

Welcome to the readme of EL Locus Solus' Else library for Pd

# ELSE - EL Locus Solus' Externals

---

## Version 1.0 beta-20 (Released May 8th 2019)

Needs Pd 0.49-0 or above

This project is still in a beta phase, where drastic changes may occur and backwards compatibility is not guaranteed until a final release is available. Find the latest and all releases at: <https://github.com/porres/pd-else/releases>

---

Copyright © 2017–2019 Alexandre Torres Porres <porres@gmail.com>

This work is free. You can redistribute it and/or modify it under the terms of the Do What The Fuck You Want To Public License, Version 2, as published by Sam Hocevar.

See License.txt <https://github.com/porres/pd-else/blob/master/License.txt> and <http://www.wtfpl.net/> for more details

Other licenses may apply for specific objects and this is informed in the source code (example: the [giga.rev~] object).

---

## Acknowledgements

Special thanks to Flávio Luis Schiavoni, for helping me out in a few things when I needed during my first coding attempts and contributing to this library with the objects: [median~] and [keyboard].

I'd also like to thank my cyclone buddy Matt Barber, for developing the "magic" code I'm using here and also contributing with the [float2bits], [brown~], [gray~] and [pinknoise~] objects.

---

"EL Locus Solus" is run by me, Alexandre Torres Porres. It organizes cultural events/concerts and music technology courses <http://alexandre-torres.wixsite.com/el-locus-solus> where a Live Electronics tutorial is provided with examples in Pure Data for its courses. These have just been translated and completely rewritten to english with plans of being accompanied by a book. The first versions are available at: <https://github.com/porres/Live-Electronic-Music-Tutorial>.

It is advisable to also download this tutorial as it solely depends on the ELSE library and it is a great

didactic companion to this library, in which might be merged into a single download in the future.

These examples were first developed for the now abandoned Pd Extended, making extensive use of the existing objects available in Pd Extended's libraries. Even though Pd Extended had many externals, there was the need at some point for something "else" - thus, this library emerged with the goal of providing more objects to include missing functionalities in the Pd Ecosystem.

But the library grew to encompass functionalities found in other Pd objects/libraries in Pd Extended as well, with a different design and more functionalities. This was done in order to remove ALL the dependencies of the didactic material from these other libraries - with the goal to rely on just a single library that's alive (in active development) instead of many projects that are now long gone abandoned or not receiving attention. I'm also involved in maintaining Cyclone, a legacy library for Pd (see: <https://github.com/porres/pd-cyclone>). But ELSE also superseded even cyclone for the purposes of my didactic material.

The goal of ELSE also outgrew the didactic material and includes now objects not necessarily depicted in the computer music examples. Moreover, even basic elements from Pd Vanilla are being redesigned into new objects. So that's it, ELSE is becoming a quite big library and keeps growing and growing.

It will still take a little while for ELSE to stabilize into a final version. For now, it's at an early "Beta" stage of development, where drastic changes may occur and backwards compatibility is not guaranteed until a final release is available.

---

## Downloading ELSE:

Look for the latest releases in <https://github.com/porres/pd-else/releases> - but ELSE is also available via 'deken' (Pd's external manager). In Pd, just go for Help => Find Externals and search for 'else'.

## Installing ELSE:

This release needs Pd Vanilla 0.49-0 or above, it doesn't work in Pd Extended/Purr Data. ELSE comes with a set of separate binaries, so all you need to is add the "else" path to Pd via "Preferences => Path" or using the [declare] object.

## Building ELSE for Pd Vanilla:

ELSE relies on the build system called "pd-lib-builder" by Katja Vetter (check the project in: <https://github.com/pure-data/pd-lib-builder>). PdLibBuilder tries to find the Pd source directory at several common locations, but when this fails, you have to specify the path yourself using the `pdincludepath` variable. Example:

```
make pdincludepath=~/.pd-0.49-0/src/ (for Windows/MinGW add 'pdbinpath=~/.pd-0.49-0/bin/)
```

- Installing with pdlibbuilder

go to the pd-else folder and use "objectsdir" to set a relative path for your build, something like:

```
make install objectsdir=../else-build
```

Then move it to your preferred install folder for Pd and add it to the path.

Cross compiling is also possible with something like this

```
make CC=arm-linux-gnueabi-gcc target.arch=arm7l install objectsdir=../
```

## Current Object list (329 objects):

---

**ASSORTED:** [3] - [table~] - [nbang] - [meter]

**FFT:** [2] - [hann~] - [bin.shift~]

**PATCH/SUBPATCH MANAGEMENT:** [8] - [args] - [dollarzero] - [receiver] - [blocksize~] - [click] - [properties] - [window] - [loadbanger] / [lb]

**LIST MANAGEMENT:** [7] - [break] - [order] - [regroup] - [scramble] - [sort] - [merge] - [unmerge]

**MESSAGE MANAGEMENT:** [20] - [makesymbol] - [separate] - [fromany] - [toany] - [any2symbol] - [changed] - [hot] - [initmess] - [message] - [setmess] - [pack2] - [pick] - [router] - [routeall] - [routetype] - [selector] - [stack] - [trigger2] / [t2] - [sig2float~] / [s2f~] - [float2sig~] / [f2s~]

**FILE MANAGEMENT:** [1] - [dir]

**MIDI:** [17] - [sysrt.in] - [sysrt.out] - [ctl.in] - [ctl.out] - [touch.in] - [touch.out] - [pgm.in] - [pgm.out] - [bend.in] - [bend.out] - [note.in] - [note.out] - [clock] - [panic] - [mono] - [voices] - [suspended]

**MATH: FUNCTIONS:** [23] - [add~] - [add] - [median] - [avg] - [mov.avg] - [count] - [ceil] - [ceil~] - [floor] - [floor~] - [int~] - [rint~] - [rint] - [quantizer~] - [quantizer] - [fold] - [fold~] - [lastvalue] - [mag] - [mag~] - [sin~] - [wrap2] - [wrap2~]

**MATH: CONVERSION:** [26] - [hex2dec] - [bpm] - [dec2hex] - [car2pol] - [car2pol~] - [cents2ratio] - [cents2ratio~] - [ms2samps] - [ms2samps~] - [db2lin] - [db2lin~] - [float2bits] - [hz2rad] - [hz2rad~] - [lin2db] - [lin2db~] - [rad2hz] - [rad2hz~] - [ratio2cents] - [ratio2cents~] - [samps2ms] - [samps2ms~] - [pol2car] - [pol2car~] - [rescale] - [rescale~]

**MATH: CONSTANT VALUES:** [4] - [sr~] - [nyquist~] - [pi] - [e]

**MATH: RANDOM:** [7] - [randf] - [randf~] - [rands] - [randi] - [randi~] - [drunkard~] - [drunkard]

**LOGIC: [2]** - [loop] - [moses~]

**AUDIO PROCESSING: ASSORTED [21]** - [downsample~] - [conv~] - [chorus~] - [fbdelay~] - [ffdelay~] - [shaper~] - [crusher~] - [drive~] - [flanger~] - [freq.shift~] - [pitch.shift~] - [stretch.shift~] - [ping.pong~] - [rm~] - [tremolo~] - [vibrato~] - [vocoder~] - [morph~] - [freeze~] - [pvoc.freeze~] - [phaser~]

**AUDIO PROCESSING: DYNAMICS [5]** - [compress~] - [duck~] - [expand~] - [noisegate~] - [norm~]

**AUDIO PROCESSING: REVERBERATION: [9]** - [allpass.rev~] - [comb.rev~] - [echo.rev~] - [mono.rev~] - [stereo.rev~] - [free.rev~] - [giga.rev~] - [plate.rev~] - [fdn.rev~]

**AUDIO PROCESSING: FILTERS [23]:** - [allpass.2nd~] - [allpass.filt~] - [comb.filt~] - [lop.bw~] - [hip.bw~] - [biquads~] - [bandpass~] - [bandstop~] - [crossover~] - [bpbank~] - [bicoeff] - [brickwall~] - [eq~] - [highpass~] - [highshelf~] - [lowpass~] - [lowshelf~] - [mov.avg~] - [resonbank~] - [resonbank2~] - [resonant~] - [resonant2~] - [svfilter~]

**SAMPLING/PLAYING/GRANULATION: [8]** - [player~] - [gran.player~] - [pvoc.player~] - [pvoc.live~] - [rec~] - [rec.file~] - [play.file~] - [sample~]

**PHYSICAL MODELLING: [1]** - [pluck~]

**OSCILLATORS (DETERMINISTIC GENERATORS): [24]** - [cosine~] - [impulse~] / [imp~] - [impulse2~] / [imp2~] - [parabolic~] - [pulse~] - [saw~] - [saw2~] - [oscbank~] - [oscbank2~] - [sine~] - [square~] - [tri~] - [vsaw~] - [pмосc~] - [wavetable~] / [wt~] - [bl.imp~] - [bl.imp2~] - [bl.saw~] - [bl.saw2~] - [bl.sine~] - [bl.square~] - [bl.tri~] - [bl.vsaw~] - [bl.wavetable~]

**CHAOTIC GENERATORS: [24]** - [brown~] - [clipnoise~] - [crackle~] - [cusp~] - [fbsine~] - [fbsine2~] - [gbman~] - [gray~] - [henon~] - [ikedal~] - [latoocarfian~] - [lorenz~] - [lfnnoise~] - [lincong~] - [logistic~] - [quad~] - [rampnoise~] - [randpulse~] - [randpulse2~] - [standard~] - [stepnoise~] - [pinknoise~] - [xmod~] - [xmod2~]

**SIGNAL ROUTING: [11]** - [balance~] - [pan2~] - [pan4~] - [spread~] - [rotate~] - [xfade~] - [xgate~] - [xgate2~] - [xselect~] - [xselect2~] - [mtx~]

**CONTROL: [22]** - [adsr~] - [asr~] - [autofade~] - [autofade2~] - [decay~] - [decay2~] - [envelope~] - [envgen~] - [fader~] - [function~] - [slew~] - [glide~] - [glide2~] - [ramp~] - [susloop~] - [sequencer] - [sequencer~] - [impseq~] - [lfo] - [lfnnoise] - [impulse] - [pulse]

**TRIGGERS: [26]** - [above] - [above~] - [bangdiv] - [coin] - [coin~] - [dust~] - [dust2~] - [gatehold~] - [gate2imp~] - [pimp~] - [tempo] - [tempo~] - [pulsecount~] - [pulsediv~] - [sh~] - [schmitt] - [schmitt~] - [status] - [status~] - [trig.delay~] - [trig.delay2~] - [toggleff~] - [timed.gate~] - [match~] - [trig2bang~] - [trighold~]

**SIGNAL ANALYSIS: [11]** - [changed~] - [changed2~] - [detect~] - [lastvalue~] - [median~] - [peak~] - [maxpeak~] - [rms~] - [mov.rms~] - [vu~] - [zerocross~]

**GUI: [21]** - [mtx.ctl] - [biplot] - [pic] - [colors] - [function] - [slider2d] - [display] - [display~] - [out~] - [gain~] - [gain2~] - [keyboard] - [graph~] - [spectrograph~] - [meter~] - [meter2~] - [note] - [mix2~] - [mix4~] - [setdsp~] - [openfile]

**EXTRA: [2]** - [output~] - [cmul~]