

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

ELSE - EL Locus Solus' Externals

Version 1.0 beta-8 (Released: March 8th, 2018)

Needs Pd 0.48-1 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. Find latest releases at: <https://github.com/porres/pd-else/releases>

Copyright © 2017–2018 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

Acknowledgements

Special thanks to Flávio Luis Schiavoni, for helping me out in a few things, teaching me how to code 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] and [gray~] objects.

"EL Locus Solus" is run by Alexandre Torres Porres, who organizes cultural events/concerts and teaches computer music courses; website: <http://alexandre-torres.wixsite.com/el-locus-solus>

EL Locus Solus offers a computer music tutorial with examples in Pure Data for its courses. These are in portuguese as of yet, with plans for english translation and formatting into a book. There are over 350 examples for now and they were designed to work with Pd Extended 0.42-5, making extensive use of the existing objects available in Pd Extended's libraries.

Even though extended has quite a large set of external libraries and objects, at some point there was the need of something "else". Thus, this library emerged and EL Locus Solus is now not only offering computer music examples, but also the "ELSE" library, with extra objects for its didactic material and courses.

ELSE also started by stealing some ideas and stuff from SuperCollider, such as most of the chaotic generators UGENs. But what started as a new library to include missing functionalities in the Pd Ecosystem (Pd Extended mostly) now also grew to encompass functionalities found in other Pd objects/libraries in Pd Extended, but with a different design and more functionalities, in order to remove many dependencies of the didactic material from these old and abandoned libraries.

Hence, ELSE is now a place to put any other object I find necessary for my didactic material and work with Pd that is not already in cyclone, another library repository found in Porres' github (<https://github.com/porres/pd-cyclone>).

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 has been tested with Pd Vanilla 0.47-1, not guaranteed to work in any other version or in Pd Extended/Purr Data. ELSE comes with a set of separate binaries, so you just need to add the "else" path to Pd.

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-0.47-1/src/ (for Windows/MinGW add 'pdbinpath=~/.pd-0.47-1/bin/')
```

- Installing with pdlibbuilder

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.

Current Object list (182 objects):

MESSAGE MANAGEMENT: [9] - [break] - [fromany] - [toany] - [changed] - [order] - [setmess] - [pack2] - [routeall] - [routetype]

PATCH/SUBPATCH MANAGEMENT: [3] - [args] - [window] - [inlet2~] - [loadbanger] / [lb]

FILE MANAGEMENT: [1] - [dir]

MIDI: [1] - [clock]

CONVERSION: [18] - [any2symbol] - [cents2rato] - [cents2ratio~] - [db2lin] - [db2lin~] - [float2sig~] - [float2bits] - [hz2rad] - [hz2rad~] - [lin2db] - [lin2db~] - [rad2hz] - [rad2hz~] - [ratio2cents] - [ratio2cents~] - [rescale] - [rescale~] - [sig2float~]

MATH/LOGIC: [16] - [accum~] - [ceil] - [ceil~] - [floor] - [floor~] - [fold] - [fold~] - [lastvalue] - [loop] - [randf] - [randf~] - [randi] - [randi~] - [sin~] - [wrap2] - [wrap2~]

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

AUDIO PLAYING: [2] - [player~] - [sampler~]

AUDIO PROCESSING: [3] - [downsample~] - [allpass.rev~] - [fbdelay~]

REVERB: [1] - [plateverb~]

FILTERS (19): - [allpass.2nd~] - [lop.bw~] - [hip.bw~] - [biquads~] - [bandpass~] - [bandstop~] - [bpbank~] - [bicoeff] - [biplot] - [brickwall~] - [eq~] - [highpass~] - [highshelf~] - [lowpass~] - [lowshelf~] - [resonbank~] - [resonbank2~] - [resonant~] - [resonant2~]

OSCILLATORS (DETERMINISTIC GENERATORS): [14] - [cosine~] - [impulse~] / [imp~] - [impulse2~] / [imp2~] - [parabolic~] - [pulse~] - [sawtooth~] - [sawtooth2~] - [oscbank~] - [oscbank2~] - [sine~] - [square~] - [triangular~] - [vsaw~] - [pmosc~]

CHAOTIC GENERATORS: [21] - [brown~] - [clipnoise~] - [crackle~] - [cusp~] - [fbsine~] - [fbsine2~] - [gbman~] - [gray~] - [henon~] - [iked~] - [latoocarfan~] - [lorenz~] - [lfnnoise~] - [lincong~] - [logistic~] - [quad~] - [rampnoise~] - [randpulse~] - [standard~] - [stepnoise~] - [xmod~]

PHYSICAL MODELLING: [1] - [pluck~]

SIGNAL ROUTING: [10] - [balance~] - [pan2~] - [pan4~] - [rotate~] - [xfade~] - [xgate~] - [xgate2~] - [xselect~] - [xselect2~] - [mtx~]

CONTROL: [14] - [adsr~] - [asr~] - [autofade~] - [autofade2~] - [decay~] - [decay2~] - [fader~] - [glide~] - [glide2~] - [ramp~] - [susloop~] - [sequencer~] - [impseq~] - [lfo]

GUI: [14] - [mtx.ctl] - [slider2d] - [display] - [display~] - [out~] - [gain~] - [gain2~] - [keyboard] - [graph~] - [meter~] - [meter2~] - [mix2~] - [mix4~] - [setdsp~]

TRIGGERS: [19] - [coin~] - [dust~] - [dust2~] - [gatehold~] - [gate2imp~] - [pimp~] - [tempo] - [tempo~] - [pulsecount~] - [pulsediv~] - [sh~] - [trig.delay~] - [trig.delay2~] - [togedge~] - [toggleff~] - [timed.gate~] - [match~] - [trig2bang~] - [trighold~]

SIGNAL ANALYSIS: [10] - [changed~] - [changed2~] - [elapsed~] - [lastvalue~] - [median~] - [peak~] - [rms~] - [vu~] - [past~] - [zerocross~]

EXTRA: [1] - [output~]