

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

ELSE - EL Locus Solus' Externals

Version 1.0 beta-11 (Released August 30th 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], [brown~], [gray~] and [pinknoise~] objects.

"EL Locus Solus" is run by me, Alexandre Torres Porres, and is a thing that organizes cultural events/concerts and music technology 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 and are being currently translated and completely rewritten to english with plans of being accompanied by a book. There are about 325 examples for now and they were first designed to work with Pd Extended 0.42-5, making extensive use of the existing objects available in Pd Extended's libraries.

Even though Pd Extended's libraries form a large set of external objects, at some point there was the need of something "else". Thus, this library emerged with the goal of providind extra objects. 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 (libraries from Pd Extended mostly) has now grown to encompass functionalities found in other Pd objects/libraries in Pd Extended, but with a different design and more functionalities, in order to remove ALL the dependencies of the didactic material from these libraries.

The reasons behind this is to rely on just a single library that was not abandoned instead of many projects that are now long gone abandoned. The consequence is that the tutorial is now being rewritten from scratch and is also expanding as there are plenty new things possible with the new objects from ELSE.

I'm also involved in maintaining Cyclone, a legacy library for Pd (see: <https://github.com/porres/pd-cyclone>). There was the idea of relying also on cyclone for this didactic material, but ELSE is now also superseding even cyclone, at least 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.

Hence, ELSE is now a coming to be a quite big library, and keeps growing and growing, and it will take a while to stabilize. We can say then it's still in an early stage of development, that's why it's still in Beta stage, where drastic changes may occur and backwards compatibility is not guaranteed until a final release.

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-0.48-1/src/ (for Windows/MinGW add 'pdbinpath=~/.pd-0.48-1/bin/')
```

- Installing with pdlibbuilder

use "objectsdirectory" 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 (261 objects):

ASSORTED: [2] - [nbang] - [table~]

PATCH/SUBPATCH MANAGEMENT: [7] - [args] - [blocksize~] - [click] - [properties] - [window] - [loadbanger] / [lb] - [initmess]

MESSAGE MANAGEMENT: [13] - [break] - [fromany] - [toany] - [any2symbol] - [changed] - [order] - [hot] - [setmess] - [pack2] - [routeall] - [routetype] - [sig2float~] - [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: [27] - [add~] - [add] - [median] - [count] - [ceil] - [ceil~] - [floor] - [floor~] - [int~] - [rint~] - [rint] - [quantizer~] - [quantizer] - [fold] - [fold~] - [lastvalue] - [mag] - [mag~] - [power] - [power~] - [randf] - [randf~] - [randi] - [randi~] - [sin~] - [wrap2] - [wrap2~]

MATH: CONVERSION: [22] - [hex2dec] - [bpm] - [dec2hex] - [car2pol] - [car2pol~] - [cents2ratio] - [cents2ratio~] - [db2lin] - [db2lin~] - [float2bits] - [hz2rad] - [hz2rad~] - [lin2db] - [lin2db~] - [rad2hz] - [rad2hz~] - [ratio2cents] - [ratio2cents~] - [pol2car] - [pol2car~] - [rescale] - [rescale~]

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

LOGIC: [1] - [loop]

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

AUDIO PROCESSING: [12] - [downsample~] - [allpass.rev~] - [fbdelay~] - [ffdelay~] - [shaper~] - [crusher~] - [drive~] - [noisegate~] - [norm~] - [freqshift~] - [rm~] - [tremolo~]

AUDIO PROCESSING: REVERB: [1] - [plateverb~]

AUDIO PROCESSING: FILTERS (22): - [allpass.2nd~] - [lop.bw~] - [hip.bw~] - [biquads~] - [bandpass~] - [bandstop~] - [crossover~] - [bpbank~] - [bicoeff] - [biplot] - [brickwall~] - [eq~] - [highpass~] - [highshelf~] - [lowpass~] - [lowshelf~] - [mov.avg~] - [resonbank~] - [resonbank2~] - [resonant~] - [resonant2~] - [svfilter~]

OSCILLATORS (DETERMINISTIC GENERATORS): [24] - [cosine~] - [impulse~] / [imp~] - [impulse2~] /

[imp2~] - [parabolic~] - [pulse~] - [saw~] - [saw2~] - [oscbank~] - [oscbank2~] - [sine~] - [square~] - [tri~] -
[vsaw~] - [pmosc~] - [wavetable~] / [wt~] - [bl.imp~] - [bl.imp2~] - [bl.saw~] - [bl.saw2~] - [bl.sine~] -
[bl.square~] - [bl.tri~] - [bl.vsaw~] - [bl.wavetable~]

CHAOTIC GENERATORS: [23] - [brown~] - [clipnoise~] - [crackle~] - [cusp~] - [fbsine~] - [fbsine2~] -
[gbman~] - [gray~] - [henon~] - [ikedam~] - [latoocarfian~] - [lorenz~] - [lfnnoise~] - [lincong~] - [logistic~] -
[quad~] - [rampnoise~] - [randpulse~] - [randpulse2~] - [standard~] - [stepnoise~] - [pinknoise~] - [xmod~]

PHYSICAL MODELLING: [1] - [pluck~]

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

CONTROL: [18] - [adsr~] - [asr~] - [autofade~] - [autofade2~] - [decay~] - [decay2~] - [envelope~] -
[envgen~] - [fader~] - [function~] - [glide~] - [glide2~] - [ramp~] - [susloop~] - [sequencer] - [sequencer~] -
[impseq~] - [lfo]

TRIGGERS: [24] - [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: [10] - [changed~] - [changed2~] - [elapsed~] - [lastvalue~] - [median~] - [peak~] -
[rms~] - [mov.rms~] - [vu~] - [zerocross~]

GUI: [18] - [mtx.ctl] - [pic] - [colors] - [envelope] - [slider2d] - [display] - [display~] - [out~] - [gain~] - [gain2~]
- [keyboard] - [graph~] - [meter~] - [meter2~] - [mix2~] - [mix4~] - [setdsp~] - [openfile]

EXTRA: [1] - [output~]