ELSE - EL Locus Solus' Externals

for the Pure Data programming language

Version: 1.0-0 RC-7 (release candidate #7) With Live Electronics Tutorial.

Released: March 1st 2023

About ELSE

This version of ELSE needs **Pd 0.53-2** or above.

ELSE is a big library of externals that extends the performance Pure Data (Pd) - Miller S. Puckette's realtime computer music environment (download Pd from: http://msp.ucsd.edu/software.html).

ELSE provides a cohesive system for computer music, it also serves as a basis for an Live Electronics Tutorial by the same author, yours truly, Alexandre Torres Porres. This library's repository resides at https://github.com/porres/pd-else/. This tutorial is also found as part of the download of the ELSE library. Look for the 'Live-Electronics-Tutorial' folder inside it and also check its README on how to install it.

Note that you can also download Camomile with support for ELSE externals, see https://github.com/emviveros/Camomile-ELSE/releases.

ELSE is also part of PlugData by Timothy Schoen, which is a fork of Pd that loads as a standalone or VST with a revamped GUI. ELSE has received collaboration from Tim and others involved with PlugData and many objects have been included in ELSE just so they are supported in PlugData. See: https://github.com/timothyschoen/PlugData

This project is still in an experimental phase (currently at a 'release candidate' phase), where changes may occur and backwards compatibility is not guaranteed until a final and more stable release is available.

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. Instructions on how to install and 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 what version of ELSE you have when you open Pd. You can also just load the 'else' external for that same purpose, check its help file.

It is important to stress this library runs in Pd Vanilla 0.52-1 or above and is not compatible to forks like the long dead "Pd Extended" and its new reincarnations "Pd-L2ork/Purr Data". Nevertheless, ELSE is included in the PlugData fork --> https://github.com/timothyschoen/PlugData, and a port of ELSE to Purr Data is currently underway --> https://github.com/agraef/purr-data/tree/pd-else.

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 (assuming the unpacked Pd package is in ~/pd-0.53-1; for Windows/MinGW add `pdbinpath=~/pd-0.53-1/bin/`):

```
make pdincludepath=~/pd-0.53-2/src/
```

Installing with pdlibbuilder

Go to the pd-else folder and use "objectsdir" to set an *absolute* path for your build, something like:

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

Then move it from there 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=~/else-build
```

Installing sfont~ and plaits~

For technical reasons these objects reside in their own subdirectories of the ELSE source directory, so a normal build as described above will skip them. If you want to build and install these separately, check the "sfont~" and "plaits~" folders and their READMEs with instructions.

But the easiest (and recommended) way is to build and install those objects from the toplevel source directory using the special 'sfont' and 'plaits' targets:

```
make sfont plaits pdincludepath=~/pd-0.53-1/src/
```

```
make sfont-install plaits-install objectsdir=~/else-build
```

This also makes sure that [sfont~] and [plaits~] get added to the else library directory instead of their own subdirectories, so that the included help patches work without further ado.

More About ELSE

"EL Locus Solus" is run by yours truly, Alexandre Torres Porres, and 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 been recently translated and completely rewritten to english with plans of being accompanied by a book. The latest releases are available at: https://github.com/porres/Live-Electronic-Music-Tutorial. This tutorial solely depends on the ELSE library and is a great didactic companion to this library. Both the library and the tutorial are provided as a single download, directly via Pure Data or GitHub.

The examples from the first incarnation of this tutorial 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 much 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. See below in this document a list of alternatives to Cyclone provided by ELSE.

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.

ELSE has been in active development for over 5 years but it will still take a little while for it to stabilize into a final version. For now, it's at a "Release Candidate" stage of development, where changes may occur and backwards compatibility is not guaranteed until a final release is available.

Acknowledgements

Flávio Luis Schiavoni helped me out in a few things when I first started coding and collaborated with the objects: [median~] and [keyboard]. Lucas Cordiviola is an active tester and has helped countless times

with compilation issues for windows.

I'd also like to thank my Cyclone buddies Derek Kwan and Matt Barber, cause I started learning how to code externals with them as part of the cyclone team. Other developers of cyclone need to be praised, like Czaja, the original author, as I did steal quite a bit from cyclone into ELSE. I'd like to give a special thanks for Matt Barber for developing the "magic" in cyclone that I'm using here and also collaborating to ELSE with the objects: [float2bits], [brown~], [gray~], [perlin~], [pink~] and [blip~].

Lucarda is always helpful testing stuff for Windows and more. Seb shader is a tcl/tk master that helped me a lot with this (which I know next to nothing) and is responsible for the [keycode] object.

Kudos and thanks to my buddy Esteban Viveros for helping with the compilation of ELSE for other systems as well as ELSE for Camomile. See: https://github.com/porres/Camomile-ELSE.

Albert Graef from Purr Data is doing a great job helping port ELSE so it builds and runs on Purr Data (still experimental and a long way to go). He's also doing several valuable contributions to the code and being helpful in general in this process - check https://github.com/agraef/purr-data/releases/tag/2.19.2+ELSE.

For Last, Timothy Schoen is doing an amazing jaw dropping job with PlugData, a project based on camomile that includes ELSE. Timothy has also helped me fix many ELSE related issues and coded first versions of compiled band limited oscillators. PlugData is also a fork of Pd with a revamped GUI. See: https://github.com/timothyschoen/PlugData.

Current Object list (474 objects):

ASSORTED: [03]

- [else]
- [chrono]
- [datetime]

TABLE: [03]

- [tabgen]
- [tabreader]
- [tabreader~]

FFT: [02]

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

TUNING/NOTES: [19]

- [scales]
- [sclae2freq]
- [scala]
- [autotune]
- [autotune2]

- [makenote2]
- [retune]
- [eqdiv]
- [cents2scale]
- [scale2cents]
- [frac2cents]
- [cents2frac]
- [frac2dec]
- [dec2frac]
- [midi2freq]
- [freq2midi]
- [pitch2note]
- [note2pitch]
- [note2dur]

PATCH/SUBPATCH MANAGEMENT: [20]

- [args]
- [meter]
- [presets]
- [dollsym]
- [receiver]
- [retrieve]
- [blocksize~]
- [click]
- [properties]
- [fontsize]
- [canvas.active]
- [canvas.bounds]
- [canvas.gop]
- [canvas.pos]
- [canvas.edit]
- [canvas.file]
- [canvas.vis]
- [canvas.setname]
- [canvas.zoom]
- [loadbanger] / [lb]

GENERAL MESSAGE MANAGEMENT: [29]

- [format]
- [swap2]
- [nmess]
- [unite]
- [separate]
- [symbol2any]
- [any2symbol]
- [buffer]

- [changed]
- [hot]
- [initmess]
- [message]
- [default]
- [pack2]
- [pick]
- [limit]
- [spread]
- [router]
- [route2]
- [routeall]
- [routetype]
- [selector]
- [stack]
- [store]
- [morph]
- [interpolate]
- [sig2float~] / [s2f~]
- [float2sig~] / [f2s~]
- [pipe2]

LIST/MESSAGE MANAGEMENT: [19]

- [break]
- [order]
- [combine]
- [delete]
- [group]
- [iterate]
- [insert]
- [scramble]
- [sort]
- [reverse]
- [rotate]
- [replace]
- [sum]
- [stream]
- [slice]
- [merge]
- [unmerge]
- [amean]
- [gmean]

FILE MANAGEMENT: [01]

- [dir]

MIDI: [20]

- [midi]
- [midi.learn]
- [sysrt.in]
- [sysrt.out]
- [ctl.in]
- [ctl.out]
- [touch.in]
- [touch.out]
- [pgm.in]
- [pgm.out]
- [bend.in]
- [bend.out]
- [note.in]
- [note.out]
- [midi.clock]
- [noteinfo]
- [panic]
- [mono]
- [voices]
- [suspedal]

OSC: [05]

- [osc.route]
- [osc.format]
- [osc.parse]
- [osc.send]
- [osc.receive]

MATH: FUNCTIONS: [32]

- [add~]
- [add]
- [median]
- [avg]
- [mov.avg]
- [count]
- [gcd]
- [lcm]
- [frac.add]
- [frac.mul]
- [ceil]
- [ceil~]
- [factor]
- [floor]
- [floor~]

- [trunc]
- [trunc~]
- [rint~]
- [rint]
- [quantizer~]
- [quantizer]
- [fold]
- [fold~]
- [lastvalue]
- [mag]
- [mag~]
- [sin~]
- [wrap2]
- [wrap2~]
- [op~]
- [op]
- [cmul~]

MATH: CONVERSION: [28]

- [hex2dec]
- [bpm]
- [dec2hex]
- [car2pol]
- [car2pol~]
- [cents2ratio]
- [cents2ratio~]
- [ms2samps]
- [ms2samps~]
- [db2lin]
- [db2lin~]
- [float2bits]
- [hz2rad]
- [lin2db]
- [lin2db~]
- [deg2rad]
- [rad2deg]
- [pz2coeff]
- [coeff2pz]
- [rad2hz]
- [ratio2cents]
- [ratio2cents~]
- [samps2ms]
- [samps2ms~]
- [pol2car]
- [pol2car~]
- [rescale]

- [rescale~]

MATH: CONSTANT VALUES: [04]

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

LOGIC: [01]

- [loop]

AUDIO PROCESSING: ASSORTED [24]

- [downsample~]
- [conv~]
- [chorus~]
- [del~]
- [fbdelay~]
- [ffdelay~]
- [rdelay~]
- [shaper~]
- [crusher~]
- [drive~]
- [power~]
- [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 [25]:

- [allpass.2nd~]
- [allpass.filt~]
- [bitnormal~]
- [comb.filt~]
- [lop.bw~]
- [hip.bw~]
- [biquads~]
- [bandpass~]
- [bandstop~]
- [crossover~]
- [bpbank~]
- [bicoeff2]
- [brickwall~]
- [eq~]
- [highpass~]
- [highshelf~]
- [lop2~]
- [lowpass~]
- [lowshelf~]
- [mov.avg~]
- [resonbank~]
- [resonbank2~]
- [resonant~]
- [resonant2~]
- [svfilter~]

BUFFER/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: SINTHESIZERS: [02]

- [sfont~]
- [plaits~]

SYNTHESIS: GRANULAR SYNTHESIS: [01]

- [grain.synth~]

SYNTHESIS: PHYSICAL MODELLING: [01]

- [pluck~]

SYNTHESIS: OSCILLATORS (DETERMINISTIC GENERATORS): [26]

- [cosine~]
- [impulse~] / [imp~]
- [impulse2~] / [imp2~]
- [parabolic~]
- [pulse~]
- [saw~]
- [saw2~]
- [oscbank~]
- [oscbank2~]
- [sine~]
- [square~]
- [tri~]
- [gaussian~]
- [vsaw~]
- [pmosc~]
- [wavetable~] / [wt~]
- [blip~]
- [bl.osc~]
- [bl.imp~]
- [bl.imp2~]
- [bl.saw~]
- [bl.saw2~]
- [bl.square~]
- [bl.tri~]
- [bl.vsaw~]
- [bl.wavetable~]

SYNTHESIS: CHAOTIC/STOCHASTIC/NOISE GENERATORS: [26]

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

CONTROL: MOUSE/KEYBOARD INTERACTION [4]

- [mouse]
- [canvas.mouse]
- [keycode]
- [keymap]

CONTROL: FADER/PANNING/ROUTING: [15]

- [fader~]
- [autofade~]
- [autofade2~]
- [balance~]
- [pan2~]
- [pan4~]
- [pan8~]
- [spread~]
- [rotate~]
- [xfade~]
- [xgate~]
- [xgate2~]

- [xselect~]
- [xselect2~]
- [mtx~]

CONTROL: SEQUENCERS: [9]

- [euclid]
- [score]
- [score2]
- [pattern]
- [sequencer]
- [sequencer~]
- [impseq~]
- [rec]
- [rec2]

CONTROL: ENVELOPES [6]

- [adsr~]
- [asr~]
- [decay~]
- [decay2~]
- [envelope~]
- [envgen~]

CONTROL: RAMP, LINE GENERATORS / LINE SMOOTHENING [13]

- [ramp~]
- [susloop~]
- [function~]
- [slew]
- [slew2]
- [slew~]
- [slew2~]
- [lag~]
- [lag2~]
- [glide]
- [glide2]
- [glide~]
- [glide2~]

CONTROL: RANDOM/STOCHASTIC: [18]

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

- [rand.dist]
- [histogram]
- [rand.hist]
- [markov]
- [drunkard~]
- [drunkard]
- [brown]
- [randpulse]
- [randpulse2]
- [lfnoise]
- [stepnoise]
- [rampnoise]

CONTROL: CONTROL RATE LFOs [5]

- [lfo]
- [phasor]
- [pimp]
- [impulse]
- [pulse]

CONTROL: TRIGGERS: [27]

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

- [trig2bang~]
- [trighold~]

CONTROL: TRIGGERS: CLOCK [8]

- [clock]
- [metronome]
- [metronome~]
- [polymetro]
- [polymetro~]
- [speed]
- [tempo]
- [tempo~]

ANALYSIS: [15]

- [changed~]
- [changed2~]
- [detect~]
- [lastvalue~]
- [median~]
- [peak~]
- [tap]
- [range]
- [range~]
- [maxpeak~]
- [rms~]
- [mov.rms~]
- [vu~]
- [zerocross~]
- [beat~]

GUI: [36]

- [numbox~]
- [drum.seq]
- [bicoeff]
- [pad]
- [messbox]
- [mtx.ctl]
- [biplot]
- [zbiplot]
- [pic]
- [colors]
- [function]
- [circle]
- [slider2d]
- [display]

- [out1~]
- [out~]
- [out4~]
- [out8~]
- [gain~]
- [gain2~]
- [button]
- [keyboard]
- [graph~]
- [range.hsl]
- [multi.vsl]
- [spectrograph~]
- [meter~]
- [meter2~]
- [meter4~]
- [meter8~]
- [note]
- [mix2~]
- [mix4~]
- [setdsp~]
- [openfile]
- [oscope~]

ALTERNATIVES TO CYCLONE

ELSE offers alternatives to objects from the Cyclone library (a library that clones objects from MAX/MSP). The objects that have no similar counterpart in ELSE (at least so far) are: anal / buddy / capture / capture~ / coll / cycle / decide / decode / frameaccum~ / framedelta~ / funbuff / funnel / flush / forward / histo / kink~ / linedrive / prob / match / maximum / minimum / mousefilter / next / offer / peak / prob / pv / spray / substitute / teeth~ / trough / universal / vectral~

But, here are some considerations: - maximum/minimum => [array max] [array min] - for [capture~] you can use [print~] - for [kink~] you can use [function~] - [spike~] => [else/status~] + [else/detect~] => [threshold~] + [timer] - [anal] is usually used with [prob] for markov chains, but you can use [else/markov] instead - [teeth~] is just a comb reverberator which can be constructed with [else/ffdelay~] + [else/fbdelay~] - [forward] => just message boxes with ";" - [flush] => [poly] has a flush option, so does [else/voices] - for [coll], you can use [text] which miller considers is a better design, but it's simpler.

Alternatives:

- +=~ / [plusequals~] => else/add~
- operators >~ / <~ / %~ / etc => else/op~ => [expr~]
- bitwise operators (bitand~ / bitor~, etc) => [expr~]
- accum => else/add - acos/acosh/acos~/acos~/asin/asin~/atan~/atan2~/asin/asinh~/cosh/cosh~/cosx~/sinh/sinh~/sinx~/tanh/tanh~/tanx~ => [expr]/[expr~]
- active => else/canvas.active

- append => [list append]
- atob/atodb~/dbtoa/dbtoa~ => db2lin/lin2db/db2lin~/lin2db/~
- bangbang => else/loadbanger => trigger
- borax / else/noteinfo
- average~ / avg~ => else/mov.avg~
- bondo => else/hot
- buffer~ => else/sample~
- buffir~ => else/conv~
- counter => else/count
- changed => else/changed~
- click~ => else/impseq~
- clip/clip~ => clip/clip~
- comb~ => else/comb.rev~
- count~ => else/ramp~
- cross~ => else/crossover~
- cycle~ => else/wavetable~ => tabosc4~
- curve~ => else/envgen~
- cartopol/poltocar/cartopol~/poltocar~ => else car2pol/pol2car/car2pol~/pol2car~
- degrade~ => else/crusher~
- drunk => else/drunkard
- delay~ => else/ffdelay~
- delta~ => [rzero~ 1]
- deltaclip~ => else/slew~ => slop~
- downsamp~ => else/downsample~
- edge~ => else/status~ => threshold~
- fromsymbol => else/symbol2any / else/separate
- grab => else/retrieve
- gate => else/router
- gate~ => else/xgate~
- index~ => else/ramp~
- iter => else/iterate
- join => else/merge
- listfunnel => else/order
- loadmess => else/initmess
- line~ => else/envgen~ => vline~
- lookup~ => else/shaper~
- lores~ => else/lowpass~
- mean => else/mov.avg
- matrix~ => else/mtx~
- maximum~/minimum~ => max~/min~/expr~ (totally unnecessary external)
- minmax~ => else/range~
- mstosamps~/samps2ms~ => else/ms2samps ms2samps~ samps2ms samps2ms~
- midiflush => else/panic
- midiformat/midiparse => midi in/out objects (else/note.in/note.out, etc)
- mtr => else/rec
- mousestate => else/mouse

- onebang => else/nmess
- onepole~ => lop~
- overdrive~ => else/drive~
- peakamp~ => else/peak~
- pak => else/pack2
- past => else/above
- peek~ => tabwrite
- phaseshift~ => else/allpass.2nd~
- phasewrap~ => else/wrap2~ => wrap~
- pink~ => else/pinknoise~
- play~ => else/tabplayer~
- poke~ => else/tabwriter~
- pong/pong~ => else/fold / else/wrap2 / else/fold~ / else/wrap2~
- pow~ => pow~ (totally unnecessary external)
- prepend => else/insert => [list prepend]
- round / round~ => else/quantizer / else/quantizer~
- rand~ => else/rampnoise~
- record~ => else/tabwriter~
- reson~ => else/bandpass~
- scale / scale~ => else/rescale / else/rescale~
- seq => else/midi
- speedlim => else/limit
- spell => [list fromsymbol]
- split => else/spread
- sprintf => else/format => makefilename
- sustain => else/suspedal
- switch => else/selector
- sah~ => else/sh~ => samphold~
- selector~ => else/xselect~
- slide~ => else/lag2~
- snapshot~ => else/s2f~ => snapshot~
- svf~ => else/svfilter~
- table => array
- tanh~ (again) => else/drive~
- thresh~ => else/schmitt~
- train~ => else/pulse~
- trapezoid~ => else/envelope~
- triangle~ => else/vsaw~
- trunc~ => else/trunc~
- thresh => else/combine
- togedge => else/status
- tosymbol => else/any2symbol / else/unite
- unjoin => else/unmerge
- urn => else/rand.seq
- uzi => else/loop
- xbendin/sbendin2/xbendout/xbenout2 => else/bend.in / else/bend.out

- xnotein/xnoteout => else/note.in / else/note.out
- wave~ => else/wavetable~
- zerox~ => else/zerocross~
- zl => several dedicated objects include functionalities from it, such as: else/group, else/scramble, else/sort, else/reverse, else/rotate, else/sum, else/slice and else/stream

GUI:

- comment => else/note
- scope~ => else/oscope~