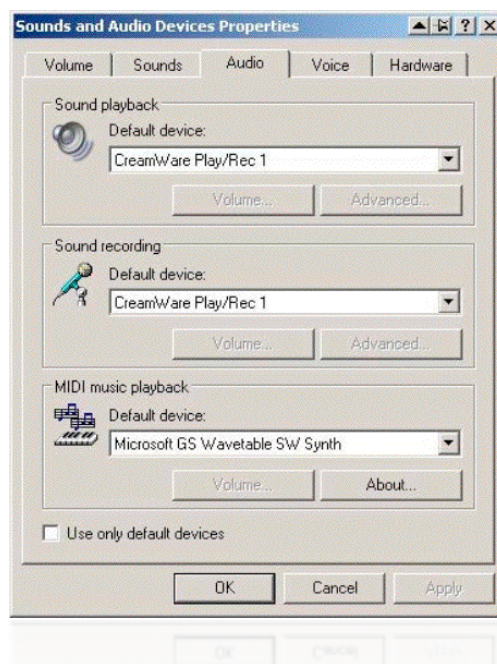
Using MME and DirectSound Drivers with Pulsaret on Windows

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.

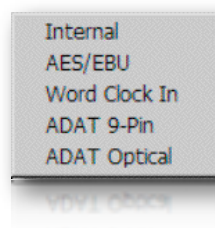
For example, after the Pulsaret ReWire Device is registered, Cubase will have a Pulsaret menu item in the Devices menu. When you choose it you will see a list of the audio outputs from Pulsaret. They will default to the off state. Click on any of the buttons to activate that channel. Once the channel 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.

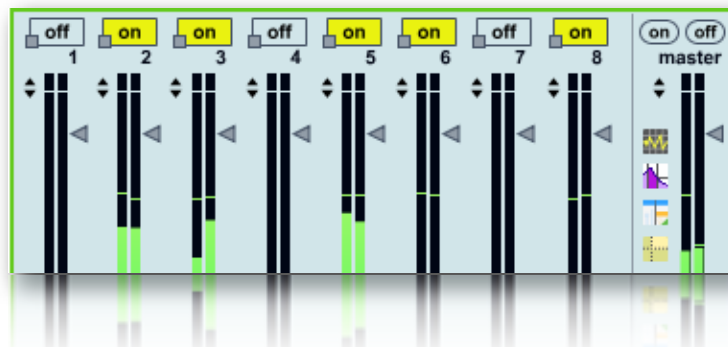
Many ASIO drivers have other settings you can edit in a separate window. Click the Open ASIO Control Panel button at the bottom of the DSP Status window to access these settings. If your interface card has a control panel in its ASIO driver, the documentation for the interface should cover its operation.

_on/off stream triggering

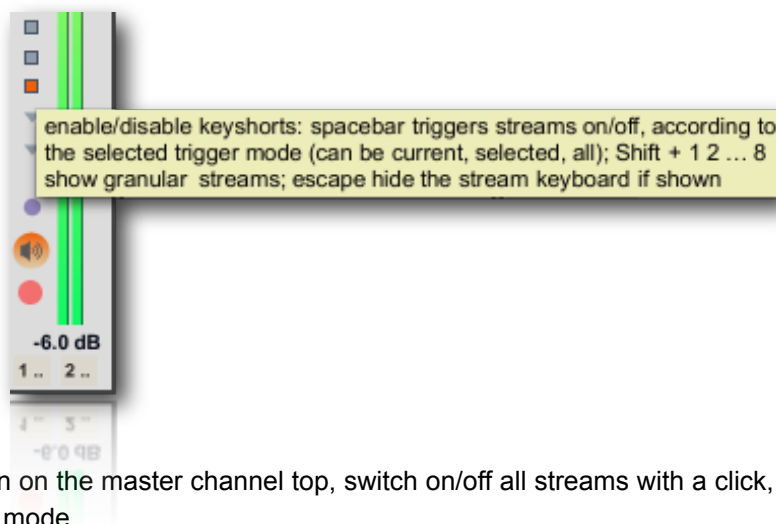
There are four ways to trigger (i.e. turn-on/off) streams,



- 1) clicking on/off button on the mixer streams channels or clicking on/off in the streams;
- 2) selecting from pop-up menù trigger-mode then using **spacebar on the keyboard** or MIDI CC (Control Change).
- 3) **current** mode, will trigger selected granular stream;
- 4) **selected** only work on the selected streams, infact each mixer on/off button contains a ocher little one. Checking it will be “arm” the streams for the on/off sync. In the example below, streams 2, 4, 6 and 8 will be turned on/off together:
- 5) **all**, trigger the 8 streams just pressing spacebar;

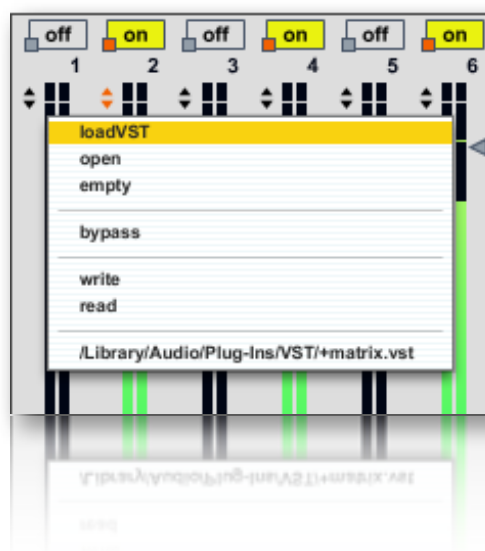


You can bypassing keyboard spacebar keyshort by preferences button (third from the top).



N.B. the on/off button on the master channel top, switch on/off all streams with a click, these work regardless of the type of trigger mode.

_VST slots



Load and manage VST plugin is easy too, **loadVST** open a browse file to choose a .vst file on Mac and .dll on Windows; **open** show the VST window; **empty** cleans the VST slot loading a default which means no plug-in; **bypass** bypassing effect; **write** open browse file for saving VST bank file; **read** is the opposite; the last voice is the path **name for the current VST loaded**. Clicking on it, you will **open** VST GUI (the same open). All settings of the VST loaded are exported inside of Pulsaret project. Some VST are **not fully compatible** and may not successfully export their content.

_master-rec

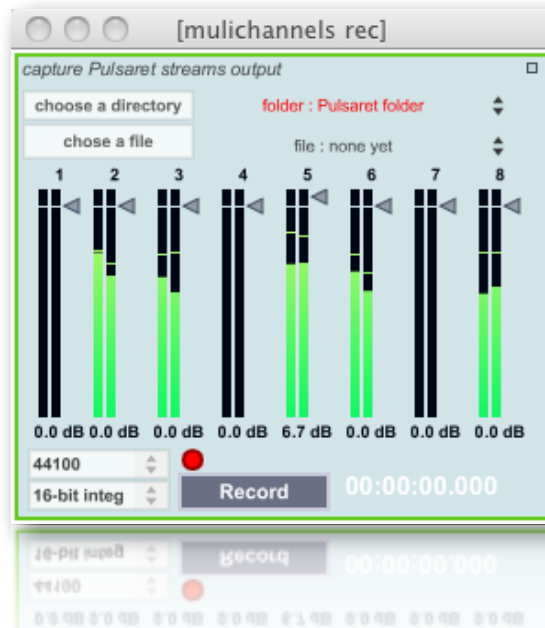
To enable master rec, you need clicking on the red button beside Record. You will arm your master channels, then formatting output file and see master record level and starting.



- 1) Horizontal fader rescale post master signal;
- 2) Set soundfile sampling rate and type: the default sampling rate is the current DSP sampling rate, the default type is 16 bits;
- 3) Adjust audio level: each recording channel can be modulated separately, and all channels will be modulated together by the Master gain slider;
- 4) Start recording: by default, the object is set to create a new audio file each time one 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 location where the file is recorded is the Pulsaret application folder. It is possible to change the location by clicking into the "Choose a directory button", and select a specific place;
- 5) Create a specific file into a specific place: If one wishes to set a different name and/or location for each new file, it is possible to use the "Open a 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;
- 6) Toggle between auto-record and specific one: once a file location has been defined, it is possible to toggle between the two "directory" and "file" modes, by selecting the "Current : ..." pop-up menu under the directory and file buttons. The mode which is currently active has a yellow color;
- 7) Toggle between float/no float window:

N.B. master-rec, receive the signal before being sent to a physical device, so even if the output disable (off), master-rec will continue to receive the signal.

_multichannels-rec



mulichannels-rec it work like master-rec, but it able to write a file multichannels (16) containing all 8 granular streams outputs (i.e. stereo channels (2) x 8 streams = 16 channels).

N.B. channels-rec, capture stream logical output, so even if the output disable (off), it will continue to receive the signal normally.

N.B. Naturally you can enable master-rec and multichannels rec at the same time.

N.B. Only SR, bit deep quantization and arm-recording are saved in the project.

Granular Streams

_params explanations

In this section we examine the granulation parameters, please refer to web for further details about granular synthesis.

You can interact with Pulsaret controls in the following ways:

- 1) Click and drag in the slider (or dial) to change the value;
- 2) Hold down the Command key (Macintosh) or the Control key (Windows) for more precise mouse control;
- 3) Click on the control name then up/down arrow to scroll the slider;
- 4) Click and drag on the control value, the box number below the parameter;
- 5) Click on the control value, then enter a value followed by the enter key;
- 6) With a MIDI CC (control change), see MIDI/OSC chapter in this manual;
- 7) With a OSCmessage, see MIDI/OSC chapter in this manual;

Sliders handle deterministic values, while the dials (knobs) random values. The combination (deterministic-random) produces a value that is passed to the granulator.

For example the length value is scaled with rnd_lengt has follows:

$$\text{length_result} = \text{length} + (\text{birnd}(\text{rnd_length} * \text{length}));$$

generally, the random range is expressed in %.

length slider = grain duration in milliseconds;

rnd_length dial = random range deviation, in %;

density slider = grains per second;

rnd_density dial = random range deviation, in % (as rnd_length);

cps slider = grain frequency in Hz;

keyboard button = (small button in the middle of pitch control) show a chromatic keyboard, for the chromatic pitch transposition. Middle C (60 MIDI note) is unchanged pitch;

rnd_cps dial = random range deviation, in % (12 semitone);

glisson slider = glissando semitones inside of the grain;

freeze button = (small button in the middle of glisson control) switch between the value 0 (freezes) and the last control value;

rnd_glisson dial = randomly moves semitones glisson (-/+ 12 semitones);

morph = morphing between two waveshapes (left=wave_1, right=wave_2, middle=mix 50%);

rnd_morph = morphing randomly between two waveshapes, in %;

volume slider = grains amplitude in decibel;

pan dial = random stereo grains distribution, expressed in degrees;

distance slider = stereo field width ;

autopanning = autopanning select mode: pan (manually), jittering (random), lfo (low frequency modulation)

panning rotate frequency = work only in lfo: set the rotation pan frequency in Hz, you can choose a shape among those available (see windowing section for more details).

_general settings

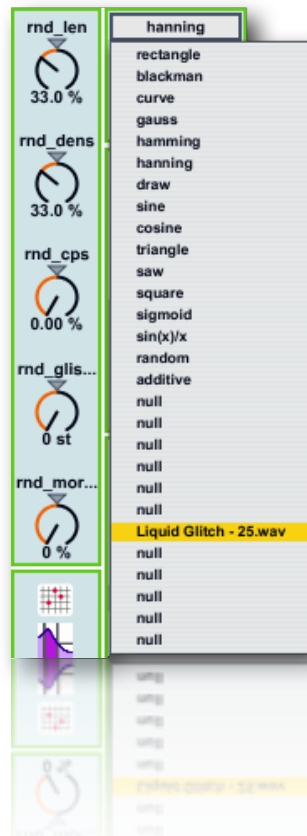
You can manage some settings for the eight stream:



- 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.

_envelope select

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 pop-up menùs.

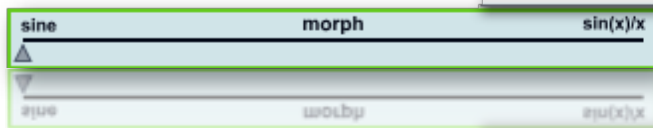


see “Windowing” for more explanation.

_waves 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

rectangle
blackman
curve
gauss
hamming
hanning
draw
sine
cosine
triangle
saw
square
sigmoid
sin(x)/x
random
additive
null
null
null
null
null
null
Liquid Glitch - 25.wav
null
null
null
null
null



ung
ung
ung
ung
ung
Liquid Glitch - 25.wav
ung
ung

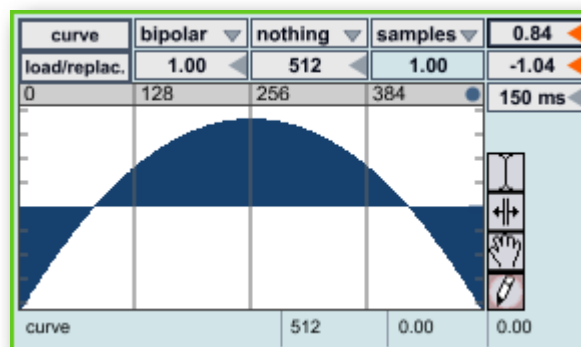
Windowing

_general

The windowing module, provide classics envelops and prototypes used for smoothing the grain amplitude. Changing or drawing a shape, you can get interesting effects (pseudo-convolution), in Pulsaret you can draw an envelope in real time, or load an audio file from disk (up to 12 audio files).

_select shape

There are 16 pre-generated prototypes, some of them have additional control parameters. When you select one of these windows: “curves, gauss or additive”, will be shown their controls.



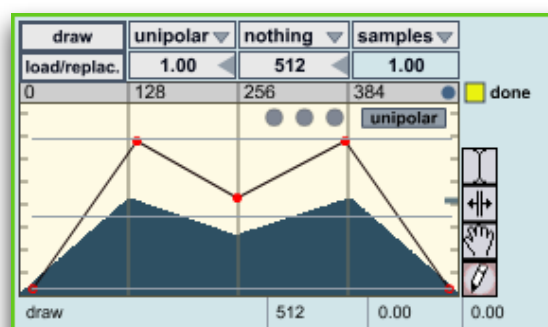
For example, “**curve**” show two box-number (most top right -0.01 and 2.00) to control the middle/border deformation. Gauss show a Gaussian standard deviation. Both have an important parameter, resolution update time in milliseconds. Since they work in real time, moving the deformations params we’ll do heavy operation in the buffer, it’s important limit the speed of operations.

“**additive**” will show four box-numbers: “additive” create harmonics waves. It works similarly to Csound Gen19. Each partial is automatically added to the previous, you can control the following parameters for each partial: “partial number” of the harmonic series; “partial amplitude”; “partial offset” normalized (da 0 a 1) and “partial phase” normalized (da 0 a 1).

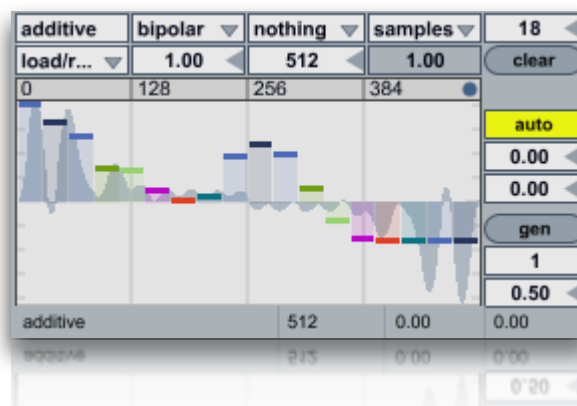
N.B. the waveform created by "additive" will not be saved in the project.

_draw

Very interesting is the envelope “draw” (default), when selected appear yellow button (draw). This allows you to superimpose a pad where you can track the envelope segments. You can add new break-points clicking on the 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 shape used like default grain envelope. You can change unipolar/bipolar view mode, although you need change too in the buffer from the menu, for a coherent view.

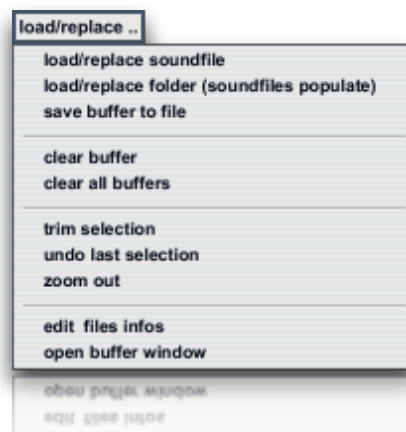


_additive



These subroutines generate composite waveforms made up of weighted sums of simple sinusoids. Each slider is the specific amplitude contribute partial. You can change harmonics number for the spectrum (top right **numbox 18**), **clear** all amplitude contributes of spectrum. Through the **auto** button you can enable amplitude autorescale, thus, the resulting shape it will be always normalized to one. The two belows numberbox, setups **phase** and **offset** for sinusoidal gen. Furthermore the gen button put a fine amplitude (0.5) on the specified partial num (1 belows).

_buffer menage



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 soundbuffers audiofiles (aiff, wave mp3 supported). Also you can drop on the wave pad a soundfiles 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 time display of file;

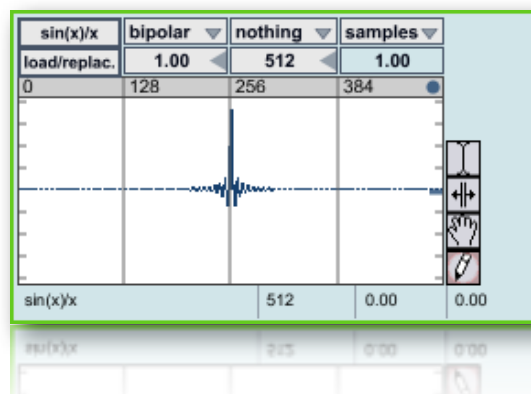
edit files infos = show a text-edit with information on loaded files;

open buffer window = opens the wavepad buffer window, or brings it to the front if it is already open. The window is resizable but not editable, **scrolling the mouse over time it switches to scrub mode**;

normalize (1.00) = scale the sample values in the buffer so that 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 than 0 are saved (loaded) in the project. Normalized value 0 it can have two meanings, original file amplitude if the params never it is modified or normalized to 0 (actually 0.001 amplitude, -60 db).

_wave pad

The WavePad is the windowing buffer used to generate waveshape synthesis, Pulsaret windowing can upload up to 12 audio files. You can load an audio file from the pop-up menu (load/replace), or simply drag & drop a sound file on the pad. There are supports WAV, Next/Sun, AIFF, MP3¹ file informations will be displayed below, loading sound files, the names will be showed in the popup menu, replacing the "null" string.



On the right side of the pad, you can choose the type of action: select, loop, move and draw. The tools **select** and **loop** allows you to scroll the buffer manually (scubbing).

loop has a dual function, compared to select, you can control the length grains, dragged up and down. **move** allows you to scroll the waveform inside the pad, also moving the mouse up/down for time zoom. scanning looped the selection (wraparound); **draw** allows you to draw waveform directly on the pad.

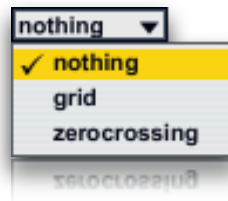


Moving the handle up/down, you can zooming in/out the signal amplitude. Note that this is not a destructive rescaling but only a visual effect, the time zoom out button, resets the entire time display of file.

_wave pad display

¹ Density uses QuickTime to convert a media file (including MP3 files) into the sample memory of a wavePad and requires that QuickTime be installed on your system. If you are using Density on Windows, we recommend that you install QuickTime and choose a complete install of all optional components.

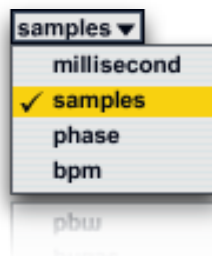
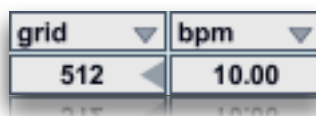
Snap-mode, sets wavepad selection range. Snap causes the start and end points of the selection to automatically move to specific points in the buffer, defined by the snap mode. possible arguments are none, grid, and zero crossing.



nothing = disables snap to allow free selection;

grid = Specifies that the selection start and end points should snap to the vertical grid lines, as set by the grid message. Since the spacing of the grid lines is affected by the current time measurement unit, snap to grid will be affected by these parameters as well;

zerocrossing = instead of snapping the selection to a uniform grid, this mode searches for zero-crossings of the buffer data. These are defined as the points where a positive sample follows a negative sample, or vice-versa. This can be useful to find loop and edit points;



The numbox (10.00) specifies that the selection start and end view points should snap to the vertical grid lines, as set by the grid message. since the spacing of the grid lines is affected by the current time measurement unit, and by the offset value (if an offset has been specified), snap to grid will be affected by these parameters as well.

i.e. Sets the unit of time measurement used by the display.

N.B. for instance if time unit **phase** is selected, the grid value it must be between 0 and 1.

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;

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;

bipolar switch wiew mode unipolar/bipolar;

In addition there are two items: "reset shape" and "reset all shapes" that they are explained alone.

`_resize buffer`

As for the first 16 prototypes, you can resize the buffer length (in samples), 512 is the default.

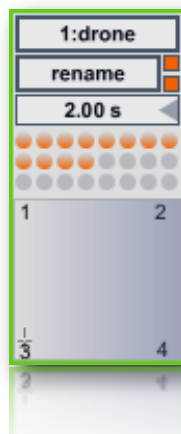
N.B. when you resize, it resets the default shape, all changes as (pencil draw or crop) will be lost but only "normalize" will be preserved. "resetshape" and "resetallshapes", refer to the prototypes (first 16), restores the default shape. "clear" e "clearall", clean the buffer by writing 0, they work on everyone, including sndN.

Snapshots

`_store/recall`

A snapshot is a photo of all the parameters of the graphical interface (GUI), each stream can store up to 100 snapshots. In Pulsaret there are nine modules "snapshots": one for the main and eight for the streams, can be called simultaneously and independently of all snapshots.

You can also make transitions between two or more snapshots in the different streams and main simultaneously or between all subscribed clients of the all Pulsaret parameters.

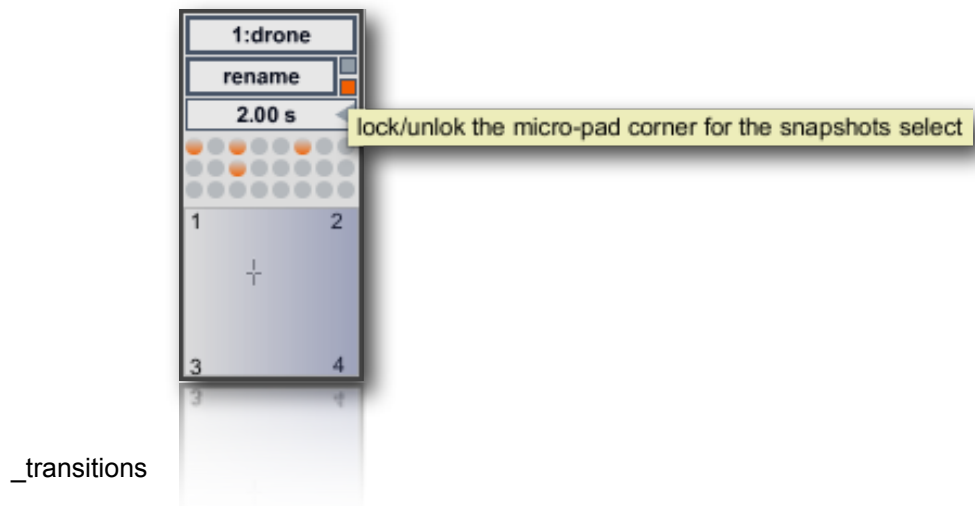


When you find a Interesting "sound", just shift + click on a button to record the preset (snapshot), recording the preset button turns orange. You can recall a snapshot, simply clicking on the buttons, all the widgets enabled to work with snapshots, will be restored with its value.

Alternatively, a snap can be stored from menù, just digit a snap number in the dialog box, see below for more explanations.

`_micro pad`

You can use the micro-pad to obtain intermediate values (interpolation). On the pad edges, you can set four snaps. To do this you must first unlock the snaps-select through the low square orange button, while the up botton enable/disable auto recall snapshots, see below for more details.

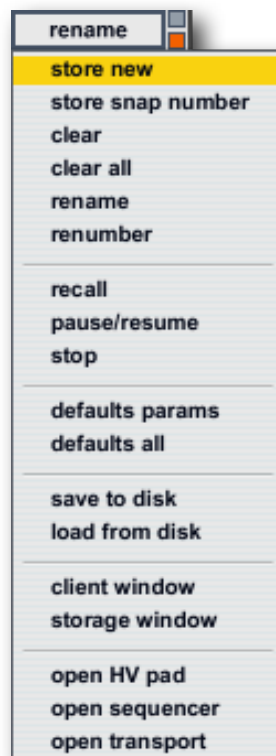


To recall a snapshot in a given time, you must use the menu recall, Is important to understand that the transition occurs from the **current parameters positions**, toward selected snapshot.

actual gui widgets positions >>> (toward) selected snapshot (in a given time)

You can change the transition-time (pop-up menu below snapshot select), transitions can occur simultaneously on different streams and/or on the main. If autorecall (first square button from right) is enabled, simply select from the menu a snap to start the transition.

_menage

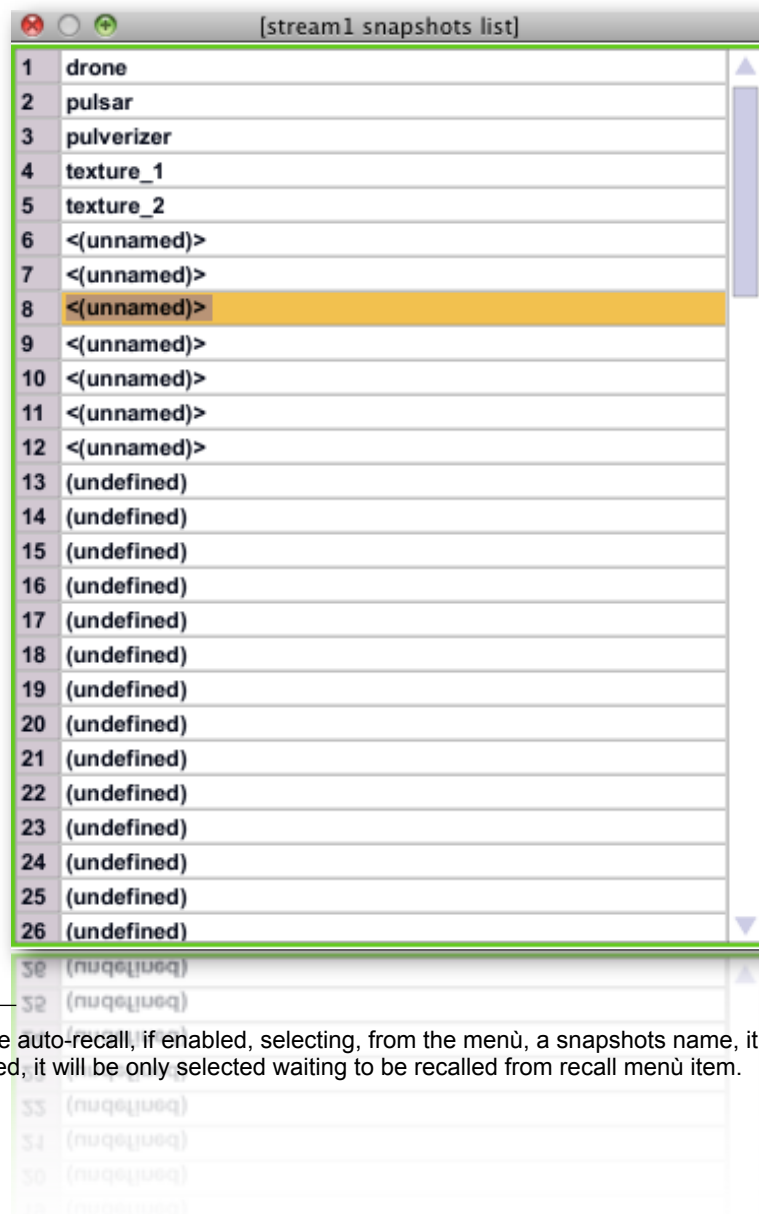


- store new snapshot using the next empty preset slot;
- store snap number following optionally you can digit a snapshot name in double quotes;
- clear selected snapshot;

- clear all snapshot;
- rename snapshots list;
- renumber stored snaps into consecutive, beginning with 1;
- recall² a snapshot in a given time;
- pause/resume transition toward selected snap;
- stop transition cancel current transition;
- default_params values, only for the transitions streams/main subscribed-widgets;
- defaults value for all streams/mainwidgets;
- save snap bank to disk;
- load snap bank from disk;
- default all gui widgets values;
- open menage client window;
- open data storage window;
- open Hyper Vectorial window (HV_Pad);
- open snapshots sequencer improviser (see sequencer below);
- open global transport time (see transport below);

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 open a window for fast renaming snapshots:



² you can enable/disable auto-recall, if enabled, selecting, from the menu, a snapshots name, it will be recalled on the transition time, if disabled, it will be only selected waiting to be recalled from recall menu item.

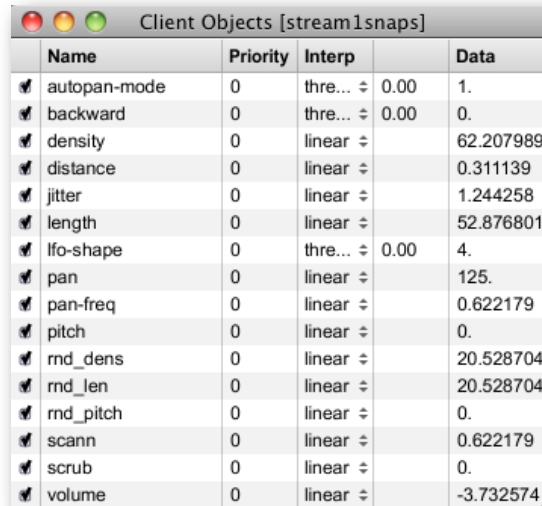
double clicks on the <(unnamed)> to edit text, then rename it.

Clicking/scrubbing on the left snapshots numbers, you achieve a fast snapshots recalling

N.B. Slot 99 is default/defaults_all values, slot 100 is the last GUI state when you save/load a project.

_exclude/include a client from the transition

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



Name	Priority	Interp	Data
✓ autopan-mode	0	thre... ⇅ 0.00	1.
✓ backward	0	thre... ⇅ 0.00	0.
✓ density	0	linear ⇅	62.207989
✓ distance	0	linear ⇅	0.311139
✓ jitter	0	linear ⇅	1.244258
✓ length	0	linear ⇅	52.876801
✓ lfo-shape	0	thre... ⇅ 0.00	4.
✓ pan	0	linear ⇅	125.
✓ pan-freq	0	linear ⇅	0.622179
✓ pitch	0	linear ⇅	0.
✓ rnd_dens	0	linear ⇅	20.528704
✓ rnd_len	0	linear ⇅	20.528704
✓ rnd_pitch	0	linear ⇅	0.
✓ scann	0	linear ⇅	0.622179
✓ scrub	0	linear ⇅	0.
✓ volume	0	linear ⇅	-3.732574

the client window (accessible from the "client-window" menu) shows the current subscriptions, priority, interpolation, enable state and data belonging to subscribed widgets.

a few types of interpolation are available:

off: no interpolation;

thresh, ithresh: threshold "interpolation", requires additional argument to set 'fade' thresh

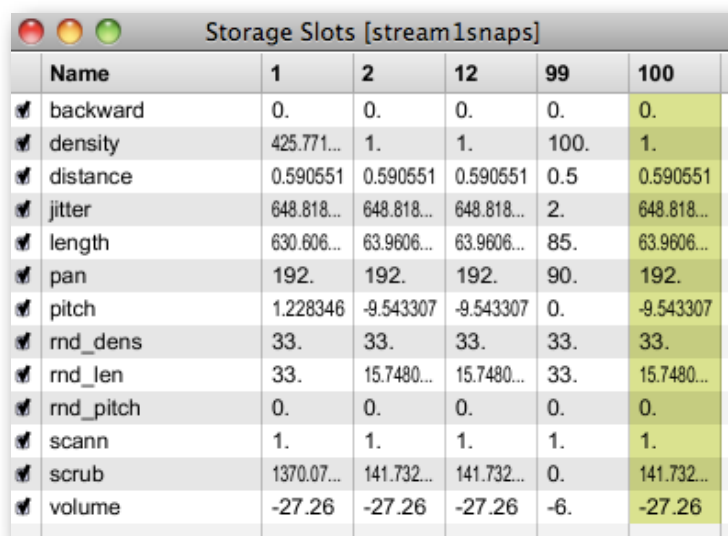
thresh: 'fade' < thresh = value a; fade >= thresh = value b

ithresh: 'fade' < thresh = value b; fade >= thresh = value a

pow: power curve interpolation, additional argument determines the exponent of the power curve

table: lookup table, additional (see below) argument specifies the name of a table to perform a lookup from.

Tables are assumed to have values from 0-100, representing 'fade' * 100. The table values are interpolated if 'fade' doesn't fall on a distinct table value.



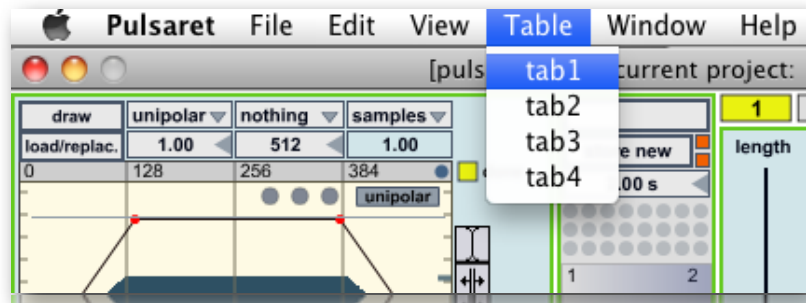
Name	1	2	12	99	100
✓ backward	0.	0.	0.	0.	0.
✓ density	425.771...	1.	1.	100.	1.
✓ distance	0.590551	0.590551	0.590551	0.5	0.590551
✓ jitter	648.818...	648.818...	648.818...	2.	648.818...
✓ length	630.606...	63.9606...	63.9606...	85.	63.9606...
✓ pan	192.	192.	192.	90.	192.
✓ pitch	1.228346	-9.543307	-9.543307	0.	-9.543307
✓ rnd_dens	33.	33.	33.	33.	33.
✓ rnd_len	33.	15.7480...	15.7480...	33.	15.7480...
✓ rnd_pitch	0.	0.	0.	0.	0.
✓ scann	1.	1.	1.	1.	1.
✓ scrub	1370.07...	141.732...	141.732...	0.	141.732...
✓ volume	-27.26	-27.26	-27.26	-6.	-27.26

The storagewindow displays any stored presets. Active sets are displayed in bold. Active, but changed, sets 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.

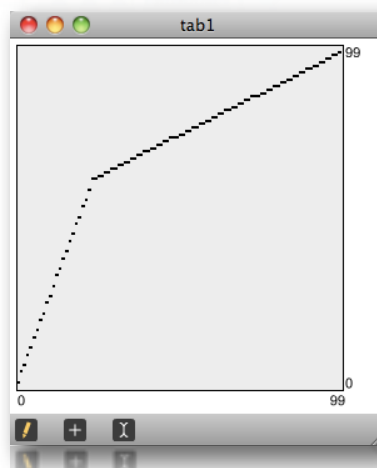
N.B. As the “granular streams” have their own snapshots/transitions, even on the main there are also, work equally.

_draw a transition curve

In Pulsaret 1.0 some functions are only available from the application menù, such functions, like tab/curve write, it's accessible from menù **Table**, we can define four functions, these can be used to interpolate a specific client widget for the transition as follows:



draw a shape on the pad for the transition between snapshots (interpolating)



open clients-objects window and select table from interp pop-up menù, digit the tab name (tab1, tab2, tab3 or tab4).

	Name	Priority	Interp		Data
<input checked="" type="checkbox"/>	backward	0	threshold ⇅ 0.50		0.
<input checked="" type="checkbox"/>	density	0	table ⇅ tab1		1.
<input checked="" type="checkbox"/>	distance	0	exponential c... ⇅ 2.00		0.590551
<input checked="" type="checkbox"/>	jitter	0	exponential c... ⇅ 0.50		648.818909
<input checked="" type="checkbox"/>	length	0	inverted thres... ⇅ 0.00		63.960632
<input type="checkbox"/>	pan	0	linear ⇅		192.
<input type="checkbox"/>	pitch	0	linear ⇅		-9.543307
<input checked="" type="checkbox"/>	rnd_dens	0	table ⇅ tab1		33.
<input checked="" type="checkbox"/>	rnd_len	0	table ⇅ tab1		15.748032
<input type="checkbox"/>	rnd_pitch	0	linear ⇅		0.
<input type="checkbox"/>	scann	0	linear ⇅		1.
<input checked="" type="checkbox"/>	scrub	0	table ⇅ tab2		141.732285
<input checked="" type="checkbox"/>	volume	0	linear ⇅		-27.26

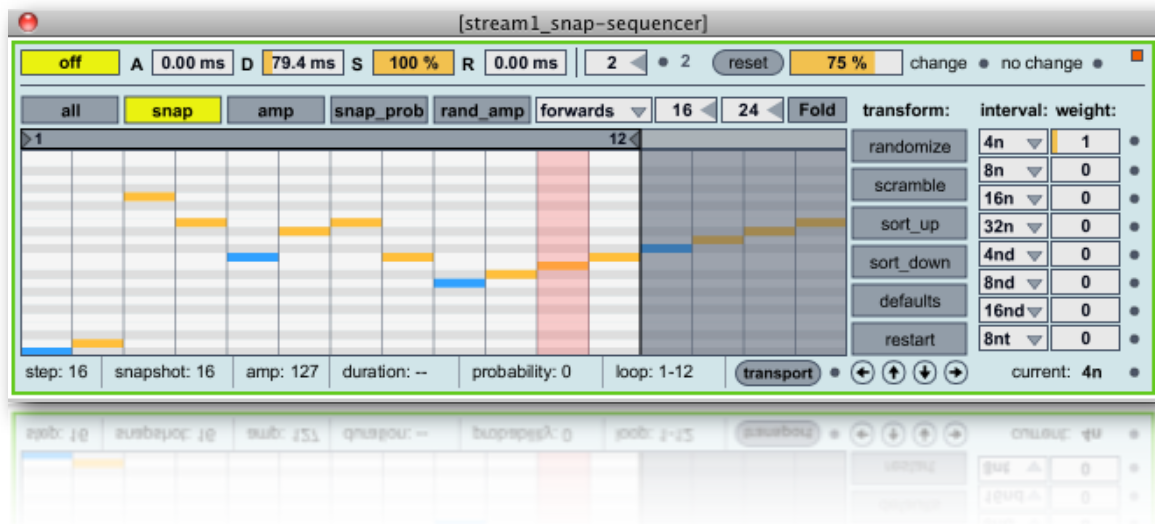
the example we used to map tab1: “density”, “rnd_density”, “rnd_length”, tab2 for “scrubb”; we unsubscribe from transitions: “pan”, “pitch”, “rnd_pitch” and “scann”; other mappings are possible:

- none, no interpolation;
- linear interpolation. Presets recalled will be interpolated using a standard linear algorithm;
- Threshold, takes argument (float), which sets the threshold. Snapshots 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;
- Inverted threshold, takes 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;

- exponential curve 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 an argument (tab1, tab2, tab3 or tab4), which specifies the name of a table to use for curve lookup. Presets recalled will recall data between the two specified presets, along the curve described in the table;

Snapshots sequencer improviser unit



In Pulsaret 1.0 is available a interesting snapshots rhythm generator and step sequencer. The aim of this device is to control the snapshot recall by improvising with different rhythmic pulses that trigger the snapshots contained in a step-sequencer.

`_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 subpatcher). 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 subpatcher). If the program decides to change the time interval, a trig is sent.

`_time intervals`

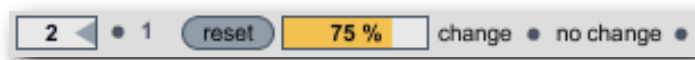
up to eight time intervals can be selected. The choice is performed with a random function, according to a weight probability for each possible interval - an interval with a weight of 2 will have twice as many chances to be selected by the random procedure that an interval that has a weight of 1.

`_step sequencer`

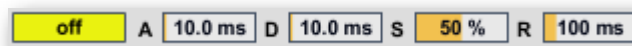
when a given time interval is chosen, it is used as a step rate for driving the step-sequencer. The step-sequencer is synced to global transport, so the steps are played in accordance with transport tempo. The snap-sequencer has many editing modes that allow for modifying the snapshots content (snap,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

`_output`

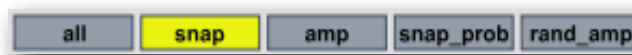
the snap-sequencer outputs a list of data each time a step is triggered. Each data type (snap, amp, amp rand...) is interpreted in order to recall a snapshot, with additional possibilities such as random amplitude of envelope and snap recall probability to spice things up a little.



you can set the beats number change interval with own probability weight, on the right it's shows if the random parts it's to change or not.

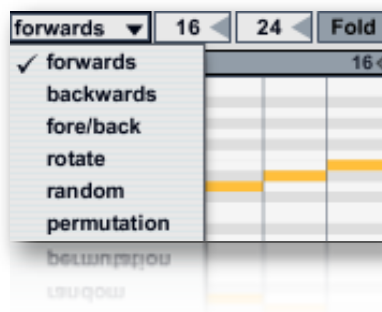


When enabled, the sequencer work also like amplitude envelope for the stream or main (master ch.), you can define a normal ADSR step sequence for draw an envelope duration and shape. If disabled the stream (or main master signal), pass through normally.



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.

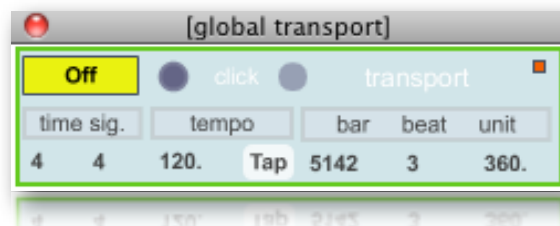


Sequencer play mode read steps in the explained way. You can chose the number of the step (columns) and snapshots range (row), then fold/unfold lets you chose to display all possible snapshots, or only a specific set of snapshots.



Some kinds of trasformations are available, use these four bottom arrow keys to shift the sequence up and down (snap number) or left and right (in time). Hold down the [shift] key to move in four step increments.

`_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

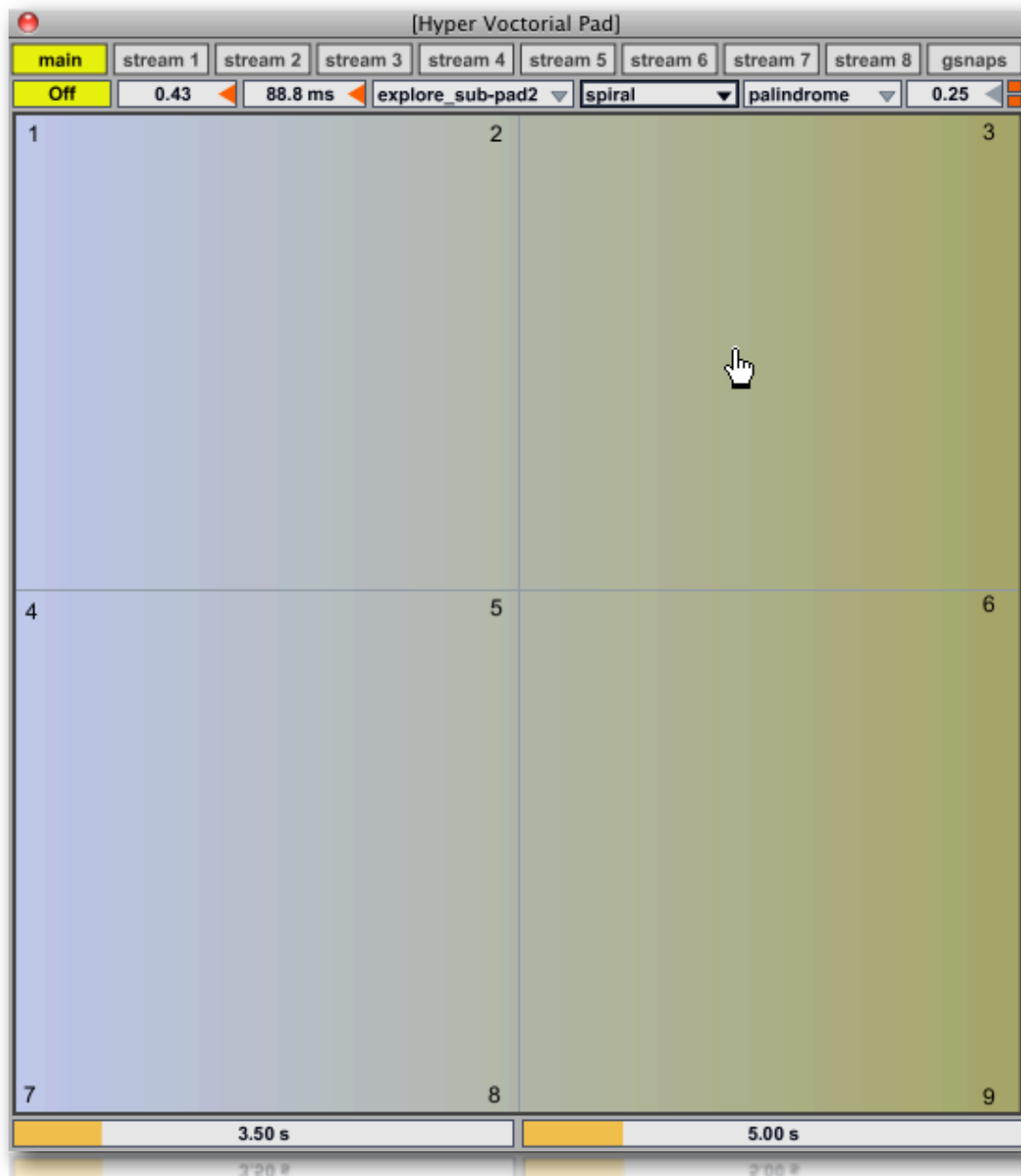
1/128
1/64
1/64D
1/32T
1/32
1/32D
1/16T
1/16
1/16D
1/8T
1/8
1/8D
1/4T
✓ 1/4
1/4D
1/2T
1/2
1/2D
1/1T
1/1

In Pulsaret you can find also this formula of value, they are the same but shorted:

Hyper Vectorial Pad

(<http://www.csounds.com/maldonado/>) thanks Gabriel Maldonado

_general explanation



In Pulsaret 1.0, the 10 HV_pad (i.e. Hyper Vector Pad) they are grouped within an unique window. You can access from Hyper Vectorial Pads menù item (cmd + P or ctrl + P on Windows).

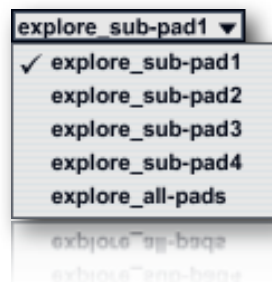
Hyper Vectorial Pad window contains ten vectorials pads **main**, **streams** (8) and **global snapshots**.

HV_Pad extends the functions of micro-pad, interpolating up to 9 snapshots, also allows to modify many parameters with a single mouse motion. Normally the user interacts with these areas with the mouse or through self-scanning exploration, however a remote MIDI control is possible (not yet), by assigning the X/Y axes to one or more incoming MIDI CC (Control Change) messages.

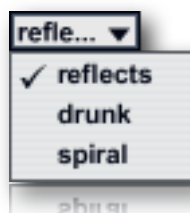
HV_pad combines four pads, each pad has two adjacent sides, sharing the same snapshots to their corners.

_chose pad and exploration mode

HV_pads generally are explored with the mouse, but you can also automate the scanning pointer (automatic pad-explorer) as described below:



Choose a sub-pad which to use for scanning, also you can use the entire HV_pad (entire area: four pads together).



then chose a scanning metod: reflects causes a reflection when the cursor reaches the edge;
dunk moves randomly, creating unpredictable paths;
spiral can be centripedal, centrifugal or palindrome.

_X/Y time

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



They assume a different meaning for each method used, in **reflects**: X/Y time to reach edge (the time it takes to move from side to side);

drunk: max time deviation (random range);

in **spiral**: X (left) time for each spiral, Y (right) total spiral time to reach the pad area;

The spiral, will enable two related parameters:



centripedal moves from outside to inside, when it reaches the limit again, repeat;

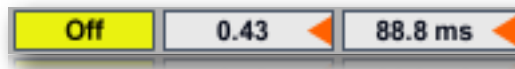
centrifugal does the opposite;

palindrome is a combination of both;

The phase (number box) implies: 0: no spiral formation; 0.25 circular; > 0.25 crushed ; > 0.5 clockwise rotation.

_settings

The last part is about turn on the auto-exploration system and others general settings.
N.B. auto-exploration is possible only when the DSP is on (see above DSP_settings),



Yellow on/off button, start/stop auto-exploration, then the two numberbox (from left) are X/Y step resolution, sets the horizontal/vertical tracking ratio for pointer movements. Values greater than one cause the pointer to move more quickly when dragged; values less than one cause it to move more slowly; the last (most right) is the interpolation update time in msec (be careful with short times).

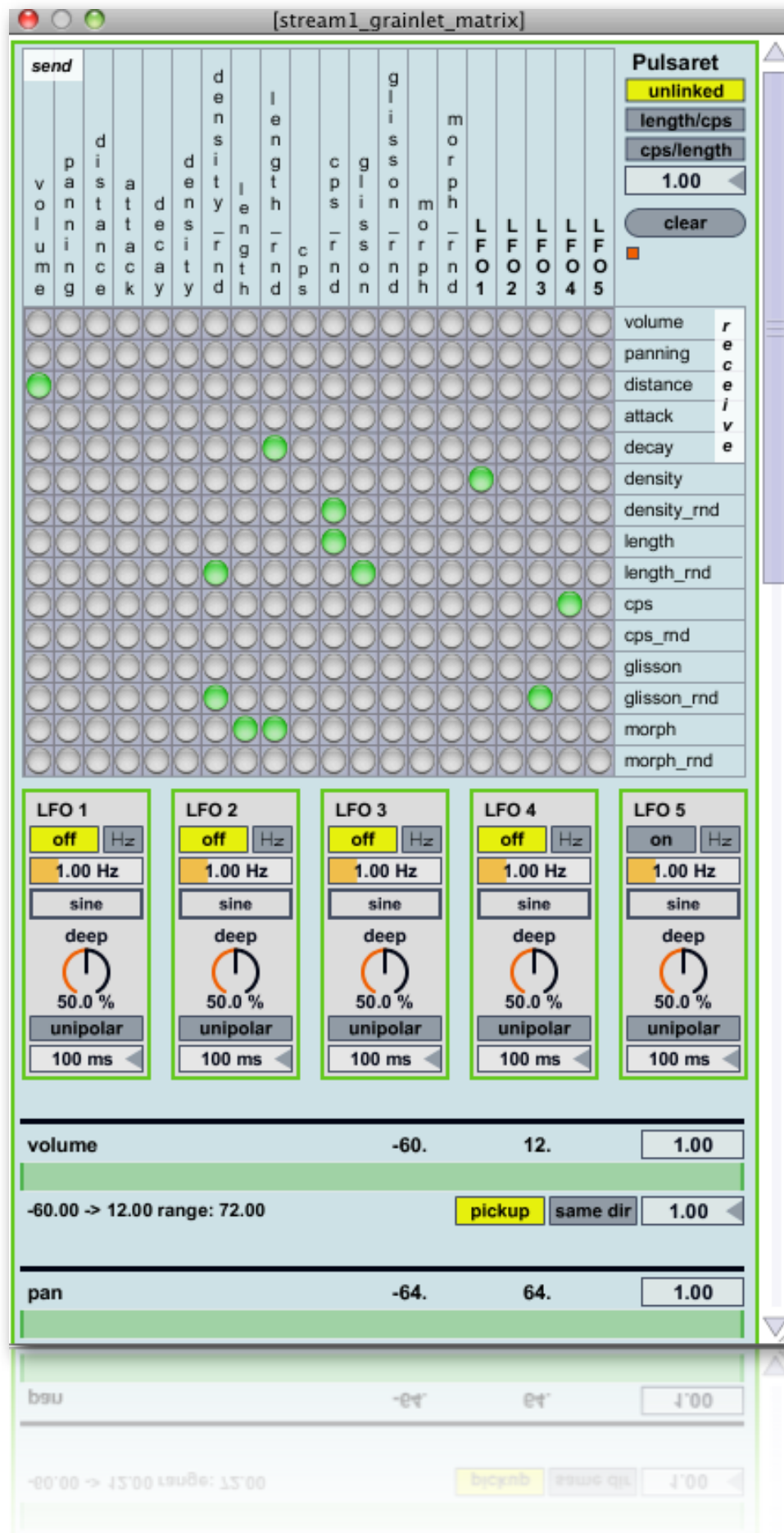


the little up button on the most right sets float/no float window below you can lock/unlock the pads corner for snapshot selection, see micro-pad above for more details.

Matrix

_general explanation

Pulsaret 1.0 implements a matrix parameter linkage, used 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.

_pulsaret linkage



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

$$\text{cps} = (1/\text{length}) * \text{cycleN}$$

$$\text{length} = (1/\text{cps}) * \text{cycleN}$$

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 = $1/100 = 0.01 * 2 = \mathbf{0.02 \text{ Sec}}$ (ossia 20 Ms)

for the cps/length linkage:

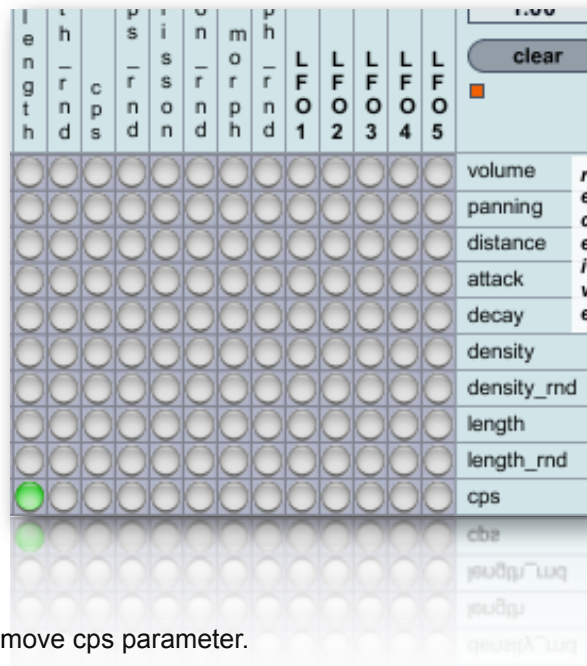
cycleN = 3

length = 45 Ms = 0.045 sec

cps = $1/0.045 * 3 = \mathbf{66.66 \text{ Hz}}$

_parameters rescale

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

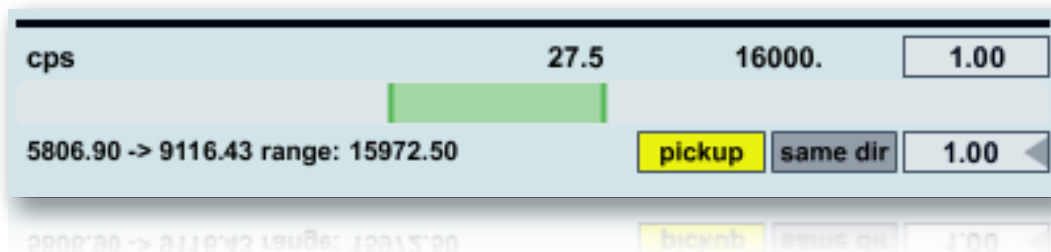


all length range is employ to move cps parameter.

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. After you have assign matrix linkage, you need enable LFO clicking on the on button.

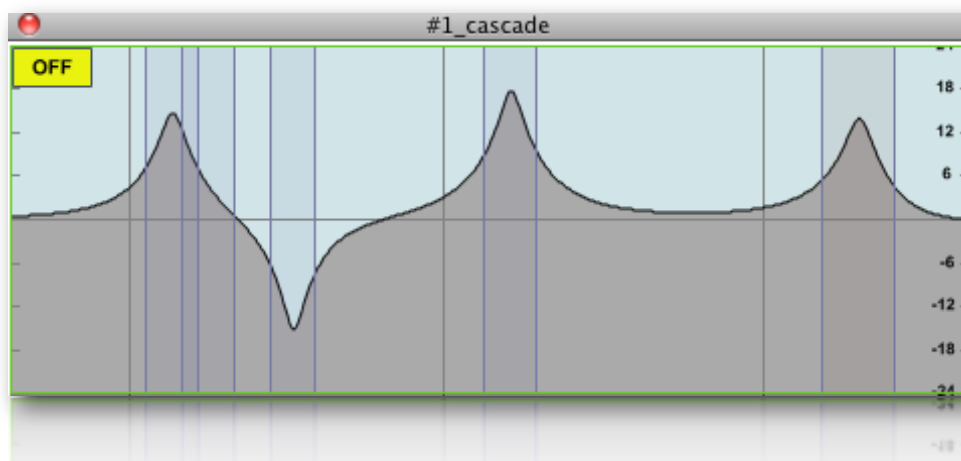


The LFO params are (from top to bottom):

- 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.

Filter

_general explanation

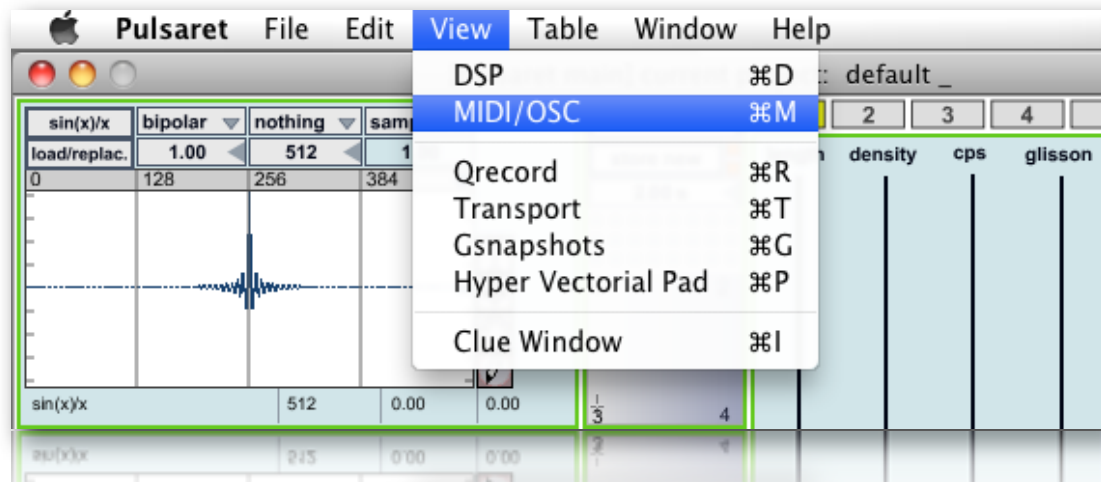


Every streams and Master main output, have own Cascaded series of five biquad filters, cascade filters the signal by a series of two-pole, two-zero (i.e. biquad) filters, often referred to as "second order sections", they can be bypassed by the on/off button.

MIDI/OSC I/O

_settings

MIDI/OSC I/O is accessible only from the application menù (CMD + M or CTRL + M on Windows):



The header is divided into two parts: MIDI (Music Instrument Digital Interface) and OSC (Open Sound Control).



MIDI:

MIDI I/O, provide a synthesis parameters control via 7 bit CC (Control Change). If you have a motorized MIDI controller, Pulsaret can be synchronized with device. When you change stream number by the stream selection, Pulsaret send values params via MIDI data.

In order to chose the I/O ports you can use the two pop-up menùs, a list of available devices will be displayed, click on **refresh** buttun to update devices list, usefull after plug-in/out an device(s).

In order to enable MIDI out, you need select individually each control, to do it quickly, just click **midiout** to select/deselect all widgets for midi output (sync). Analogous **contin** enable/disable all continuously midi out data. It means that Pulsaret data are sent when you move any GUI (Graphics User Interface) parameter.

Generally MIDI data are sent only when you release the mouse button (key up), unlike continuously, which sends data whenever the switch changes state.

The down slider (Pulsaret 1.0) “**MIDI out update rate**”, expressed in milliseconds, speedlimit MIDI output data. Text string is “**MIDI monitor**” to view raw coming MIDI data. The tiny top central button, set float/no float window.

OSC:³

like MIDI, OSC configure the I/O network protocol. Dispatch messages through an OpenSound Control address hierarchy with pattern matching. Values are serialized and sent over the network as OSC compatible UDP;

³ OSC is a CNMAT Max objects, can be found at: <http://www.cnmatic.berkeley.edu/MAX>



Input port (8000) set input port to send messages to at host;

Output port (7998) set output port to send messages;

Oranges buttons enable/disable respective OSC I/O.

Curiosity: default I/O ports: 3009 1974 is the author birthday.

host (127.0.0.1 yellow text string) set output host by ip address or hostname (**localhost** = 127.0.0.1).

Bottom there is text string, like MIDI monitor, show OSC I/O data, also you can copy and paste selected text (OSC path) on a Pulsaret widget.

If you need monitoring Pulsaret output OSC, you can make a loopback with the same input output number port in the localhost.

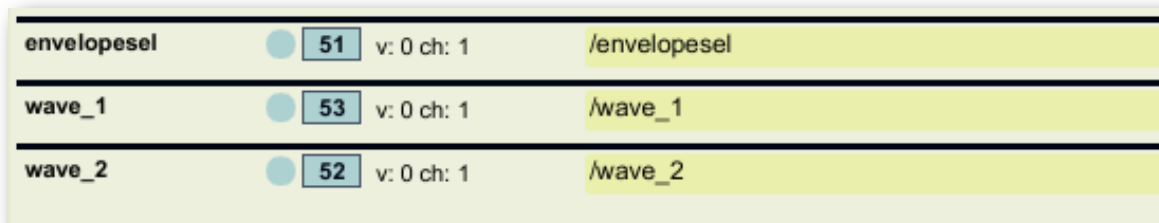
Naturally you can use professional MIDI/OSC router monitor like **OSCulator** ⁴.

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

_raw, scale, trigger

At the present, only CC (Control Chnge) can be assigned to Pulsaret parameters, we have three types of controls:

1: **CC_raw**, 2: **CC_scale**, 3: **CC_trigger**



Inside of any controls, either raw, scale or trigger, they are shown two MIDI/OSC assignments values. Generally in OSC it's showed only text string adress while MIDI shows the CC number and value-channel.

CC_raw reads from MIDI/OSC Input, simply returns integer values (between 0 and 127 for MIDI), circle button is learn function, clicking on it turns red and waits to feel a MIDI CC, when it happens the the control is set up on the incoming CC number and MIDI channel.

Now the Pulsaret parameter is ready to be controlled via MIDI.

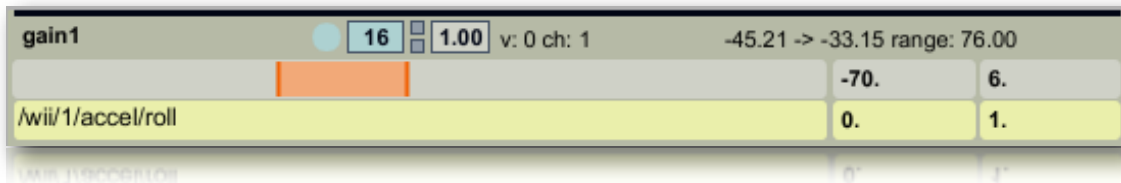
Reported Control Change (number 51 in the example above), of course can be manually changed. Below are showed value (v) and channel (ch) information. at the present they can't be changed manually but only by learn function.

⁴ <http://www.osculator.net/> 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 [iPhone](#) or [Lemur](#) can do just the same... with great ease of use.

OSC data are received at the following address. Naturally you can change it. In the above example, we control Pulsaret **envelopesel** with a **Nintendo Wii controller** (OSC data), it's achievable just copy and paste from the OSC monitor the string of adress.

N.B. raw control, doesn't send Midi OUT values.

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 of $-70 \div +6$ (76 units). You know that MIDI has a 7 bit resolution, then covers a range of 128 only units ($0 \div 127$),



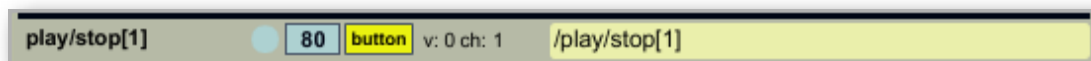
in the above example, the orange horizontal selection scale the 128 steps of CC 16 on a part of the entire parameter range. You can also resize the total range by the box number ($-70 \div 6$), so the current selection will report to current range, you can chose a different range settings min/max range of selection by click and digit a new value, then press enter or dragging the numbox. On the right of CC number, we have the numbox (1.00) is the exponential base factor, default 1.0 = linear (not curved) > 1.00 exponential curve.

OSC works in the same way, it **assumes default values between 0 and 1**. You cal also change OSC input range. For instance the above example, we control the gain1 params through **/wii/1/accel/roll** OSC data from OSCulator and Nintendo Wii, rescaling the range for controlling only a middle part of parameter.

Also the example above, the MIDI Out Sync is disabled (the top of two little buttons next to the CC number) and is also disabled "continuously MIDI Out" (below), in this way the MIDI output will not be sent. Enabling it, MIDI data it will sent only when you release the mouse button (key up), unlike continuously, which sends data whenever the switch changes state.

N.B. CC = -1 as to receive all controller numbers on a specific channel.

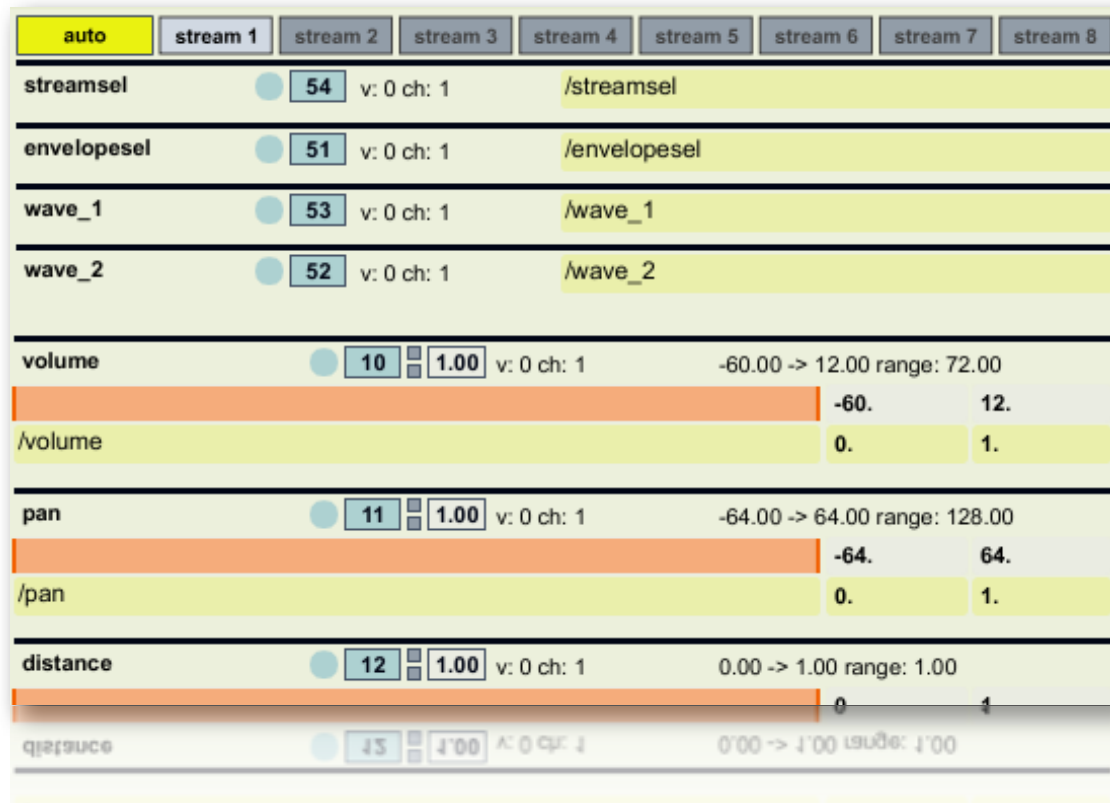
N.B. unlike MIDI, OSC it send continuously the data.



CC_trigger acts as a toggle/button, all the incoming values > 0 will return "true" =1 and the only v0 value will return "false" = 0. You can choose how to use the value: either toggle or button.

Toggle, switch between off and on (0 and 1); button, flash on when the CC value is greater than 0, end 0 when the CC value is 0. OSC need no more explanations.

_streams controls switch



Considering that Pulsaret have 8 granular streams, it would be difficult to manage 8 X N MIDI assignments CC_params. An similar 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...

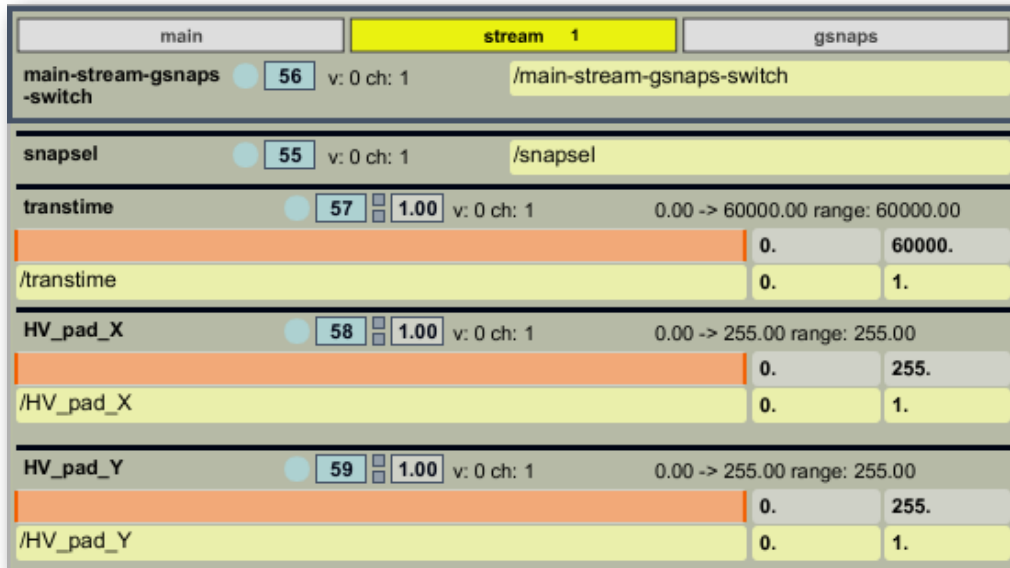
I will think to this in the next releases, at the present MIDI CC, can controlling only and always a **selected granular stream**.

N.B. MIDI/OSC **streamselect** map, is crucial because it allows to bring the stream in the foreground to be controlled via MIDI/OSC.

Like saying, MIDI controls they works only on the **current selected granular stream** (foreground). Selecting a different one, will automatically be able to receive MIDI/OSC on selected data.

In order to disable auto-switch MIDI stream control, you can deselect header auto button, then chese one of the 8 granular stream. Thus when you change granular stream (from de main GUI), MIDI it will continue controlling always the same stream.

_common MIDI params



Some Pulsaret params are common, for example snapshots tratements module: **main**, **streams** (8) and **gsnaps**, total 10. Idem for Hyper Vectorials Pads.

When you open Hyper Vectorial Panel from the main menù, you discover 10 vectorials pads. MIDI controls it complys auto/manual-switch explained above. Moreover we have others two HV_pad: **main** and **global snapshots**. In order to pass MIDI control over it, you need select manually one of two, when selected stream (central default), you will see stream number according to stream selected.

Also you can control selection switch via **main-stream-gsnap-switch** MIDI/OSC.

When you connect a MIDI controller **not motorized**, you should disable MIDI out, so as not to inhibit the "pickup" (see below) system. When the MIDI Out Sync is disabled, Pulsaret work as follows:

_sync/pick-up

An important function is the "pickup value", this is inspired by Ableton Live. Pickup works when a Pulsaret parameter value, does not correspond to its MIDI CC value. Therefore when a MIDI message was received, Pulsaret expect that the CC value reaches the widget.



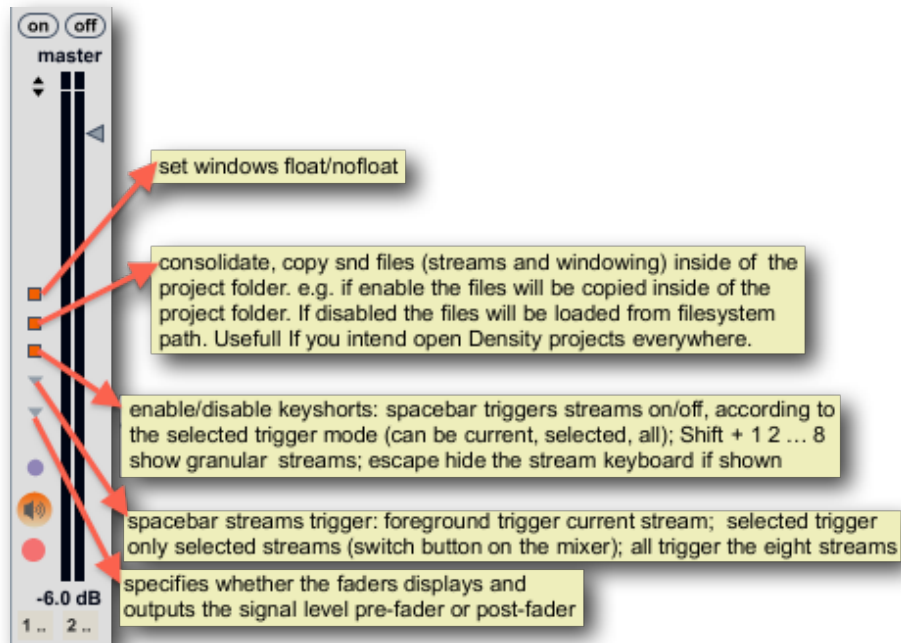
N.B. If MIDI Out Sync is enabled, the pickup system is automatically disabled, so if you do not have a motorized MIDI controller, (like Behrenger BCF2000, for instance) MIDI Out Sync must be disabled.

This section lists all the common controls to the two environments main/streams, Pulsaret allows you to control only one stream at once, only the stream in the foreground.

General

_settings

So far we have analyzed the only main-mixer, "main" also implements other important functions, such as saving the project etc ...

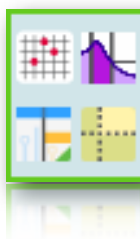


_fast icons

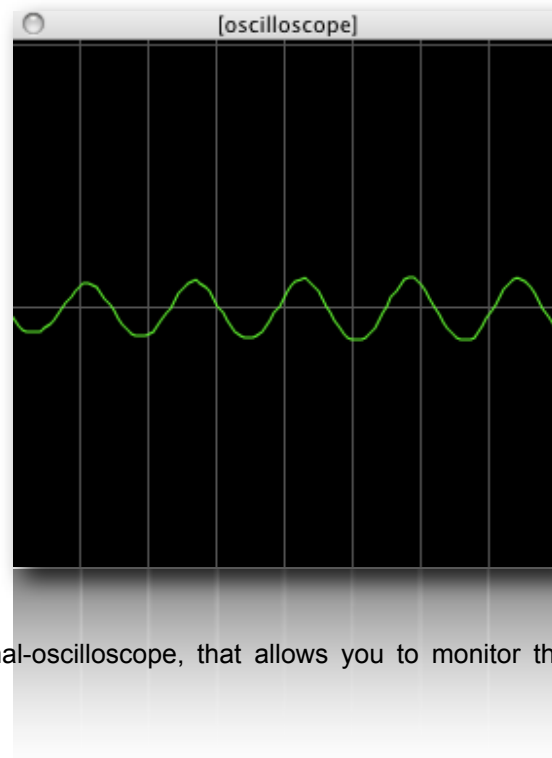
a fast way to open sub windows is by clicking on the icons, we have in the master slider, on the top, an "signal oscilloscope" (see below), then a cascade filter (it work on master signal), snapshots improviser sequencer unit and HV_pads. N.B. HV_pads contains all 8 streams pads, "main pad" and "global pad" too.



In the same way, streams, naturally cascade filter work on stream signal and we have on the top left, the Matrix grainlet linkage (see Matrix for more explanations).



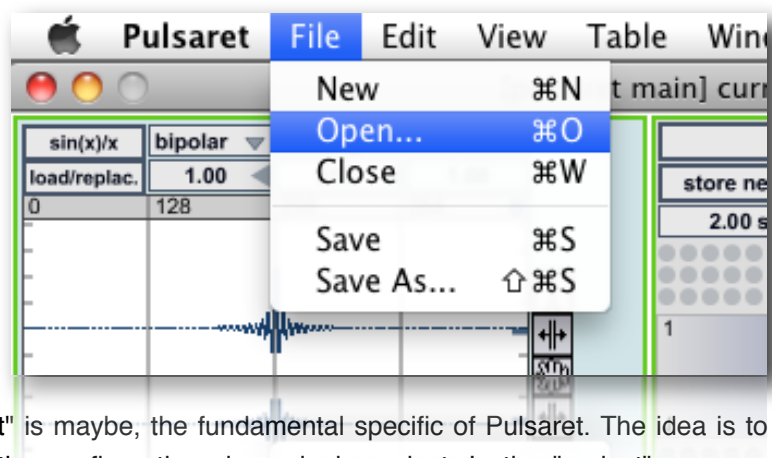
_oscilloscope



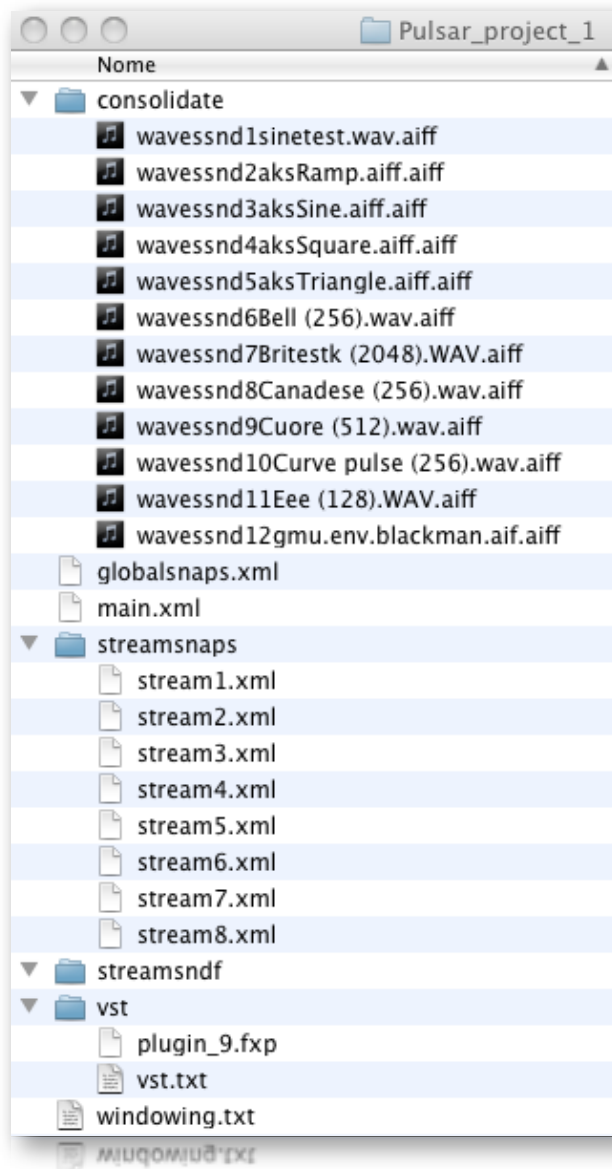
oscilloscope is a simply signal-oscilloscope, that allows you to monitor the visual progression of master waveform.

Discarding "snapshots-transitions", we have already explained Granular streams, The most important 1.0 feature is **consolidate**. When consolidate is enabled, the sound files loaded in the streams slots (12 available) and/or in the windowing (12 available), they will be exported inside project folder. If disabled the project will include only system path. For example when you save a project without consolidate soundfiles, you will be able to make it, checking consolidate button and therefore saving the project still.

_save/load project



Management "project" is maybe, the fundamental specific of Pulsaret. The idea is to record all the presets (snapshots) and all the configurations in a single project. In the "project" are saved the last state of all parameters and settings, then when you reload, returns the last configuration. The project is exported as a folder, with the following root:



main.xml contains the general configuration and main snapshots;
globalsnaps.xml contains ulteriorly all Pulsaret widgets snapshots;

streamN.xml contains the streams snapshots;

streamN.txt contains the data collector. The collector records the path of soundfiles loaded, and information such: zoom, normalize, length selection, file sample length, sampling rate, time duration etc...

windowing.txt as above;

N.B. opening a project on another computer you can not properly access path soundfiles. Therefore you need to **consolidate** sound files in the prject folder. See above for more explanations.

consolidate folder contain a copy of Pulsaret windowing soundbuffer, in the files saved on the project folder, it's added (prefix) an ID. The ID it's based on prefix "shape" and the slot number (of 12 slots), moreover it's postfixed (append) the ".aiff" extension

N.B. you cannot use \ or / symbol in the project name folder.

e.g

shapesnd12filename.wav.aiff

for windowing module

When you **Save** newly the project without consolidate checked flag, the originals files paths they will be restored;

vst folder contains all presets for the VST plug-ins loaded, the plugins bank files (.fxp) they will be created only for the allocated vst slots;

The items **Save Save As...** are only available form the application menù, when you save a new project, you must write a name. Remember that the name refers to the **project folder**, saving it again will overwrite all data.

In order to load a Pulsaret project, **Open...** from application menu item, it will open file system window where to be able to choose a project folder.

N.B. you may not select a subfolders or invalid project files, you need simply select the only entire folder of the project.

N.B. You can quickly load your project, **drag the project folder on the Pulsaret GUI**.

We have already seen **MIDI/OSC I/O** and **DSP** settings, please see above.

_status bar



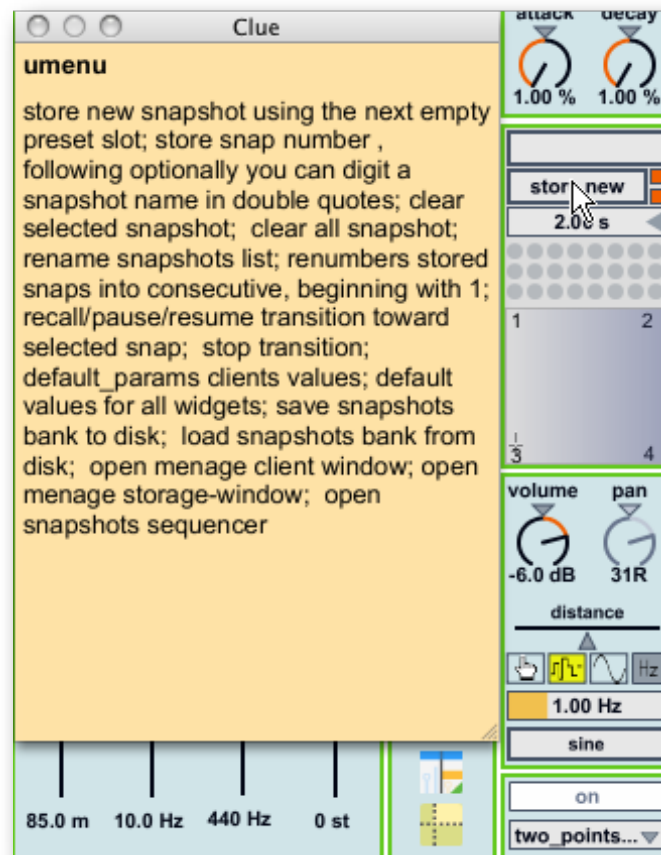
status bar is subdivided into two part, on the left we have main and general DSP infos, on the right we have selected stream wave pad infos and some others stream infos (like grain overlaps) and Matrix pickup value.

On the left SR (sampling rate) it can be dynamically changed from the status bar, the current sampling rate it will be saved on the Pulsaret project, in fact when you consolidate soundfiles in the project it's important re-load project SR properly, to avoid down/up sampling pitch change. All the others are some general informations like I/O vector size, signal vector size, overdrive state (enable/disable) and cpu utilization. The orange button on the left, enable/disable cpu and overdrive monitoring. The empty space (center) show "pickup value", see **MIDI/OSC** for more explanations. the two circle leds they show **transport global activity** (left) and **MIDI input activity** (right) but they have a double function, infact clicking over they, you will open directly global transport and MIDI/OSC.

The stream status bar is similar, enabling the orange button you will monitoring grain overlaps factor, depending by density/length values (see Granular streams for details), the other follows information (on the right) are relating to sound file showed in the in wave pad: file name, file length in samples, sound file sampling rate, sound file length in milliseconds (see wave pad for details).

_clue window

In Pulsaret 1.0 ver. I've removed the annoying Hints messages, they always appeared when mouse is over of Pulsaret widgets, soiling the GUI during realtime performance. Now all the hints are visualized inside a **clue window**.



The clue window (cmd + I or ctrl + I on window) is a floating window that displays documentation for the item under the mouse. You can use it to obtain reference information for objects or learn what something does in the Pulsaret user interface.

You can chose to close it when you will know Pulsaret features, expecially during performance.

_history

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

- .eight granular streams simultaneously
- .envelope/windowing manage up to 16 pre-generated shape (prototypes)
- .envelope/windowing loading and manage 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
- .multichannels I/O mapping
- .quick-record export master channel and/or multichannel file streams, progressive file autaname, 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 manage: 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 menu 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 sndfiles consolidate (copy buffers in the prj folder)
- .every stream can deforming own windowing envelope (attack/decay)
- .global presets, manage 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

_acknowledgments

I would like to thank:

Alessandro Grego, Eugenio Giordani, Massimiliano Vitti, Giacomo Pirozzolo, Alessandro Barbetti, Han Otten
A special thanks to Renato Alberti, Felix Petrescu, Eric Lyon and Carmine Cella

Bibliography:
Curtis Roads "Microsound"
The MIT Press
Cambridge, Massachussets

The Author:

Alessandro Petrolati
apeSoft

ape@kagi.com

www.densitygs.com

www.alessandro-petrolati.it