ELSE - EL Locus Solus' Externals

for the Pure Data programming language

Version: 1.0-0 beta-37 With Live Electronics Tutorial

Released: March 13th, 2021

About ELSE

This version of ELSE needs *Pd 0.51-3* or above, download Pure Data from: http://msp.ucsd.edu/software.html

ELSE is a big library for Pure Data that provides a cohesive system for computer music, it also serves as a basis for an Live Electronics Tutorial by the same author: Alexandre Torres Porres. This tutorial is also found as part of the download of the ELSE library. Note that you can also download Camomile with support for ELSE externals, see https://github.com/emviveros/Camomile-ELSE/releases/tag/beta36

This project is still in a beta phase, where changes may occur and backwards compatibility is not guaranteed until a final release is available.

This library's repository resides at https://github.com/porres/pd-else/.

Downloading ELSE:

You can get ELSE from https://github.com/porres/pd-else/releases - where all releases are available, but ELSE is also found via Pd's external manager (In Pd, just go for Help => Find Externals and search for 'else'). In any case, you should download the folder to a place Pd automatically searches for, and the common place is the ~/documents/pd/externals folder.

Note that since version 1.0-0 beta 28, the downloads of ELSE also contain a "live eletronics tutorial" as part of the package (as mentioned above). Look for the 'live-electronics-tutorial' folder inside it and also check its README on how to install it.

Instructions on how to build ELSE are provided below.

Installing ELSE:

ELSE comes as a set of separate binaries and abstractions, so it works if you just add its folder to the path or use [declare -path else]. ELSE comes with a binary that you can use load via "Preferences => Startup" or with [declare -lib else], but all that this does is print information of whaty version of ELSE you got wehn you open Pd. You can also just load the 'else' external for that same purpose, check its help file.

It's important to stress this release needs Pd Vanilla 0.51-1 or above (Pd Extended/Purr Data aren't supported).

More About ELSE

"EL Locus Solus" 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. This tutorial it solely depends on the ELSE library and it is a great didactic companion to this library. Both the library and the tutorial are provided as a single download, directly from Pure Data or GitHub.

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 Ecossystem.

But the library grew to encompass functionalities found in other Pd objects/libraries from old 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 superseeds cyclone for the purposes of this 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.

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-0.51-1/src/
(for Windows/MinGW add 'pdbinpath=~/pd-
0.51-1/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-gnueabihf-gcc
target.arch=arm7l install
objectsdir=../
```

Acknowledgements

Special thanks to Flávio Luis Schiavoni, for helping me out in a few things when I first started coding and collaborating 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 collaborating with the [float2bits], [brown~], [gray~], [perlin~] and [pinknoise~] objects.

Current Object list (399objects):

ASSORTED: [04]

- [table~]
- [meter]
- [euclid]
- [else]

FFT: [02]

- [hann~]
- [bin.shift~]

TUNING/NOTES: [09]

- [autotune]
- [autotune2]
- [retune]
- [eqdiv]
- [frac2dec]
- [dec2frac]
- [midi2freq]
- [pitch2note]
- [note2pitch]

PATCH/SUBPATCH MANAGEMENT: [18]

- [args]
- [dollsym]
- [receiver]
- [blocksize~]
- [properties]
- [fontsize]
- [canvas.active]
- [canvas.bounds]
- [click]
- [canvas.gop]
- [canvas.pos]
- [canvas.edit]

- [canvas.vis]
- [canvas.setname]
- [canvas.wname]
- [canvas.name]
- [canvas.zoom]
- [loadbanger] / [lb]

MESSAGE MANAGEMENT: [23]

- [format]
- [nbang]
- [unite]
- [separate]
- [symbol2any]
- [any2symbol]
- [buffer]
- [changed]
- [hot]
- [initmess]
- [message]
- [pack2]
- [pick]
- [limit]
- [spread]
- [router]
- [routeall]
- [routetype]
- [selector]
- [stack]
- [trigger2] / [t2]
- [sig2float~] / [s2f~]
- [float2sig~] / [f2s~]

LIST/MESSAGE MANAGEMENT: [15]

- [break]
- [order]
- [combine]
- [group]
- [iterate]

- [insert]
- [scramble]
- [sort]
- [reverse]
- [rotate]
- [sum]
- [stream]
- [slice]
- [merge]
- [unmerge]

FILE MANAGEMENT: [01]

- [dir]

MIDI: [18]

- [midi]
- [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]
- [suspedal]

MATH: FUNCTIONS: [26]

- [add~]
- [add]

- [median]
- [avg]
- [mov.avg]
- [count]
- [gcd]
- [lcm]
- [ceil]
- [ceil~]
- [factor]
- [floor]
- [floor~]
- [int~]
- [rint~]
- [rint]
- [quantizer~]
- [quantizer]
- [fold]
- [fold~]
- [lastvalue]
- [mag]
- [mag~]
- [sin~]
- [wrap2]
- [wrap2~]

MATH: CONVERSION: [27]

- [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~]
- [op~]

MATH: CONSTANT VALUES: [04]

- [sr~]
- [nyquist~]
- [pi]
- [e]

MATH: RANDOM: [04]

- [rand.f]
- [rand.f~]
- [rand.i]
- [rand.i~]

LOGIC: [01]

- [loop]

AUDIO PROCESSING: ASSORTED [23]

- [downsample~]
- [conv~]
- [chorus~]
- [del~]

- [fbdelay~]
- [ffdelay~]
- [rdelay~]
- [shaper~]
- [crusher~]
- [drive~]
- [flanger~]
- [freq.shift~]
- [pitch.shift~]
- [stretch.shift~]
- [ping.pong~]
- [rm~]
- [tremolo~]
- [vibrato~]
- [vocoder~]
- [morph~]
- [freeze~]
- [pvoc.freeze~]
- [phaser~]

AUDIO PROCESSING: DYNAMICS [05]

- [compress~]
- [duck~]
- [expand~]
- [noisegate~]
- [norm~]

AUDIO PROCESSING: REVERBERATION: [09]

- [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: [13]

- [player~]
- [gran.player~]
- [pvoc.player~]
- [pvoc.live~]
- [grain.sampler~]
- [grain.live~]
- [batch.rec~]
- [batch.write~]
- [rec.file~]
- [play.file~]
- [tabplayer~]

- [tabwriter~]
- [sample~]

SYNTHESIS: GRANULAR SYNTHESIS: [01]

- [grain.synth~]

SYNTHESIS: PHYSICAL MODELLING: [01]

- [pluck~]

SYNTHESIS: 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~]

SYNTHESIS: CHAOTIC/NOISE GENERATORS: [25]

- [brown~]
- [clipnoise~]
- [perlin~]
- [crackle~]
- [cusp~]
- [fbsine~]
- [fbsine2~]
- [gbman~]
- [gray~]
- [henon~]
- [ikeda~]
- [latoocarfian~]
- [lorenz~]
- [lfnoise~]
- [lincong~]
- [logistic~]
- [quad~]
- [rampnoise~]
- [randpulse~]
- [randpulse2~]
- [standard~]
- [stepnoise~]
- [pinknoise~]
- [xmod~]
- [xmod2~]

SIGNAL ROUTING: [12]

- [balance~]
- [pan2~]
- [pan4~]
- [pan8~]
- [spread~]
- [rotate~]
- [xfade~]
- [xgate~]
- [xgate2~]
- [xselect~]
- [xselect2~]

- [mtx~]

CONTROL: [26]

- [mouse]
- [canvas.mouse]
- [adsr~]
- [asr~]
- [autofade~]
- [autofade2~]
- [decay~]
- [decay2~]
- [envelope~]
- [envgen~]
- [fader~]
- [function~]
- [lag~]
- [lag2~]
- [glide~]
- [glide2~]
- [ramp~]
- [susloop~]
- [drum.seq]
- [sequencer]
- [sequencer~]
- [impseq~]
- [lfo]
- [phasor]
- [impulse]
- [pulse]

CONTROL: RANDOM: [09]

- [rand.seq]
- [markov]
- [drunkard~]
- [drunkard]
- [randpulse]
- [randpulse2]
- [lfnoise]

- [stepnoise]
- [rampnoise]

TRIGGERS: [28]

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

ANALYSIS: [13]

- [changed~]
- [changed2~]
- [detect~]
- [lastvalue~]

- [median~]
- [peak~]
- [range]
- [range~]
- [maxpeak~]
- [rms~]
- [mov.rms~]
- [vu~]
- [zerocross~]

GUI: [33]

- [gui]
- [pad]
- [messbox]
- [mtx.ctl]
- [biplot]
- [pic]
- [colors]
- [function]
- [circle]
- [slider2d]
- [display]
- [display~]
- [out1~]
- [out~]
- [out4~]
- [out8~]
- [gain~]
- [gain2~]
- [button]
- [keyboard]
- [graph~]
- [range.hsl]
- [spectrograph~]
- [meter~]
- [meter2~]
- [meter4~]
- [meter8~]

- [note]
- [mix2~]
- [mix4~]
- [setdsp~]
- [openfile]
- [oscilloscope~]

EXTRA: [02]

- [output~]
- [cmul~]