

GranuLab

Real-Time Granular Synthesizer

by [rasmus ekman](#)

[granular synthesis](#)

[program overview](#)

[patches](#)

[store output sound](#)

[output options](#)

[midi support](#)

[version info & contact](#)

What is Granular Synthesis? [>>> granny overview](#)

w o t s i t ?

Granular synthesis is the technology of creating complex sound by playing back many short and relatively simple sound fragments with varying parameters.

f o o d f o r c h o p

Just using a sine waveform as base material, and controlling length, pitch and density of grains, several types of sound can be generated.

It is however more exciting to use a concrete sound sample. Good raw material includes speech, instrument or ambient sounds, or indeed synthetic sounds produced with Csound, **GranuLab** or some other program. In this case, granulation means cutting up the sound into short segments and controlling how these are played back.

p r o s a n d c o n s

The advantage of granulation is that you have control over where each grain begins reading in the soundfile. This means that you can do time-stretching and repitching of sounds in real time, without using computationally intensive FFT analysis/resynthesis.

From a formal viewpoint granulation is a bit of a fudge, and sometimes it is impossible to create a clean effect which has neither echoic nor a metallic or dredgy sound.

But in any case which is just the point any sound can be scrambled beyond recognition with very little effort.

GranuLab Overview

[>>> grain params](#)

[>>> versions](#)

basic operation

On starting **GranuLab** you'll just see a mass of sliders and some buttons. The sliders which control grain generation are grouped in thematic boxes. Each long slider is the main controller of a parameter, beside it are one or two half-length sliders which do random or [amplitude-modulation variation](#) of the longer slider's value.

You can load a 16-bit WAV mono or stereo soundfile into **GranuLab** for processing. With stereo files you can decide which channel(s) to use for input to the grain stream (under the [Command](#) menu).

There are five basic parameters in **GranuLab**:

[Soundfile section](#) - which part of the current sound file is used.

[Soundfile playback rate](#) - Controls the rate by which the starting point of each succeeding grain is incremented. This is the key to time-stretching and time-compression.

[Grain rate and grain length](#) - These parameters cooperate to determine **grain density**. Low density (less than 1-1.5) means chopped sound. High density allows many echo and pseudo-filtering effects.

[Grain pitch](#) - The internal pitch of each grain. The pitch may be changed during playing time of the grain, by a simple glissando parameter.

[Grain envelope](#) - attack and decay time for each grain, in percent of grain length.

Besides the grain parameters, there is also a section for [amplitude and panning](#) of the output of each grain stream.

Each basic grain parameter may be modulated, either by random offset from the set value, or by the amplitude of the point in the sound file where the grain is taken from (see [amplitude modulation](#) below).

SHIFT+click on a slider to restore the default value. Restore all sliders at once by clicking the [Default patch](#).

Slider output is always shown in the status bar, together with relevant information.

The setting of all sliders may be stored as a [patch](#), which may be recalled later.

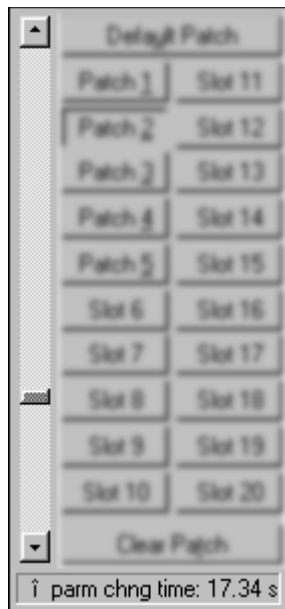
GranuLab may also be controlled by external [MIDI signals](#).

GranuLab 8 can have up to [8 streams simultaneously](#).

amplitude modulation

Most parameters may be controlled by the amplitude of the input sound. The amplitude envelope of the sound is taken as the file is read in to **GranuLab** (the left channel of stereo files). The amplitude value used for each grain is synchronised with the soundfile at the point where the grain will end its fade-in phase. Some of the amp modulations create quite unique connections in the sound.

the portamento slider

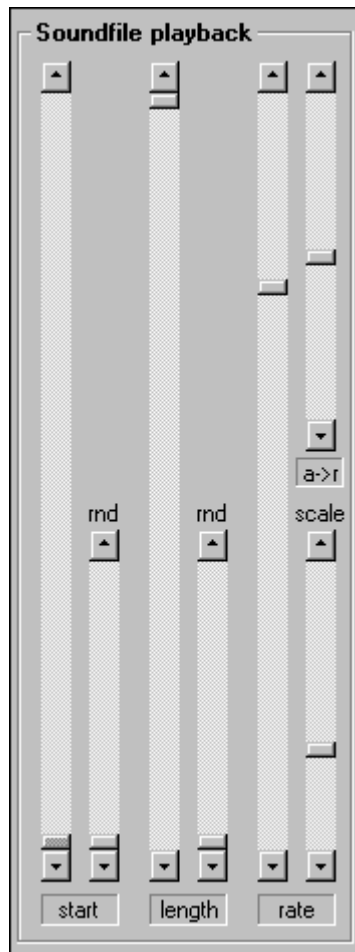


The slider in the [Patch banks](#) box (far right) controls the time which it will take for any change of slider values (or patch) to take full effect.

If the time is set to eg 30 seconds, and you click a patch button (or move a slider), the grain generation values will slowly transform from the current values towards the new patch (or slider) value. By changing the portamento time and clicking again on a patch button or slider, you can speed up or slow down the rate of change at any time.

Soundfile Playback

>>> [grain density](#)



(The parameters in this section are not really applicable to the built-in sine wave, so use mainly with loaded sound files.)

Sliders **start** and **length** control which part of the sound is used. The selected part of the sound is looped.

The **rate** group of sliders are the key to time-stretching.

scale: This slider controls the range of the main rate slider. At the default setting, the soundfile playback rate may be varied between +/-2 times normal speed. If the scale is increased, the playback rate range may be up to +/-22 times normal playback.

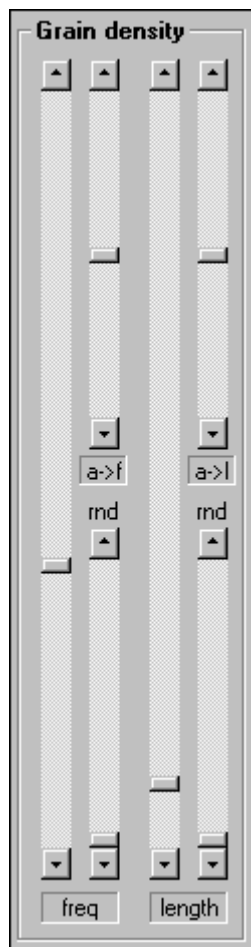
In the maximum range, the precision of the rate setting deteriorates, so it seemed useful to have two controls for playback rate.

a->r: The amplitude of the input sound can be used to

control playback rate. This can be used to good effect eg with speech, making vowels relatively longer or shorter.

Grain Density

>>> [grain_pitch](#)



The frequency at which new grains are generated (**freq**), multiplied by the length of each grain (**length**), determines grain density. The total density will be displayed in the status bar, rightmost pane. In [GranuLab 8](#), the density of all streams follows the density of the current stream.

Density: 5.513 (9.518)

Since each grain is faded up and out, density 1.5 - 2 is usually the minimum for smooth non-grainy playback. With thicker density, echo and chorus effects may be created.

A high grain frequency may create a pitch which interferes with the pitch of the sound (for good and for bad).

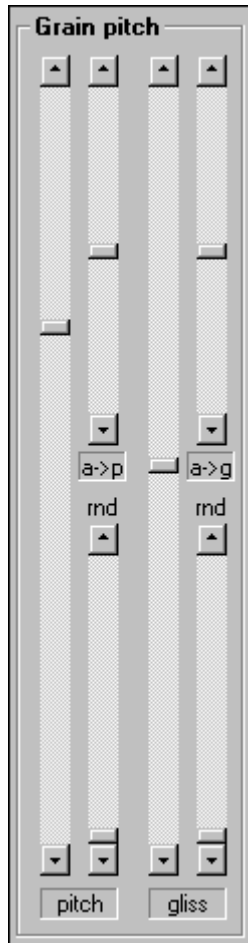
Warning! You need to be careful with grain **freq** and **length**: Your computer can only manage a certain grain density, and when this is exceeded, the sound will begin chopping. This will at the same time weigh down the processor so that the **GranuLab** program interface (sliders and all) gets no time from Windows for normal operation. This

will make it seem to get completely stuck. **In this situation, hit the Q key on your keyboard to stop sound output.**

See also: [I/O options](#).

Grain Pitch

>>> [envelope](#)



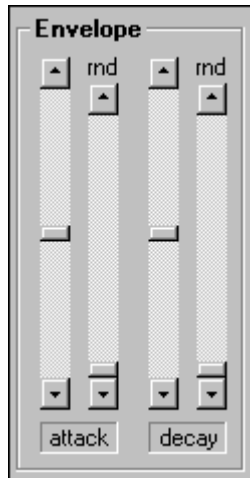
The [soundfile playback rate](#) controls where in the soundfile each new grain begins reading.

Grain **pitch** decides at which rate the grain will play back its fragment of the sound. This works like a normal sampler, or like interfering with the speed of a gramophone turntable, but since the starting point of each grain is controlled separately, the Donald Duck or tape slow-down effects do not appear with short grains.

a->p: Amplitude modulation of pitch can be used to enhance drum loops, etc.

Grain **gliss** controls pitch change during the play time of each individual grain. The grain pitch may slide up to +/- 8 octaves away from the pitch at the beginning of the grain.

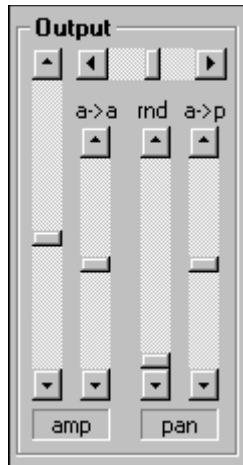
Envelope >>> [amplitude & panning](#)



The envelope sliders control how large portion of each grain is spent for fade-in and fade-out, in percents of total grain length.

Amplitude and Panning >>> [patch banks](#)

GranuLab 8 version



The leftmost slider sets global output amplitude.

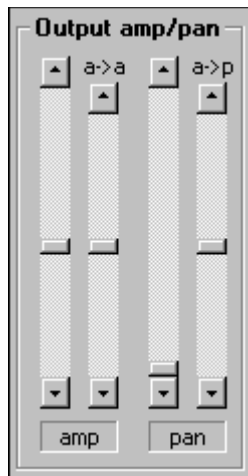
The **a->a** slider gives a compression expansion effect, as amplitude of input sound controls the amplitude of grains;

The top horizontal slider sets base panning in output stereo image.

The **rnd** slider sets the amount of left-right randomisation of grains. This is most useful with mono soundfiles output in stereo.

The slider **a->p** sets the amount of randomisation controlled by amplitude of input sound.

GranuLab 1.0 version



The leftmost slider sets global output amplitude.

a->a: gives a compression expansion effect, as amplitude of input sound controls the amplitude of grains;

The **pan** sliders only control the amount of left-right randomisation of grains. This is most useful with mono soundfiles output in stereo.

a->p: Panning randomisation is controlled by amplitude of input sound. The two panning sliders are not so useful together.

Patch Banks

[>>> gesture window](#)

patches



When you have changed the slider values to some interesting configuration, hold down the SHIFT key on your keyboard and click on a slot button in the Patch banks box.

Each patch slot holds one full set of slider values. To recall a patch, just click on the button.

The *Default* configuration of the sliders gives (roughly) straight playback of a sound file. The default value for each slider can be recalled by SHIFT+clicking it.

banks



There are eight banks, accessible from the **A** through **H** buttons.

Each bank holds up to twenty patches (only 16 on 640 x 480 screens), and may be saved to a file.

The name in the text-edit field is used as a (Windows 95) long file

name, so only use valid file-name symbols.

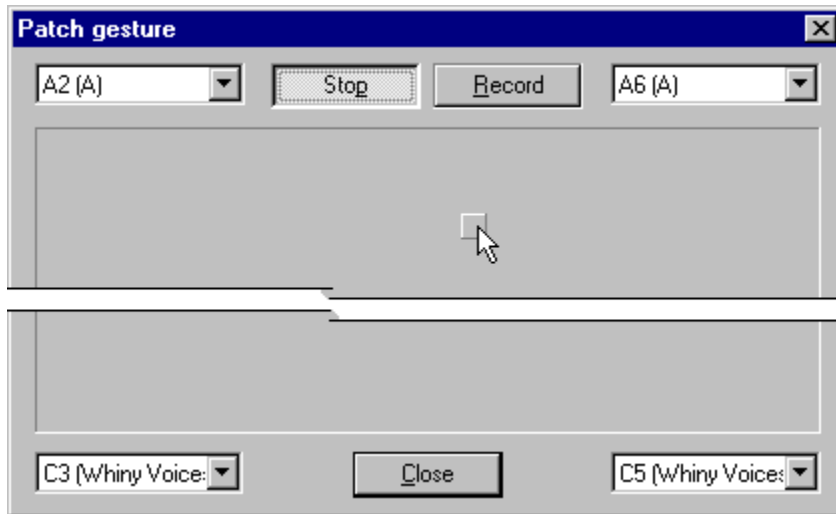
The patch bank files are by default all saved to the **\Patches** folder in **GranuLab's** home folder.

You may change the active patch folder only by loading a new patch from another folder. (I wanted minimal use of file dialogs, as they interfere with output sound.)

Gesture Window

[>>> granny overview](#)

[>>> store sound](#)



(window not shown in its entirety)

This is a special window where you may assign a [patch](#) to each corner, and then crossfade between the four patches by mouse control. This window is opened from the [Command | Open Gesture Window](#) menu. All currently available patches at the time of opening the window appear in the drop-down list boxes in the corners.

If you SHIFT+click when closing the gesture window, the last played patch mix is imported into **GranuLab's** main window.

GranuLab 8 only

In **GranuLab 8** the Gesture Window is non-modal, meaning it does not take exclusive control over the interface. Normal operation is therefore possible: Loading soundfiles, changing active stream tab, storing patches etc. The generated gesture patch will be applied to the topmost grain stream.

The Gesture Window may be folded up or out by double-clicking in its title bar, to get it out of the way temporarily.

To refresh the set of patches in the Gesture Window dropdown boxes, the dialog must be closed and then reopened. You cannot store or open patch files while this dialog box is open.

Multiple Streams [>>> granny overview](#) [>>> versions](#)

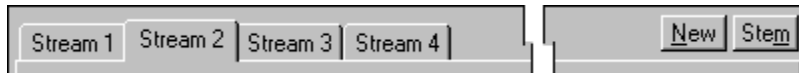
(Please note: These features are available in **GranuLab 8** only.)

See also [GranuLab 8 features](#).

s t r e a m s

The whole set of grain parameters applied to a soundfile, outputting sound, is called a *grain stream*.

In **GranuLab 8**, there may be up to 8 grain streams. These are accessible from tabs in the interface. Each tab reveals a complete, independent set of grain parameters. These may be applied to the same soundfile, in or out of sync, or to a different soundfile for each stream.

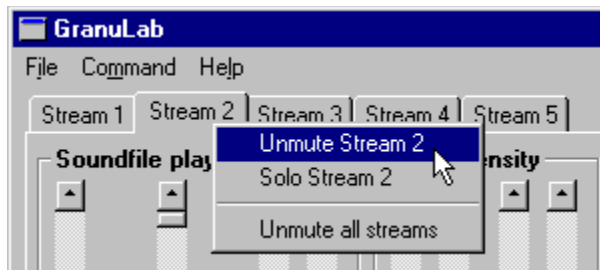


When you hit the **New** button, the current stream is copied to a new stream. (This is the only way of guaranteeing that a soundfile is used by two different streams in sync.)

Stem will remove the currently visible stream.

s o l o a n d m u t e

Right-click a stream tab label to show a popup menu which offers standard solo and mute options. The stream tab does not need to be topmost.

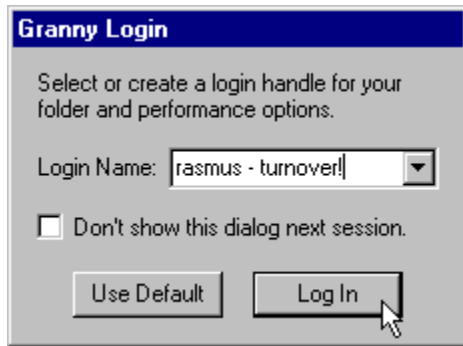


Login Window [>>> granny overview](#) [>>> versions](#)

(Please note: These features are available in **GranuLab 8 Pro** only.)

See also [GranuLab 8 features](#).

multiple preference configuration storage



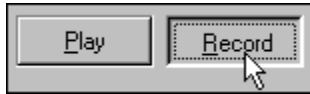
The **Login Window** by default opens at program startup, and lets you log in under any name. All preferences (file paths and output/MIDI options etc) will be saved between program runs separately for each login. This may be convenient in a multi-user environment.

During program run, a new login can be opened or created from the [File | New Login...](#) menu.

To delete a login handle, select the entry, then hold down CTRL+SHIFT and click the **Log In** button

Not tested in a network setup, so this may be limited to local use.

Storing Output Sound to File [>>> i/o options](#)



Hit the **Record** button to store the generated output sound to a WAV file.

NOTE: This button will be renamed to when live input becomes available.

There are two options for the name of the output file: Either it is always called **Granny.wav** and is overwritten every time you hit **Record**; or it is called **Granny##.wav**, where **##** are two digits which are incremented for each new recording. This option is set under menu [Command | Use Numbered Out Files](#). The output file folder can be set by a dialog invoked from menu [Command | Output Soundfile Options | Select Soundfile Folder](#).

You can select output sample rate, mono/stereo etc independently of the format of the input sound file. The input sound is completely resampled, so its sample rate does not affect performance.

tip:

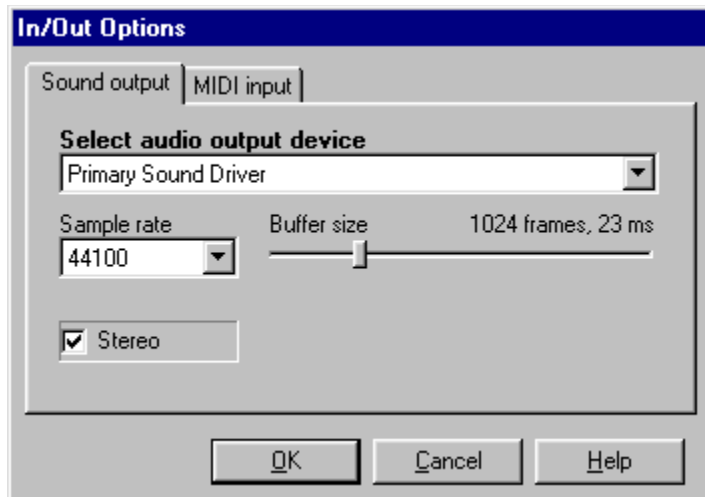
By recording a file, you can capture very dense grain structures even though your computer cannot keep up with generating the sound.

Make sure that the option **Soft stop** in [Command | Grain Generation Options](#) menu is NOT checked. While playing, use the **Q key** on the keyboard to stop sound output, but do not stop recording! As soon as sound output stops, you can click a patch slot button (or move any sliders). The slider values will change to their new settings in file time - according to the state of the [portamento slider](#) - rather than in real time.

When you hit the **Play** button to restart sound production, the recorded file will not have the chopped clicky sound that you hear while recording the only problem is timing when to change patches while the output sound is chopped and delayed (since the processor cannot keep up).

I/O Options [>>> store sound](#) [>>> midi support](#)

sound options [>>> midi options](#)



This dialog is available under the [Command](#) menu. Here you select sound card and MIDI input device.

GranuLab 8 supports DirectSound output. This may give around 20-30 milliseconds response time at lowest, but performance will vary.

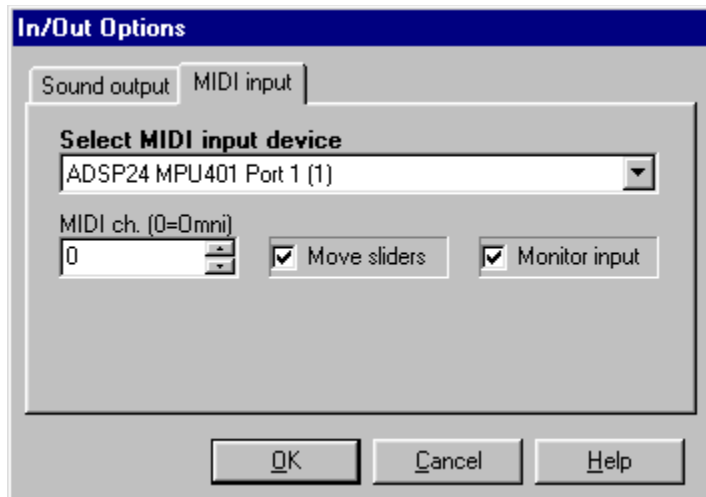
GranuLab 1.0 supports only the old Windows Multimedia drivers.

If you play **GranuLab** from a MIDI keyboard, you'll want as small sound buffer as possible, but this will make the sound break up at a lower [density](#). With lower sampling rate you can also shorten the buffers means slower response from the program, unless you also change the buffers.

The option **Soft stop** will make **GranuLab** fade out grains gracefully before stopping sound output this is to avoid clicks in the output sound. This is really necessary for MIDI note playing, and preferable at most other times too. **Soft stop** has the following consequences:

1. It will take **GranuLab** the latency time to quit performance. There will then be a release time for MIDI notes, which you can control only by minimising the output buffers.
2. This fade-out will also appear in any recorded sound file, so if you do not want them there, you will have to put up with clicks each time sound output stops. There is currently no way around this.

MIDI options



The **Midi ch.** box selects a midi channel to listen to. If zero is chosen Granny will listen to all MIDI channels. This is useful also when checking if any MIDI messages are reaching the program.

The **Move sliders** option will move the sliders visibly along with any MIDI controller input. This used to affect program performance slightly (on a 486), but should now probably be left checked.

If **Monitor input** is selected, information about all MIDI input is displayed in the status bar. This does not affect program performance measurably, so it is recommended to keep this checked at all times.

using GranuLab with other MIDI programs

You may want to generate MIDI controller events in another program, and feed this to Granny. This is perfectly feasible, using a "MIDI loopback" program (**Hubi's Loopback** is recommended. It can be found by searching for that string eg in the Google search engine).

Note that any program generating real-time MIDI will be competing with **GranuLab** for CPU time. Many MIDI programs, (like Cakewalk or Cubase), will be quite aggressive about their system priority when playing, so that **GranuLab** is starved for processing time. This will lead to severe chopping of output sound, even for lower densities (on a 200 MHz Pentium MMX the maximum density immediately decreased by two-thirds). You should therefore try to find some fairly simple (or older) program to generate your MIDI events, or preferably use external hardware.

MIDI support >>> [i/o options](#)

patch selection

Program change messages 0-20 selects the corresponding patch in the current bank. 0 is the default patch (they may appear as 1-21 in your MIDI gear, then default = 1, but I think you'll manage). Select bank **A-H** with program change numbers 30-37 (or 31-38).

MIDI controllers

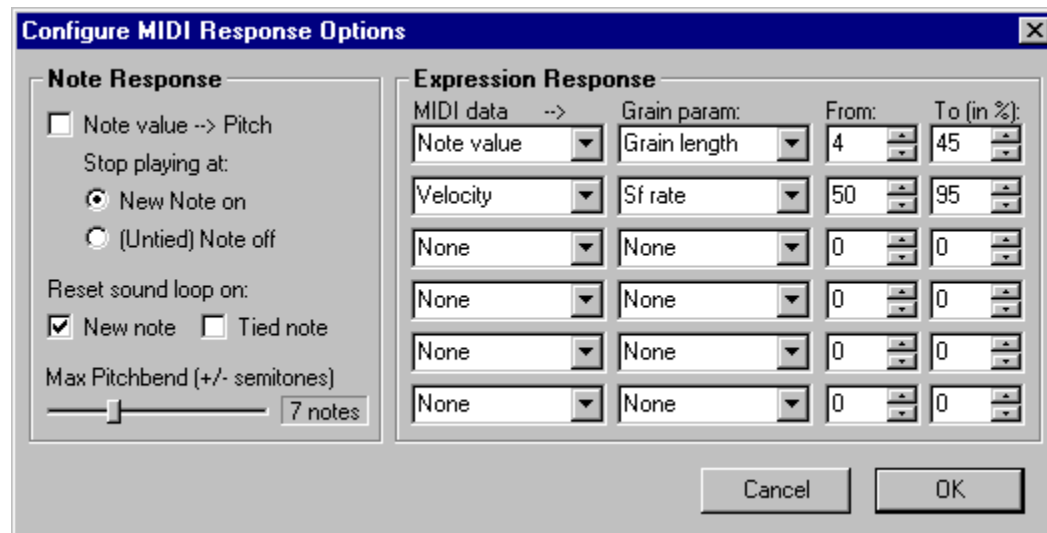
The sliders for grain generation may be set by MIDI controllers. Controllers 4-31 control the sliders from left to right (4 = soundfile loop start; 31 = portamento time). Since each controller has only 128 different values, controllers 36-63 do precision control (14-bit). They are only useful sometimes see [MIDI Controllers](#) for assignment and default values.

Warning: Don't mix mouse input, MIDI controllers, notes or several expression options for the same sliders; this is not supported. That is, grain parameters will change abruptly, and you may have to put up with unintuitive program behaviour.

MIDI note input (GranuLab 1.0 only)

You can use MIDI note input to trigger **GranuLab 1.0**. The note values use middle C (Note value 60) for original soundfile pitch. If the pitch slider is in the lower half (reversed grain playback), MIDI notes will also use reversed playback.

MIDI response (GranuLab 1.0 only)



In this dialog you can choose to let common MIDI note data (ie velocity/aftertouch and note/pitchbend) control some of the grain generation parameters. The MIDI data are kept within a range of parameter slider's length, which is given as a percentage. Thus, if you want pitchbend to control grain gliss over the positive half of the grain gliss slider, you set the values From = 50%; To = 100% (or inverse). If you use the expression values to control [grain frequency](#) or [grain length](#), it's advisable to use the same MIDI expression type to control the other parameter inversely else you will probably get too high densities.

Note: In **GranuLab 8**, the MIDI response has yet to be redesigned -- the interaction between MIDI

channels and streams will be resolved for **GranuLab2**. ((If you push, it might happen. So far I've had no feedback on this feature whatsoever...))

MIDI Controllers

Table of **GranuLab** MIDI input controller assignment and default values.

Crude (fine)	- Slider	- Crude default values
4 (36)	- Start position	- 0
5 (37*)	- Start position, random	- 0
6 (38)	- Loop length	- 127
7 (39*)	- Loop length, random	- 0
8 (40)	- Playback rate	- 111
9 (41*)	- Playback rate, scale	- 0
10 (42*)	- Playback rate, amp mod	- 64
11 (43)	- Grain frequency	- 33
12 (44*)	- Grain frequency, random	- 0
13 (45*)	- Grain frequency, amp mod	- 64
14 (46)	- Grain length	- 11
15 (47*)	- Grain length, random	- 0
16 (48*)	- Grain length, amp mod	- 64
17 (49)	- Pitch	- 87
18 (50*)	- Pitch, random	- 0
19 (51*)	- Pitch, amp mod	- 64
20 (52)	- Glissando	- 64
21 (53*)	- Glissando, random	- 0
22 (54*)	- Glissando, amp mod	- 64
23 (55*)	- Attack	- 42
24 (56*)	- Attack, random	- 0
25 (57*)	- Decay	- 42
26 (58*)	- Decay, random	- 0
27 (59*)	- Output amp	- 57
28 (60*)	- Grain amp, amp mod	- 64
29 (61*)	- Grain random panning	- 0
30 (62*)	- Grain rand pan, amp mod	- 64
31 (63)	- Panning (GranuLab 8)	- 64
32 (64)	- Parameter change time	- 2

(fine controllers marked with * will generally not be useful, since they make a marginal contribution)

In GranuLab 1.0 and earlier versions, panning does not exist, so Parameter change time uses controller 31.

Table version date: Jan 07, 1999

Contact & Commercial Info [>>> registration](#)

contact

mail: rasmus.ekman@abc.se

web: <http://hem.passagen.se/rasmuse/Granny.htm>

Below some of the information on the web is reproduced. Note that this may be outdated, so please refer to the above link for current information.

version information & pricing

GranuLab 1.0: This program is publicly distributed from the GranuLab home page.

It is freeware for individual home users. Any commercial use of the program incurs a licence fee. This includes using it to decorate your magazine's cover CD-rom, or charging people in any way to use or access a machine with **GranuLab** installed.

See [How to Register](#).

GranuLab 8

GranuLab 8 (or "GranuLab Transitional"): This program is like running up to 8 copies of **GranuLab 1.0** simultaneously. It also has **DirectSound** support and some other useful features.

GranuLab 8 is at present only sent out to registered users by email.

See [GranuLab 8 Features](#), [Multiple Streams](#), [Login Window](#), and [How to Register](#).

GranuLab 2

GranuLab2: This is a projected future complete rebuild. See the [development plan](#).

You may register ahead of time to get this program. On registration you will get **GranuLab 8**. When Granny^2 is ready you will receive any and all versions of that program.

Present registration fee for individual home users: \$20.

Fee for professional users, institutions or [commercial entities](#): \$50.

GranuLab2 is presently vapourware. The price will be raised when it is distributed.

See further [How to Register](#).

By "**commercial entities**" I mean of course normal business enterprises (like a studio charging bands more than a nominal fee); also professional music producers, educational institutions which charge their students directly, and organisations which run ritualised or frequent member donation schemes. Music producers who sell less than, say, 5000 copies of their albums, and who attract less than a couple of hundred paying people for concerts are not considered commercial entities in this sense.

GranuLab 8 Features

[>>> contact](#)

[>>> registration](#)

- * This version will run up to [8 grain streams simultaneously](#), accessible by tabs in the interface. Each stream can use a different soundfile. Streams may be muted and soloed.
- * **DirectSound** support. I get 20-30 ms latency, but performance will vary.
- * Soundfiles can be loaded at will during performance.
- * The pro version offers separate login for multiple users. See [Login Window](#).
- * The [Patch Gesture Window](#) is non-modal. This means that you can change stream and set/use patches while this window is open. The patch generated in the Gesture Window will be applied to the currently visible stream, and may be stored as a new patch at any time.
- * [Panning](#) is redesigned to allow stereo placement of streams.
- * **MIDI note->pitch input does not work at present!** The whole MIDI handling will be reviewed at some later time. [MIDI controller](#) input to sliders should work however.

GranuLab2 Development Plan [>>> contact](#) [>>> registration](#)

Document date of this version: Feb 10, 2003.

Preliminary and subject to changes. The time schedule leans towards the optimistic and may be postponed at any time.

- * Live input - Early Spring 2003
- * (ASIO support as soon as I can make it work at home)
- * More control options - bit by bit, starting in Spring 2003
- * New format for patches (XML) - Spring 2003
- * Simple filter on sound input - Spring 2003
- * Grain frequency modulation (not just up/down gliss) - Spring-Summer
- * New interface - Summer-Autumn (also hike the reg price)
- * New version of GranuLab 1.0, with new interface - Summer 2003

Still free of charge, still single stream, has everything above except maybe live input

- * Remake MIDI control - Summer-Autumn
- * 32-band filter on input sound, each outputs a grain stream - Summer-Autumn 2003
- * external soundfile can control 32-band filter; this allows granular vocoder effects - Autumn 2003 (hike the reg fee again here)

How to Register >>> [contact](#)

Document date of this version: Feb 10, 2003.

The information and terms stated in this document may change at any time.

Please make sure that you have the most recent version of this help file before acting on the information below. You will find it packed with the most recent public version of **GranuLab**.

[Go to the GranuLab home page](#) (click this link if you are currently on-line).

overture

This document covers two common cases:

- (a) You're a happy home user / institution and want [Granny 8](#) plus the future [GranuLab2](#). Please read on.
- (b) You want to include the present binary on the cover CD-rom of your magazine, so you've been encouraged to register. I suggest \$10 per thousand readers.
See address below.

registration info

Online registration will be available shortly (written Feb 2003)

You can still register **GranuLab** by sending well concealed cash to the following address:

Rasmus Ekman
Bjurholmsgatan 16
116 38 Stockholm
Sweden

(This address should be valid at least until end 2005.)

Swedish residents can also use Postgiro, please mail me for info.

price:

US \$20 for individual home users.

US \$50 for professionals, commercial entities, and users with a setup where more than one copy of GranuLab may be run simultaneously on different computers.

The prices will change by the first beta release of GranuLab².

you will get

* **GranuLab 8**, the 8-stream version of **GranuLab 1.0**.

* All upcoming versions of **GranuLab2** (alphas, betas, and the final tool).

Please note that **GranuLab** will only be distributed by email for the time being. You should therefore provide a (preferably stable) email address to send it to.

rasmus ekman

rasmus.ekman@abc.se

